Predicting Patient Mortality from ICU Data through a Quorum of Neural Networks

02-518 Final Project

Rishav Dutta

1 Abstract

Determining how severe the condition of a patient in the ICU can help doctors evaluate how to allocate their resources. This paper attempts to take the data from the first 48 hours of a patient's time in the ICU and make the prediction of mortality. We use data from the PhysioNet 2012 challenge and the model was evaluated on the test set provided. We use the patient's data and global averages to replace missing data and use an RDF based encoding to preprocess the data. Through randomized training sets and random oversampling, we train a quorum of neural networks that each learn slightly differently. Then through a threshold based voting scheme, we predict the mortality of a sample. This model earned an event 1 score (as defined by the PhysioNet challenge) of 0.538 which would have placed 1st in the competition.

2 Introduction

The intensive care unit (ICU) in a hospital is the wing devoted to diagnosis and care of severe, but possibly reversible, medical conditions. This includes both severe illnesses and life-threatening injuries, both of which require constant monitoring and care for the patient. Those in the intensive care unit have many of their vital organ systems monitored by expensive medical machines and processes. Furthermore, many failing organ processes

need to be simulated through specialized equipment in order to ensure the body functions return to normal. ICU care is extremely expensive for the patient. The specialization of the equipment, higher concentration of available staff, and long lengths of stay can contribute to upwards of $4300 US dollars per day per patient [1].

In order to best tailor the care to the patients' needs and to best allocate resources, it is essential that hospitals and doctors can best predict ICU outcomes of patients. This paper attempts to take ICU patient data from the PhysioNet/Computers in Cardiology Challenge 2012 [7] and develop a predictor for in-hospital patient deaths from the first 48 hours of data since the patients admittance. It is vital to make the mortality prediction so that the decision can be made how to best allocate ICU resources to those who need it most.

A commonly used metric for ICU patients is the SAPS II score (Simplified Acute Physiology Score). The SAPS II score is calculated 24 hours after the admission to the ICU, and the score is only calculated one time. The measurement results in two different scores, an integer score between 0 and 163 and a predicted mortality between 0% and 100%. This metric can be used in order to predict a patients potential mortality during their time in the ICU.

This paper will be analyzing data taken within the first 48 hours of the patients admittance to the ICU. While this model will not account for real time, time-series inputs, the model presented can potentially be extended for such a modification in the future.

3 Background and Related Work

3.1 Background

The 2012 PhysioNet/Computers in Cardiology Challenge presented numerous entrees that used the same dataset to attempt to solve a similar problem. The goal of the challenge was to present a predictor that takes the ICU data and outputs if the patient was likely to die in hospital or not. The papers that submitted to this challenge have proposed various methods of analyzing the same data, with the intent of maximizing both scores.

3.2 Data

The data that will be used in this paper is the same data that was used in the PhysioNet challenge. The data consists of 12,000 records from ICU stays of 48 hours or more, and only the first 48 hours data are recorded. For each record, there are up to 37 potential variables that could be recorded for each patient at least once during the 48 hours. It is possible that not any of the variables are recorded at all for certain patients. Each variable that is recorded has an associated time-stamp attached to the value. Each record also has a set of six general descriptor variables that describe static features taken at admission. This is `recordID`, `age`, `height`, `weight`, `ICU type`, and `gender`. Finally, each record also has an associated outcome related descriptor that describes the outcome of the ICU stay for the patient.

The time series variables are all values taken from ICU machines during the time of stay of the patient. It is possible that the variable was recorded at some point during the stay but never again recorded. This may lead to "incomplete" data with respect to that variable. It was left up to those attempting the challenge to determine how to address the missing or incomplete data.

3.3 Related Work

There are quite a few papers that attempt to solve the same problem. Algorithms described by these papers are optimized to best predict the probability of death for a patient using very limited patient data. This paper will attempt use the first 4000 samples as training data points just as the competitors in the PhysioNet Challenge did.

An interesting problem with the dataset is the existence of potential missing values as not all the values are recorded every hour. Lee et al. [3] and Pollard et al. [6] dealt with missing values by replacing them with the mean value for a patient's age and gender for that variable i.e. if temperature was not recorded it was replaced with 36.4 °C. Macas et al. did not preprocess the missing data, instead they removed the features that had numerous missing values [4]. Xia et al. suggest that the missing data were not collected because the doctor considered that variable to be normal for the patient [8]. Thus the variable should be set to a value in the normal range (for the patient's age and gender). McMillan et al. considered the missing value patterns in the data to be important features

in training the model as it is present for a potentially significant reason [5]. Interpolating the missing data is a crucial step in training an accurate model.

ICU outcome prediction needs to respect the fact a patient's condition changes with respect to the time spent in the ICU [7]. Hence the algorithm must somehow preserve the time-series nature of the data. Joshi et al. do so through Radial Domain Folding [2]. They suggest encoding the time series data so that a patient's state is linked with a set of elements x_{ij} where i represents the measurement and j represents the patient's state [2]. Macas et al. preserve this nature by generating new features for each patient corresponding to changes in the data over time. For example, a generated feature is the slope of the line of sodium levels in the blood [4]. Xia et al. did something similar in that they used linear regression in order to estimate the direction of motion of a certain value in addition to using the last data point as a significant feature [8].

Once the data is preprocessed, the goal is to learn the function that best classifies the outcome of ICU patients. Lee et al. focus on the importance of placing the correct data into the missing slots, and hence train models that do so differently and then perform regression on the features they created [3]. Macas et al. use a simple linear Bayes classifier whose features were selected through a wrapper based feature selection algorithm [4]. Xia et al. use a feature extraction method to find the top features, and then use those features to train a hundred neural networks. These networks would then vote on the outcome [8].

Xia et al. have done the work most comparable to ours, as both papers use neural networks to solve the problem. The way we generate the features for the neural networks are significantly different. Xia et al. use a voting scheme for a set of 100 neural networks, which is significantly different from our threshold based quorum that averages votes. Furthermore, they do not randomize the training set for each neural network, so their networks learn much differently than ours.

4 Methods

4.1 Initial Representation and Missing Data

The data that we work with is time-stamped labelled variable and values. Each patient (training sample) has a list of tuples in the form (t, k, v) where t is a time, k is a measured variable, v is a measured value. The type of the measurement can be one of

37 possible variables. The time is in the form $hh : mm$ where hh is the hour and mm the minute since admittance to the ICU. Just because a certain k appears at least once in the sample does not mean it will appear again as the physician chose not to measure this variable (potentially for a variety of reasons). Furthermore, it is possible that not all the k will appear in the sample as well.

To best see the evolution of the patient over time, we represent each sample as a matrix DF of 37 columns and 48 rows. Any $x_{ij} \in DF$ represents the measured value of variable j in hour i. If a value is not measured in hour i for variable j then x_{ij} will be NaN. If there are multiple measurements of variable j within the same hour, then x_{ij} is the max of all the measurements taken during that hour. We take the max because we want to estimate the worst state of the patient during the hour.

In order to best improve the model, we decide to replace the $NaNs$ under a set of assumptions. It is not necessarily a valid assumption to suggest that the jth variable was not measured because the patient was at normal levels during that hour. Rather, it is possible that variables are measured at different intervals that are greater than one hour. Hence, we take a different approach to interpolate the missing data in Algorithm 1.

for *all* $x_{ij} == NaN$ **do**
  if $\exists k$ *such that* $k > i$ **then**
    $x_{ij} := x_{kj}$
  else
    $x_{ij :=}$ normal level for variable k
  end
end

Algorithm 1: NaN Replacement

We only replace a value with the normal level if the variable was not measured at all or if the variable was not measured for the remainder of the 48 hours. We make the assumption that if the variable was not measured for the remainder of the 48 hours, the decision was made that the variable has returned to normal values and does not need to be measured for the rest of the ICU stay. This is a reasonable assumption to make because the doctors decided it was more important to record other measurements indicating this one is insignificant to the current state of the patient.

4.2 Preprocessing and Feature Selection

An individual sample DF now is now contains no missing values, however the values in the sample vary based on the units of the variable. In order to standardize the units, we encode the data similarly to [2]. Like Joshi et al., we encode the data $x_{ij} in DF$ through a modified z-score defined in (2) $z'(x_{ij})$ in order to show the degrees of deviation from the normal for each point. Looking through this perspective allows the model to potentially learn the relationships between severely deviated variables which can correspond to failing organ systems [2]. (2) describes how we calculate the z-score.

$$z(x_{ij}) = \frac{x_{ij} - \mu_j}{\sigma(x_j)} \tag{1}$$

(1) is we how to calculate the z-score for the point x_{ij}. μ_j is the mean normal value for the variable j for the age and gender of the patient. We describe $\sigma(x_j)$ to be the value of the standard deviation of the variable j for the age and gender of the patient. We then use this z-score in order to calculate the modified z-value (2).

$$z'(x_{ij}) = \begin{cases} 0, & -1 \leq z(x_{ij}) \leq 1 \\ z(x_{ij}) - 1, & z(x_{ij}) > 1 \\ z(x_{ij}) + 1, & z(x_{ij}) < 1 \end{cases} \tag{2}$$

We then transform each sample DF into zDF such that for an element x_{ij} in DF, the corresponding k in zDF, $k_{ij} = z'(x_{ij})$. This transformation yields zDF a unit-less representation of the deviation from the expected value for each measured variable at each hour. This yields a potential 1776 features. However, we will make the assumption that it is not important to look at each of the values, but rather look at how the values change over the stay in the ICU.

To model the time series features, we extract mean, the maximum, the minimum, the last value, and the slope from each of the columns in zDF. We also extract the mean, the maximum, the minimum, and the slope of the last 12 values from each of the columns as well. We estimate the slope over time through linear regression of the values vs the

hour. Hence from each of the 37 columns, we extract 9 values leading to 333 features for each sample. Furthermore, for each of the extracted values e_i we perform the following transformation TRAN(e_i)(3). We perform this transformation in order to reduce the range of potential values in zDF so that each neural network is less sensitive.

$$TRAN(e_i) = \begin{cases} 0, & e_i = 0 \\ log(e_i), & e_i > 0 \end{cases} \tag{3}$$

In order to pick the features from the 333 that are most relevant, we perform a filter based feature selection algorithm based on mutual information in order to choose the top 30 features to use in the model. 30 was the chosen number because it performed best in preliminary testing. Once the 30 features were selected, only those features were used in model training.

4.3 Training Set Selection

The training data from the PhysioNet 2012 Competition has only 14 percent of the training data consistent of patients who died in-hospital (we consider this a positive example)[7]. In order to make sure the model learned properly, we decided to over sample the positive examples at an expected ratio of 1 to 3. The following randomized algorithm is how we generated the training set for each neural network (Algorithm 2)

.

while *number of training examples* < 4000 **do**
  r = $randInt(0, 200)$;
  if $r \,\%3 == 0$ **then**
    add positive example to training examples
  else
    add negative example to training examples
  end
end

Algorithm 2: Training Set Selection

In developing the test set in this way, we ensure that a positive example is given slightly higher weight so the model learns a positive example better and we ensure the model will not just guess negative for every test case in order to minimize the loss. Furthermore, we ensure that each of the models that we train will have a slightly different exposure to the data so that in voting we have a diverse set of models predicting the outcome.

4.4 Model Development

The neural network used over a single training set could very often hit a local minimum during gradient descent and thus have very poor results. In order to avoid this problem, we randomize the training set for 20 different neural networks and we take the average of their preliminary outputs in deciding the final decision. Each network outputs either 1 or 0, hence the average a of the 20 networks is a value between 0 and 1. If $a > t$, where t is an experimentally obtained threshold value, then we vote for 1, otherwise we vote for 0. From performance experiments and cross validation, the threshold value we found was 0.7.

Each of the 20 neural networks were trained on a different training sets as created by Algorithm 2. In expectation there were 3 times as many negative examples as positive examples in each of the 20 training sets, but the exact number of positive examples different for each of the sets. The 4000 samples from Set A of the PhysioNet challenge were split into the positive and negative examples. There are 554 positive samples in Set A and 3446 negative samples. Each of these sets were significantly over sampled in order to generate the model with the best performance.

After training set selection, the neural networks were trained over 200 iterations on the top 30 features with a learning rate of 0.001. Through initial experimentation it was found the ideal network has 15 nodes in the first hidden layer and 15 nodes in the second hidden layer. After each of the networks are trained on their respective training sets, the model is evaluated on two separate test sets, Set B and Set C each consisting of 4000 examples. The networks vote either 1 or 0 on each sample and if the average vote is greater than t, then the model outputs 1, otherwise the model outputs 0.

Pred \ True	Death	Survival
Death	TP	FP
Survival	FN	FP

Table 1: Confusion Matrix for Scoring

5 Results

The quorum model had a testing accuracy of 85.85% on the test set (set-C from the PhysioNet Challenge). More importantly, the model performed well with respect to sensitivity and precision. The PhysioNet 2012 challenge defined an event 1 score as a measure of model performance [7]. Table 1 describes the confusion matrix of scoring. The best classifier makes no mistakes predicting deaths or survivals and hence would have 0,s in the right diagonal (FN = 0, and FP = 0). The event 1 score as defined in [7] is:

$$s = min\left(\frac{TP}{TP+FN}, \frac{TP}{TP+FP}\right)$$

The score is the minimum of the fraction of in-hopsital deaths predicted, and the fraction of correct predictions of in-hospital deaths. This higher the event 1 score, the better the model has learned how to correctly identify patient mortality. A SAPS-1 based model yields an event 1 score of 0.29 and random guessing yields an event 1 score of 0.139 on the test set. Our model obtains an event 1 score of 0.538. We also found that in order to maximize the event 1 score, the quorum model could not just do a pure vote. Rather, it needs to hit a threshold of 70% votes for in-hospital death to predict that output. The following is a confusion matrix of our model.

$$\begin{pmatrix} 315 & 270 \\ 277 & 3138 \end{pmatrix}$$

The score of 0.538 is significantly better than the SAPS-I and SAPS-II based model which are currently used. The score of 0.538 would have also scored the highest in the PhysioNet 2012 competition (second place had a score of 0.535) suggesting the RDF encoding and quorum based networks do provide an improvement to existing models.

6 Discussions

The methods that we use in order to predict ICU deaths involve significant pre-processing of the time series data. However, the number of features actually used to train the model is fairly small. While the model does have a slightly higher event 1 score, and thus a better performance, among other models that have attempted to solve the same problem, our model does not predict a probability of mortality. Through modifications to our model we can transform it to do so. Still, the work we have done here is useful to doctors and hospital staff working in the ICU; categorizing patients based on accurate in-hospital mortality predictions can allow a better redistribution of resources.

The model can also be tailored to work with the personal information of the patient. Currently, the model uses global averages and standard deviations for age, gender, and weight groups in order to perform the RDF encoding. By replacing that with personal averages and standard deviations, the model can learn exactly what is normal for this individual patient and adjust the severity accordingly. This would allow for an improved personalized medicine based solution to ICU treatment.

There are some improvements that can be made to make the model better. Currently the model takes a long time to pre-process and train. This is extremely problematic in an ICU environment, especially one with personalized data, because time is critical when working with critically ill patients. Improvements can be made in order to bring down the preprocessing time and training time so that this model can be modified to work in an ICU environment.

Finally, the quorum model of neural networks based on randomized training data is an interesting method of prediction. The randomized training data means that each neural network will potentially get a different subset of the data, and hence will learn how to predict slightly differently. The quorum model exposes neural networks to different parts of the training data so it is highly likely some percentage of the quorum will learn the target function correctly.

7 Conclusions and Future Work

In this paper we discuss how we can use a Quorum-based neural network approach with some time-series analysis in order to classify patients based on their 48 hour ICU data. Our algorithm relies on the fact that the data is interpolated by our assumptions (Algorithm 1) encoded with respect to a modified Z-score, transforming the data from raw values into a matrix of severity for each variable. We then train a quorum through randomized training set development (Algorithm 2) in order to attempt to learn the complex relationship between the first 48 hours of the ICU patient's admittance and their eventual mortality.

The model we have presented does well with respect to the models that currently attempt to solve this problem. In future work we can work to make this model real-time, taking live patient data and classifying the outcome accordingly. In doing so, we hope to work towards constructing a personalized medicine based solution that helps both hospitals and their patients.

References

[1] Halpern and Pastores. Critical care medicine beds, use, occupancy and costs in the united states: a methodological review. *Critical Care Medicine*, 43(22), November 2015.

[2] Joshi and Szolovits. Prognostic physiology: Modeling patient severity in intensive care units using radial domain folding. *AMIA Annu Symp Proc*, November 2012.

[3] Ho Vikalo Lee, Arzeno and Ghosh. An imputation-enhanced algorithm for icu mortality prediction. *Computing in Cardiology*, 39, 2012.

[4] Odstrcilık Macas, Kuzilek and Huptych. Linear bayes classification for mortality prediction. *Computing in Cardiology*, 39, 2012.

[5] Esbroeck Rubinfeld McMillan, Chia and Syed. Icu mortality prediction using time series motifs. *Computing in Cardiology*, 39, 2012.

[6] Williams Harris Martinez Pollard, Harra and Fong. 2012 physionet challenge: An artificial neural network to predict mortality in icu patients and application of solar physics analysis methods. *Computing in Cardiology*, 39, 2012.

[7] Scott Silva, Moody and Celi. Predicting mortality of patients in. intensive care: The physionet/computing in cardiology challenge 2012. *Computing in Cardiology*, 39, March 2012.

[8] Petrie Xia, Daley and Zhao. A neural network model for mortality prediction in icu. *Computing in Cardiology*, 39, 2012.