PhysioNet 2012 Challenge: Predicting Mortality of ICU Patients using a Cascaded SVM-GLM Paradigm

Luca Citi, Riccardo Barbieri

Dept. Anesthesia, Massachusetts General Hospital – Harvard Medical School, Boston, MA, U.S.A.
Dept. Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, U.S.A.

Abstract

The focus of the PhysioNet/CinC Challenge 2012 is to develop methods for patient-specific prediction of in-hospital mortality using general descriptors recorded at the time of admission to the ICU and up to 37 time-series measurements collected during the first 48 hours after admission. We developed an algorithm that uses both general descriptors and time-series measurements to predict the in-hospital death (IHD) of ICU patients in Event 1, and to provide a probability estimate of IHD in Event 2. Both aggregated variables and general descriptors were used as features of quadratic Support Vector Machine (SVM) classifiers. Six SVMs were trained using, for each one, all the positive examples plus, in turn, one sixth of the negative examples in the training set. Finally, a Generalized Linear Model with probit link was used to predict the probability of IHD for Event 2 using the raw outputs of the six SVMs as regressors. A positive binary prediction of IHD for Event 1 was made when the probability estimate was higher than an optimized threshold. Official final results of the challenge reported that our entry achieved an Event 2 score of 17.88, which is the best score out of the total 23 submissions, and Event 1 score of 0.5345 (second best score).

1. Introduction

In-hospital mortality is known to depend on the patient's demographics and on the presenting conditions, as well as on easily and routinely performed tests and measurements recorded in the first hours from admission. Prediction models, such as the APACHE IV [1], SAPS 3 [2], and MPM0 III [3] scores, have been developed in the last three decades, primarily to compare the efficacy of medications, guidelines and protocols, on a population basis.

The focus of the 2012 Computing in Cardiology (CinC) challenge is to stimulate the development of methods for patient-specific prediction of in-hospital death (IHD) of intensive care unit (ICU) patients [4]. While state-of-the-art scores emphasize simple calculations based on a sparse number of common ICU observations, the challenge dataset is made of a rich set of variables and no restriction has been imposed on the complexity of the prediction algorithm that the participants can implement. This seems very reasonable in an era of digital information where computers have proven to be excellent tools for discovering patterns and extracting information from large data sets.

2. Methods

2.1. Description of the dataset

The total data used for the challenge consist of records from 12,000 ICU stays. Each record contains general descriptors recorded at the time of admission to the ICU (age, gender, weight, height, and type of ICU) and up to 37 time-series measurements (for example, the diastolic/mean/systolic arterial blood pressure and lab tests) that may be observed (never, once, or more than once) during the first 48 hours after admission. For each time series measurement, the associated time stamp indicating the time elapsed since admission, was also recorded. Two subsets, A and B, each one made of 4,000 of the 12,000 records, were available to the participants. For subset A the binary outcome of each stay was also provided, taking value 1 in case of in-hospital death and 0 if the patient survived the hospitalization. Subset C was only available to the organizers and was used to assess the final scores.

In order to keep the algorithm simple and to prevent overfitting in the choice of the final candidate, we decided to use only set A to train the algorithm. Using also set B might have improved the fitting of the normalization coefficients (see section 2.3), and also the training of the SVMs, employing semi-supervised techniques [5]. In a real-case scenario this would represent the case where the data collected during the ICU stay are available while the outcome of the hospitalization is unknown, maybe because the subject was transferred to a different ward or hospital.

2.2. Variables used

Our first decision was to use all the variables provided and let the classification algorithm deal with possibly redundant or uninformative ones. We made the following exceptions: a) by mistake, we removed the variable *MechVent* that we thought uninformative; b) we combined together the variables related to non-invasive and invasive measures of the arterial blood pressure; c) the *Weight* variable which is both a general descriptor recorded at admission and a time series, was only considered a time-series measurement; d) we introduced a new variable *CumUrine* which is the cumulative sum of the *Urine* measurements. As a result we ended up with 4 descriptors and 30 possible time-series measurements.

2.3. Normalization of variables

Although all 34 variables are non-negative, they have very different ranges and scales of values, as well as different probability distributions. For example, while the variable *Temp* has a distribution comparable to that of a Gaussian with most of the observations between 32 and 42 C, the distribution of *Urine* looks more like a log-normal and its range extends from a just a few millilitres up to several thousands (see figure 1). This can make the classification harder and less robust. In fact, in general, classifiers work best if the features have comparable ranges and, possibly, a Gaussian-like distribution. A standard approach is to linearly rescale the variables such that their ranges extend from -1 to $+1$. For heavy-tailed distributions, like the one related to *Urine*, one should take the log first and then rescale, whereas for distributions like that of *Temp* this first step is not necessary. Rather than manually deciding on a variable by variable basis, we employed an automated procedure that attempts to apply the the appropriate transformation in order to normalize any variable in the spectrum ranging from purely normal to purely log-normal. We implemented this procedure using the following steps:

- for a given variable type X (e.g., *Urine*) collect all its valid measurements from all records and obtain a series of sorted occurrences $x_1 \leq x_2 \leq \cdots \leq x_N$;
- find the empirical quantile of each occurrence as $q_i = (i - 1/2)/N$ for $1 \leq i \leq N$;
- find the indices corresponding to the 1^{st} and 99^{th} percentiles: $i_{\text{L}} = \min\{i \,|\, q_i > 0.01\}$ and $i_{\text{U}} = \max\{i \,|\, q_i < 0.99\}$;
- store the values of $x_{\text{L}} = x_{i_{\text{L}}}$ and $x_{\text{U}} = x_{i_{\text{U}}}$ for later use;
- create a matrix of regressors R where each line is made of $[1,\, x_i,\, \log(1 + x_i)]$, and a vector of targets y with elements $\Psi^{-1}(q_i)/3$ for $i_{\text{L}} \leq i \leq i_{\text{U}}$, where $\Psi(\cdot)$ is the CDF of a standard Gaussian distribution;
- find the vector, w, of weighting coefficients minimizing the mean square error: $w = (R^{\text{T}}R)^{-1}(R^{\text{T}}y)$.

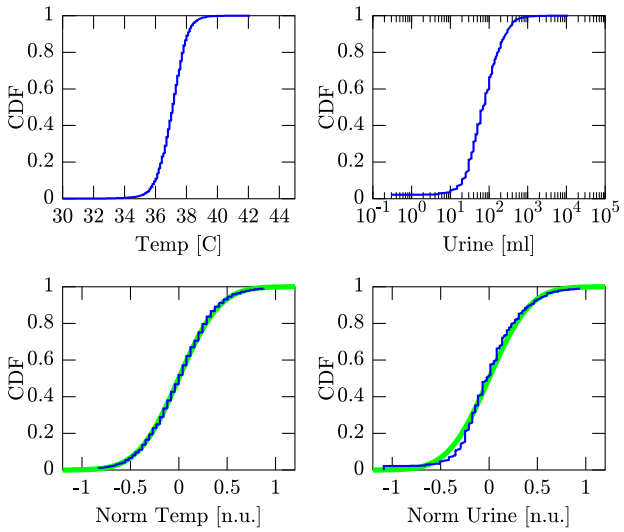

Figure 1. This figure helps explain the motivation and the effects of the normalization step described in section 2.3. The top plots show the empirical CDF of two variables, *Temp* on the left and *Urine* on the right, that have very different ranges and probability distributions (please note that the top right plot uses a logarithmic scale for the abscissas). The CDF of *Temp* was based on approximately 86,000 measurements available in set A while that of *Urine* was based on approximately 140,000. The bottom plots show how, after the variable normalization step, the transformed variables (blue lines) are much closer to a Gaussian distribution (green thick lines) and have similar ranges.

We used this procedure on the data from the training set to find the optimal weighting coefficients for all variables. We then used those coefficients to transform the variables both during the training and the test phase, according to the following steps:

- for any new measurement x of a given variable type, retrieve the corresponding x_{L}, x_{U}, and w;
- clip x to the range $[x_{\text{L}}, x_{\text{U}}]$ (this reduces the effect of possible outliers and constrains the transformed value approximately to the range $[-1, +1]$)
- normalize the new measurement using the transformation $z = w_1 + w_2\, x + w_3\, \log(1 + x)$.

As clearly shown in figure 1, the algorithm is able to automatically normalize the cumulative distribution function (CDF) of both example variables.

2.4. Feature creation

We used three of the normalized descriptors, *Age*, *Gender* and *Height*, directly as features for the classifier. As the fourth descriptor, *ICUType*, is categorical (i.e., it takes one of four values: Coronary Care Unit, Cardiac Surgery

Recovery Unit, Medical ICU, or Surgical ICU) rather than imposing an unnatural ordering, we split it into four corresponding binary variables of which one and only one is non-zero for each record.

The time-series measurements were measured a different number of times for each record, ranging form zero to a few tens. For this reason, we were confronted with the challenging task of finding a fixed number of informative features able to parsimoniously capture the distribution and possible trends of a variable number of points of the time-series. The approach that we took was to split the 48 hours of observation into two periods of 24 hours. For each period we computed the minimum, the mean and the maximum value assumed in any of the measurements. This way, the information about the average value of the measurements, but also about the variability between measurements and even possible trends (indicating whether the subject's conditions during the second 24 h are improving or deteriorating, w.r.t. the first 24 h) were made available to the following classification step.

2.5. Missing variables

As not all the descriptors and time-series were available for all records, we had to deal with the problem of missing values. If one variable (either a descriptor or a time-series) was never recorded for a given record, we used the approach called "imputation" and replaced its feature/s with value zero. Because of the normalization step, this approximately corresponds to replacing the missing raw variable with a measure of central tendency, which corresponds to the arithmetic mean for Gaussian-distributed variables and to the geometric mean for log-normal ones. In some cases, the time-series measurement were taken only in the first 24 h or only during the next 24 h. In this case, replacing with zero all the features related to the period with missing measurements could possibly create a non-existing improvement or deterioration trend. Instead, we duplicated the values from the available period, assuming stationarity conditions as default in absence of further measurements.

2.6. Classification

For the classification stage, we used support vector machines (SVMs) [6, 7] because of their robustness to noisy data and their excellent ability to deal with large datasets, i.e. datasets with an abundant number of possibly redundant or even uninformative features. This allowed us to obtain good performance without the need of a feature selection stage. Specifically, we used ν-SVMs [8] from the library "libsvm" [9]. Compared to the more commonly used C-SVMs, ν-SVMs have the advantage of being easier to tune (in most cases the default choice of the parameter, $\nu = 0.5$, works surprisingly well) and of presenting a statistically appealing interpretation of the regularization parameter ν (see [8]). We used a second order polynomial kernel $K(v_1, v_2) = (\gamma\, v_1^{\mathrm{T}} v_2 + 1)^2$. The main reason is that with a polynomial kernel, in case of missing values, the imputation with zero corresponds to performing the dot product $v_1^{\mathrm{T}} v_2$ in a sub-space made of the dimensions corresponding to the features that are available for both v_1 and v_2. This is not the case, for example, with a radial basis function (RBF) kernel.

As the dataset is unbalanced, i.e. the number of IHD is roughly one seventh of the total number of cases, we decided to split the set of survivors into six subsets and train six different machines. Each machine is trained on all positive examples (IHD) and one of the subsets of negative ones (survivors).[1] We explored different values of the tuning parameters ν and γ, and discovered that our machine was not too sensitive to their values. Therefore we decided to train the six machines with different parameter pairs (ν, γ) taken from the Cartesian product $\{0.52, 0.56\} \times \{10^{-1.7}, 10^{-1.3}, 10^{-0.9}\}$.

2.7. Generalized linear model

We used the classifiers in an unconventional way: instead of taking their thresholded binary outputs, we used their raw outputs to fit a probabilistic model yielding a probability estimate of the subject's IHD, as requested for Event 2 of the challenge. Specifically, we used the outputs of the six classifiers, together with a constant term, as regressors (predictor variables) in a generalized linear model (GLM) with probit link [10]. During the training phase, we fitted the model, finding a vector c of coefficients of the regressors, $y = [1, y_1, y_2, \ldots, y_6]$, according to the link function $\Psi^{-1}(\mathrm{E}[\mathrm{IHD}]) = c^{\mathrm{T}} y$ where $\Psi(\cdot)$ is the CDF of a standard Gaussian distribution and $\mathrm{E}[\cdot]$ is the expected value operator. During the test phase, the probability estimate of the subject's IHD is simply $\Psi(c^{\mathrm{T}} y)$.

We decided to sort the classifiers' outputs in ascending order, $y_1 \leq y_2 \leq \cdots \leq y_6$, inside the vector y because this allows for the implementation of robust statistics of the outputs of the six classifiers. For example, the median operator corresponds to $c = [c_0, 0, 0, \frac{1}{2}, \frac{1}{2}, 0, 0]$, the simple mean to $c = [c_0, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}]$, while a trimmed mean could be something in between.

In order to provide a binary outcome as requested by Event 1 of the challenge, we simply predicted in-hospital death whenever $\Psi(c^{\mathrm{T}} y) > \theta$, where θ was an optimized threshold yielding the maximum Event 1 score on the training set (see figure 2).

[1] Here and in the following, the terms "positive" and "negative" should be intended as the sign of the classifier's output ($+1$ for IHD, -1 for survivors), which is the opposite of the meaning they have in daily language where a positive outcome of an hospitalization is, of course, when the patient survives.

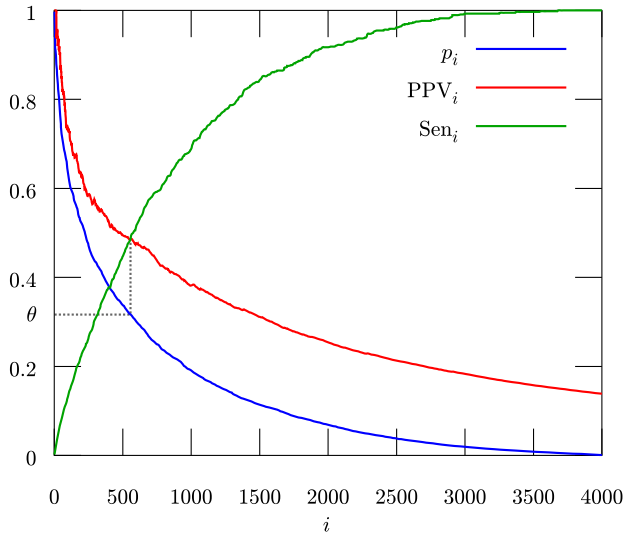

Figure 2. The figure helps explain the procedure used to find the optimal threshold that we used to produce the Event 1 output, given the Event 2 output. The blue line shows the set of Event 2 outputs for the training set (set A) sorted in decreasing order. For each value, we determine what the positive predictive value (PPV=TP/(TP+FP), red line) and what the sensitivity would be (Sen=TP/(TP+FN), green line) if that value of Event 2 output were chosen as threshold. We then select the value that maximizes the Event 1 score, i.e., the minimum between PPV and Sen (please note that this value is obtained for i approximately equal to the number of positives in the training set: 554).

3. Results

We submitted ten algorithms for phases one and two of the challenge. At the end of phase two, the organizers provided the results on set B of all entries from all participants. This algorithm scored second for Event 2, with a score of 13.24. A different entry of ours (not presented here but with the same backbone as this one) obtained a second place on Event 1 with a score of 0.5270.

We selected the algorithm presented here as entry for the actual competition. It was tested on set C by the organizers achieving, according to the official final scores, an Event 2 score of 17.88, which is the best score of all submissions (23 official and 39 in total), and an Event 1 score of 0.5345 (second place, just 0.0008 below the best score).

4. Conclusions

In this paper, we have presented our submission to the 2012 CinC challenge, with the goal of predicting subject-specific in-hospital death of ICU patients. Final results ranked our algorithm first in Event 2 and second in Event 1.

The good performance of our algorithm demonstrates that a sound analysis of the probabilistic structure of the data, combined with robust machine learning techniques and a GLM framework, is a successful strategy to yield accurate predictions in terms of probability estimates of the subject's IHD. As such, this paradigm provides a solid base for developing a computational tool to be used in clinical settings in order to offer patient-specific critical information to medical staff and and guide their supervision activities, therapeutic actions, and life-support interventions.

References

[1] Zimmerman JE, Kramer AA, McNair DS, Malila FM. Acute physiology and chronic health evaluation (APACHE) IV: hospital mortality assessment for today's critically ill patients. Crit Care Med May 2006;34(5):1297–1310.

[2] Moreno RP, Metnitz PGH, Almeida E, Jordan B, Bauer P, Campos RA, Iapichino G, Edbrooke D, Capuzzo M, Gall JRL. SAPS 3–from evaluation of the patient to evaluation of the intensive care unit. part 2: Development of a prognostic model for hospital mortality at ICU admission. Intensive Care Med Oct 2005;31(10):1345–1355.

[3] Higgins TL, Teres D, Copes WS, Nathanson BH, Stark M, Kramer AA. Assessing contemporary intensive care unit outcome: an updated mortality probability admission model (MPM0-III). Crit Care Med 2007;35(3):827–835.

[4] Silva I, Moody GB, Scott DJ, Celi LA, Mark RG. Predicting mortality of patients in intensive care: The PhysioNet/Computing in Cardiology challenge 2012. In Computing in Cardiology. Krakow, 2012; .

[5] Chapelle O, Schölkopf B, Zien A (eds.). Semi-Supervised Learning. Cambridge, MA: MIT Press, 2006.

[6] Vapnik V. The Nature of Statistical Learning Theory. 2 edition. Springer, 2000.

[7] Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000.

[8] Schölkopf B, Smola A, Williamson R, Bartlett P. New support vector algorithms. Neural computation 2000; 12(5):1207–1245.

[9] Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2011;2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[10] Aldrich J, Nelson F. Linear probability, logit, and probit models. Sage Publications, 1984.

Address for correspondence:

Luca Citi
Dept of Anesthesia, Critical Care, and Pain Medicine
Massachusetts General Hospital
55 Fruit Street, Jackson 4
Boston, MA, 02114, U.S.A.
E-mail: lciti@neurostat.mit.edu , lciti@ieee.org