Learning Score Systems for Patient Mortality Prediction in Intensive Care Units via Orthogonal Matching Pursuit

Aadirupa Saha*, Chandrahas Dewangan†§, Harikrishna Narasimhan*, Sriram Sampath‡, Shivani Agarwal*
*Department of Computer Science and Automation, Indian Institute of Science, Bangalore–560012, India
Email: {aadirupa.saha, harikrishna, shivani}@csa.iisc.ernet.in
†Veveo India Pvt. Ltd., Bangalore–560008, India
Email: dewangan.chandrahas@gmail.com
‡Department of Critical Care Medicine, St. John's Medical College Hospital, Bangalore–560034, India
Email: sriram.sampath123@gmail.com

***Abstract*—The problem of predicting outcome of patients in intensive care units (ICUs) is of great importance in critical care medicine, and has wide implications for quality control in ICUs. A dominant approach to this problem has been to use an ICU score system such as, for example, the Acute Physiology and Chronic Health Evaluation (APACHE) system, and the Simplified Acute Physiology Score (SAPS) system, to compute a certain severity score for a patient from a set of clinical observations, and apply a logistic regression model on this score to obtain an estimate of the probability of mortality for the patient; owing to their simplicity, these methods are widely used by clinicians. However, existing ICU score systems are built from a fixed set of patient data, and often perform poorly when applied to a patient population with different characteristics; also, with changes in patient characteristics, a score system built from a given patient data set becomes suboptimal over time. Moreover, most of these score systems are built using semi-automated procedures that require some amount of manual intervention, making it difficult to adapt them to a new patient population.**

Thus there is a huge need for adaptive methods that can automatically learn predictive models from a given set of patient data, tailored to perform well on similar patient populations. Indeed, there has been much work in recent years on applying various machine learning methods to this problem; however these methods learn different representations from the score systems preferred by clinicians. In this work, we develop a machine learning method based on orthogonal matching pursuit that automatically learns a score system type model, which enjoys the benefits of both worlds: like other machine learning methods, it is adaptive; like standard score systems, it uses a representation that is easy for clinicians to understand. Experiments on real-world patient data sets show that our method outperforms standard ICU score systems, and performs at least as well as other machine learning methods that employ more complex representations. As an added advantage of using the OMP approach, one can use a group-sparse variant of OMP which allows learning models with similar performance using a smaller number of clinical observations; we include experiments with this as well.

Keywords- Intensive care units, mortality rate prediction, score systems, orthogonal matching pursuit, logistic regression

I. Introduction

The problem of predicting mortality of patients in intensive care units (ICUs) has been extensively studied in critical care medicine for several decades, and has wide implications in monitoring the quality of care provided in ICUs, and in comparing ICUs across different demographics [9], [27].

A widely used approach to this problem are methods based on ICU score systems that use a simple score lookup table to compute a certain severity score for a patient from a set of clinical observations, and apply a logistic regression model on this score to obtain an estimate of the probability of mortality for the patient (see Figure 1 for an example of a score table); popular ICU score systems include the Acute Physiology and Chronic Health Evaluation (APACHE) [10]–[12], [28], and the Simplified Acute Physiology Score (SAPS) systems [15], [21]. Owing to their simplicity and ease of use, score systems are very popular among clinicians in practice.

However, existing ICU score systems, which are designed from a fixed set of patient data, are often found to perform poorly when used to make predictions for a patient population with different characteristics [2], [13], [23], [26]; for example, score systems built from western patient data are known to perform poorly on Indian patients [22], [25]. Also, due to changes in patient characteristics, a score system built using a given patient data set becomes suboptimal over time [9], [27]. Moreover, the score tables prescribed in these systems are often designed either manually using domain expertise [6], [10], [12], or by using a semi-automatic procedure that requires a certain amount of manual intervention [11], [14], making it difficult to adapt these systems to a new patient population.

There is thus an enormous need for adaptive methods that can automatically learn predictive models from a given set of patient data, tailored to perform well on similar patient populations [13], [18]. While there has been a lot of work in recent years in applying several machine learning methods for this problem, ranging from logistic regression to support vector machines to neural networks [3], [18], [20], these methods learn representations that are different from the score systems preferred by clinicians.

In this work, we develop a machine learning method that automatically learns a score system type model from given patient data, and thus enjoys the benefits of both worlds: like other machine learning methods, it is adaptive; like standard score systems, it uses a representation that is easy for clinicians to understand. Our method uses a variant of the orthogonal matching pursuit algorithm for the logistic loss [17] to learn a score system model that is a weighted sum of indicator functions defined in terms of a sparse set of feature-threshold pairs; the final mortality rate estimation model is obtained by applying a sigmoid transformation to the learned score model.

§Work done while at the Indian Institute of Science, Bangalore.

Variables	SOFA Score 0	1	2	3	4
Respiratory Pao_2/FIo_2, mm Hg	>400	≤400	≤300	≤200†	≤100†
Coagulation Platelets $\times 10^3/\mu L$‡	>150	≤150	≤100	≤50	≤20
Liver Bilirubin, mg/dL‡	<1.2	1.2-1.9	2.0-5.9	6.0-11.9	>12.0
Cardiovascular Hypotension	No hypotension	Mean arterial pressure <70 mm Hg	Dop ≤5 or dob (any dose)§	Dop >5, epi ≤0.1, or norepi ≤0.1§	Dop >15, epi >0.1, or norepi >0.1§
Central nervous system Glasgow Coma Score Scale	15	13-14	10-12	6-9	<6
Renal Creatinine, mg/dL or urine output, mL/d‖	<1.2	1.2-1.9	2.0-3.4	3.5-4.9 or <500	>5.0 or <200

Figure 1: Snapshot of the score lookup table used in the Sequential Organ Failure Assessment (SOFA) ICU score system [6]. Each entry in the table contains a feature interval, with the row indicating the corresponding patient feature, and the column indicating the score assigned to the interval.

Our experiments on real-world patient data sets obtained from different populations reveal that the proposed method performs significantly better than standard ICU score systems when trained with the same set of features used by these systems; we also show that our method often performs better than other machine learning methods for this problem that employ more complex representations. Using a variant of our method for learning score system models using a limited number of features, we are also able to obtain score systems that perform comparable to or better than standard score systems using only a subset of the features prescribed by them.

A. Related Work

There has been much work in the past several decades in designing ICU score systems for evaluating the severity of illness in patients in ICU, and in using these scores to predict patient mortality rates [4], [6], [7], [10], [11], [14], [15], [19]. One of the most popular among these is the Acute Physiology and Chronic Health Evaluation (APACHE) score system [10]–[12], [28], which over the years has evolved from an initial version [12] in 1981 that used a simple score model involving 34 clinical variables, to the current version (APACHE-IV) [28] which uses a more complex model over a larger set of 142 clinical variables; other popular score systems include the Simplified Acute Physiology Score (SAPS) series [15], [21], and the Mortality Probability Model (MPM) series [7], [16].

These score systems primarily differ in the clinical variables used, which can range from routine physiological variables to disease-specific variables [27], and the methodology used to design the score computation model, which can vary from a manual procedure based on domain expertise [6], [10], [12], to one based on statistical analysis of data [7], [21], [28]. There are also other related ICU score systems which primarily aim to measure the severity of dysfunction of a patient's organ systems, such as for example, the Sequential Organ Failure Assessment (SOFA) [6], Logistic Organ Dysfunction (LOD) [14], and Multiple Organ Dysfunction (MOD) [19], but which nevertheless are useful for predicting patient mortality rates.

Paper Organization. We start with some preliminaries and background on ICU score systems in Section II. We describe our OMP based method for learning score systems in Section III. We present experimental evaluations of the proposed method in Section IV.

II. Preliminaries and Background

We now give our problem setup and some background on score systems used for mortality rate prediction in ICUs.

A. Problem Setup and Notations

We consider a binary class probability estimation problem where there is an instance space $\mathcal{X} \subseteq \mathbb{R}^d$ consisting of patient instances represented by d clinical observations or features, and a label space $\mathcal{Y} = \{\pm 1\}$, where 1 denotes that a patient died in ICU and -1 denotes that he survived in ICU. We are given a training sample $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)) \in (\mathcal{X} \times \{\pm 1\})^N$ consisting of observations for N patients along with their survival outcomes. Assuming as usual an underlying (unknown) distribution D over $\mathcal{X} \times \{\pm 1\}$ from which examples are drawn with the corresponding conditional probability distribution $\eta(\mathbf{x}) = \mathbf{P}(y = 1 \,|\, \mathbf{x})$, our goal is to then learn from the training sample S a model $\widehat{\eta}_S : \mathcal{X} \to [0, 1]$ that estimates the probability of mortality for a new patient; often, $\widehat{\eta}_S$ is constructed by applying a suitable transformation to a real-valued function.

Notations. We use x_{ij} to denote the j^{th} feature of patient instance $\mathbf{x}_i$ in S, and x_j to denote the j^{th} feature of a new instance $\mathbf{x} \in \mathcal{X}$. For any $r \in \mathbb{N}$, we denote $[r] = \{1, \ldots, r\}$. $\mathbf{1}(\cdot)$ denotes the indicator function, where $\mathbf{1}(\phi)$ is 1 if the predicate ϕ is true and 0 otherwise.

B. Score Systems for ICU Mortality Rate Prediction

As mentioned earlier, a popular approach for mortality rate prediction is to use a score system to predict a severity score for a patient, and learn a logistic regression model to convert this score into an estimate of the patient's mortality. An ICU score system typically comes with a score lookup table, designed apriori using a fixed patient data set, that divides each patient feature into a certain number of intervals, and assigns a score to each interval; the severity score for a patient is calculated as a sum of scores accumulated over different features. More formally, for a given patient instance $\mathbf{x} \in \mathcal{X}$, a score system computes a real-valued severity score using:

$$f_{\text{severity}}(\mathbf{x}) = \sum_{j=1}^{d} \sum_{k=1}^{m_j} \alpha_k^j \mathbf{1}\big(x_j \in (a_k^j, a_{k+1}^j]\big), \qquad (1)$$

where $(a^j_1, a^j_2], \ldots, (a^j_{m_j}, a^j_{m_j+1}]$ are m_j feature intervals corresponding to patient feature j, and $\alpha^j_1, \ldots, \alpha^j_{m_j} \in \mathbb{R}$ are the corresponding scores; for example, in the score lookup table for the SOFA ICU score system given in Figure 1, one can see that the feature values for the feature in the first row (ratio of partial pressure arterial oxygen and fraction of inspired oxygen or $\text{Pao}_2/\text{Fio}_2$) are divided into five intervals: $(0, 100], (100, 200], (200, 300], (300, 400], (400, \infty]$, which are assigned scores $4, 3, 2, 1, 0$ respectively. The final mortality rate estimation model is obtained using a training sample S by fitting a logistic regression model to the severity scores predicted as above using the given score table:

$$\widehat{\eta}_S(\mathbf{x}) = \sigma_{\text{sigmoid}}(c_S f_{\text{severity}}(\mathbf{x}) + d_S),$$

where $\sigma_{\text{sigmoid}}(s) = \frac{1}{1+e^{-s}}$ is the sigmoid function, and parameters $c_S, d_S \in \mathbb{R}$ are learned using training sample S.

Different ICU score systems differ in the choice of patient features used, and the methodology used to design the score lookup table. For example, the APACHE II ICU score system uses several physiological variables, the age of the patient, and certain features derived from the patient's past health record; the score lookup table here is designed manually based on clinical assessments and known physiological relationships between various variables [10]. On the other hand, the LOD ICU score system uses features that measure the severity of dysfunction of various organ systems, with the score table designed using a multi-stage logistic regression analysis on a given patient data set [14]. Indeed, most of these score systems come with static score tables designed from a fixed patient data set using procedures that are at best semi-automatic, and are hence difficult to apply on a new patient data set; clearly, there is a need for adaptive methods that can automatically learn a score system from a given patient data set.

III. Learning Score System Type Models for ICU Mortality Rate Prediction

We now describe a new machine learning method that can automatically learn score system type models from a given patient data set. Our approach makes use of an orthogonal matching pursuit based technique to learn a model that is defined in terms of a sparse set of feature-threshold pairs.

We begin by defining for each feature $j \in [d]$, a set of m_j user-specified thresholds $\Theta_j = \{\theta_{j,1}, \theta_{j,2}, ..., \theta_{j,m_j}\} \subset \mathbb{R}$, and the corresponding set of feature-threshold pairs $\mathcal{P} = \{(j, \theta) \,|\, j \in [d],\, \theta \in \Theta_j\}$. We then rewrite the score system model in Eq. (1) as an equivalent weighted sum of indicator functions defined in terms of a subset of feature-threshold pairs in $\mathcal{P}$, instead of feature intervals:

$$f(\mathbf{x}) = \sum_{\tau=1}^{t} \alpha_\tau \mathbf{1}(x_{j_\tau} \leq \theta_\tau), \tag{2}$$

where $1 \leq t \leq |\mathcal{P}|$, $\boldsymbol{\alpha}_{1:t} = [\alpha_1, \ldots, \alpha_t]^\top \in \mathbb{R}^t$, and $\{(j_\tau, \theta_\tau)\}_{\tau=1}^t \subseteq \mathcal{P}$; let $\mathcal{F}_{\text{SS}}$ be the set of all such score system models. Our goal is to learn a model in $\mathcal{F}_{\text{SS}}$ that minimizes the mortality rate estimation error on the training sample measured in terms of the logistic loss:

$$\widehat{f}_S \in \operatorname*{argmin}_{f \in \mathcal{F}_{\text{SS}}} \sum_{i=1}^{N} \log(1 + \exp(-y_i f(\mathbf{x}_i)), \tag{3}$$

Algorithm 1 LogitOMP-SS

1: **Input:**
2: $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)) \in (\mathcal{X} \times \{\pm 1\})^N$
3: $\mathcal{P} = \{(j, \theta) \,|\, j \in [d],\, \theta \in \Theta_j\}$
4: Regularization parameter $\lambda > 0$
5: Precision parameter $\epsilon > 0$ for stopping criterion
6: **Define:**
7: $\mathbf{b}(j, \theta) \equiv \big[\mathbf{1}(x_{1,j} \leq \theta), ..., \mathbf{1}(x_{N,j} \leq \theta)\big]^\top, \forall (j, \theta) \in \mathcal{P}$
8: **Initialize:**
9: $\mathcal{I}_0 = \phi$
10: $\eta_0(i) = \frac{1}{2}$, $\forall i \in [N]$
11: $t = 1$
12: **Loop**
13: $r_t(i) = \eta_{t-1}(i) - \mathbf{1}(y_i = 1)$, $\forall i \in [N]$
14: $(j_t, \theta_t) = \operatorname*{argmax}_{(j,\theta) \in \mathcal{P} \setminus \mathcal{I}_{t-1}} \frac{|\mathbf{b}(j,\theta)^\top \mathbf{r}_t|}{||\mathbf{b}(j,\theta)||_2}$
15: **If** $\left(\frac{|\mathbf{b}(j_t,\theta_t)^\top \mathbf{r}_t|}{||\mathbf{b}(j_t,\theta_t)||_2} \leq \epsilon\right)$
16: **Break**
17: **End If**
18: $\mathcal{I}_t = \mathcal{I}_{t-1} \cup \{(j_t, \theta_t)\}$
19: $(\widehat{\boldsymbol{\alpha}}_{1:t}, \widehat{\beta}) =$
20: $\operatorname*{argmin}_{\boldsymbol{\alpha}_{1:t} \in \mathbb{R}^t, \beta \in \mathbb{R}} \sum_{i=1}^N \log\left(1 + e^{-y_i\left(f_{\boldsymbol{\alpha}_{1:t}}(\mathbf{x}_i) + \beta\right)}\right) + \frac{\lambda}{2}\|\boldsymbol{\alpha}_{1:t}\|_2^2,$
21: where $f_{\boldsymbol{\alpha}_{1:t}}(\mathbf{x}) = \sum_{\tau=1}^t \alpha_\tau \mathbf{1}\big(x_{j_\tau} \leq \theta_\tau\big)$
22: $\eta_t(i) = \sigma_{\text{sigmoid}}\big(\sum_{\tau=1}^t \widehat{\alpha}_\tau \mathbf{1}(x_{i,j_\tau} \leq \theta_\tau) + \widehat{\beta}\big)$, $\forall i \in [N]$
23: $t = t + 1$
24: **End Loop**
25: **Output:** $\widehat{\eta}_S(\mathbf{x}) = \sigma_{\text{sigmoid}}\big(\sum_{\tau=1}^{t-1} \widehat{\alpha}_\tau \mathbf{1}(x_{j_\tau} \leq \theta_\tau) + \widehat{\beta}\big)$

where the final mortality rate estimation model is given by $\widehat{\eta}_S(\mathbf{x}) = \sigma_{\text{sigmoid}}\big(\widehat{f}_S(\mathbf{x})\big)$.

Solving the above optimization problem exactly can be computationally expensive when the number of feature-threshold pairs in $\mathcal{P}$ is large. Moreover, in practice, one would find it convenient to use score system models that involve a small set of feature-threshold pairs. We therefore pose the above problem as a sparse learning problem over a 'blown-up' feature space containing a boolean feature defined in terms of the indicator function in Eq. (2) for each feature-threshold pair in $\mathcal{P}$; we solve this problem using a variant of the (greedy) orthogonal matching pursuit (OMP) algorithm for optimizing the logistic loss [17] (see Algorithm 1)[1].

Before describing our algorithm, we shall find it convenient to define for each feature-threshold pair $(j, \theta) \in \mathcal{P}$, the boolean vector $\mathbf{b}(j, \theta) = \big[\mathbf{1}(x_{1,j} \leq \theta), ..., \mathbf{1}(x_{N,j} \leq \theta)\big]^\top \in \{0, 1\}^N$. At each iteration t, the proposed algorithm (termed LogitOMP-SS) computes a residual vector $\mathbf{r}_t \in \mathbb{R}^N$ containing differences between the mortality rates predicted by the current model and the true outcomes for each patient in S (line 13), and selects the feature-threshold pair (j_t, θ_t) that best describes this residual difference, and more specifically, for which the corresponding boolean vector $\mathbf{b}(j_t, \theta_t)$ yields the maximum projection onto the residual vector (line 14); the scores $\boldsymbol{\alpha}_{1:t}$ for the t feature-threshold pairs chosen till now (along with a bias term $\widehat{\beta}$) are then refitted by solving a (regularized)

[1] We use a slight variant of the logistic orthogonal matching pursuit algorithm in [17] that incorporates regularization in the model refitting step.

logistic regression optimization problem over S (line 20). The algorithm terminates when the maximum projection value computed in line 14 falls below a precision ϵ (line 15), with the final mortality rate estimation model given by a sigmoid transformation on the learned score system model (line 24).

The above algorithm imposes no restriction on the number of patient features used to construct a score system model. However, it is often desirable in practice to use models that yield good prediction accuracy with a small number of clinical observations [18]. We therefore also consider using a 'group' sparse OMP variant of this algorithm [17] (referred to as LogitGOMP-SS) that allows us to learn score systems using a limited number of features. In particular, instead of having our algorithm choose a single feature-threshold pair in each iteration, we now allow our method to operate on groups of feature-threshold pairs, with each group containing all feature-threshold pairs corresponding to a single feature. The modified algorithm aims to choose a sparse set of groups, and thus a small set of features, with the projection computation step for a group now involving computing the ℓ_2-norm of a vector of projection values of each feature-threshold pair in the group; the algorithm is terminated when the desired number of features are chosen.

In the above algorithms, the set of thresholds Θ_j for each feature j can be chosen by clustering the feature values in the training set into m_j clusters; the thresholds $\theta_{j,1}, \ldots, \theta_{j,m_j}$ can then be set respectively to the highest values in these clusters.[2]

IV. Experimental Results

We now present experimental comparison of the proposed OMP based method (LogitOMP-SS) with standard ICU score systems and several other machine learning methods used for ICU mortality prediction. We also provide results on the group OMP variant of our method (LogitGOMP-SS) for learning score systems with limited number of features.

We applied the above methods on two sets of real-world ICU patient data pertaining to two different populations: the first data was a de-identified data set from St John's Medical College Hospital, Bangalore, collected from 2006 to 2014, and contained 29 clinical observations for 3499 patients[3]; the second data, provided as a part of the CinC Challenge 2012 [1], was a subset of MIMICS-II database [24] collected at a tertiary teaching hospital in Boston from 2001 to 2008, and contained 42 observations for 4000 patients.

We compared the proposed LogitOMP-SS method with four standard ICU score systems, namely, APACHE-II [10], SAPS-II [15], LOD [14] and SOFA [6]. The St. John's patient data contained the relevant clinical observations for the APACHE II and LOD score systems, while the CinC data contained the relevant clinical observations for the APACHE II, SAPS II, and LOD score systems.[4] In each case, we evaluated our method on the same set of features used by the score system that we compared with, yielding five versions of the above data sets (see Table I).[5] We also included for comparison three baseline machine learning methods: regularized linear logistic regression [3], regularized kernel logistic regression (with RBF kernel), and kernel RankSVM (RBF kernel) with the output scores calibrated using Platt scaling [8], [18] (each applied to the entire set of features).

Table I: ICU data sets used.

Data set	# patients	# features
St-John's-APACHE-II	3499	27
St-John's-LOD	3499	23
CinC-APACHE-II	4000	27
CinC-SAPS-II	4000	29
CinC-SOFA	4000	15

We measured the performance of the mortality rate estimation model $\widehat{\eta} : \mathcal{X} \rightarrow [0,1]$ learned by the different methods using four standard evaluation metrics: the area under the ROC curve (AUC) used widely for ICU mortality rate estimation problems, the logistic loss (LogLoss) and Brier score (BS) for measuring mortality rate estimation error of the learned model, and the (range normalized) HosmerLemeshow (HL) statistic for measuring the calibration performance of the learned model [1]. For a set of test examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, the first three performance measures are given by

$$\text{AUC} = \frac{1}{n^+ n^-} \sum_{i:y_i=1} \sum_{j:y_j=-1} \Big[\mathbf{1}\big(\widehat{\eta}(\mathbf{x}_i) > \widehat{\eta}(\mathbf{x}_j)\big) + \tfrac{1}{2}\mathbf{1}\big(\widehat{\eta}(\mathbf{x}_i) = \widehat{\eta}(\mathbf{x}_j)\big) \Big];$$

$$\text{LogLoss} = \frac{1}{n} \sum_{i=1}^{n} \Big[\mathbf{1}(y_i = 1) \log(\widehat{\eta}(\mathbf{x}_i)) + \mathbf{1}(y_i = -1) \log(1 - \widehat{\eta}(\mathbf{x}_i)) \Big];$$

$$\text{BS} = \frac{1}{n} \sum_{i=1}^{n} \big(\eta(\mathbf{x}_i) - \mathbf{1}(y_i = 1)\big)^2,$$

where $n^+ = \sum_{i=1}^{n} \mathbf{1}(y_i = 1)$ is the number of positive test examples, and $n^- = n - n^+$ is the number of negative test examples. For calculating the HL statistic measure [5], we divide the probability range $[0,1]$ into g equi-frequency bins (obtained by dividing the patients into g quantiles, sorted according to $\widehat{\eta}$), and for each bin k, count the number of patient examples o_k assigned to the bin that turn out to be positive, and compute the average probability π_k predicted by $\widehat{\eta}$ for patients in the bin; the (range normalized) HL statistic for $\widehat{\eta}$ is then given by

$$\text{HL} = \frac{1}{\pi_{\max} - \pi_{\min}} \sum_{k=1}^{g} \frac{(o_k - n_k \pi_k)^2}{n_k \pi_k (1 - \pi_k)},$$

where $\pi_{\max} = \max_{1 \leq k \leq g} \pi_k$, $\pi_{\min} = \min_{1 \leq k \leq g} \pi_k$, and n_k is the number of patients assigned to bin k; in our experimental evaluations, g was set to 10. Note that for the AUC, higher values imply better performance, while for the other three performance measures, lower values are better.

[2] In our experiments, the number of thresholds m_j was set to the same value for all features, which was tuned using a validation set.

[3] This study had received clearance from the Institutional Ethical Review Board of the hospital.

[4] The patient data available to us did not contain clinical observations for the organ insufficiency, chronic diseases and vasopressor attributes in the APACHE-II, SAPS-II and SOFA score systems respectively; while applying these score systems to patient instances, as prescribed in these systems, the missing features were assigned a (normal) score of 0.

[5] The features used in these data sets correspond to the worst (high and low) values of clinical observations recorded within 24 hours of admission of each patient to ICU, as recommended by the score systems used.

All results reported here were averaged over 10 random 2:1 train-test splits of the given data sets. The tunable parameters for the different methods were chosen using a validation set consisting of a part of the training set, with the parameter yielding the highest AUC chosen in each case.[6]

A. Comparison of LogitOMP-SS with APACHE-II system

We present results comparing the APACHE-II ICU score system with the one learned by LogitOMP-SS using the same set of features used in APACHE-II. Table II and III contain results respectively on the St-John's-APACHE-II and CinC-APACHE-II data sets; in each case, the first and second best entries in a column are written in bold face. As can be seen, on both data sets, LogitOMP-SS performs significantly better than APACHE-II on almost all evaluation measures, indicating that score systems that are learned from a given patient data set do indeed perform well on patients from the same population, and clearly perform better than a static score system that is designed for a different population. In fact, the proposed method also outperforms all the other machine learning methods on the second data set, while in most cases, performs comparable to or better than the other machine learning methods on the first data set.

	AUC	LogLoss	BS	HL
LogitOMP-SS	0.7047	0.4925	0.1599	**25.15**
APACHE-II	0.6607	0.5112	0.1673	37.97
Linear Logistic Regression	0.7047	**0.4907**	**0.1593**	**22.40**
Kernel Logistic Regression	**0.7069**	0.5046	**0.1582**	30.85
RankSVM + Platt Scaling	**0.7067**	**0.4914**	0.1597	34.23

Table II: Comparison of LogitOMP-SS with APACHE-II score system and other ML methods on St-John's-APACHE-II data.

	AUC	LogLoss	BS	HL
LogitOMP-SS	**0.9377**	**0.2115**	**0.0650**	**16.15**
APACHE-II	0.9106	0.2457	0.0765	**14.12**
Linear Logistic Regression	0.9143	0.2440	0.0732	50.53
Kernel Logistic Regression	**0.9339**	**0.2207**	**0.0671**	17.49
RankSVM + Platt Scaling	0.9322	0.2283	0.0703	47.13

Table III: Comparison of LogitOMP-SS with APACHE-II score system and other ML methods on CinC-APACHE-II data

B. Comparison of LogitOMP-SS with SAPS-II system

We next present results comparing the proposed method with the SAPS-II score system. Our results on the CinC-SAPS-II data set, presented in Table IV, clearly shows that on most performance measures, the score system learned by the LogitOMP-SS method (using the same features as in SAPS-II) performs better than the SAPS-II system and other machine learning methods, with SAPS-II performing better only on the HL statistic; among the other ML methods, kernel logistic regression and RankSVM perform comparable to each other on most measures and better than linear logistic regression.

[6]In particular, we divided the training set into two parts in the ratio 3:2, used the first part for training the given algorithm with different parameter values, and the second part as a validation set to choose the best parameter. For the proposed methods, the number of thresholds for different features was set to the same value and chosen from $\{5, 10, 15, 20\}$; the set of feature thresholds were chosen by applying the k-means clustering algorithm to the feature values; and the precision parameter ϵ was set to 0.1. The regularization parameter for all methods were chosen from $\{10^{-6}, \ldots, 10^{3}\}$, while the RBF kernel width parameters in kernel logistic regression and RankSVM were chosen from $\{10^{-2}, \ldots, 10^{2}\}$.

	AUC	LogLoss	BS	HL
LogitOMP-SS	**0.9432**	**0.2034**	**0.0620**	**13.66**
SAPS-II	0.8802	0.2779	0.0860	**12.25**
Linear Logistic Regression	0.9120	0.2441	0.0732	29.55
Kernel Logistic Regression	0.9301	0.2295	**0.0688**	26.70
RankSVM + Platt Scaling	**0.9313**	**0.2288**	0.0692	51.57

Table IV: Comparison of LogitOMP-SS with SAPS-II score system and other ML methods on CinC-SAPS-II data.

C. Comparison of LogitOMP-SS with SOFA system

The next ICU score system that we consider is the SOFA system; the results on the CinC-SOFA data set are given in Table V. Here again, we find that the score system learned by LogitOMP-SS gives better performance than the one prescribed by SOFA; moreover, the proposed method beats the other machine learning methods on all evaluation measures, with RankSVM and kernel logistic regression again yielding the second best performance in most cases.

	AUC	LogLoss	BS	HL
LogitOMP-SS	**0.8667**	**0.2878**	**0.0876**	**16.40**
SOFA	0.8119	0.3254	0.0994	20.64
Linear Logistic Regression	0.8453	0.3090	0.0946	**18.14**
Kernel Logistic Regression	0.8527	0.3041	**0.0921**	28.89
RankSVM + Platt Scaling	**0.8549**	**0.3012**	0.0923	26.56

Table V: Comparison of LogitOMP-SS with SOFA score system and other ML methods on CinC-SOFA data.

D. Comparison of LogitOMP-SS with LOD system

We next apply the LogitOMP-SS method on the St-John's-LOD data set, and compare it against the LOD ICU score system. As seen in Table VI, the score system learned by LogitOMP-SS performs considerably better than the one given by LOD, and also beats the other machine learning methods on most evaluation measures, with kernel logistic regression following close suit; on the HL statistic, we find linear logistic regression performing better than all other methods.

	AUC	LogLoss	BS	HL
LogitOMP-SS	**0.7015**	**0.5006**	**0.1639**	**30.41**
LOD	0.6319	0.5262	0.1724	57.00
Linear Logistic Regression	0.6815	**0.5082**	0.1664	**21.86**
Kernel Logistic Regression	**0.6900**	0.5084	**0.1600**	36.04
RankSVM + Platt Scaling	0.6892	0.5096	0.1668	51.73

Table VI: Comparison of LogitOMP-SS with LOD score system and other ML methods on St-John's-LOD data.

E. Evaluation of LogitGOMP-SS for learning score systems with limited number of features

We now present an evaluation of the group OMP variant of the proposed method (LogitGOMP-SS) on the St-John's-APACHE-II and CinC-SAPS-II data sets for learning score systems from a limited number of features. It is clearly seen from Tables VII and VIII that the score systems produced by LogitGOMP-SS using a subset of the features in these data sets often yield performance comparable to or better than the corresponding score systems that use the entire set of features; in particular, the score system learned by LogitGOMP-SS on the CinC-SAPS-II data set with 15 features performs better than the SAPS-II system on most evaluation measures, despite using only half the number of features used by SAPS-II.

	AUC	LogLoss	BS	HL
LogitGOMP-SS ($k = 10$)	0.6395	0.5199	0.1699	**34.85**
LogitGOMP-SS ($k = 15$)	0.6515	0.5158	**0.1684**	42.52
LogitGOMP-SS ($k = 20$)	**0.6593**	**0.5127**	**0.1673**	41.14
APACHE-II	**0.6607**	**0.5112**	**0.1673**	**37.97**

Table VII: Evaluation of LogitGOMP-SS for different number of features (denoted here by k) on St-John's-APACHE-II data, and comparison with APACHE-II score system (27 features).

	AUC	LogLoss	BS	HL
LogitGOMP-SS ($k = 10$)	0.8682	0.2867	0.0872	**20.11**
LogitGOMP-SS ($k = 15$)	**0.9046**	**0.2568**	**0.0790**	31.52
LogitGOMP-SS ($k = 20$)	**0.9235**	**0.2364**	**0.0723**	55.57
SAPS-II	0.8802	0.2779	0.0860	**12.25**

Table VIII: Evaluation of LogitGOMP-SS for different number of features (denoted here by k) on CinC-SAPS-II data, and comparison with SAPS-II score system (29 features).

V. Conclusions

ICU score systems are widely used by clinicians to predict mortality rate of patients in ICUs. However, the score systems used in practice are designed from a fixed patient data set, and are often suboptimal when applied to a patient population with different characteristics; due to semi-automatic procedures often employed to design these score systems, it becomes difficult to adapt them to a new patient population. Since there is a need for adaptive methods that can automatically learn from data models of the type preferred by clinicians, we have developed an orthogonal matching pursuit based machine learning method that can learn a score system type prediction model from given patient data. Our experiments on real-world patient data sets reveal that the proposed method performs better than the standard score systems, and also in many cases outperforms other machine learning methods for this problem.

Acknowledgments

SA gratefully acknowledges support from DST, Indo-US Science and Technology Forum, and an unrestricted gift from Yahoo. HN is supported by a Google India PhD Fellowship in Machine Learning. SS acknowledges the contribution of the Medical Informatics team of the St John's Research Institute, especially Dr. Kedar A., Abhijeet W. and Prof. Tony Raj in designing and maintaining the database, and Ms. Veena and Ms. Sunitha for data collection for this study.

References

[1] Predicting mortality of ICU patients: The PhysioNet/Computing in Cardiology Challenge 2012. http://www.physionet.org/challenge/2012/.

[2] G. Apolone, G. Bertolini, R. D'Amico, G. Iapichino, A. Cattaneo, G. De Salvo, and R.M. Melotti. The performance of SAPS II in a cohort of patients admitted to 99 Italian ICUs: Results from GiViTI. *Intensive Care Medicine*, 22(12):1368–1378, 1996.

[3] D. Bera and M.M. Nayak. Mortality risk assessment for ICU patients using logistic regression. *Computing in Cardiology*, 2012.

[4] D. Cullen, J.M. Civetta, B.A. Briggs, and L.C. Ferrara. Therapeutic intervention scoring system: A method for quantitative comparison of patient care. *Critical Care Medicine*, 2(2):57–60, 1974.

[5] Hosmer D.W. and S. Lemeshow. *Applied logistic regression*. John Wiley & Sons, 2004.

[6] F.L. Ferreira, D.P. Bota, A. Bross, C. Mélot, and J.L. Vincent. Serial evaluation of the SOFA score to predict outcome in critically ill patients. *Journal of the American Medical Association*, 286(14), 2001.

[7] T. L. Higgins, D. Teres, W.S. Copes, B.H. Nathanson, M. Stark, and A.A. Kramer. Assessing contemporary intensive care unit outcome: An updated mortality probability admission model (MPM0-III). *Critical Care Medicine*, 35(3):827–835, 2007.

[8] T. Joachims. A support vector method for multivariate performance measures. In *ICML*, pages 377–384. ACM, 2005.

[9] M.T. Keegan, O. Gajic, and B. Afessa. Severity of illness scoring systems in the intensive care unit. *Critical Care Medicine*, 39(1):163–169, 2011.

[10] W.A. Knaus, E.A. Draper, D.P. Wagner, and J.E. Zimmerman. APACHE II: A severity of disease classification system. *Critical Care Medicine*, 13(10):818–829, 1985.

[11] W.A. Knaus, D.P. Wagner, E.A. Draper, J.E. Zimmerman, M. Bergner, P.G. Bastos, C.A. Sirio, D.J. Murphy, T. Lotring, and A. Damiano. The APACHE III prognostic system. Risk prediction of hospital mortality for critically ill hospitalized adults. *Chest Journal*, 100(6), 1991.

[12] W.A. Knaus, J.E. Zimmerman, D.P. Wagner, E.A. Draper, and D.E. Lawrence. APACHE-acute physiology and chronic health evaluation: A physiologically based classification system. *Critical Care Medicine*, 9(8):591–597, 1981.

[13] J. Labarère, R. Bertrand, and M.J. Fine. How to derive and validate clinical prediction models for use in intensive care medicine. *Intensive Care Medicine*, 40(4):513–527, 2014.

[14] J.R. Le Gall, J. Klar, S. Lemeshow, F. Saulnier, C. Alberti, A. Artigas, and D. Teres. The logistic organ dysfunction system: A new way to assess organ dysfunction in the intensive care unit. *Journal of the American Medical Association*, 276(10):802–810, 1996.

[15] J.R. Le Gall, S. Lemeshow, and F. Saulnier. A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study. *Journal of the American Medical Association*, 270(24):2957–2963, 1993.

[16] S. Lemeshow, D. Teres, J. Klar, J.S. Avrunin, S.H. Gehlbach, and J. Rapoport. Mortality probability models (MPM II) based on an international cohort of intensive care unit patients. *Journal of the American Medical Association*, 270(20):2478–2486, 1993.

[17] A. Lozano, G. Swirszcz, and N. Abe. Group orthogonal matching pursuit for logistic regression. In *AISTATS*, pages 452–460, 2011.

[18] O. Luaces, F. Taboada, G.M. Albaiceta, L.A. Domínguez, P. Enríquez, and A. Bahamonde. Predicting the probability of survival in intensive care unit patients from a small number of variables and training examples. *Artificial Intelligence in Medicine*, 45(1):63–76, 2009.

[19] J.C. Marshall, D.J. Cook, N.V. Christou, G.R. Bernard, C.L. Sprung, and W.J. Sibbald. Multiple organ dysfunction score: A reliable descriptor of a complex clinical outcome. *Critical Care Medicine*, 23(10), 1995.

[20] G. Meyfroidt, F. Güiza, J. Ramon, and M. Bruynooghe. Machine learning techniques to examine large patient databases. *Best Practice & Research Clinical Anaesthesiology*, 23(1):127–143, 2009.

[21] R.P. Moreno, P. Metnitz, E. Almeida, B. Jordan, P. Bauer, R.A. Campos, G. Iapichino, D. Edbrooke, M. Capuzzo, and J.R. Le Gall. 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 Medicine*, 31(10):1345–1355, 2005.

[22] A. Nimgaonkar, D.R. Karnad, S. Sudarshan, L. Ohno-Machado, and I. Kohane. Prediction of mortality in an Indian intensive care unit. *Intensive Care Medicine*, 30(2):248–253, 2004.

[23] E. Paul, M. Bailey, A. Van Lint, and V. Pilcher. Performance of APACHE III over time in Australia and New Zealand: A retrospective cohort study. *Anaesthesia and Intensive Care*, 40(6):980, 2012.

[24] M. Saeed, M. Villarroel, A.T. Reisner, G. Clifford, L. Lehman, G. Moody, T. Heldt, T. Kyaw, B. Moody, and R.G. Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[25] S. Sampath, M.P. Fay, and P. Pais. Use of the logistic organ dysfunction system to study mortality in an Indian intensive care unit. *National Medical Journal of India*, 12(6):258–260, 1999.

[26] I.J. Timm. Automatic generation of risk classification for decision support in critical care. In *ECAI/IDAMAP*, pages 38–41. Citeseer, 1998.

[27] J.L. Vincent and R. Moreno. Clinical review: Scoring systems in the critically ill. *Critical Care*, 14(2):207, 2010.

[28] J.E. Zimmerman, A.A. Kramer, D.S. McNair, and F.M. Malila. Acute physiology and chronic health evaluation (APACHE) IV: Hospital mortality assessment for today's critically ill patients. *Critical Care Medicine*, 34(5):1297–1310, 2006.