Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference

Extreme Logistic Regression: A Large Scale Learning Algorithm with Application to Prostate Cancer Mortality Prediction

Che Ngufor and **Janusz Wojtusiak** and **Andrea Hooker** and **Talha Oz** and **Jack Hadley**
George Mason University, Fairfax VA 22030
{cngufor,jwojtusi, toz, jhadley1}@ gmu.edu, apd9a@virginia.edu

Abstract

With the recent popularity of electronic medical records, enormous amount of medical data is being generated every day at an exponential rate. Machine learning methods have been shown in many studies to be capable of producing automatic medical diagnostic models such as automated prognostic models. However, many powerful machine learning algorithms such as support vector machine (SVM), Random Forest (RF) or Kernel Logistic Regression (KLR) are unbearably slow for very large datasets. This makes their use in medical research limited to small to medium scale problems. This study is motivated by an ongoing research on prostate cancer mortality prediction for a national representative of US population where the SVM and RF took several hours or days to train whereas simple linear methods such as logistic regression or linear discriminant analysis take minutes or even seconds. Because, most real-world problems are non-linear, this paper presents a large scale algorithm enabling a recently proposed least squares extreme logistic regression to learn very large datasets. The algorithm is shown on a case study of mortality prediction for men diagnosed with early stage prostate cancer to provide very fast and more accurate result than standard statistical methods.

1 Introduction

Accurate survival prediction and prognostic risk factor identification is essential to offering appropriate care for men with early stage prostate cancer. The first question that a newly diagnosed man with prostate cancer asks his physician is: “What are my survival chances?” The answer to this question is not very straight forward because survival is dependent on the inter-play of multiple factors. These factors can be roughly divided into three main categories: (1) patient specific factors (e.g age and comorbidities), (2) tumor specific factors (e.g PSA and Gleason score), (3) treatment types (e.g radiation or watchful waiting). Knowing the effect of these risk factors on mortality, the physician has to then decide whether immediate aggressive treatment is necessary or not.

The determination of the effects of the various prostate cancer risk factors on mortality is a difficult task even for the most experienced prostate cancer urologist. Despite the importance of timely and accurate prediction of prostate cancer mortality for medical decision making, it is known that physicians are very poor judges of prognostic (Chow et al. 2001). Physicians are overly optimistic in survival predictions and this action may adversely affect the quality of care offered to patients.

Machine learning on the other hand provide several indispensable tools for the intelligent analysis of medical data. Electronic medical records has been popularized in the past few years and many hospitals are equipped with inexpensive devices that collect and store data in large amounts on a daily basis. Machine learning algorithms can be used to learn these data and produce prognostic models. These models can be created as automatic such that when new data is gathered the model can update itself to learn the new data.

One draw back of machine learning however, is that the most powerful algorithms in this field are very slow to train on large datasets. KLR for example is a very powerful algorithm that is very competitive with state-of the art methods like SVM and RF. However, like most kernel learning machines, training time can be overly long due to the expensive kernel computation and parameter tuning. These slow learners have made the use of machine learning in healthcare to be limited to small to medium scale problems. This paper was motivated by such problems encountered with the training of SVM and RF in an ongoing research project on prostate cancer mortality prediction using a very large cohort study of men. These algorithms took several hours or days to train and make predictions. In particular, selecting optimal training parameters for SVM was extremely painful and time consuming. The researchers in the project had to resort to simple linear and non-optimal methods like logistic regression or linear discriminant analysis whose training took minutes or even seconds.

To address the problem of slow learning, recently an extremely fast and accurate method to train KLR was proposed in (Ngufor and Wojtusiak 2013a) based on a simple approximation of the logistic function and the extreme learning machine (ELM) (Huang et al. 2012). Instead of the the traditional KLR that uses kernels to evaluate dot products of data points in a feature space, Ngufor and Wojtusiak (2013a) proposed to explicitly build a feature space through a Single hidden Layer Feed-forward Network (SLFN) and uses it to construct an *"ELM-Kernel"* as done in ELM for training KLR. The approach was used to train iterative re-weighted least squares kernel logistic regression (IRLS-KLR) and the results obtained were comparable to SVM but with a much faster training speed. Further, based on a simple approximation of the logistic function, the authors converted IRLS-KLR into a non-iterative least-squares KLR (LS-KLR) that was found to be very fast and outperformed SVM on most of the experiments performed in the paper. Extreme learning method was then extended to LS-KLR and an even faster algorithm called least-squares extreme logistic regression (LS-ELR) was obtained.

LS-ELR can efficiently learn medium to large scale problems, however, as the number of training points increases the ELM-kernel size increases. Therefore solution by LS-ELR can be prohibitive for very large scale problems due to memory requirements and other computational complexities. This paper proposes an extension to LS-ELR enabling it to learn very large datasets. The complexity of the new algorithm depends only on the dimension of the SLFN hidden layer feature space which can be significantly less than the number of training data points.

The extended algorithm was applied to predict mortality for a large population of men from the SEER-Medicare database diagnosed with early stage prostate cancer. To appropriately account for censoring, each observation was weighted using the inverse-probability of censoring weighting (IPCW) obtained through the Nelson-Aaalen estimator of the survival function. The effect of several risk factors such as age, Elixhauser comorbidities, aggregated Clinical classification Software (CCS), tumor characteristics and initial treatment types on mortality was assessed by progressively increasing the information level of the models generated.

With respect to scalability and accuracy of the algorithm, numerical results show that the extended algorithm is very fast, accurate and compares favorably in terms of weighted-accuracy to traditional cox-proportional hazard model. Adding more detailed diagnosis information in the models resulted in increasing accuracy of the models.

Unless otherwise mentioned, the following notations will be adopted throughout the paper: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ is training data with $x \in \mathbb{R}^d$ a d-dimensional column vector and $y \in \{0, 1\}$ is the class label of x. $\mathbf{I}_n$ and $\mathbf{1}_n$ are the $n \times n$ identity matrix and $n \times 1$ column vector of ones respectively.

The paper is hence organized as follows: Section 2 briefly reviews IRLS-KLR and its approximation to obtain LS-KLR. The extension of ELM to LS-KLR is briefly described in Section 3, and the section ends by presenting the proposed large scale learning algorithm. Section 4 describes the experimental set-up for the case study of mortality prediction. Results are reported in Section 5 and Section 6 concludes the paper.

2 Kernel Logistic Regression

One major disadvantage of classical logistic regression (LR) is that it is a linear classifier. It assumes that the outcome or log-odds is a linear function of the independent variables. However, most real world classification problems are non-linear, and so LR cannot capture any non-linearity that may exist in the data.

Using the "kernel trick" however, a kernelized version of LR, or Kernel Logistic Regression (KLR) can be constructed. A mapping function $\boldsymbol{\phi} : x \in \mathbb{R}^d \to \boldsymbol{\phi}(x) \in \mathbb{R}^{d_f}$ is chosen to convert the non-linear relationship between the response and the independent variable into a linear relationship in a higher (and possibly infinite) dimensional feature space. $\boldsymbol{\phi}$ is however usually unknown, but dot products in the feature space can be expressed in terms of the input vector through the kernel function: $\mathbf{K}(x, y) = \boldsymbol{\phi}(x) \cdot \boldsymbol{\phi}(y)$.

Using $\boldsymbol{\phi}$, the "logit" transformation can be written as

$$\log(\pi/(1-\pi)) = \boldsymbol{\phi}(x) \cdot \boldsymbol{\beta} + b$$

where $\pi = \mathbf{Pr}(y = 1|x; \boldsymbol{\beta}) = 1/(1 + \exp(-\boldsymbol{\phi}(x) \cdot \boldsymbol{\beta} - b))$ is the class posterior probability and b is a bias term. Given the training data $\mathcal{D}$, the regularized negative log-likelihood function to maximize can be written as

$$l(\beta) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \frac{\gamma}{2}\sum_{i=1}^{n}\Big[y_i(\boldsymbol{\phi}(x_i) \cdot \boldsymbol{\beta} + b) - \log\left(1 + \exp(\boldsymbol{\phi}(x_i) \cdot \boldsymbol{\beta} + b)\right)\Big]. \quad (1)$$

As with LR, maximum likelihood estimates of the parameters of KLR can be obtain through iterative methods such as the Newton-Raphson algorithm.

Iterative Re-weighted Least Squares KLR

Assume for the moment that the map $\boldsymbol{\phi}(x)$ includes a constant term 1 i.e $\boldsymbol{\phi}(x) \equiv (\boldsymbol{\phi}(x), 1)$ and $\boldsymbol{\beta} \equiv (\boldsymbol{\beta}, b)$. After taking the first and second partial derivative or Eqn. 1 with respect to the parameters $\boldsymbol{\beta}$, a Newton-Rahpson update formula for the parameters is given by

$$\begin{aligned}\boldsymbol{\beta}^{new} &= \boldsymbol{\beta}^{old} + \left(\boldsymbol{\phi} \cdot \mathbf{W} \cdot \boldsymbol{\phi} + \frac{1}{\gamma}\mathbf{I}_{d_f}\right)^{-1} \cdot \boldsymbol{\phi} \cdot (\mathbf{Y} - \boldsymbol{\pi}) \\ &= \left(\boldsymbol{\phi} \cdot \mathbf{W} \cdot \boldsymbol{\phi} + \frac{1}{\gamma}\mathbf{I}_{d_f}\right)^{-1} \cdot \boldsymbol{\phi} \cdot \mathbf{W} \cdot \mathbf{z} \quad (2)\end{aligned}$$

where $\mathbf{W}$ is an $n \times n$ diagonal matrix of weights with i'th element $w_i = \pi_i(1 - \pi_i)$, $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_n)$ and $\mathbf{z} = \boldsymbol{\phi} \cdot \boldsymbol{\beta}^{old} + \mathbf{W}^{-1} \cdot (\mathbf{Y} - \boldsymbol{\pi})$.

Equation 2 can be cast into the following constrained optimization problem (Ngufor and Wojtusiak 2013a)

$$\min_{\boldsymbol{\beta}} L = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\gamma}{2}\sum_{i=1}^{n} w_i \varepsilon_i^2 \quad (3)$$

$$\text{subject to :} \quad z_i = \boldsymbol{\phi}(x_i) \cdot \boldsymbol{\beta} + b + \varepsilon_i, \ \forall \ i = 1, \ldots n. \quad (4)$$

where the bias term b has been re-introduced. The Lagrangian for this optimization problem is given by

$$\mathcal{L}(\boldsymbol{\beta}, b, \boldsymbol{\alpha}, \boldsymbol{\varepsilon}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\gamma}{2}\sum_{i=1}^{n} w_i \varepsilon^2 - \sum_{i=1}^{n} \alpha_i \left(\boldsymbol{\phi}(x_i) \cdot \boldsymbol{\beta} + b + \varepsilon_i - z_i\right) \quad (5)$$

where $\boldsymbol{\alpha} = (a_1, \ldots, \alpha_n) \in \mathbb{R}^n$ is the Lagrange multipliers. The optimality condition can be derived as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = 0 \implies \boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}(x_i) \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{n} \alpha_i = 0 \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial \varepsilon_i} = 0 \implies \alpha_i = w_i \gamma \varepsilon_i, \ \forall \ i = 1, \ldots n \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \implies \boldsymbol{\phi}(x_i) \cdot \boldsymbol{\beta} + b + \varepsilon_i = z_i, \ \forall \ i = 1, \ldots n \tag{9}$$

Using Eqns. 6 and 8 to eliminate $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$ from Eqn. 9 gives the linear system

$$\begin{pmatrix} \mathbf{K} + \frac{1}{\gamma}\mathbf{W}^{-1} & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{z} \\ 0 \end{pmatrix} \tag{10}$$

where $\mathbf{K} = \boldsymbol{\phi} \cdot \boldsymbol{\phi}$ and

$$\mathbf{z} = \mathbf{K}\boldsymbol{\alpha} + b\mathbf{1}_n + \mathbf{W}^{-1}(\mathbf{Y} - \boldsymbol{\pi}). \tag{11}$$

The IRLS-KLR algorithm proceeds iteratively, updating $\boldsymbol{\alpha}$ and b according to Eqn 10 and then updating $\mathbf{z}$ according to Eqn. 11. As demonstrated in (Ngufor and Wojtusiak 2013a), this recursive training can be unbearably slow for even small to medium scale datasets.

Least Squares Kernel Logistic Regression

Based on an initial approximating idea of the logistic function first introduced in (Ngufor and Wojtusiak 2013b), the authors in (Ngufor and Wojtusiak 2013a), approximated the right hand side of Eqn. 10 by substituting for $\boldsymbol{\pi}$ its first order Taylor expansion there by converting IRLS-KLR into a non-iterative least-square or LS-KLR algorithm given by the linear system

$$\begin{pmatrix} \mathbf{K} + \frac{4}{\gamma}\mathbf{I}_n & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} 4(\mathbf{Y} - \frac{1}{2}\mathbf{1}_n) \\ 0 \end{pmatrix}. \tag{12}$$

3 Extreme Learning Machine

Extreme learning machine (ELM) (Huang et al. 2012) is a learning algorithm based on the was proposed in (Huang, Zhu, and Siew 2006) based on the SLFN in which the input weights and biases are randomly selected and the output weights obtained through a minimal norm least squares solution. Since the weights and biases are randomly generated no iterative tuning was required and this significantly reduce computational time for both model training and parameter selection.

The output function of ELM for generalized SLFNs is given by

$$f(x) = \sum_{i=1}^{p} \boldsymbol{\beta}_i \phi_i(x) = \boldsymbol{\phi}(x)\boldsymbol{\beta}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ is the vector of output weights between the hidden layer of p nodes and the output nodes and $\boldsymbol{\phi}(x) = (\phi_1(x), \ldots, \phi_p(x))^T$ is the output row vector of the hidden layer with respect to the input x. It can be seen that ϕ maps the data from a d-dimensional feature space to a p-dimensional hidden layer feature space.

The original implementation of ELM uses minimal norm least squares method instead of standard optimization. Huang et al. (2012) introduced regularization and transformed the ELM algorithm into a constrained optimization algorithm. After defining the constrained-optimization-based ELM and applying the optimality condition, Huang et al. (2012) obtained a simple linear system for the output weights given by

$$\boldsymbol{\phi}^T \left(\boldsymbol{\phi}\boldsymbol{\phi}^T + \frac{1}{\gamma}\mathbf{I}_n \right) \boldsymbol{\beta} = \mathbf{Y}. \tag{13}$$

4 Extreme Logistic Regression

The learning process of ELM proceeds in two steps: (1) the input data points are mapped into the hidden-layer with the input weights and biases randomly generated, (2) a regularized least squares solution is obtained through Eqn 13 where $\boldsymbol{\phi}$ is the hidden-layer output matrix. The hidden-layer output matrix is used to defined a randomized kernel or *ELM-kernel* given by $\boldsymbol{\phi}\boldsymbol{\phi}^T$.

Commonly used ELM hidden-layer output map functions include the Sigmoid function $\phi(\mathbf{w}, b_0, x) = 1/(1+\exp(-\mathbf{w} \cdot x - b_0))$ with $\mathbf{w} \in \mathbb{R}^d$, $b_0 \in \mathbb{R}$ and the Gaussian function $\phi(\mathbf{w}, b_0, x) = \exp(-b_0\|x - \mathbf{w}\|^2)$ with $b_0 > 0$. The values $\{(\mathbf{w}_i, b_{0_i})\}_{i=1}^n$ are randomly generated according to any continuous probability distribution.

A number of authors (Frénay and Verleysen 2010; Liu, He, and Shi 2008) have derived extreme learning methods for SVM by simply plugging in the ELM-kernel for the standard SVM kernel and obtained similar generalization performance as SVM but at a much lesser computational cost. No such extension existed for the KLR until recently where Ngufor and Wojtusiak (2013a) applied the ELM-kernel technique to KLR and also obtained similar generalization performance but with a significantly faster training speed. They called the method "Extreme Logistic Regression" (ELR). Precisely, the ELM-kernel $\mathbf{K} = \boldsymbol{\phi}\boldsymbol{\phi}^T$ is substituted for the kernels in Eqns. 10 and 12 producing IRLS-ELR and LS-ELR respectively.

The interest of this paper is on ELR, since unlike SVM and ELM, ELR yield class posterior probability outcomes based on maximum likelihood criterion, whereas SVM and ELM outputs class decision scores. This can be very useful in many real-world applications such as in medical diagnosis where class probabilities are more important than class decisions. For example in survival analysis, given a patient's data, a survival model attempts to determine the probability of the event occurring or not. A mere decision of whether the event occurs or not is not very useful for a urologist who wants to decide whether to administer an aggressive treatment or not.

A Large Scale Learning Algorithm

Though it is a nice property in kernel methods that explicit representation of the map function $\boldsymbol{\phi}$ and the weights $\boldsymbol{\beta}$ are not required, this however may lead to systems that grows as the number of sample points. For example, the solutions for IRLS-KLR and LS-KLR represented by Eqns. 10 and 12 involves solving linear systems with the $n \times n$ kernel matrix $\mathbf{K}$. Such systems are appropriate for small to medium scale problems. For large scale problems, the solution can be prohibitive due to the expensive computation of the kernel matrix and memory requirements. On the other hand, in ELM methods, the explicit representation of $\boldsymbol{\phi}$ and hence the weights $\boldsymbol{\beta}$ whose dimension depends on the dimension of the hidden-layer feature space p makes it possible to solve a much smaller system as will now be shown.

With LS-ELR as defined by Eqn. 12, as the sample size increases, it may become difficult to solve the linear system due to the increasing size of the $n \times n$ ELM-kernel $\mathbf{K} = \boldsymbol{\phi}\boldsymbol{\phi}^T$. A different system can be obtain by taking advantage or the fact that the weights $\boldsymbol{\beta}$ can be computed explicitly. The optimality conditions represented by Eqns. 6 -8 shows that

$$\boldsymbol{\alpha} = (\boldsymbol{\phi}^T)^{\dagger}\boldsymbol{\beta}, \quad \mathbf{1}_n^T\boldsymbol{\alpha} = 0 \quad \text{and} \quad \boldsymbol{\alpha} = \gamma\mathbf{W}\boldsymbol{\varepsilon}$$

where $\boldsymbol{\phi}^{\dagger}$ is the Moore-Penrose generalized inverse of the matrix $\boldsymbol{\phi}$. LS-ELR uses $\mathbf{W} = \mathbf{I}_n$ and replaces $\boldsymbol{\pi}$ by its first order Taylor expansion in Eqn. 9 leading to

$$\boldsymbol{\varepsilon} + \frac{1}{4}\boldsymbol{\phi}\boldsymbol{\phi}^T\boldsymbol{\alpha} + \frac{1}{4}b\mathbf{1}_n = \mathbf{Y} - \frac{1}{2}\mathbf{1}_n.$$

Substituting the expressions for $\boldsymbol{\varepsilon}$ and $\boldsymbol{\alpha}$ and multiplying both sides by $\boldsymbol{\phi}^T$ gives the linear system

$$\boldsymbol{\phi}^T\left(\frac{1}{4}\boldsymbol{\phi} + \frac{1}{\gamma}(\boldsymbol{\phi}^T)^{\dagger}\right)\boldsymbol{\beta} + \frac{b}{4}\boldsymbol{\phi}^T\mathbf{1}_n = \boldsymbol{\phi}^T(\mathbf{Y} - \ \mathbf{1}_n)$$
$$\mathbf{1}_n^T(\boldsymbol{\phi}^T)^{\dagger}\boldsymbol{\beta} = 0.$$

Using the generalized matrix inverse identity $(\boldsymbol{\phi}^T)^{\dagger} = \boldsymbol{\phi}(\boldsymbol{\phi}^T\boldsymbol{\phi})^{\dagger}$ and noting that if the p columns of $\boldsymbol{\phi}$ are randomly and independently chosen, then the above linear system can be written in a concise form as

$$\begin{pmatrix} \boldsymbol{\phi}^T\boldsymbol{\phi} + \frac{4}{\gamma}\mathbf{I}_p & \boldsymbol{\phi}^T\mathbf{1}_n \\ \mathbf{1}_n^T\boldsymbol{\phi}(\boldsymbol{\phi}^T\boldsymbol{\phi})^{-1} & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{\beta} \\ b \end{pmatrix} = \begin{pmatrix} 4\boldsymbol{\phi}^T(\mathbf{Y} - \frac{1}{2}\mathbf{1}_n) \\ 0 \end{pmatrix}. \tag{14}$$

Unlike the solution of Eqn 12, the solution of Eqn. 14 involves solving a linear system with a $p \times p$ matrix where $p \ll n$.

A similar system for ELM is also derived in (Huang et al. 2012) and given by

$$\left(\phi^T\boldsymbol{\phi} + \frac{1}{\gamma}\mathbf{I}_p\right)\boldsymbol{\beta} = \boldsymbol{\phi}^T\mathbf{Y}. \tag{15}$$

5 Experiments

This section describes the experimental set-up for a case study of survival predictions using the large scale algorithm presented in this paper. Five algorithms: LS-ELR, ELM, Logistic Regression (LG), Naive Bayes (NB) and Cox proportional hazard model (COX) are compared in terms of time scalability and two accuracy measures.

Model Selection

LS-ELM is trained using Eqn. 14 while ELM is trained using Eqn 15 with the Sigmoid function as the ELM hidden-layer map function. In both algorithms, γ and p are two parameters that need to be selected during training. However it has been shown in many works on ELM that setting p to a sufficiently large number such as $p \geq 10^3$ is sufficient to get good generalization. Further, the computational cost in choosing optimal values for γ is significantly lessen due to the simplicity of the models. In the numerical experiments, γ is selected by a standard 10-fold cross-validation using the training set.

Data Source and Study Population

Data for this study were from the Surveillance, Epidemiology, and End Results (SEER) linked with Medicare claims database. SEER combines tumor registry data covering a population based sample of approximately 26% of the US population with claims data from Medicare (Warren et al. 2002). Specifically, in the presented work the target population were individuals diagnosed with prostate cancer starting 2002 through 10 years of follow up. Claims for these individuals were pulled from the Medicare Provider Analysis and Review (MedPAR, Part A), Carrier Claims (Physicians/Suppliers, Part B) and the Out Patient claims files. Individuals were eligible for the study if they were continuously enrolled in Medicare Part A or Part B for at least a year prior to 1'st cancer diagnosis and were at least 66 years old at 1'st diagnosis. Individuals with more than one prostate cancer diagnosis where excluded from the study. A total of 105,000 individuals were identified for potential inclusion in the study.

The outcome variable of interest is death due to all causes. By the end of the 10 years follow up 24% of the patients had died and 0.04% died of prostate cancer.

Measurement of Comorbidity

An important prostate cancer risk factor of interest in this study are the individuals comorbid conditions. Medicare claims prior to 1'st prostate cancer diagnosis were used to compute comorbidities. ICD-9-CM codes available in the claims data have been aggregated to form:

- Elixhauser comorbidity (Elixhauser et al. 1998). A comprehensive list of 30 comorbidity measures. Unlike other comorbidity measures such as the Chalson comorbidity, the Elixhauser comorbidity is not simplified into a weighted index because each comorbidity was found to affect the outcome differently. Among the various existing comorbidity indices, the Elixhauser index has been found to be a better predictor of mortality (Sharabiani, Aylin, and Bottle 2012).
- Clinical Classifications Software (CCS) for ICD-9-CM (Elixhauser, Steiner, and Palmer 2008). The CCS collapses the over 14,000 ICD-9-CM codes into a smaller number of clinically meaningful categories. The single level classification scheme which groups ICD-9-CM codes into mutually exclusive categories was used in this

study. Among the 285 CCS categories, 56 were dropped because their frequency was less than 0.05%.

- Age. Age at diagnosis was also included in the predictive models. The average age for the data considered was 74 years.

In computing the comorbidities, a comorbid condition was identified if its corresponding ICD-9-CM diagnosis or procedure code appears more than once in the Carrier and Outpatient claims and more than 30 days apart, or appeared in the MedPAR claims at least once.

Tumor Risk Factors

Tumor related information such as the clinical stage, diagnostic prostate specific antigen (PSA) level, Gleason score and Prostatic acid phosphatase (PAP) were extracted for all individuals in the study. The PAP was included in the study due to its recent popularity in the medical literature as a significant prognostic factor for patients with intermediate and high-risk prostate cancer (Al Taira et al. 2007).

Initial Treatment Types

Initial treatment types (within 3 months of diagnosis) for all patients were identified in the Medicare files by the Healthcare Common Procedure Coding System (HCPCS) codes. The treatments types was categories into two main groups: Aggressive therapy (surgery, external beam radiation therapy, brachytherapy, cryotherapy, or hormone replacement) and Non-Aggressive therapy (conservative management or active surveillance). Non-Aggressive therapy was defined as the non-presence of any code defining Aggressive therapy or if the SEER data reported no treatment was administered.

About 81% of patients had some Aggressive treatment within 3 months of diagnosis.

Performance Measure

In learning imbalanced data such as the survival data in this study, the overall classification accuracy is often not an appropriate measure of performance. A trivial classifier that predicts every case as the majority class can still achieve very high accuracy. The traditional approach is to use metric such as Recall, Precision, and Weighted Accuracy. These measures can be derive directly from the confusion matrix given in Table 1 as: Recall $= a/(a+c)$, Precision $= a/(a+b)$, Weighted-Accuracy $= \lambda a/(a+c) + (1-\lambda)c/(b+c)$

Table 1: Confusion Matrix

		Predicted Class		
		Positive	Negative	Total
True Class	Positive	a	b	$a+b$
	Negative	c	d	$c+d$
	Total	$a+c$	$b+d$	n

For a physician to make informed decision about treatment options, it is desirable to have a classifier that gives high prediction accuracy for deaths while maintaining reasonable accuracy for survival. The Weighted-Accuracy is a good performance measure in such cases. The weights λ can be adjusted to suit the particular problem.

Another popular performance measure commonly used in survival analysis is the concordance index (C-Index) (Harrell, Lee, and Mark 1996). The C-Index is interpreted as the probability that, given two randomly drawn patients, the patient who experience the event first has a predicted higher probability of the event occurring. Such pairs of patients are called concordant. The C-index is computed as the frequency of all concordant pairs among all pairs of patients in the test data.

The Weighted-Accuracy and C-index was used to evaluate the algorithms. However, because ELM does not output probabilities which is required for the computation of C-Index, only the Weighted- Accuracy will be reported for this algorithm. The weight in Weighted-Accuracy was set equal to the observed mortality rate in the training set i.e $\lambda = 0.24$.

IPCW Analysis

The prostate cancer data used in this study is censored, i.e it includes a survival time (in months) and an indicator variable indicating if the outcome of interest (death) has occurred within the 10 years follow up. The analysis therefore has to take into account the fact that for some observations, the follow up may end before the event occurs or are lost to follow-up. Simply ignoring the censored observations may lead to bias estimates. IPCW analysis (Robins, Rotnitzky, and Zhao 1994) is a method proposed to account for the censoring.

The dependence of censoring on the outcome event is used to up or down-weight the censored observations. Thus in order to avoid bias in the estimates, one can reweight the censored observations using the censoring probability or the probability of not being censored.

In this work, the Nelson-Aaalen estimator of the censoring probability conditioning on the independent variables is used to weight the censored observations in the training set while weights for the uncensored is set to 1.

6 Results

To investigate the effect of age, comorbidities, tumor risk factors and initial treatment types, 6 models were constructed in increasing complexity. The independent variable for the respective models were as follow:

- M1: Age + Elixhauser comorbidity,
- M2: Age + CCS
- M3: M1 + Tumor Risk Factors
- M4: M2 + Tumor Risk Factors
- M5: M3 + Initial Treatment Types
- M6: M4 + Initial Treatment Types

Table 2 and 3 shows the performances of the 5 classifiers for each of the 6 model specifications M1-M6. Overall, LS-ELR and ELM performed better than the other models with respect to the Weighted-Accuracy measure. A slight advantage can be seen for LS-ELR. Cox proportional harzards model outperformed all models with respect to the C-Index.

logistic regression performed poorly when evaluated using the Weighted-Accuracy, but its performance improves for the C-Index. It should be noted however that, experiments without the IPCW analysis for logistic regression produced significantly better results with the 2 performance measures. Because this study aimed at accounting for censoring the results are not reported.

The last row in both tables is the total computational time for models M1-M6. Clearly, LS-ELR and ELM outperformed the other models in time scalability. LS-ELR runs about 13.5 times faster than Naive Bayes, 3.2 times faster than Cox model and 2.8 times faster than logistic regression and more accurate than these models when using the Weighted-Accuracy.

LS-ELR and ELM performed best when the Elixhauser comorbidities, tumor risk factors and initial treatment types are included as independent variables. The best results for these models was obtained with model M5. A similar result can be seen for Naive Bayes especially for the C-Index. On the other hand, Cox model and Logistic regression seems to performed better with the CCS comorbidities.

Table 2: Performance Comparison: Weighted-Accuracy

Models	LS-ELR	ELM	LR	NB	COX
M1	0.77	0.76	0.61	0.67	0.72
M2	0.75	0.75	0.62	0.65	0.72
M3	0.78	0.78	0.62	0.66	0.73
M4	0.75	0.76	0.62	0.65	0.73
M5	0.79	0.79	0.61	0.67	0.72
M6	0.76	0.75	0.65	0.65	0.73
time (sec)	67.08	66.15	188.12	909.38	208.52

Table 3: Performance Comparison: C-Index

Models	LS-ELR	LR	NB	COX
M1	0.69	0.67	0.68	0.71
M2	0.66	0.67	0.67	0.72
M3	0.73	0.70	0.69	0.75
M4	0.67	0.69	0.67	0.76
M5	0.75	0.70	0.70	0.75
M6	0.70	0.69	0.65	0.76
time (sec)	95.50	223.05	1774.63	207.39

7 Conclusions

The application of machine learning tools in healthcare research is often limited to small scale problems primary due to the slow learning rate of most of its effective algorithms. This study presented a large scale algorithm permitting LS-ELR to learn very fast on very large data sets. The complexity of the presented algorithm depends only on the dimension of the SLFN hidden layer feature space which can be fixed for most problem sizes.

The performance of LS-ELR is tested on a large case study of mortality prediction for men newly diagnosed with prostate cancer from the SEER-Medicare database. Numerical results show that the algorithm is extremely fast and can be more accurate in survival predictions than standard statistical methods.

References

Al Taira, M.; Merrick, G.; Wallner, K.; and Dattoli, M. 2007. Reviving the acid phosphatase test for prostate cancer. *Oncology* 21(8):1.

Chow, E.; Harth, T.; Hruby, G.; Finkelstein, J.; Wu, J.; and Danjoux, C. 2001. How accurate are physicians' clinical predictions of survival and the available prognostic tools in estimating survival times in terminally iii cancer patients? a systematic review. *Clinical Oncology* 13(3):209–218.

Elixhauser, A.; Steiner, C.; Harris, D. R.; and Coffey, R. M. 1998. Comorbidity measures for use with administrative data. *Medical care* 36(1):8–27.

Elixhauser, A.; Steiner, C.; and Palmer, L. 2008. Clinical classifications software (ccs). *Book Clinical Classifications Software (CCS)(Editor edˆ eds)*.

Frénay, B., and Verleysen, M. 2010. Using svms with randomised feature spaces: an extreme learning approach. In *ESANN*.

Harrell, F.; Lee, K. L.; and Mark, D. B. 1996. Tutorial in biostatistics multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine* 15:361–387.

Huang, G.-B.; Zhou, H.; Ding, X.; and Zhang, R. 2012. Extreme learning machine for regression and multiclass classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 42(2):513–529.

Huang, G.-B.; Zhu, Q.-Y.; and Siew, C.-K. 2006. Extreme learning machine: theory and applications. *Neurocomputing* 70(1):489–501.

Liu, Q.; He, Q.; and Shi, Z. 2008. Extreme support vector machine classifier. In *Advances in Knowledge Discovery and Data Mining*. Springer. 222–233.

Ngufor, C., and Wojtusiak, J. 2013a. Extreme logistic regression. Submitted.

Ngufor, C., and Wojtusiak, J. 2013b. Learning from large-scale distributed health data: An approximate logistic regression approach. *ICML 13: Role of Machine Learning in Transforming Healthcare*.

Robins, J. M.; Rotnitzky, A.; and Zhao, L. P. 1994. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association* 89(427):846–866.

Sharabiani, M. T.; Aylin, P.; and Bottle, A. 2012. Systematic review of comorbidity indices for administrative data. *Medical care* 50(12):1109–1118.

Warren, J. L.; Klabunde, C. N.; Schrag, D.; Bach, P. B.; and Riley, G. F. 2002. Overview of the seer-medicare data: content, research applications, and generalizability to the united states elderly population. *Medical care* 40(8):IV–3.