Active Learning Based Survival Regression for Censored Data

Bhanukiran Vinzamuri
Dept. of Computer Science
Wayne State University
Detroit, Michigan 48202
bhanukiranv@wayne.edu

Yan Li
Dept. of Computer Science
Wayne State University
Detroit, Michigan 48202
rock_liyan@wayne.edu

Chandan K. Reddy
Dept. of Computer Science
Wayne State University
Detroit, Michigan 48202
reddy@cs.wayne.edu

ABSTRACT

Time-to-event outcomes based data can be modelled using survival regression methods which can predict these outcomes in different censored data applications in diverse fields such as engineering, economics and healthcare. Predictive models are built by inferring from the *censored* variable in time-to-event data, which differentiates them from other regression methods. Censoring is represented as a binary indicator variable and machine learning methods have been tuned to account for the censored attribute. Active learning from censored data using survival regression methods can make the model query a domain expert for the time-to-event label of the sampled instances. This offers higher advantages in the healthcare domain where a domain expert can interactively refine the model with his feedback. With this motivation, we address this problem by providing an active learning based survival model which uses a novel model discriminative gradient based sampling scheme. We evaluate this framework on electronic health records (EHR), publicly available survival and synthetic censored datasets of varying diversity. Experimental evaluation against state of the art survival regression methods indicates the higher discriminative ability of the proposed approach. We also present the sampling results for the proposed approach in an active learning setting which indicate better learning rates in comparison to other sampling strategies.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications-Data Mining; G.3 [**Mathematics of Computing**]: Survival analysis.

General Terms

Algorithms, Design, Performance

Keywords

Active learning; Survival analysis; Cox regression; Healthcare

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2662065.

1. INTRODUCTION

Time-to-event data is widely noticed in many real world applications ranging from engineering to economics to healthcare [3, 28]. In this data, the time is measured until the occurrence of the event of interest. The time measured is the prediction attribute in time-to-event data. The other components of this data include the covariates and a binary censoring indicator variable. Censoring occurs when an observation is incomplete due to some random cause which is independent of the event of interest. The most frequent form of censoring is *right censoring* where subjects are followed until some time, at which the event has yet to occur, but then the subject takes no further part in the study. Censoring differentiates time-to-event data from other commonly observed forms of data.

Active learning from censored data can be very useful in a wide range of applications where a domain expert (oracle) can be involved in the model building process. In healthcare applications, the survival model can select instances by learning from a small labelled set of instances and then query the expert to receive the time-to-event label before including it in the model. This expert feedback can help in refining the model which is particularly useful for healthcare applications such as predicting *30-day* readmission risk [20, 21]. In such applications, the domain expert can integrate domain knowledge into the survival model to build a more robust model.

Active learning from censored data is particularly challenging because the model must choose an instance from both censored and uncensored set of instances in the dataset and query the expert to obtain the time-to-event label. In general censored data mining tasks, censored instances are either deleted or the missing values are imputed to convert it into an *uncensored* problem. An important challenge here lies in utilizing the censored instance completely while building the active learning based survival regression model without deleting or modifying the instance.

Over the past few years, data mining methods have been tuned to predict from censored data. Machine learning methods such as neural networks [25], random forests [15] and support vector machine [13] based approaches have been applied to deal with censored data. These methods in particular can handle non linear relations between the covariates in censored data. Survival regression methods such as Cox proportional hazards [8] and Accelerated failure time (AFT) [26] model are also used to build regression models from censored data.

Cox regression differs from other methods mentioned above because it estimates the relative risk rather than the absolute risk of occurrence of the event. In the healthcare scenario, this is highly useful for a doctor to compare two patients from the same cohort to identify who is at a relatively higher risk. Cox regression also has a simple formulation which consists of just estimating two quantities (i) the unspecified baseline hazard function and (ii) a linear function of the set of covariates.

In this paper, we present an Active Regularized Cox regression (ARC) framework which effectively integrates active learning and Cox regression using a novel model discriminative gradient sampling strategy and robust regularization. Regularization helps in providing good generalizability in ARC and the model discriminative gradient sampling encourages selecting appropriate instances to be labelled by the domain expert. ARC is tested on censored electronic health records (EHR), synthetic censored and publicly available survival datasets. Experimental results over **10** different datasets indicate that ARC outperforms other competing methods on **8** datasets and attains very competitive AUC values. To our knowledge, this is the first work which combines active learning with Cox regression for predicting time-to-event outcomes in the 30-day readmission problem [20, 21] for heart failure.

1.1 Our Contributions

The main contributions of this paper are as follows:

1. Propose an Active Regularized Cox regression (ARC) framework which effectively integrates active learning and regularized Cox regression. ARC uses a novel model discriminative gradient based sampling strategy to select instances to label during the active learning process. In addition, we propose a scalable and efficient coordinate majorization descent (CMD) optimization method for solving regularized Cox regression.

2. Develop a unified ARC framework which encapsulates three regularized Cox regression algorithms which include the kernel elastic net Cox (KEN-COX) [11], elastic net Cox (EN-COX) [10] and LASSO-COX [9] regression algorithms.

3. Demonstrate the performance of ARC on various synthetic and real survival datasets. In addition, we conduct experiments using real electronic health records for heart failure diagnosed patients. We evaluate the performance using domain specific discriminatory metrics such as survival AUC (concordance index) and mean squared error. The active learning curves are also plotted over several datasets which illustrate the effectiveness of the model discriminative gradient based sampling strategy in ARC.

This paper is organized as follows. In Section 2, the related work in the areas of using machine learning approaches in survival regression are discussed. Specifically, we emphasize the work done in the area of integrating machine learning techniques with Cox regression. In Section 3, the Cox regression algorithm is introduced, and the associated terminology is explained in detail.

In Section 4, the algorithm for the CMD based regularized Cox regression ($RegCox$) is provided and the proposed ARC framework is explained. The model discriminative gradient based sampling strategy used in this approach is also explained. In Section 5, experimental analysis is conducted to evaluate ARC against different kinds of survival regression algorithms. In Section 6, we provide the conclusions and some interesting directions for future research.

2. RELATED WORK

In this section, we present the related work in the area of using machine learning methods for survival analysis. In the survival analysis framework, Cox regression has garnered significant interest from researchers in the clinical and machine learning communities [7].

- *Regularized Cox Regression*: Regularization was one of the first few methods to be integrated with Cox regression. LASSO-COX is a regularized Cox regression approach which introduces the L_1 norm penalty in the Cox log-likelihood loss function [9]. Elastic net Cox (EN-COX) further adds the elastic net regularizer to the log-likelihood loss function in Cox regression [10]. LASSO-COX and EN-COX performed better than Cox regression on a wide range of datasets. More recently, robust regularizers such as the kernel elastic net (KEN-COX) and OSCAR have also been integrated within the Cox regression framework [11]. Experimental results indicate that the R^2 and MSE values for KEN-COX were better than EN-COX for EHR datasets.

  In [12], the problem of diabetes risk prediction was tackled using real patient data. For the risk prediction, the authors used methods such as LASSO-COX and Cox regression coupled with strong feature selection mechanisms. They also applied different variants of Cox and other machine learning techniques such as k-nearest neighbour method to obtain highly discriminative models.

- *Machine Learning for Survival Data*: Standard machine learning algorithms cannot handle censoring in survival analysis. Hence, machine learning methods have been modified to handle censoring. Support vector regression to handle censoring (SVRC) is an approach where the standard SVM quadratic programming problem is modified by introducing the censored target variable into it [13, 14]. This SVM framework handles censored data by minimizing the regularized empirical risk with respect to this data dependent loss function to obtain a SVM decision function for censored data. This formulation uses an inverse probability of censoring weighting scheme.

  Random Survival Forests (RSF) is a random forests method for censored survival data [15]. Its difference from random forests lies in the fact that in RSF the splitting criterion in growing the tree must explicitly involve survival time and censoring information. A survival tree is grown for each bootstrap sample and the prediction error is calculated for the ensemble.

- *Optimizing performance metrics*: Other approaches of integrating machine learning into survival analysis include optimizing the survival AUC criterion directly to

build robust models. Boosting the concordance index for survival data is an approach where the concordance index metric is modified into an equivalent smoothed criterion using the sigmoid function. Using this and the gradient of this smoothed survival concordance criterion, a gradient boosting algorithm is run to iteratively generate ensembles [16, 17]. Boosting has also been applied to Cox regression to build a CoxBoost framework for high dimensional micro array data [18]. In contrast to this approach, our ARC framework employs a model discriminative gradient based sampling strategy in active learning.

Ranking in survival analysis is based on developing an approach that learns models by directly optimizing the concordance index [19] . In this paper, the authors focus on maximizing the log-sigmoid and the exponential bounds on the concordance index respectively.

In contrast to the above mentioned methods, our ARC framework aims at obtaining good instances to label through a model discriminative gradient based sampling strategy. Active learning for supervised regression tasks [32, 33] have been developed on different real world datasets. However, existing active learning methods fail to handle time-to-event data and censoring. We address this critical problem in the active learning literature through our ARC framework.

3. SURVIVAL ANALYSIS

Survival analysis is a statistical discipline that deals with censored data and it tries to extract patterns which quantify the relation between the covariates and the risks. More specifically it aims at quantitatively evaluating the effects of covariates and predict event times in the cohort from the knowledge of the covariates. The main complications in survival analysis are caused by the statistical noise which is primarily due to *censoring*.

Censored times C_i are associated with each instance i along with observed time for the event O_i. The failure time for instance i, T_i is set to the minimum of O_i and C_i. If $O_i \leq C_i$ this indicates that the event of interest has occurred within the censoring time. However, if O_i is unknown then T_i is set to C_i and the instance is censored. Censoring is included in the computation of Cox regression using the risk set R_i which is calculated using T_i where $(T_i = min(O_i, C_i))$.

Censoring can also be explained in the context of medical problems such as readmission prediction. In this problem, an event is defined as the onset of heart failure readmission within 30 days of discharge from the previous admission. For example, if a patient was not readmitted after discharge from the previous admission different cases can arise.

The censored cases can be identified as (i) the patients whose follow up details were lost over time or (ii) the patients who were not readmitted within the time period of follow up until the end of the study (which if fixed to 30 in this case). This is commonly called the right censoring setting, which is the most frequently studied censoring phenomena in survival analysis.

Cox regression is one of the most widely used survival analysis methods. It is a semi parametric regression model which can accommodate both discrete and continuous measures of event times. It assumes that conditioned on the covariates X all risks are statistically independent, and that the hazard probability of the primary risk for individuals with covariates X is a function of the following parametrized form.

Table 1: Notations used in this paper

Name	Description
X	n x m matrix of feature vectors.
T	n x 1 vector of failure times.
K	number of unique failure times.
δ	n x 1 binary vector of censored status.
R_i	set of all patients at risk at time T_i $(T_j > T_i)$.
β	m x 1 regression coefficient vector
$L(\beta)$	partial log-likelihood
$h(t\|X)$	conditional hazard probability
$h_0(t)$	base hazard rate
$S_0(t)$	base survival rate
$S(t\|X)$	conditional survival probability
Ke	column wise kernel matrix

$$h(t|X) = h_0(t) \times exp(X \cdot \beta) \tag{1}$$

$$h(t|X) = \underbrace{h_0(t)}_{\text{base hazard rate}} \times \underbrace{exp(X_1\beta) \times \ldots \times exp(X_m\beta)}_{\text{proportional hazards}} \tag{2}$$

Here $X \cdot \beta = \sum_{\mu=1}^{m} X_\mu \beta_\mu$ with time independent parameters $\beta = (\beta_1, \ldots \beta_m)$. The function $h_0(t)$ is called the base hazard rate. It is the base hazard rate one would find for the trivial covariates $X = (0, 0, \ldots 0)$. The proportional hazards (PH) assumption in Cox regression also basically states that different covariates contribute each an independent multiplicative factor to the primary risk hazard rate.

The effect of covariates are taken to be mutually independent and also independent of time. However, it is easy to incorporate time-dependent covariates also into the Cox regression model. In Cox regression, the goal is to find the most probable parameters $\beta = (\beta_1, \ldots, \beta_m)$ and the most probable base hazard function $h_0(t)$.

β is estimated using maximum likelihood estimation over the partial log-likelihood function. The base hazard function on the other hand is estimated using Equation (3). This base hazard function is estimated for an arbitrary time t after calculating β. During estimation the Cox regression model does not assume knowledge of absolute risk and estimates only the relative risk.

This model is also referred to as the CoxPH (Proportional Hazards) model because of the proportional hazards assumption which states that the hazard for any individual is a fixed proportion of the hazard for any other individual.

$$h_0(t) = \sum_{T_i \leq t} \frac{\delta_i}{\sum_{j \in R_i} exp(X_j\beta)} \tag{3}$$
$$S_0(t) = exp(-h_0(t))$$
$$S(t|X_i) = S_0(t) \times exp(X_i\beta)$$

In Equation (3), the formulae for estimating the base survival function $S_0(t)$ and the conditional survival probability $S(t|X_i)$ are provided. This function models the probability of survival for an instance whereas the hazard probability models the probability of occurrence of the event of

interest for an instance. Cox regression is one of the most popular survival regression models and its simple formulation makes it easier to integrate it with different data mining techniques.

4. ACTIVE LEARNING WITH REGULARIZED SURVIVAL ANALYSIS

In this section, we explain the proposed Active Regularized Cox regression (ARC) framework. In Section 4.1, we explain a simple regularized Cox regression algorithm (*RegCox*) which uses the elastic net regularizer. A scalable coordinate majorization descent (CMD) based algorithm for solving this problem is provided.

In Section 4.2, the model discriminative gradient based sampling strategy used in active learning is explained. In Section 4.3, the ARC framework which combines active learning and regularized Cox regression using model discriminative gradient based sampling is explained.

4.1 RegCox: Regularized Cox Regression

Cox regression models have the tendency to overfit the dataset, which limits their generalizability to different scenarios [30]. Regularization is used to overcome the overfitting tendency of the models. The corresponding problem can be solved using unconstrained optimization methods such as gradient descent and coordinate descent (CD).

However, in practice these methods do not scale well. To alleviate this problem, we present a coordinate majorization descent (CMD) based algorithm for solving *RegCox* which is more efficient and scalable than the regular CD solver.

$$L(\beta) = n^{-1} \sum_{i=1}^{K} -X_i\beta + log(\sum_{m \in R_i} exp(X_m\beta)) \quad (4)$$

$$L_j^{'}(\beta) = n^{-1} \sum_{i=1}^{K} \{-X(i,j) + \frac{\sum_{m \in R_i} X(m,j) exp(X_m\beta)}{\sum_{m \in R_i} exp(X_m\beta)}\}$$

In this section, we present the *RegCox* framework which is a generic regularized Cox regression framework which can use any standard regularizer such as the LASSO, elastic net and kernel elastic net. We consider solving *RegCox* here with the specific instance of the elastic net regularization.

In Equation (4), $L(\beta)$ is the partial log-likelihood loss function in Cox regression and $L_j^{'}(\beta)$ is the gradient of log-likelihood with respect to the j^{th} attribute. $G(\beta)$ is the composite function consisting of the log-likelihood and regularization term.

$$G(\beta) = L(\beta) + \sum_{j=1}^{m} \lambda(\alpha|\beta_j| + \frac{1}{2}(1-\alpha)\beta_j^2) \quad (5)$$

$$G(\beta_j) = L(\beta_j, k \neq j) + \lambda(\alpha|\beta_j| + \frac{1}{2}(1-\alpha)\beta_j^2)$$

To apply CMD optimization, we define the objective function $G(\beta_j)$ in Equation (5) for fixed λ, α and β_k. The majorization minimization principle [22] is applied here and instead of minimizing $G(\beta_j)$ in Equation (5) an update of β_j is found such that the univariate function $G(\beta_j)$ is decreased. To write this updating formula for β_j some additional notation is defined using D_j in Equation (6).

$$D_j = \sum_{i=1}^{K} \frac{1}{4n} \{\max_{m \in R_i}(X(m,j)) - \min_{m \in R_i}(X(m,j))\}^2 \quad (6)$$

$$\beta_j^{new} = \frac{S(D_j\beta_j - L_j^{'}(\beta), \lambda\alpha)}{D_j + \lambda(1-\alpha)}$$

$$S(z,t) = (|z| - t)_+ sign(z)$$

In Equation (4), the formulae for computing the j^{th} component of the log-likelihood gradient vector is provided. We use this notation to represent this gradient ($L_j^{'}(\beta) = \frac{\partial}{\partial\beta} L_j(\beta)$). R_i represents the risk set at time point i. K represents the number of unique failure times.

λ is the regularization parameter and α is the elastic net parameter ($0 < \alpha < 1$). $S(z,t)$ is the soft thresholding function. The equation for estimating the regression coefficient vector β^{new} in *RegCox* using coordinate majorization descent (CMD) optimization is also provided.

In Algorithm 1, the regression coefficient vector for the j^{th} coordinate is estimated by keeping all other coordinate values fixed. The regularization parameter λ is determined through cross validation. The LASSO-COX is another instance of *RegCox* which we consider in our ARC framework. LASSO-COX [9] can be considered as a special case of the elastic net regularizer for the value of α set to 1.

The third regularized Cox regression algorithm we consider in *RegCox* is the kernel elastic net Cox regression (KEN-COX). Kernel elastic net Cox regression supplements EN-COX [10] with a column wise kernel matrix information. A RBF kernel matrix (Ke) is computed over the features (columns) of the dataset, and this information is plugged into the elastic net regularizer. The formulation is provided in Equation (7). In this formulation, we use a notation where $X(:,i)$ represents the i^{th} column vector of the matrix X.

KEN-COX can also be solved by using the CMD procedure used for solving *RegCox*. The only modification required in Algorithm 1 is modifying the denominator in the equation for estimating β_j^{new}. The details and algorithm for solving KEN-COX are provided in [11].

Algorithm 1 Regularized Cox Regression (RegCox)

Require: Training Feature Vectors X, Censored variable δ, Time-to-event T, Regularization parameter λ

1: Initialize β
2: **repeat**
3: Compute $L(\beta)$, $G(\beta)$ from X, T ,λ and α using Equations (4), (5)
4: **for** $j = 1, \ldots, m$ **do**
5: Set the objective function $G(\beta_j)$ and apply the CMD procedure
6: Compute the updating factor D_j for computing β_j^{new} using Equation (6)
7: $\beta_j^{new} = \frac{S(D_j\beta_j - L_j^{'}(\beta), \lambda\alpha)}{D_j + \lambda(1-\alpha)}$
8: **end for**
9: Update $\beta = \beta^{new}$
10: **until** Convergence of β
11: Output β
12: Output hazard function using $h_0(t)$, β and δ

$$\beta = \min_{\beta} L(\beta) + \lambda(\alpha \parallel \beta \parallel_1) + \lambda(1-\alpha)\beta^T Ke\beta \quad (7)$$

$$Ke(i,j) = exp(\frac{- \parallel X(:,i) - X(:,j) \parallel_2^2}{2\sigma^2})$$

4.2 Model Discriminative Gradient Based Sampling Strategy

In this section, we explain the model discriminative gradient based sampling strategy used by *RegCox* in ARC. In general regression problems, solving for the optimal parameter β which can minimize the empirical error is a widely used search approach. In this approach, the parameters are repeatedly updated according to the negative gradient of the loss $L(\beta)$ with respect to each training example (X_i, T_i, δ_i). The equation for obtaining β is provided in Equation (8). In this equation, α is called the learning rate.

$$\beta = \beta - \alpha \frac{\partial L_{X^+}(\beta)}{\partial \beta} \quad (8)$$

In active learning, model change is estimated after adding a new example X^+ to the training data with censored status δ^+ and time-to-event value T^+. The empirical risk on the enlarged training set $D^+ = D \cup (X^+, T^+, \delta^+)$ is defined using Equation (9).

$$C(X^+) = \alpha \frac{\partial L_{X^+}(\beta)}{\partial \beta} \quad (9)$$

The goal of our sampling strategy in active learning is then to choose the example that could maximally change the current model and this selection function can be formulated as

$$X^* = argmax_{X^+ \in pool} \parallel C(X^+) \parallel \quad (10)$$

However, in practice we do not know the true label (time-to-event) (T^+) of the sampled data point X^+ in advance. Therefore, we are not able to estimate the model change directly. Instead the expected change is calculated over all possible K unique time-to-event labels from $\{T_1, T_2, \ldots, T_K\}$ to approximate the true change.

$$X^* = argmax_{X \in pool} \sum_{k=1}^{K} h(T_k|X) \parallel \frac{\partial L_X(\beta)}{\partial \beta} \parallel \quad (11)$$

The impact of adding an instance X from the pool to the training data is calculated in Equation (11). The absolute value of the gradient of the loss function with respect to the instance is weighted by the hazard probability $h(T_k|X)$ for that instance. This value is accumulated over all unique time-to-event values to obtain an estimate of the impact of X on the model. Finally, the instance X^* which can induce the maximum model change over all the instances in the pool is selected and assumed to be the *most discriminative* instance for active learning. This explains our model discriminative gradient based sampling strategy.

4.3 Proposed ARC Algorithm

In Algorithm 2, the basic ARC framework is explained. In line 3, the *RegCox* model is built using the training data and time-to-event values. In lines 4-6, the model is applied to all the instances in the unlabelled pool where Equation (11) is applied. In lines 7-8, the instance which makes the highest impact on the model is selected and the time-to-event label for this instance is requested. Finally, in lines 8-10, the training data is updated to build the model at the end of the current active learning round.

Algorithm 2 ARC Algorithm

Require: Training Set $Train$, Unlabelled pool $Pool$, Time-to-event T, Censored status δ, Active learning rounds max

1: $p = 1$
2: **repeat**
3: $Model = RegCox(Train, \delta, T)$
4: **for** each instance in $Pool$ **do**
5: Use model discriminative gradient sampling for each instance in $Pool$
6: **end for**
7: $X^* = argmax_{X \in pool} \sum_{k=1}^{K} h(T_k|X) \parallel \frac{\partial L_X(\beta)}{\partial \beta} \parallel$
8: Query domain expert for label (time-to-event) of X^*
9: $Train \leftarrow Train \cup X^*$
10: $Pool \leftarrow Pool \setminus X^*$
11: $p = p + 1$
12: **until** $p \neq max$

Convergence and Complexity of ARC: The coordinate majorization descent (CMD) method mentioned earlier is used in *RegCox* and it is known to converge efficiently [22] which guarantees the convergence of ARC. However, convergence rates may vary with the kind of regularizer used among LASSO, EN and KEN. The time complexity of Cox regression is O(mK) where m is the number of columns, K is the number of unique time-to-event values. The complexity of ARC can be computed as O($nmK + nK$) where n is the number of instances. The additional nK term here is because of the model discriminative gradient sampling step which is applied on the pool of unlabelled instances.

5. EXPERIMENTAL RESULTS

In this section, we present the experimental results obtained after applying ARC on various diverse datasets. Several real and synthetic survival datasets are used along with electronic health records to assess the performance of ARC. The data processing is explained in the experimental setup subsection. We provide different results which assess the goodness of fit, discriminative ability and learning rates respectively. The ARC framework is implemented in C++ using the Eigen Matrix library [31]. The code for ARC is available at [29]. This includes the code for ARC and the preprocessed datasets (except the proprietary ones from Henry Ford Health System).

5.1 Experimental Setup

5.1.1 Datasets used and data pre-processing

In this section, we demonstrate the performance of ARC on the following datasets.

- *Survival datasets*: Breast, Primary biliary cirrhosis (PBC) and Colon are survival datasets which are used directly from the standard survival R package. PBC data is from the Mayo Clinic trial in primary biliary cirrhosis (PBC) of the liver conducted between 1974 and 1984. Breast cancer dataset is from the German Breast Cancer Study Group. Colon cancer dataset is obtained from the survival R package. These datasets

have the time-to-event and censored attributes provided along with the covariate values.

- *EHR datasets*: We consider electronic health records (EHR) for heart failure diagnosed patients for our analysis. This dataset was obtained for patients diagnosed with primary heart failure from Henry Ford Health System, Detroit, Michigan, USA for a duration of 10 years. For pre-processing this data, we construct features for all the distinct lab variables. To tackle the problem of multiple lab values for the same patient, we represent each lab by a set of summary statistics and apply a logarithm transformation on these values to normalize them.

  Time-to-event (30 day readmission) values are calculated using the prior admission and discharge dates. Patients are *right censored* using the 30 day readmission study period. This implies that if the difference between the last known follow up date and the previous admission date for a patient exceeds 30 days without the onset of a heart failure readmission, then this patient is right censored.

  We present a snapshot of the distribution of readmission probabilities over this EHR dataset. In Figure 1, the readmission probabilities are plotted over a small sample of the EHR dataset for 30, 60 and 90 day readmission for heart failure. An EN-COX model was trained on 200 random instances from one of our EHR datasets and the predicted survival probability values were obtained on a validation sample of 1000 instances. The hazard probabilities are plotted using the equations provided in Section 3. This plot can help the readers understand the readmission trends present in this EHR dataset.

- *Synthetic datasets*: We generate synthetic datasets by setting the pairwise correlation ρ between any pair of covariates to vary from -0.5 to 0.5. We generate the feature vectors using this correlation and a normal distribution $N(0,1)$. Feature vectors of different dimensionality are generated to construct four synthetic datasets. For each of these synthetic datasets, the generated failure times T are calculated using a Weibull distribution with γ set to 1.5. The Weibull distribution is used here to generate positive responses (failure times) to suit the constraints of synthetic survival data. Censoring for each dataset was set randomly to achieve 40% censoring in each synthetic dataset.

5.1.2 Evaluation Metrics

Survival AUC which is also known as the concordance index is used widely in the field of survival analysis [24, 19]. It can be interpreted as the fraction of all pairs of patients whose predicted survival times are correctly ordered among all patients that can actually be ordered.

The motivation behind using this evaluation metric in survival analysis lies in the fact that in clinical decision making the physicians and researchers are often more interested in evaluating the relative risk of a disease between patients with different covariates, than the absolute survival times of these patients.

Survival AUC is the probability of concordance between the predicted and the observed survival. It can be written as in Equation (12). Survival AUC is equivalent to the area under the time-dependent ROC curve, which is a measure of the discriminative ability of the model at each time point under consideration.

In Equation (12), the survival AUC (concordance index) is computed using an indicator function where the predicted values ($S(T^{'}|X_i)$) are the conditional probabilities of survival computed at time $T^{'}$. The indicator function $\mathrm{I}_{a<b}$ is 1 if $a < b$ or 0 otherwise. $S(T^{'}|X_i)$ is estimated using the equations given in Section 3. In Equation (12), num represents the number of comparable pairs.

$$S_{AUC} = \frac{1}{num} \sum_{T_i \in uncensored} \sum_{T_j > T_i} \mathrm{I}_{S(T'|X_i) < S(T'|X_j)} \quad (12)$$

$$rMSE = \sqrt{\frac{\sum_{i=1}^{n} (\delta_i - (exp(X_i^T \beta) h_0(T')))^2}{n}}$$

The other metric we used for evaluation is the root mean squared error ($rMSE$). This is computed using the formula given in Equation (12). In this equation, the hazard function $h_0(T^{'})$ is obtained using the equations from Section 3.

While presenting the experimental results in this paper, we test ARC in an academic setting without the involvement of a real domain expert. The instances which are sampled through the model discriminative gradient based sampling scheme in ARC are automatically assigned to their appropriate time-to-event labels by our program.

Table 2: # Instances, # Features and Active Learning Sampling Size in Dataset

Dataset	# Inst	# Feat	Train(Samp Size)
Breast	686	10	100 (20)
PBC	311	19	50 (10)
Colon	888	15	200 (20)
HF1	5675	98	500 (100)
HF2	4379	98	500 (100)
HF3	3543	98	500 (100)
HF4	2826	98	500 (50)
Syn1	500	15	100 (15)
Syn2	500	50	100 (15)
Syn3	100	50	50 (1)

5.2 Comparison of ARC with other Survival Regression Algorithms

In Table 2, we provide the details of the datasets considered for our experiments. In this table, the sample size indicates the number of instances which are queried and labelled at the end of each iteration. The last column signifies the initial training size selected for active learning, along with the sample size queried at the end of each active learning round for that dataset.

In Table 3, we provide the survival AUC (concordance index) values obtained after running the ARC framework on several real life survival datasets and heart failure (EHR) dataset. In the EHR datasets, we use the following notation; HF 1-4 corresponds to four subsequent readmission datasets for the patients diagnosed with primary heart failure. Each of these dataset records the entire EHR for that particular admission for the patient.

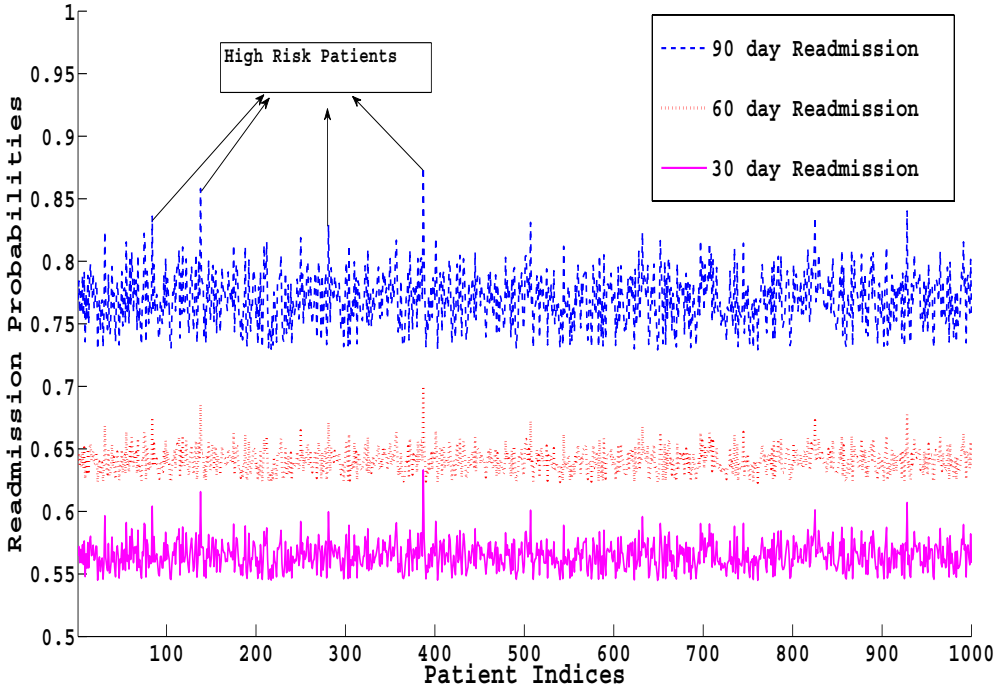


Figure 1: Readmission Probabilities for Patients computed within 30, 60 and 90 days post discharge from index hospitalization.

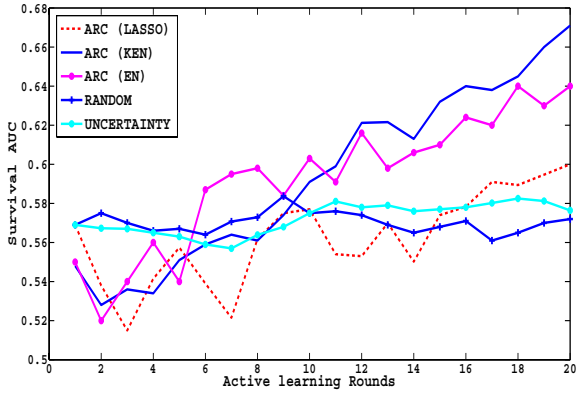
Table 3: Comparison of Survival AUC values of ARC with other survival regression algorithms

Dataset	LASSO-COX	EN-COX	CoxBoost	RSF	BoostCI	ARC(LASSO)	ARC(EN)	ARC(KEN)
Breast	0.61	0.63	0.67	0.68	0.69	0.65	0.6856	**0.734**
Colon	0.651	0.65	0.62	0.60	0.64	0.738	0.735	**0.859**
PBC	0.735	0.759	0.86	**0.863**	0.79	0.81	0.825	0.862
HF1	0.54	0.55	0.59	0.58	0.59	0.60	0.64	**0.671**
HF2	0.56	0.5822	0.60	0.61	0.601	0.66	0.68	**0.71**
HF3	0.533	0.553	0.59	0.59	0.58	0.575	0.58	**0.601**
HF4	0.54	0.55	0.58	0.569	0.56	0.585	0.581	**0.645**
Syn1	0.59	0.628	0.60	0.61	0.589	0.7823	0.838	**0.92**
Syn2	0.801	0.815	0.86	**0.94**	0.93	0.86	0.867	0.921
Syn3	0.67	0.688	0.64	0.64	0.664	0.73	0.78	**0.81**

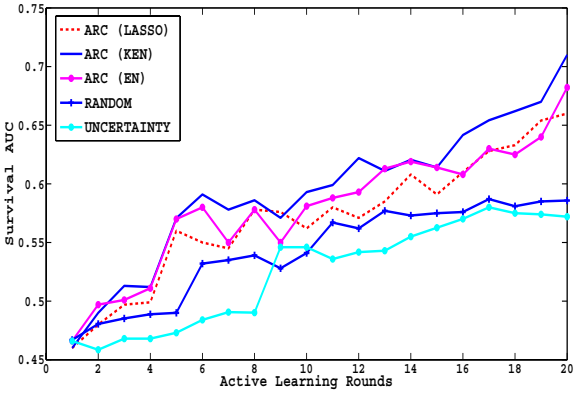
We employ a notation through the remaining part of this paper to represent different active learning algorithms in ARC. ARC (LASSO) represents integrating LASSO-COX with active learning. Similarly ARC (EN) and ARC (KEN) represent integrating EN-COX and KEN-COX with active learning respectively. The performance of these different algorithms in ARC is compared to that of LASSO-COX [9], EN-COX [10], Boosting Cox Regression (CoxBoost) [18], Random Survival Forests (RSF) [15] and Boosting on Concordance Index (BoostCI) algorithms [16].

The *fast-cox* package is used for the LASSO-COX and EN-COX algorithms [27]. Fastcox is implemented in R and offers an effective algorithm for obtaining the entire regularization path of the EN-COX algorithm. CoxBoost and Random Survival Forests are run using the publicly available CoxBoost and rsf R packages respectively. The BoostCI algorithm was implemented in R based on the pseudo-code provided in this paper [16]. KEN-COX uses an additional σ parameter in its RBF kernel which is set to 0.3 for all the experiments.

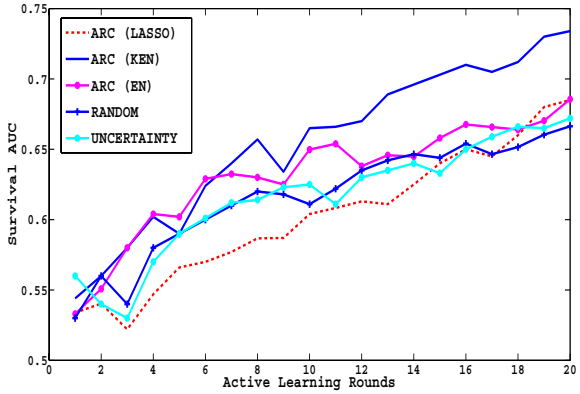
The results in Table 3 show that for **8** out of **10** datasets considered,(ARC) obtains higher concordance index values in comparison to other survival regression algorithms. This better performance of ARC is attributed to the fact that it selects informative instances during the initial active learning rounds. This directly helps in obtaining models with higher discriminative ability.



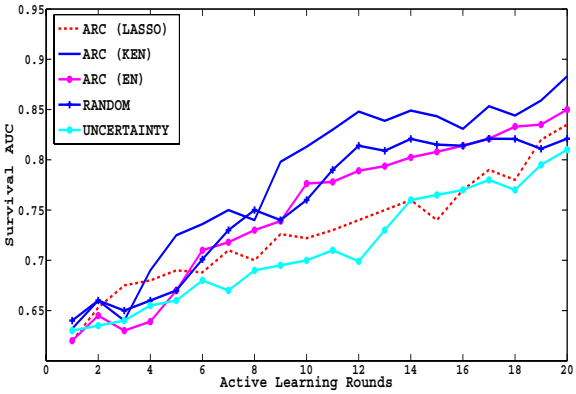
(a) Heart Failure EHR 1



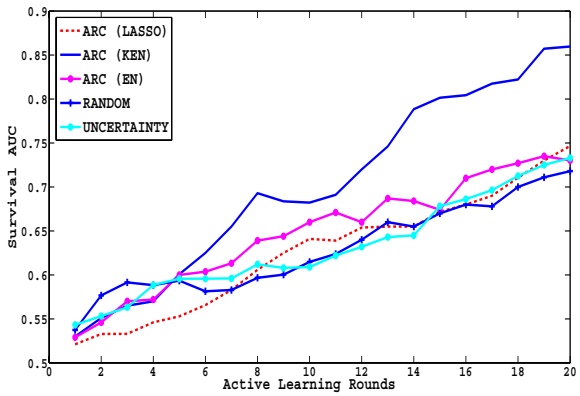
(b) Heart Failure EHR 2



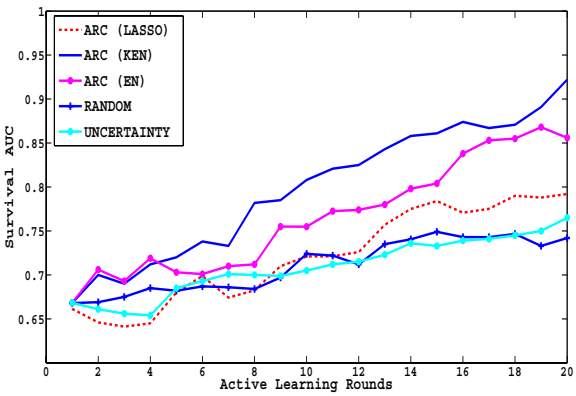
(c) Breast



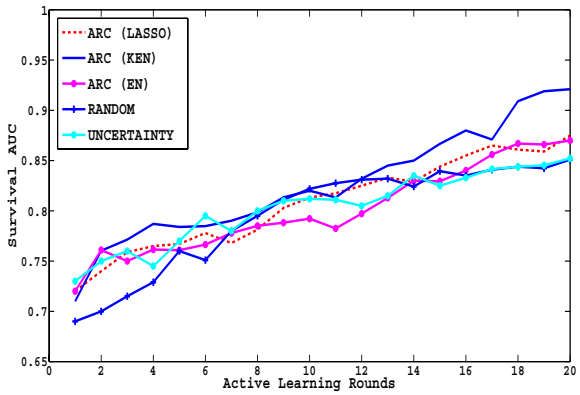
(d) Primary biliary cirrhosis



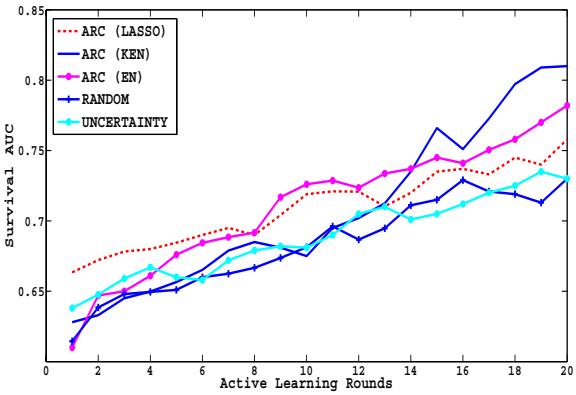
(e) Colon



(f) Synthetic 1



(g) Synthetic 2



(h) Synthetic3

Figure 2: Comparison of the active learning rates of ARC with UNCERTAINTY and RANDOM sampling over EHR, survival and synthetic datasets.

5.3 Comparison of Sampling Strategies in ARC

The sampling strategies evaluated in ARC in this experiment are the following:

1. **RAND**: Randomly sample instances from the pool and update the training data.
2. **Uncertainty based Sampling**: Sample those instances from the pool, which the model is most uncertain about [23].
3. **Model Discriminative Gradient based Sampling**: Sample from the pool that instance which causes the greatest change in the absolute value of the gradient of the loss function evaluted at that instance scaled by the hazard probability over all the unique time-to-event values. The Equation for this is provided in Section 4.2.

In Figure 2, the learning curves are plotted over **20** active learning rounds for **8** datasets. Depending on the size of the dataset being considered, we set the sampling size for each round in batch mode active learning. For each dataset, we consider integrating LASSO-COX, EN-COX and KEN-COX in the ARC framework. For plotting the curve for uncertainty based sampling, we used the predicted survival probabilities to determine those instances the model is most uncertain about. For the random setting, instances were chosen randomly at the end of each active learning round. The x-axis represents the number of active learning rounds. The y-axis represents the concordance index (Survival AUC).

The learning curves for the heart failure EHR data indicate that ARC (KEN) and ARC (EN) obtain significantly better AUC values than other methods, with ARC (LASSO) being marginally better than RAND and UNCERTAINTY. We observe some aberration in the learning curves due to the non-linearity and skewed distribution of EHR data. The learning curves for Breast, PBC and Colon indicate that ARC (KEN) still achieves the highest AUC value with ARC (LASSO) or ARC (EN) being the second best. This suggests that qualitative instances are being sampled from the pool and added to the training data in the active learning rounds. The results over all the datasets also show the effectiveness of ARC based sampling in comparison to uncertainty and random sampling.

5.4 Goodness of fit in ARC

In this section, we compare the performance of ARC(LASSO), ARC(EN) and ARC(KEN). The root mean square error (rMSE) values for the survival regression models are calculated after **20** active learning rounds and the final values are reported. The rMSE is used to assess the goodness of fit obtained by the Cox regression model. It is calculated using Equation (12). The standard deviation values are also provided in Table 4.

The results in Table 4 show that ARC(KEN) provides the best fit (lowest rMSE) amongst all the ARC based algorithms. We attribute this to the fact that the kernel elastic net is a more robust regularizer in comparison to the elastic net and lasso. It uses additional pairwise feature similarity information through the column wise kernel matrix (Ke) and supplements the elastic net penalty. This makes it more effective at capturing correlation in the dataset than other competing approaches.

Table 4: Comparison of rMSE ± std values of ARC

Dataset	ARC(LASSO)	ARC(EN)	ARC(KEN)
Breast	3.08±0.117	2.96±0.113	**2.54±0.09**
Colon	4.69±0.15	3.6±0.12	**1.73±0.05**
PBC	6.70±0.38	4.6±0.26	**3±0.17**
HF1	1.30±0.04	1.32±0.04	**1.29±0.04**
HF2	1.33±0.02	1.42±0.02	**1.26±0.019**
HF3	**1.41±0.023**	1.48±0.024	**1.41±0.023**
HF4	**1.30±0.024**	1.43±0.026	**1.30±0.024**
Syn1	3.35±0.134	3.76±0.1504	**3.25±0.13**
Syn2	**3.23±0.129**	3.9237±0.156	3.28±0.131
Syn3	2.92±0.41	3.25±0.45	**2.58±0.364**

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an Active Regularized Cox regression (ARC) framework which integrates active learning with Cox regression using a novel model discriminative gradient based sampling strategy. In healthcare applications such as readmission risk prediction, ARC can identify patient records to be labelled by a domain expert which can help in building survival models with expert feedback. In ARC, the domain expert provides a time-to-event label for the instance sampled by the model. This labelled instance is then added to the training data at the end of each active learning round and the model is updated with the sampled instance.

We conducted several experiments to study the performance of ARC using three regularized Cox regression algorithms on various synthetic and real datasets. Experimental results indicate that ARC(KEN) is more effective than ARC(LASSO) and ARC(EN). The survival AUC values obtained from ARC(KEN) were also observed to be higher than those obtained from ARC(LASSO) and ARC(EN).

We plan to extend this work by studying the inclusion of the Accelerated Failure Time (AFT) model [26] in the active learning scenario. AFT model is a linear survival regression model which is applicable when the proportional hazards (PH) assumption is violated in certain domains. We would also like to integrate other existing methods such as transfer learning with Cox regression to build transfer learning based survival regression models.

Acknowledgements

This work was supported in part by the U.S. National Science Foundation grants IIS-1231742 and IIS-1242304.

7. REFERENCES

[1] P. McCullough , E. F. Philbin , J. A. Spertus , S. Kaatz , K. R. Sandberg and W. Weaver. Confirmation of a heart failure epidemic: findings from the Resource Utilization Among Congestive Heart Failure (REACH) study. *Journal of the American College of Cardiology*, 39(1): 60–69, 2002.

[2] J. P. Klein and M. L. Moeschberger. *Survival analysis: techniques for censored and truncated data.* Springer, 2003.

[3] D. W. Hosmer Jr, S. Lemeshow, and S. May. *Applied survival analysis: regression modeling of time to event data*, volume 618. Wiley-Interscience, 2011.

[4] C.-F. Chung, P. Schmidt, and A. D. Witte. Survival analysis: A survey. *Journal of Quantitative Criminology*, 7(1):59–98, 1991.

[5] H. Koul and V. Susarla and J. R. Van. Regression analysis with randomly right-censored data. *The Annals of Statistics*, 1276–1288, 1981.

[6] D. Cohn, Z. Ghahramani and M. I. Jordan. Active learning with statistical models. *arXiv preprint cs/9603104*, 1996.

[7] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.

[8] P. Sasieni. Cox regression model. *Encyclopedia of Biostatistics*, 1999.

[9] R. Tibshirani et al. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

[10] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for cox proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

[11] B. Vinzamuri and C. K. Reddy. Cox regression with correlation based regularization for electronic health records. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 757–767. IEEE, 2013.

[12] H. Neuvirth, M. Ozery-Flato, J. Hu, J. Laserson, M. S. Kohn, S. Ebadollahi, and M. Rosen-Zvi. Toward personalized care management of patients at risk: the diabetes case study. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 395–403. ACM, 2011.

[13] F. M. Khan and V. B. Zubek. Support vector regression for censored data (SVRC): a novel tool for survival analysis. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 863–868. IEEE, 2008.

[14] L. Evers and C.-M. Messow. Sparse kernel methods for high-dimensional survival data. *Bioinformatics*, 24(14):1632–1638, 2008.

[15] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. Random survival forests. *The Annals of Applied Statistics*, pages 841–860, 2008.

[16] A. Mayr and M. Schmid. Boosting the concordance index for survival data–a unified framework to derive and evaluate biomarker combinations. *PloS one*, 9(1):84483, 2014.

[17] Y. Chen, Z. Jia, D. Mercola, and X. Xie. A gradient boosting algorithm for survival analysis via direct optimization of concordance index. *Computational and mathematical methods in medicine*, 2013.

[18] H. Li and Y. Luan. Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data. In *Bioinformatics*, 21(10):2403–2409,2005

[19] H. Steck, B. Krishnapuram, C. oberije, P. Lambin, and V. C. Raykar. On ranking in survival analysis: Bounds on the concordance index. In *Advances in Neural Information Processing Systems*, pages 1209–1216, 2007.

[20] A. F. Hernandez, M. A. Greiner, G. C. Fonarow, B. G. Hammill, P. A. Heidenreich, C. W. Yancy, E. D. Peterson, and L. H. Curtis. Relationship between early physician follow-up and 30-day readmission among medicare beneficiaries hospitalized for heart failure. *JAMA: The Journal of the American Medical Association*, 303(17):1716–1722, 2010.

[21] D. Kansagara, H. Englander, A. Salanitro, D. Kagen, C. Theobald, M. Freeman, and S. Kripalani. Risk prediction models for hospital readmission. *JAMA: The Journal of the American Medical Association*, 306(15):1688–1698, 2011.

[22] K. Lange and D. Hunter and Y. Ilsoon. Optimization transfer using surrogate objective functions *Journal of computational and graphical statistics*, 1–20, (9), 2012.

[23] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

[24] M. Gönen and G. Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005.

[25] E. Biganzoli. and P. Boracchi. and E .Marubini. A general framework for neural network models on censored survival data. *Neural Networks*, 15(2), 209–218, 2002

[26] LJ. Wei. The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. *Statistics in Medicine*, 11(14),1871–1879,1992

[27] Y. Yang and H. Zou. A Cocktail Algorithm for Solving The Elastic Net Penalized Cox Regression in High Dimensions. Statistics and Its Interface, 2012.

[28] K. Richard. Proportional hazard regression models and the analysis of censored survival data. *Applied Statistics*, 227–237, 1977.

[29] http://dmkd.cs.wayne.edu/survival

[30] ACC. Coolen and L. Holmberg. Principles of Survival Analysis. *Oxford University Press, 2013/14*.

[31] G. Guennebaud and B. Jacob. Eigen v3. http://eigen.tuxfamily.org, 2010.

[32] A. I. Schein and L. H. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007.

[33] R. Castro, R. Willett, and R. Nowak. Faster rates in regression via active learning. In *NIPS*, volume 3, 2005.