The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Feature Selection and Case-Based Reasoning for Survival Analysis in Bioinformatics

Isabelle Bichindaritz, Charles Englebert
State University of New York
7060 NY-104
Oswego, NY, USA

Angelina Regua, Leszek Kotula
Upstate Medical University
766 Irving Ave.
Syracuse, NY, USA

Abstract

The development of microarray technology has made it possible to assemble biomedical datasets that measure the expression profile of thousands of genes simultaneously. However, such high-dimensional datasets make computation costly and can complicate the interpretation of a predictive model. To address this, feature selection methods are used to extract biological information from a large amount of data in order to filter the expression dataset down to the smallest possible subset of accurate predictor genes. Feature selection has three main advantages: it decreases computational costs, mitigates the possibility of overfitting due to high inter-variable correlations, and allows for an easier clinical interpretation of the model. In this paper we compare three methods of feature selection: iterative Bayesian Model Averaging (BMA), Random Survival Forest (RSF) and Cox Proportional Hazard (CPH) and five methods of survival analysis: Analysis Random Survival Forest (RSF), Cox Proportional Hazard (CPH), Alan Additive Filter (AAF), DeepSurv (neural network), and CbrSurv (case-based reasoning), which we introduce in this paper. Features selected by these methods are compared with a hand selected set of features. All the data we used came from the Metabric breast cancer dataset. Our results indicate that feature selection improves the performance of survival analysis methods. Overall, the best survival analysis performance was obtained by combining RSF for feature selection and CbrSurv, closely followed by DeepSurv, for survival prediction.

Introduction

The goal of this project was to assess the impact of a subset of genes and proteins on breast cancer severity. It focused on analyzing publicly available oncology genetic datasets to determine the effects of a group of genes or proteins on a specific type of cancer, its severity, pathways associated, and subgroups of patients with differential risk factors. Research on cancer heavily relies on genetic information. The goal of survival analysis is to estimate the life expectancy of patients from genetic and/or clinical information. This consists in practice in dealing with survival curves which represent the chance for patients of being still alive as a function of time. This project introduces CbrSurv, a novel survival analysis method based on case-based reasoning and compares it with Deep Learning methods (DeepSurv), Random Survival Forests (RSF) and with classical survival methods Cox Proportional Hazards (CPH) and Alen Additive Filter (AAF). Vast amounts of genetic data are now available for researchers working on cancer. The dimension of data-sets can be extremely high. Some data-sets contain 100 as many features as observations, which can lead to overfitting, interpretation problems and high computational cost. For these reasons, bioinformatics researchers apply feature selection methods to reduce the dimension of the data-sets by keeping the most relevant features only. This project worked with three of those methods, Bayesian Model Averaging (BMA), Random Survival Forest (RSF) and Cox Proportional Hazards (CPH).

Survival Analysis

Survival Analysis Principles

Survival analysis is about predicting how likely an event is to happen over time (Klein and Moeschberger 2003). All survival methods provide a survival curve which represents how likely the event is to have not already happened over time. In our case the event we are interested in is the death of the cancer patient. In survival analysis, we do not necessarily know how long has each patient lived as the experiment may have stopped before they are all dead.

The individuals in a population who have not been subject to the death event are labeled as right-censored. We observe either the survival time, if we have the death date, or a censured time if we do not have the date of the death but only the date of the last visit to the doctor. An instance in the survival data is usually represented as (x_i, t_i, δ_i) where x_i is the feature vector, t_i is the observed time, δ_i is the indicator: 1 is for an uncensored instance, which means the patient is dead, and 0 is for a censored instance, which is a patient being alive. The survival function $S(t) = P(O > t|x)$ represents the probability of being still alive after time t, where O represents the survival time. We can also define the hazard function λ as :

$$\lambda(t) = \lim_{\delta t \to 0} \frac{P(t < O \leq t + \delta t | O > t)}{\delta t} \quad (1)$$

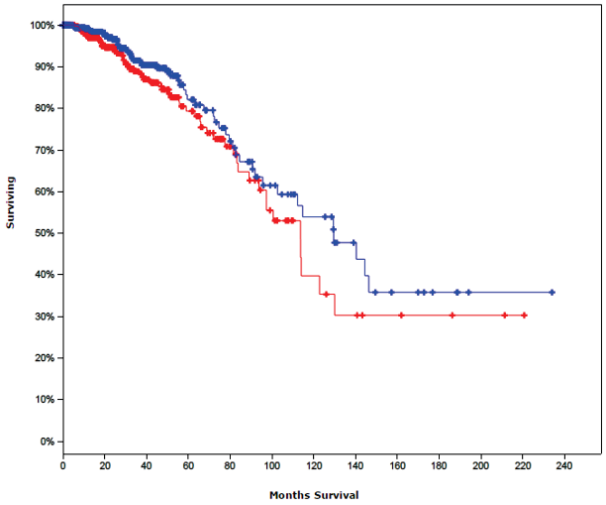

Figure 1: Survival curves estimated with Kaplan-Meier estimate. They represent how likely patients with or without alteration on query genes are surviving disease free over time.

This function represents how the risk of an event per time unit changes over time. Also, as

$$\lambda(t) = \frac{-S'(t)}{S(t)} \tag{2}$$

We get :

$$S(t) = exp(-\int_0^t \lambda(z)dz) \tag{3}$$

for any survival function.

The point of survival analysis is making estimations on data-sets where the target is unknown for certain patients. For those patients only a minimum value of their life duration is known.

Kaplan-Meier Estimator

The Kaplan Meier method permits to estimate an average survival curve related to a population. It consists in defining the survival function as follows :

$$S(t) = \prod_{t_i<t} \frac{n_i - d_i}{n_i} \tag{4}$$

where d_i is the number of death events at time t and n_i is the number of subjects at risk of death just prior to time t. This method does not take any co-variance into account such as gene expressions or age of patients. It only takes as input the survival or censured times of the entire population and predicts an average curve relative to the whole population. cBioportal.org uses this function to display survival curves which permit to make an estimation of the effect of particular gene alterations. The genes can be selected through a user interface and are called query genes.

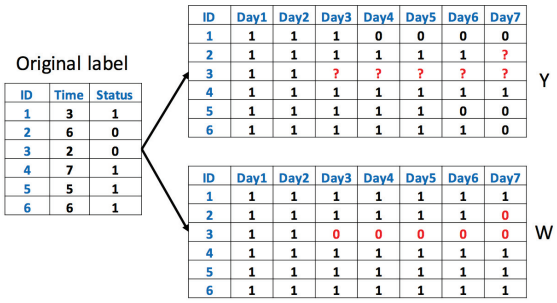

Figure 2: Illustration of generating Y and W from the original label in a simple survival data-set.

Cox Model

The Cox Proportional hazards model is among the most popular methods for survival analysis. It models the hazard function as (Lin 2012) :

$$\lambda(t|x) = \lambda_0(t)exp(\beta^T x) \tag{5}$$

where $\beta = (\beta_1, .., \beta_2)^T$ is a vector of parameters and $\lambda_0(t)$ is a baseline hazard function. It must be evaluated separately. We can also define $h(x) = \beta^T x$ as a risk function.

However, the Cox model has some limitations. The ratio of risk between two individuals does not depend on time as we assume λ_0 equal for every individual. In other words, every survival curve has the same shape. The baseline hazard function λ_0 has to be estimated separately, which induces more errors. Also, in the real world there are too many complex interactions and scenarios that can affect the event of interest in various ways. Thus, in practice, choosing an appropriate theoretical distribution to approximate survival data is very difficult, if not impossible. Multi-task learning can permit to overcome these weaknesses.

Multi-task Learning

The primary motivation of transforming the survival analysis into a multi-task learning problem is that the dependency between the outcomes at various time-points can be accurately captured through a shared representation across related tasks in this multi-task transformation, which could reduce the prediction error on each task. Therefore, multi-task learning methods permit to overcome the weaknesses of Cox Proportional hazards.

Multi-task learning is a sub-field of machine learning in which multiple learning tasks are solved at the same time, while exploiting commonalities and differences across tasks. Multi-task learning permits to improve the performance of multiple classification tasks by learning them jointly (Argyriou and Evgeniou 2007; **?**). To apply it to our problem of survival analysis we must apply a transformation to the problem (Li and Wang 2016). It consists in replacing each label into a k-column vector as shown on figure 2.

Y is the target matrix and W the indicator matrix. Then a natural way to solve the multi-task learning problem is by determining $\hat{B}$ such as :

$\hat{B} = \text{argmin}_B \frac{1}{2}.\|Y - XB\|_F^2 + R(B)$

where $R(B)$ is a regulation term which helps to avoid over-fitting. To handle the question marks in the target matrix, we resort to W and redefined our optimization problem as :

$\hat{B} = \text{argmin}_B \frac{1}{2}.\|\Pi_W(Y - XB)\|_F^2 + R(B)$
where Π_W is a function defined as :
$(\Pi_W(U))_{i,j} = \begin{cases} U_{i,j} & \text{if } W_{i,j} = 1, \\ 0 & \text{if } W_{i,j} = 0. \end{cases}$

Alan Additive Filter

Alan Additive filter works the same way as Cox Proportional Hazard but instead it uses as regression function (Aalen 1989):

$$\lambda(t) = \beta_0(t) + \sum_{i=1}^{T} \beta_i(t).x_i \tag{6}$$

Random Survival Forest (RSF)

Random Forest is a method that operates by constructing a multitude of decision trees at training time and outputting the class (in case of a classification analysis) or the mean regression value (in case of a regression analysis). This popular machine learning method was adapted to survival analysis in the R package $randomForest$ (Ehrlinger 2016). Random forest also permits to rank features and so it provides another feature selection method.

DeepSurv

DeepSurv is a deep learning method for survival analysis based on Faraggi and Simon network (Katzman et al. 2016). The implementation of the model is based on Theano and Lasagne. It uses a Log Likelihood as a loss function. The team which developed DeepSurv also performed feature selection on a database of breast cancer patients. With this feature selection, they were able to reach a concordance index of 0.69. DeepSurv is also able to recommend treatments. It performs this by comparing the risk with and without each treatment.

In survival analysis we do not have the label of every observation. The likelihood is used as a loss function in most survival analysis methods. It is defined as :

$$L(\beta) = \prod_{k=1}^{K} \left(\frac{exp(h_\beta(x_k))}{\sum_{m=k}^{K} exp(h_\beta(x_m))} \right)^{\delta_i} \tag{7}$$

where β is a parameterized weight of the network on which the learning is made, h_β is the risk function of a cox model, and K the number of patients in the data-set. The hazard function is then $\lambda(t|x) = \lambda_0(t)exp(h(x))$

Concordance-index

We can not use a classical loss function such as L2 as we do not possess the curve which represents the probability of dying in our data but only the time at which the patient actually died. Instead, one performance measure traditionally used is the c-index, or concordance index. (Steck et al. 2008) (Gerds et al. 2013) This measure evaluates the ordering of predicted times: how correct is the ordering? It is interpreted as follows: 0.5 is the expected result from random predictions, 1.0 is perfect concordance and, 0.0 is perfect anti-concordance (multiply predictions with -1 to get 1.0).

Case-based Survival Analysis

This paper introduces case-based reasoning (Bichindaritz and Marling 2006) as a new method for survival analysis. We formulate the case-based reasoning framework in this situation as follows.

Given an instance of survival analysis to solve as $< pb_i >$, where pb_i is the feature vector representing the problem situation of a new case, we want to reuse previously encountered solved instances in the form of $< pb_j, sol_j, \delta_j >$, where pb_j is the feature vector representing a solved problem, sol_j is the observed survival time, or solution to the problem, and δ_j is the indicator: 1 is for an uncensored instance, which means the patient is dead, and 0 is for a censored instance, which is a patient being alive.

We adopt an algorithm similar to case-based regression as presented in EAR system (Zhang and Yeung 2012), however with important modifications for survival analysis.

Algorithm 1 Overall Algorithm

Input:
New: new case to solve $< pb_i >$
CB: case base with solved cases $< pb_j, sol_j, \delta_j >$

Output:
estimated length of survival

$RetrievedCases \leftarrow NeighborhoodSelection(pb_i, CB)$
$AdaptationRules \leftarrow RuleGenerationStrategy($
$pb_i, RetrievedCases, CB)$
for all $c \in RetrievedCases$ **do**
    $RankedRules \leftarrow RuleRanking($
    $AdaptationRules, c, pb_i)$
    $ValEstimate(c) \leftarrow$
    $AdaptationCombination(RankedRules, c)$
end for
return $CombineVals(ValEstimate(c)$
for $c \in RetrievedCases)$

Retrieval

Based on the work by Jalali and Leake (2013), a local retrieval based on feature matching between memorized cases and the new case often performs as well as more complex retrieval schemes. In our case, where the number of features can be high, this type of search and retrieval has been chosen for its efficiency. The optimal number of cases is not fixed but determined by a maximal distance measure to consider, which is a parameter of our system, and which can also be set by default in the following manner. If we have *n* cases in the case base, the number of unordered pairs of different

cases is *n * (n-1) / 2*. By calculating the distance between each of these pairs of cases, and averaging these distances, we obtain an average distance *d* for the case base, which represents the radius chosen to retrieve the local neighbors of a new case.

$$RetrievedCases = \{c \in CB \; such \; that \; dist(c, New) < d\} \tag{8}$$

where dist(c, New) represents the distance between pb_j and pb_i.

Adaptation

The adaptation follows the Case Difference Heuristic approach introduced by Hanney and Keane (1996), which generates adaptation rules from pairs of cases by mapping between the differences between problem situations and the differences between the solutions. In our particular situation, we not only take into account the features of the case as represented in pb_j but also in the δ_j additional feature indicating whether the memorized case was alive or not. The adaptation proceeds through several steps :

1. Generating adaptation rules. From the retrieved set of cases, the Local cases - Local neighbors strategy described by Jalali and Leake (2013) has been chosen and adapted to survival analysis. An adaptation rule is created for each pair of cases in the retrieved cases set according to the Case Difference Heuristic:

$$IF(\Delta \; in \; features \; between \; pb_k \; and \; pb_l) \wedge(\delta_k = \delta_l) \; THEN \Delta survival \tag{9}$$

2. Adaptation rules ranking. The set of generated rules $AdaptationRules$ is then ranked according to the context of use as in EAR (**?**). To summarize this method, the adaptation context of a case is its neighborhood as previously defined, and the covariance between each feature and the case solution is calculated over the set of cases in the neighborhood. where Cov_k represents the covariance between $feature_k$ and the case solution. Given a case c in the case base, EAR calculates its adaptation context as a vector based on comparing c to the n cases in a neighborhood containing its nearest neighbor cases. For each case feature, the covariance between the feature and the case solution is calculated over the set of cases in the neighborhood. The adaptation context for a case c is defined as:

$$AdaptationContext(c) = (Cov_1, Cov_2, ..., Cov_f) \tag{10}$$

   The ranking of rules is based on the similarity between the new case and the source case. The ranking score is calculated as the euclidean distance between two element wise products (Zhang and Yeung 2012):
   1) product of the adaptation context vector of the source case to adapt and the new case, and
   2) product of the context vector of the adaptation rule and the vector representing feature differences of the composing cases of that rule.

3. Survival estimation for each retrieved case. For each case in the $RetrievedCases$ set, an adapted survival length is calculated by taking the average of the survival lengths generated by the top N adaptation rules, in ranked order. The number N is a system parameter. $ValEstimate(c)$ is simply the average of all the survival lengths generated from the adaptation rules.

4. Survival estimation for new case. The survival length for New, also represented by sol_i, is then calculated as the average of the survival lengths $ValEstimates(c)$ for all retrieved cases in its neighborhood.

Feature Selection

We want to apply feature selection before applying the model or neural network. We aim to extract biological information from high dimensional data and to filter the expression data-set down to the smallest possible subset of accurate predictor genes. It also permits to hinder the effect of the curse of dimensionality (Keogh and Mueen 2011). The improvement can be both effectiveness and efficiency.

Feature Selection Principles

There are three categories of feature selection methods: filter methods, wrapper methods and embedded methods. Filter techniques easily scale to very high-dimensional datasets, they are computationally simple and fast, and they are robust to over-fitting. They have two main disadvantages: they are independent from the problem considered and most proposed techniques are uni-variate. Uni-variate means that each feature is considered separately, ignoring feature dependencies, which may lead to redundancy between selected features. On the other hand, wrapper methods are specific to a given problem. Wrapper methods evaluate subsets of variables which allows, unlike filter approaches, to detect the possible interactions between variables. Two main disadvantages of these methods are: the increasing over-fitting risk when the number of observations is insufficient and the significant computation time when the number of variables is large. Embedded methods have been recently proposed that try to combine the advantages of both previous methods.

Bayesian Model Averaging

When several different models all fit the data but lead to different estimated effect sizes, standard errors, or predictions, we can't just choose one. We need to consider all of them regarding their respective likelihood. Bayesian averaging problems provide a way to do this.

With high dimensions, many feature subsets selected can represent the data equally well. We call $model$ a set of selected features. Bayesian Model Averaging (BMA) is a well-known feature selection method in bioinformatics. Instead of choosing a single model and proceeding as if the data were actually generated from it, BMA combines the effectiveness of multiple models by taking the weighted average of their posterior distributions. The core idea of Bayesian model averaging holds in the following equation :

number of Patients	Number of Features
1,894	24,375

$$P(\Delta|D) = \sum_{k=1}^{K} P(\Delta|M_k, D)P(M_k|D) \quad (11)$$

where Δ is the quantity of interest, D is the data and M_k is the k^{th} model. The probability of the quantities is assumed to be the mean given by all models weighted by their own probability. There are three issues at this point : obtaining the subsets of relevant models $\{M_k\}$, determining $P(\Delta|M_k, D)$ and determining $P(M_k|D)$. The BMA algorithm resolves these as described in (Yeung, Bumgarner, and Raftery 2005). Once we have $P(M_k|D)$ we can deduce the likelihood of feature x_i by :

$$P(x_i|D) = \sum_{M_k / x_i \in M_k} P(M_k|D) \quad (12)$$

The number of models to consider can be very large. If there are G candidate explanatory genes in the expression set, then there are 2^G possible models to consider. Yet the number of genes in microarray datasets varies from 10^2 to 10^4.

Bayesian Model Averaging for High Dimensional Data

The algorithm described above can only deal with data of dimension lower than 30. The usual practice of employing stepwise backward elimination to reduce the number of genes down to 30 is not applicable in a situation where the number of predictive variables is greater than the number of samples. Yeung et al (Yeung, Bumgarner, and Raftery 2005) developed an iterative BMA algorithm that takes a rank-ordered list of genes and successively applies the traditional BMA algorithm until all genes have been processed. They use Cox Proportional Hazards regression to rank genes.

Cox Proportional Hazards for feature selection

COX Proportional Hazards permit also to perform feature selection. The method consists in ranking the features in descending order of their log likelihood.

Experimental Results

Dataset

Every experiment was conducted on the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) dataset, which originally used expression profiles to identify new breast cancer subgroups in an effort to help physicians provide better treatment recommendations. This data-set consists of 24,375 gene expression data and clinical features for 1,894 patients, 43.85% of which have had an observed death due to breast cancer (with a median survival time of 1,907 days).

Experiments

We applied our three feature selection algorithms on this dataset. We also reproduced the manually selected feature set of size 16 presented by the DeepSurv authors which contains : four prognostic meta-genes (CIN, MES, LYM, and FGD3-SUSD3), the age at diagnosis, the number of positive lymph nodes, the tumor size, the ER status, the HER2 status, four indicators known to be predictive of breast cancer: ERBB2, MKI67, PGR and ESR1, and the prescribed treatment (i.e., chemotherapy, radiotherapy, or hormonotherapy) (Katzman et al. 2016). Meta-genes are sets of genes co-expressed in multiple cancer types. Those features are calculated from gene expressions. The four prognostic meta-genes were previously found to predict accurately the survival time of patients by the winners of the Sage Bionetworks-DREAM Breast Cancer Prognosis Challenge (Cheng, Yang, and Anastassiou 2013).

Test results were obtained by implementing a bootstrapping validation on an average of 100 experiments. This script automatically and randomly generates a train and a test set of $\frac{99}{100}$ and $\frac{1}{100}$ sizes respectively and saves the result and all parameters in a file. For DeepSurv however, a smaller set of experiments was performed due to time limitations, this algorithm being considerably slower than any of the others.

Selected Features

It is interesting to know which selected features are in common with the hand selected features (see Table1).

Table 1: Selected features characteristics

BMA	RSF	CPH
CHEMOTHERAPY	ER STATUS	3 CIN
RADIO THERAPY	HER2 STATUS	1 FGD3-SUSD3
HORM THERAPY		

It is interesting to note that each method seems specialized in a particular type of feature. BMA has selected only clinical features in common with the hand selected features. More exactly it has selected only clinical features related to treatment. RSF includes the two selected hormone-related features ER (estrogen) and HER2 (human epidermal growth factor receptor), which are known to be related to breast cancer. CPH selected four genes related to the meta-genes CIN and FGD3-SUSD3 (Cheng, Yang, and Anastassiou 2013).

Results Summary

We can see from Tables 2 and 3 that :

- The RSF feature selection clearly outperforms every other methods on any survival analysis methods.
- The best concordance index of 71.9 is reached with CbrSurv on a data-set of 62 features, result in close tie with DeepSurv on a set of 64 features. These results are significantly better than those of the other survival methods.

Discussion

Although considerable work has been carried out in temporal case-based reasoning (Montani and Portinale 2006), very

Table 2: Best survival analysis method for each feature selection method

Selection	Features	Analysis	CI
BMA	20	RSF	62.8
RSF	64	DeepSurv	71.7
RSF	62	CbrSurv	71.9
CPH	128	RSF	70.4

Table 3: Best feature selection method for each survival analysis method

Selection	Features	Analysis	CI
RSF	64	DeepSurv	71.7
RSF	62	CbrSurv	71.9
RSF	50	RSF	70.5
RSF	12	CPH	65.6
RSF	12	AAF	66.1

little work has addressed the specific concepts of survival analysis. Bichindaritz (Bichindaritz and Annest 2010) presented a comparison of different feature selection methods for case-based survival analysis. Her results indicated that survival prediction was improved significantly by selecting features prior to applying case-based survival. The current paper confirms these results, however with more recent methods of feature selection specialized in survival analysis. This current paper also compares CBR as a method of choice for survival analysis in comparison with deep learning. Although both methods compare similarly in terms of effectiveness, CBR is much more efficient.

Conclusion

The research carried out introduces CbrSurv, a case-based reasoning method for survival analysis using multi-task learning and gives a baseline of comparison between different selection and survival methods in the context of breast cancer. Other data-sets than Metabric should be used to further this comparison. Feature selection methods are also subject to randomness. Studying subsets of sets randomly generated by the same algorithm seems important to have more reliable estimation of the performance. Combining different feature selection methods by gathering their best features could also be explored.

Although more work is needed to reach a definitive conclusion, these results indicate that feature selection can play a helpful role when performing survival analysis on high-dimensional data. In particular, CbrSurv adaptation rule generation would need efficient optimization methods to run on large numbers of features, and DeepSurv was not able to handle the complete dataset either. In addition to the uncontested role in decreasing computational costs, the survival prediction shows improved concordance index. RSF though takes advantage of the information contained in a moderate additional number of features. Overall, the best survival analysis performance was obtained by combining RSF for feature selection and CbrSurv for survival prediction, closely followed by DeepSurv, the latter with considerably higher computational costs.

References

Aalen, O. O. 1989. A linear regression model for the analysis of life times.

Argyriou, A., and Evgeniou, T. 2007. Multi-task feature learning.

Bichindaritz, I., and Annest, A. 2010. Case based reasoning with bayesian model averaging: an improved method for survival analysis on microarray data. In *International Conference on Case-Based Reasoning*, 346–359. Springer.

Bichindaritz, I., and Marling, C. 2006. Case-based reasoning in the health sciences: What's next? *Artificial intelligence in medicine* 36(2):127–135.

Cheng, W.-Y.; Yang, T.-H. O.; and Anastassiou, D. 2013. *Biomolecular events in cancer revealed by attractor metagenes*. PLoS Comput Biol.

Ehrlinger, J. 2016. ggrandomforests: random forests for regression. *arXiv preprint arXiv:1501.07196*.

Gerds, T. A.; Kattan, M. W.; Schumacher, M.; and Yu, C. 2013. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in Medicine* 32(13):2173–2184.

Katzman, J. L.; Shaham, U.; Bates, J.; Cloninger, A.; Jiang, T.; and Kluger, Y. 2016. *Deep Survival: A Deep Cox Proportional Hazards Network*. BioMed Central.

Keogh, E., and Mueen, A. 2011. *Encyclopedia of Machine Learning*. springer.

Klein, J. P., and Moeschberger, M. L. 2003. *Survival Analysis: Techniques for Censored and Truncated Data*. springer.

Li, Y., and Wang, J. 2016. *A Multi-Task Learning Formulation for Survival Analysis*. San Val.

Lin, D. Y. 2012. *"The Robust Inference for the Cox Proportional Hazards Model"*.

Montani, S., and Portinale, L. 2006. Accounting for the temporal dimension in case-based retrieval: A framework for medical applications. *Computational Intelligence* 22(3-4):208–223.

Steck, H.; Krishnapuram, B.; Dehing-oberije, C.; Lambin, P.; and Raykar, V. C. 2008. On ranking in survival analysis: Bounds on the concordance index. In *Advances in neural information processing systems*, 1209–1216.

Yeung, K. Y.; Bumgarner, R. E.; and Raftery, A. E. 2005. *Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data*. Oxford Academic.

Zhang, Y., and Yeung, D.-Y. 2012. A convex formulation for learning task relationships in multi-task learning.