Empirical Comparison of Continuous and Discrete-time Representations for Survival Prediction

Michael Sloma Michael.Sloma@rockets.utoledo.edu
Fayeq Jeelani Syed SyedFayeq.Jeelani@rockets.utoledo.edu
Mohammadreza Nemati Mohammadreza.Nemati@rockets.utoledo.edu
Kevin S. Xu Kevin.Xu@utoledo.edu
Electrical Engineering and Computer Science Department, University of Toledo
2801 W. Bancroft St. MS 308, Toledo, OH 43606-3390, USA

Abstract

Survival prediction aims to predict the time of occurrence of a particular event of interest, such as the time until a patient dies. The main challenge in survival prediction is the presence of incomplete observations due to censoring. The classical formulation for survival prediction treats the survival time as a continuous outcome, which leads to a *censored regression* problem. Recent work has reformulated the survival prediction problem by discretizing time into a finite number of bins and then applying multi-task binary classification. While the discrete-time formulation is convenient and potentially requires less assumptions than the continuous-time approach, it also loses information by discretizing time. In this paper, we empirically investigate continuous and discrete-time representations for survival prediction to try to *quantify the trade-offs between the two formulations*. We find that discretizing time does not necessarily decrease prediction accuracy. Furthermore, discrete-time models can result in even more accurate predictors than continuous-time models, but the number of time bins used for discretization has a significant effect on accuracy and should thus be tuned as a hyperparameter rather than specified for convenience.

Keywords: Survival analysis, censored regression, multi-task learning, Cox proportional hazards model, multi-task logistic regression

1. Introduction

Survival analysis methods are typically used to analyze time-to-event data and have a long history in statistics. Many survival analysis methods can also be used for survival prediction, which aims to predict the time to event for an unseen event given a set of covariates or features for an example. The main difference between survival prediction and other prediction problems in machine learning is the presence of incomplete observations, where we have only partial information about the outcomes for some examples due to *censoring*. Typical machine learning algorithms cannot incorporate this partial information, so survival prediction algorithms need to be created to accommodate censoring.

Classical methods for survival analysis have typically treated the time to event as a continuous outcome. The classical survival prediction problem is thus formulated as a *censored regression* problem. Such approaches require some assumptions on the survival times and include both semi-parametric and parametric models. The most commonly used semi-parametric model is the Cox Proportional Hazards (CoxPH) model (Cox, 1972), which

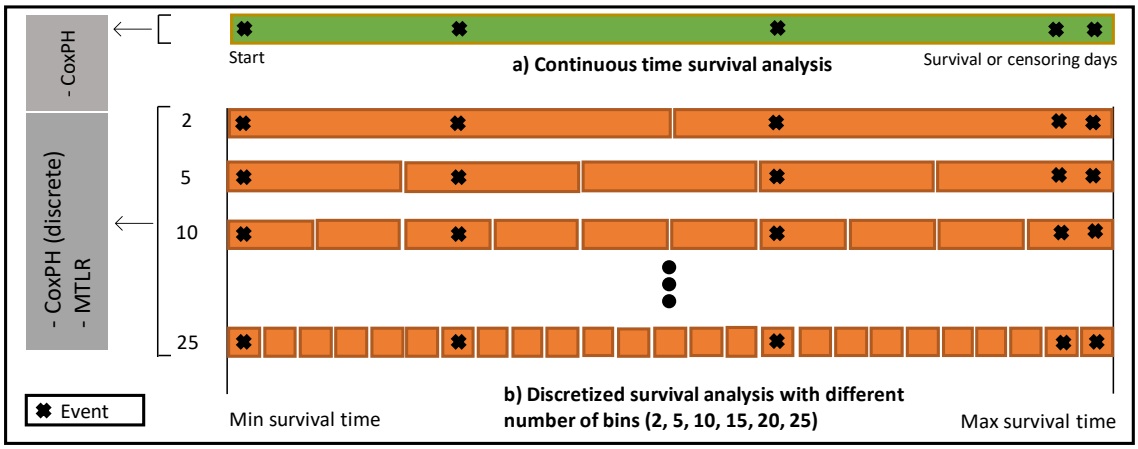


Figure 1: Illustration of time discretization into different number of bins

makes the proportional hazards assumption about survival times. Parametric models specifically assume a distribution for the survival times, such as the exponential or log-normal distribution. Both semi-parametric and parametric models can be very accurate predictors when their assumptions are satisfied, but such assumptions are commonly violated in practice.

An alternative approach to survival prediction is to discretize the survival times into a set of time bins. This is done by assuming some maximum time or horizon (e.g. 20 years) and then dividing time into equally-spaced bins (e.g. 20 bins each representing 1 year). This reformulates the survival prediction problem as a sequence of binary classification problems, which is a type of multi-task learning problem. Such an approach is both convenient and does not require any assumptions on the distribution of the survival times. This discrete-time approach forms the basis for many recently-proposed survival prediction algorithms (Yu et al., 2011; Li et al., 2016a; Lee et al., 2018; Giunchiglia et al., 2018; Ren et al., 2019; Wulczyn et al., 2020).

We examine three research questions in this paper:

RQ1 How much does discretizing time decrease the accuracy of a continuous-time survival prediction algorithm?

RQ2 How does the number of discrete time bins affect the accuracy of a discrete-time survival prediction algorithm?

RQ3 Does the added flexibility of the discrete-time formulation lead to an increase in accuracy that compensates for any decreases in accuracy to discretizing time?

To investigate these three questions, we perform an empirical comparison of survival prediction accuracy of continuous and discrete-time algorithms across four real data sets. To keep the comparison as focused as possible on the format of the times rather than model difference, we select two survival prediction models from the same family of generalized linear models: the CoxPH model and the multi-task logistic regression (MTLR) model

(Yu et al., 2011). The discretization process is illustrated in Figure 1. The CoxPH model is designed for continuous survival times, although it can be applied to discretized times as well by handling ties (Efron, 1977). On the other hand, the MTLR model requires discretized times because it formulates survival prediction as a sequence of binary classification problems.

2. Background

Survival analysis models treat the time to the event of interest, or the survival time, as a random variable. In the survival analysis literature, the distribution of survival times is mainly identified by the survival function $S(t)$ and hazard function $h(t)$ (Wang et al., 2019).

The survival function is defined as the probability that a person survives after a specific time t.

$$S(t) = P(\texttt{survival after t}) = P(T > t).$$

This is a non-increasing function equal to 0 at $t = \infty$.

The hazard function is the probability that event occurs in the next instant, given survival to time t:

$$h(t) = \frac{f(t)}{S(t)},$$

where $f(t)$ is the probability density function of the survival time.

2.1. Continuous-time Survival Prediction

Continuous-time survival analysis approaches fall into 3 main categories: parametric, semi-parametric, and non-parametric.

2.1.1. Parametric

Parametric models are ideal if a known distribution, such as exponential, gamma, Weibull, or log-normal, can be accurately fitted to the survival times (Lee and Wang, 2003). In high-dimensional settings, regularized methods can be applied to generate accurate predictions (Li et al., 2016b). However, the distribution of survival times in many survival analysis problems does not resemble one of the known distributions. As a result, the key survival analysis functions (survival, hazard, and cumulative density functions) cannot be accurately estimated as the distributions' parameters cannot be accurately estimated from the data.

2.1.2. Semi-parametric

Semi-parametric survival analysis techniques such as the Cox proportional hazard model (CoxPH) do not require specification of a particular distribution for survival time (Cox, 1972). CoxPH is one of the most widely used methods to estimate survival functions by means of fitting the log of hazards as a linear combination of subjects' covariates. This approach removes the need for baseline hazard function estimation by maximizing a partial log-likelihood function that is not dependent on the baseline hazard function.

The hazard ratio function in CoxPH model is given by

$$\log \frac{h_i(t)}{h_0(t)} = \boldsymbol{\beta}^T \boldsymbol{x}_i,$$

where the $\boldsymbol{x}_i$ is the ith subject, and $\boldsymbol{\beta}$ is the coefficient vector. To estimate the coefficient vector $\boldsymbol{\beta}$, the following partial log-likelihood function should be maximized:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{k} \left[\boldsymbol{\beta}^T \boldsymbol{x}_i - \log \sum_{j:s_j > s_i} \exp(\boldsymbol{\beta}^T \boldsymbol{x}_l) \right],$$

where k is the number of distinct event times, and s_i denotes the survival time for subject i. In the event of ties (multiple events at the same time), Efron's approach (Efron, 1977) is typically applied.

Particularly in high-dimensional settings, regularization is often applied to avoid overfitting to the training data. Perhaps the simplest regularizer is the ℓ_2 penalty, which involves minimizing $\frac{C}{2}\|\boldsymbol{\beta}\|_2^2 - l(\boldsymbol{\beta})$, where C is the regularization constant. More complex regularizers have also been proposed, including the Elastic Net (combined ℓ_1 and ℓ_2 penalties) (Simon et al., 2011).

2.1.3. Non-parametric

Suppose searching for a proper distribution that fits the survival data is time-consuming, or no distribution appropriately fits the data. In that case, non-parametric approaches can be considered to estimate the survival function. Estimating the survival function using the product-limit (PL) approach, developed by Kaplan and Meier (1958), is one of the most widely used non-parametric approaches. Unlike the parametric and semi-parametric approaches, however, the Kaplan-Meier estimator does not account for covariates and is thus not typically used for survival prediction.

2.2. Discrete-time Survival Prediction

Discrete time survival prediction treats time as discrete by dividing continuous time into intervals, for example $[0, a_1), [a_1, a_2), ..., [a_{m-1}, a_m), [a_m, \infty)$. There are several advantages of using discrete time to event setting. They do not require any proportional hazard-like assumptions on the distribution of the survival times because any discrete distribution is valid. They also allow the survival prediction problem to be formulated as a sequence of binary classification problems. Finally, compared to continuous time models, interpreting the hazard functions in discrete time models becomes easier as they are expressed as conditional probabilities, and they can handle ties easily.

To overcome the proportional hazards assumption in the CoxPH model, Yu et al. (2011) proposed a multi-task logistic regression (MTLR) approach to survival analysis and demonstrated superior performance compared to the CoxPH model on several real data sets. Rather than the hazard function, it directly models the survival function by combining local logistic regression models so that censored observations and time varying effects of features are naturally handled. The survival time s is encoded as a binary sequence of survival statuses y whose probability of observation is represented as a generalization of a logistic regression model, so that the log-likelihood of a set of uncensored patients with survival time $s_1, s_2, \ldots, s_n$ and feature vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ is given by

$$l(\theta) = \sum_{i=1}^{n} \left[\sum_{j=1}^{m} y_j(s_i)(\boldsymbol{\theta}_j^T \boldsymbol{x}_i + b_j) - \log \sum_{k=0}^{m} \exp \sum_{j=k+1}^{m} (\boldsymbol{\theta}_j^T \boldsymbol{x}_i + b_j) \right].$$

Rather than maximizing the log-likelihood, Yu et al. (2011) propose to optimize an objective with two regularizers:

$$\min_{\Theta} \frac{C_1}{2}\|\boldsymbol{\theta}_j\|^2 + \frac{C_2}{2}\sum_{j=1}^{m-1}\|\boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_j\|^2 - l(\theta),$$

where $\boldsymbol{\Theta}$ denotes the matrix of all $\boldsymbol{\theta}_j$ vectors. The two regularizers over $\|\boldsymbol{\theta}_j\|^2$ and over $\|\boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_j\|^2$ prevent overfitting and ensure smooth variation of parameters across consecutive time points, respectively. The model naturally handles the censored data by marginalizing over the unobserved variables in a survival status sequence.

Other discrete-time survival prediction approaches include another multi-task learning survival prediction method MTLSA (Li et al., 2016a), which also transforms the survival prediction problem into a series of binary classification sub problems but uses an $\ell_{2,1}$-norm penalty to enforce sparsity and prevent overfitting. More recent work has focused on deep learning-based approaches for discrete-time survival analysis (Giunchiglia et al., 2018; Lee et al., 2018; Ren et al., 2019; Wulczyn et al., 2020). Such approaches typically use the same multi-task binary classification formulation but differ in their loss functions and neural network architectures.

Another discrete-time approach reformulates the survival prediction problem not as multi-task binary classification, but as Poisson regression (Bender et al., 2020).

2.3. Related Work

Eleuteri et al. (2007) described and compared neural network-based approaches for both continuous and discrete survival analysis using modified variants of Conditional Hazard Estimating Neural Network (CHENN) (Biganzoli et al., 1998) and a Bayesian version of Partial Logistic Artificial Neural Network (PLANN) (Lisboa et al., 2003), respectively, and found no significant difference in the discrimination performance on the basis of C-index.

Kvamme and Borgan (2019) proposed an alternative discretization approach based on quantiles of the estimated event time distribution. They use two schemes: constant density interpolation (assuming constant density functions between the time-points in the discretization grid) and constant hazard interpolation (assuming constant hazard rates between the grid points) to interpolate an estimated discrete survival function for continuous time predictions.

A recent survey on machine learning approaches for survival analysis (Wang et al., 2019) covers both continuous and discrete-time approaches, but does not compare them or examine the effects of discretizing time, which we focus on in this paper.

3. Data Sets

We use 5 real data sets in this study for evaluating different survival analysis approaches. The sizes of the data sets are shown in Table 1.

- The Dutch Breast Cancer Dataset (**DBCD**): The survival data of 295 women with breast cancer is contained in this data set. This is a high-dimensional gene expression data set with 4,919 features, which is much larger than the number of instances.

Table 1: Sizes of data sets used in this study

Data Set	# of Instances	# of Features	# of Censored Instances
DBCD	295	4,919	216
NWTCO	4,028	13	3,457
FLCHAIN	7,874	42	5,705
WHASS500	500	14	285
SRTR	106,372	3,661	79,357

Around 73% of the instances are censored. A detailed description of the data can be found from (van't Veer et al., 2002; van Houwelingen et al., 2006).

- National Wilm's Tumor Study (**NWTCO**): This data set was originally presented in a study to assess the impact multiple variables on days to tumor relapse (D'angio et al., 1989). This data contains the information of $4,028$ patients, where 85.8% of the end points are censored.

- Assay of Serum Free Light Chain (**FLCHAIN**): This data set was first presented in a study aimed to measure the impact of non-clonal serum immunoglobin free light chain (FLC) on the survival time (Dispenzieri et al., 2012). It has $7,874$ instances and 72.5% of the endpoints are censored. This dataset was sourced from scikit-survival (Pölsterl, 2020).

- Worcester Heart Attack Study (**WHAS500**): This data contains the information of 500 hospitalized patients (Goldberg et al., 1986). The main objective of this study was to examine the effects of various covariates on the survival patients after their admission to the hospital. It has 14 features and 72% of the endpoints are censored. This dataset was sourced from scikit-survival (Pölsterl, 2020).

- The Scientific Registry of Transplant Recipients (**SRTR**): The SRTR data includes data on all donors, wait-listed candidates, and transplant recipients in the U.S., submitted by the members of the Organ Procurement and Transplantation Network (OPTN). The event of interest is time to transplant failure. We consider the same set of $106,372$ transplants as in Nemati et al. (2021). We use the Human Leukocyte Antigen (HLA) pairs-based feature representation they proposed, which has $3,661$ features.

4. Methods

To address the three research questions we posed in the introduction, we consider three survival prediction models:

- CoxPH model on continuous survival times.

- CoxPH model on discretized survival times.

- Multi-Task Logistic Regression (MTLR) model on discretized survival times.

The CoxPH and MTLR models both come from the same family of generalized linear models, so we believe that this is a fair comparison that focuses on the format of the survival times rather than differences in models.

To create the discretized survival times, we created uniformly spaced time bins between the minimum and maximum survival time in the data set, using a set number of bins in each of our discretized experiments. We then took the midpoint of each bin as the survival time. For example if a bin is $[15, 16)$, and a subject has a survival time of 15.76, it assigned a discretized survival time of 15.5. Discretizing times creates many ties, so we use Efron's method for handling ties (Efron, 1977) when fitting the CoxPH model, which is more accurate but slower. We discretize the survival times only on the training set (see data splits in Section 4.1) so that the original continuous survival times are used for evaluating prediction accuracy.

We experiment with the number of time bins in the range $\{5, 10, 15, 20, 25\}$ for both CoxPH on discretized survival times and MTLR. For MTLR, we also push discretization to the extreme by considering a case with only 2 time bins.

4.1. Experiment Set-up and Hyperparameter Tuning

All data was randomly shuffled with a set seed before being assigned into train/validation/test sets. The proportion of the sets was 60%/20%/20% respectively. We converted categorical variables into a one-hot encoded format. We then used scikit-learn's `StandardScaler` (Pedregosa et al., 2011) which standardizes features by removing the mean and scaling to unit variance on all columns.

To perform the hyperparameter searches, we used Ray Tune (Liaw et al., 2018). Within Tune, we used the HyperBandForBOHB scheduler (Falkner et al., 2018; Li et al., 2018) in conjunction with the TuneBOHB searcher in order to utilize a Bayesian optimization search scheme. We set all experiments to take 1 search unit so no experiments were paused and resumed in the search. Additionally, we optimize over the validation set C-index results to prevent test set leakage.

We used the CoxPH model from the scikit-survival package (Pölsterl, 2020) and the MTLR model from PySurvival (Fotso et al., 2019–)[1]. The CoxPH model used an ℓ_2 penalty, and the regularization parameter C (denoted by `alpha` in scikit-survival) was the only hyperparameter tuned for the CoxPH model. The model was fit using Newton-Raphson optimization.

Hyperparameters for the MTLR model were number of training epochs, ℓ_2 regularization parameter C_1 for the model coefficients and a second ℓ_2 regularization parameter C_2 that ensures the parameters vary smoothly across consecutive time points (denoted by `l2_reg` and `l2_smooth`, respectively, in PySurvival), and the learning rate. All parameters except for number of epochs were sampled on a log scale. The model was fit using the Adam optimizer (Kingma and Ba, 2014).

The CoxPH hyperparameter space was sampled 100 times per model type, while the MTLR hyperparameter space was sampled 5000 times. This was because the hyperparam-

1. We also tried the CoxPH implementation in PySurvival but experienced slow convergence and worse results compared to scikit-survival.

eter space for the MTLR search was substantially larger than the space for the CoxPH model, which had only 1 hyperparameter.

5. Results

The C-indices on the validation and test sets for the four datasets we examine are shown in Tables 2(a) and 2(b), respectively[2]. Prediction of survival times in the FLCHAIN data set was much easier than in the others, with C-indices around 0.9 for all models while C-indices were around 0.6 to 0.8 in the other data sets. Using these results, we turn to the three research questions posed in the introduction.

RQ1: How much does discretizing time decrease the accuracy of a continuous-time survival prediction algorithm? To answer this question, we compare the C-indices for the CoxPH model fit to the continuous survival times and for the CoxPH models fit to the discretized times. Across all data sets, there is very little difference between the two C-indices regardless of the number of time bins, both for the validation and the test sets. The minor differences are likely as a result of hyperparameter tuning rather than the time discretization. Thus, it appears that *discretizing time has minimal effect* on the accuracy of continuous-time survival prediction on real data provided that a reasonable number of bins are used.

RQ2: How does the number of discrete time bins affect the accuracy of a discrete-time survival prediction algorithm? To answer this question, we primarily examine the results across varying number of bins for the MTLR model, since we found that discretization had minimal effect on CoxPH in RQ1. Both the validation and test set C-indices shown in Table 2 show a lot of variation across the number of time bins, much more than for CoxPH. Again, some of this is attributable to hyperparameter tuning, particularly for the small DBCD data set where overfitting is a large concern. However, even for FLCHAIN, a data set with over $7,000$ instances, the test set C-index varies by 0.048, which is *one order of magnitude larger than we observed in RQ1!* For the most part, the trend we observe is that both the validation and test set C-indices increase with the number of time bins up to a certain point and then decrease after that point, as shown in Figure 2 for the NWTCO and WHAS500 data. The number of time bins where the highest C-index is achieved varies by data set, however.

RQ3: Does the added flexibility of the discrete-time formulation lead to an increase in accuracy that compensates for any decreases in accuracy to discretizing time? To answer this question, we compare the C-indices of the CoxPH model fit to the continuous times with those of the MTLR models. When examining the validation set C-indices in Table 2(a), the best MTLR model is more accurate than the CoxPH on all data sets aside except SRTR[3], and the improvement can be substantial as in the cases of the DBCD and WHAS500 data sets. When considering the test set C-indices in Table 2(b), however, this improvement does not always carry over, perhaps as a result of overfitting

2. Code and data to reproduce our results are available at https://github.com/IdeasLabUT/Continuous-Discrete-Survival-Prediction
3. For the MTLR results on the SRTR data with 10 bins and higher, the majority of runs in our hyperparameter tuning failed to converge, which may partially explain its poor accuracy.

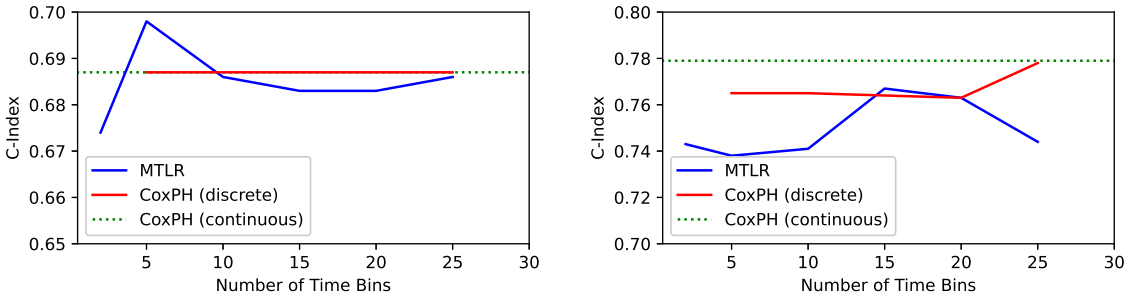
Table 2: Prediction accuracy for different models and data sets as measured by the C-index. The hyperparameters that maximize the validation set C-index are chosen for each model. These same hyperparameters are used to compute the test set C-indices. The most accurate model for each data set is shown in bold.

(a) Validation Set C-indices

Model	DBCD	NWTCO	FLCHAIN	WHAS500	SRTR
CoxPH (continuous)	0.793	0.736	0.923	0.770	0.605
CoxPH (5 bins)	0.793	0.735	0.923	0.777	0.605
CoxPH (10 bins)	0.800	0.736	0.923	0.778	**0.606**
CoxPH (15 bins)	0.792	0.736	0.923	0.778	**0.606**
CoxPH (20 bins)	0.793	0.736	0.923	0.779	0.605
CoxPH (25 bins)	0.792	0.736	0.923	0.764	0.605
MTLR (2 bins)	**0.848**	0.746	0.885	0.794	0.569
MTLR (5 bins)	0.832	**0.757**	0.911	0.816	0.570
MTLR (10 bins)	0.842	0.749	0.919	0.810	0.572
MTLR (15 bins)	0.837	0.750	**0.925**	0.809	0.572
MTLR (20 bins)	0.825	0.753	0.924	**0.827**	0.571
MTLR (25 bins)	0.842	0.751	0.921	0.814	0.550

(b) Test Set C-indices

Model	DBCD	NWTCO	FLCHAIN	WHAS500	SRTR
CoxPH (continuous)	0.809	0.687	**0.934**	**0.779**	0.611
CoxPH (5 bins)	0.813	0.687	**0.934**	0.765	**0.612**
CoxPH (10 bins)	0.813	0.687	**0.934**	0.765	**0.612**
CoxPH (15 bins)	0.814	0.687	**0.934**	0.764	0.611
CoxPH (20 bins)	0.814	0.687	**0.934**	0.763	0.611
CoxPH (25 bins)	0.813	0.687	**0.934**	0.778	0.611
MTLR (2 bins)	**0.836**	0.674	0.880	0.743	0.576
MTLR (5 bins)	0.832	**0.698**	0.916	0.738	0.559
MTLR (10 bins)	0.544	0.686	0.926	0.741	0.573
MTLR (15 bins)	0.540	0.683	0.928	0.767	0.574
MTLR (20 bins)	0.713	0.683	0.922	0.763	0.568
MTLR (25 bins)	0.684	0.686	0.918	0.744	0.554



(*a*) NWTCO (*b*) WHAS500

Figure 2: Illustration of effect of survival time discretization on test C-index for 2 data sets. On the NWTCO data, MTLR is more accurate than CoxPH when choosing 5 time bins, while MTLR is less accurate than CoxPH on the WHAS500 data regardless of the number of bins.

by choosing hyperparameters that maximize validation C-index. Specifically, the CoxPH model has higher test C-index than any of the MTLR models on the FLCHAIN, WHAS500, and SRTR data sets. Thus, there does not appear to be a clear-cut answer to this question from our experiments.

6. Discussion

Our findings for RQ1 were somewhat surprising to us, as they suggest that the *actual survival times provide little value beyond their grouping into time bins!* A possible explanation for this finding is that the continuous-time prediction model is not able to predict the order of closely-timed events, and thus providing finer-grained time information does not significantly improve prediction accuracy. Another possibility may be the choice of the C-index as the metric for accuracy. The C-index depends only on the *ordering* of predictions and not on the actual predictions themselves, so a poorly calibrated predictor could still possibly achieve a high C-index. Despite this limitation, the C-index is the most commonly used accuracy metric in survival prediction, which is why we adopted it in this study.

Our findings for RQ2 were less surprising to us. As the number of time bins gets extremely small (2 bins in the most extreme case), we are discarding a lot of information about timing of events by combining events that are not closely timed into the same bin. Thus, one might expect to see prediction accuracy increase as the number of bins increases. In the multi-task binary classification formulation used in MTLR, however, increasing the number of bins also increases the number of classification tasks, which increases the number of parameters in the model. As we found for RQ1, there is very little information to be gained in survival times beyond a certain level of granularity. Hence, we eventually begin to add more parameters without adding additional signal, which suggests that prediction accuracy should begin to decrease if the number of time bins gets too high, which we do observe.

While it is tempting to choose the time bins to represent a convenient time unit, e.g. 1 year for each bin, the trade-off for the convenience may be a non-negligible decrease in prediction accuracy. This suggests that *the number of time bins should be treated as a hyperparameter to be optimized in discrete-time survival prediction models.*

Our findings for RQ3 were somewhat inconclusive, as MTLR was more accurate than CoxPH on some data sets (provided that a good choice was made for the number of bins) and less accurate on others. The accuracy of the CoxPH predictions are highly dependent upon the validity of the assumptions behind the CoxPH model, and this is true also for other continuous-time survival prediction models including parametric approaches. This dependency on assumptions is one reason why discrete-time survival prediction models that do not have such assumptions have been introduced. It may not be possible to get a clear-cut answer to this question because the prediction accuracy of continuous-time approaches will always be dependent upon the match between the data and assumptions. However, it may certainly be possible that other, more complex non-linear discrete-time approaches (including recently-proposed deep learning models) could generally outperform continuous-time approaches across a wide variety of real data sets, and this would be an logical avenue for further study.

7. Conclusion

Our main objective in this study was to examine how several differences between continuous and discrete-time survival prediction models contribute to prediction accuracy. We formed 3 specific research questions regarding these differences and empirically investigated them across 5 real data sets.

First, we observed that discretizing time does not necessarily lead to much of a decrease in prediction accuracy when using a continuous-time survival prediction model like the CoxPH, so the discretization process itself does not seem to be limiting accuracy. On the other hand, we observed that the number of time bins used for discretization can have a significant effect on the accuracy of a discrete-time survival prediction model like the multi-task logistic regression (MTLR) (Yu et al., 2011). While the number of time bins is typically chosen for convenience (e.g. 1 time bin = 1 year), our findings suggest that the number of time bins should be treated as a hyperparameter to be tuned if maximizing prediction accuracy is the goal. Finally, we find mixed results when comparing continuous and discrete-time survival prediction algorithms in terms of prediction accuracy.

A limitation of our analysis involves our choice of models. We chose to compare the CoxPH and MTLR models because they both belong to the same family of generalized linear models, which makes for a relatively fair comparison. The discrete-time formulation appears to be more convenient, as evidenced by the many recently-proposed discrete-time survival prediction algorithms (Yu et al., 2011; Li et al., 2016a; Lee et al., 2018; Giunchiglia et al., 2018; Ren et al., 2019; Wulczyn et al., 2020), and relies on less assumptions than continuous-time approaches. We intend to extend our analysis to more sophisticated non-linear models both in continuous and discrete time.

Acknowledgments

The research reported in this publication was supported by the National Library of Medicine of the National Institutes of Health under Award Number R01LM013311 as part of the NSF/NLM Generalizable Data Science Methods for Biomedical Research Program. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

The data reported here have been supplied by the Hennepin Healthcare Research Institute (HHRI) as the contractor for the Scientific Registry of Transplant Recipients (SRTR). The interpretation and reporting of these data are the responsibility of the author(s) and in no way should be seen as an official policy of or interpretation by the SRTR or the U.S. Government. Notably, the principles of the Helsinki Declaration were followed.

References

Andreas Bender, David Rügamer, Fabian Scheipl, and Bernd Bischl. A general machine learning framework for survival analysis. *arXiv preprint arXiv:2006.15442*, 2020.

Elia Biganzoli, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini. Feed forward neural networks for the analysis of censored survival data: A partial logistic regression approach. *Statistics in Medicine*, 17(10):1169–1186, 1998.

David R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

Giulio J. D'angio, Norman Breslow, J. Bruce Beckwith, Audrey Evans, Edward Baum, Alfred Delorimier, Donald Fernbach, Ellen Hrabovsky, Barbara Jones, Panayotis Kelalis, H. Biemann Othersen, Melvin Tefft, and Patrick R. M. Thomas. Treatment of Wilms' tumor. results of the third national Wilms' tumor study. *Cancer*, 64(2):349–360, 1989.

Angela Dispenzieri, Jerry A. Katzmann, Robert A. Kyle, Dirk R. Larson, Terry M. Therneau, Colin L. Colby, Raynell J. Clark, Graham P. Mead, Shaji Kumar, L. Joseph Melton, and S. Vincent Rajkumar. Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. *Mayo Clinic Proceedings*, 87 (6):517–523, 2012.

Bradley Efron. The efficiency of Cox's likelihood function for censored data. *Journal of the American Statistical Association*, 72(359):557–565, 1977.

A. Eleuteri, M. S. H. Aung, A. F. G. Taktak, B. Damato, and P. J. G. Lisboa. Continuous and discrete time survival analysis: neural network approaches. In *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5420–5423. IEEE, 2007.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1437–1446, 2018.

Stephane Fotso et al. PySurvival: Open source package for survival analysis modeling, 2019–. URL https://www.pysurvival.io/.

Eleonora Giunchiglia, Anton Nemchenko, and Mihaela van der Schaar. RNN-SURV: A deep recurrent model for survival analysis. In *Proceedings of the 27th International Conference on Artificial Neural Networks*, pages 23–32. Springer, 2018.

Robert J. Goldberg, Joel M. Gore, Joseph S. Alpert, and James E. Dalen. Recent changes in attack and survival rates of acute myocardial infarction (1975 through 1981): The worcester heart attack study. *JAMA*, 255(20):2774–2779, 1986.

Edward L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical association*, 53(282):457–481, 1958.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Håvard Kvamme and Ørnulf Borgan. Continuous and discrete-time survival prediction with neural networks. *arXiv preprint arXiv:1910.06724*, 2019.

Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. DeepHit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 2314–2321, 2018.

Elisa T. Lee and John Wenyu Wang. *Statistical methods for survival data analysis*. John Wiley & Sons, 3rd edition, 2003.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18:1–52, 2018.

Yan Li, Jie Wang, Jieping Ye, and Chandan K. Reddy. A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1715–1724, 2016a.

Yan Li, Kevin S. Xu, and Chandan K Reddy. Regularized parametric regression for high-dimensional survival analysis. In *Proceedings of the SIAM International Conference on Data Mining*, pages 765–773. SIAM, 2016b.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

P. J. G. Lisboa, H. Wong, P. Harris, and R. Swindell. A Bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. *Artificial Intelligence in Medicine*, 28(1):1–25, 2003.

Mohammadreza Nemati, Haonan Zhang, Michael Sloma, Dulat Bekbolsynov, Hong Wang, Stanislaw Stepkowski, and Kevin S. Xu. Predicting kidney transplant survival using multiple feature representations for HLAs. *arXiv preprint arXiv:2103.03305*, 2021.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21:1–6, 2020.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4798–4805, 2019.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for Cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

Hans C. van Houwelingen, Tako Bruinsma, Augustinus A. M. Hart, Laura J. van't Veer, and Lodewyk F. A. Wessels. Cross-validated Cox regression on microarray gene expression data. *Statistics in Medicine*, 25(18):3201–3216, 2006.

Laura J. van't Veer, Hongyue Dai, Marc J. van de Vijver, Yudong D. He, Augustinus A. M. Hart, Mao Mao, Hans L. Peterse, Karin van der Kooy, Matthew J. Marton, Anke T. Witteveen, George J. Schreiber, Ron M. Kerkhoven, Chris Roberts, Peter S. Linsley, René Bernards, and Stephen H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002.

Ping Wang, Yan Li, and Chandan K Reddy. Machine learning for survival analysis: A survey. *ACM Computing Surveys*, 51(6):1–36, 2019.

Ellery Wulczyn, David F Steiner, Zhaoyang Xu, Apaar Sadhwani, Hongwu Wang, Isabelle Flament-Auvigne, Craig H Mermel, Po-Hsuan Cameron Chen, Yun Liu, and Martin C Stumpe. Deep learning-based survival prediction for multiple cancer types using histopathology images. *PLOS ONE*, 15(6):e0233678, 2020.

Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In *Advances in Neural Information Processing Systems 24*, pages 1845–1853, 2011.