Penalized Casebase in Survival Analysis

Trevor Kwan, Gabriela V. Cohen Freue, David Soave, Sahir Bhatnagar

August 29, 2022

1 Summary

A quantity of interest in clinical studies is the absolute risk of an event given a patient's covariate profile. In order for the Cox proportional hazard method to recover these absolute risk curves, the baseline hazard needs to be estimated separately, resulting in stepwise estimates in absolute risk that make the curves difficult to interpret. Casebase is an approach that transforms the data into a logistic regression, allowing for survival estimates that vary smoothly over time. We simulate data and compare the prediction performance of penalized Casebase methods with penalized and unpenalized Cox methods. Casebase methods perform better in measures of concordance and time-dependent brier scores.

2 Introduction

From a clinical viewpoint, the most important measure is often the 5 or 10 year risk of experiencing an event given a patient's covariate profile. Thus, it is important to accurately estimate the full hazard function, which can be used to estimate the cumulative incidence function (CIF) (Bhatnagar et al., 2020). Cumulative incidence, also known as absolute risk, is the probability that a person at risk of the disease at a certain time interval will be diagnosed with that disease.

One of the most popular methods used in survival analysis is the Cox proportional hazards model. It provides a flexible way of measuring the influence of covariates on the hazard function, but this flexibility comes at the cost of separating the baseline hazard from the effect of the covariates (Cox, 1972). Then to recover the cumulative incidence function (CIF), we need to separately estimate the baseline hazard, often using the Breslow estimator. This leads to stepwise estimates in the absolute risk curves that are difficult to interpret (Breslow, 1972).

Another method to obtain the absolute risk is to use parametric models such as the Weibull model or the exponential model. But these models require us to

make distributional assumptions on the survival time.

Casebase is a framework for estimating fully parametric hazard models via logistic regression (Hanley Miettinen, 2009). It compares person-moments when patients were at risk of the event "cases" with person-moments when the event of interest occurred "bases". The parametric nature of Casebase allows us to create smooth cumulative incidence curves while the data transformation allows us to consider time and censoring as well. (Bhatnagar et al., 2020).

The goal of my project is to compare Casebase methods with more classical survival analysis methods in terms of prediction performance and variable selection. We do so by designing a simulation study comparing estimators and exploring various simulation settings. We demonstrate how Casebase further overcomes the limitations of the penalized Cox.

3 Methodology

3.1 Generating the Data

The simulation settings used to generate data was based on Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent (Simon et al., 2011).

First, we generated standard Gaussian predictor data, $\boldsymbol{X}$, with p predictors and n observations. For any 2 predictors $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, we fixed the pairwise correlation at ρ, for several values of ρ, such that the correlation between any two predictor columns would not exceed ρ.

Then, we generated failure times for each observation Y_i, based on $Y_i = e^{\sum_{j=1}^{p} X_{ij}^T \beta_j + k * E_i}$, where $\beta_j = (-1)^j * e^{-2(j-1)/(p-z)}$, $E_i \sim N(0,1)$, and k is chosen based on our desired signal-to-noise ratio. The true $\boldsymbol{\beta}$ coefficients were constructed to have alternating signs and to be decreasing exponentially.

We generated censored times for each observation $C_i = e^{k * E_i}$. The resulting event time was the minimum of the failure time and censoring time, $T_i = min\{Y_i, C_i\}$. If the censoring time preceeded the failure time, $C_i < Y_i$, the observation was said to be censored.

Table 1 shows what the generated dataset would look like.

$status$	$time(\boldsymbol{T})$	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	...
0	1.420	-0.019	0.373	1.498	-1.532	1.605	...
1	0.091	-0.270	-0.633	0.283	0.706	-0.032	...
1	0.274	1.224	-0.466	0.681	1.075	0.554	...
0	2.731	-2.946	-3.041	-3.178	-4.0398	3.273	...
1	0.198	-0.080	1.402	-0.512	-0.561	-0.721	...

Table 1: Simulated Dataset

3.2 Fitting the Models

We fit 5 models for comparison: the unpenalized Cox, lasso penalized Cox, ridge penalized Cox, lasso penalized Casebase, and ridge penalized Casebase.

3.2.1 Unpenalized Cox

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ be the $p * 1$ vector of covariates associated with y_i for the i^{th} observation. And let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ be the $p * 1$ vector of regression coefficients. Denote $t_{i=1} \ldots t_m$ as the increasing list of unique failure times.

The unpenalized Cox method finds the $\boldsymbol{\beta}$ coefficients that maximize the likelihood function $L(\boldsymbol{\beta}) = \prod_{i=1}^{m} \frac{e^{\boldsymbol{x}_{j(i)}^T \boldsymbol{\beta}}}{\sum_{j \in R_i} e^{\boldsymbol{x}_j^T \boldsymbol{\beta}}}$, where $j(i)$ is the index of the observation failing at time t_i, $j \in R_i$ is the set of indices still at risk at time t_i, and $\prod_{i=1}^{m}$ multiplies through observations by order of event times. $L(\boldsymbol{\beta})$ is a partial likelihood because it does not consider probabilities for subjects who are censored and it is based on the observed order of events, meaning it ignores all information between failure times. A key property of the Cox proportional hazards model is that the baseline hazards cancel in the computation of the maximum likelihood, so we solve for the $\boldsymbol{\beta}$'s that will maximize this likelihood without needing to specify the baseline hazard.

The unpenalized Cox model performs well when we have many more observations n than predictors p. However, when $p > n$, to maximize the partial likelihood, all of the $\boldsymbol{\beta}_j$'s are sent to $\pm\infty$ (Simon et al., 2011).

3.2.2 Penalized Cox

To combat the problem of $p > n$ in unpenalized Cox, we use a penalized Cox, one penalization of which is the lasso l_1 penalization, which allows for a solution with few nonzero β_j's without the issue of β_j's being sent to $\pm\infty$. And even in the $n > p$ case, as long as p is close enough to n, this may lead to more accurate

$\boldsymbol{\beta}$ estimates than the unpenalized Cox (Simon et al., 2011).

Ridge and lasso penalization is particularly useful in $p > n$ situations, and ridge penalization particularly useful where there are many correlated predictor variables. Ridge shrinks the $\boldsymbol{\beta}$ coefficients of correlated predictors towards each other, allowing them to "borrow strength" from each other. Ridge penalization is ideal if the number of predictors is large and their $\boldsymbol{\beta}$ coefficients are all non-zero. Lasso penalization performs better when out of the true $\boldsymbol{\beta}$ coefficients many are close to zero and a smaller subset of coefficients are nonzero and larger (Friedman et al., 2008).

The elastic net penalized Cox method finds the $\boldsymbol{\beta}$ coefficients that maximize $log(L(\boldsymbol{\beta})) - \lambda(\alpha \sum_{j=1}^{p} |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2)$, where α is the relative weight of the l_1 lasso and l_2 ridge penalty, and λ controls the level of penalization. The penalty turns into a lasso penalty when $\alpha = 1$ or a ridge penalty when $\alpha = 0$ (Simon et al., 2011).

We used 10-fold cross-validation to search for the λ value out of 100 λ's that corresponded to the lowest mean cross-validation error. For each λ, the training data is split into 10 folds. We take 1 fold out as the validation set and take the remaining folds as the training set. For each of the 10 unique validation set folds, we fit the model on the training set. We do this fitting by searching for the $\boldsymbol{\beta}$ coefficients that minimize the penalized negative partial likelihood for the given λ. Then we evaluate it on the validation set to yield 10 cross-validated errors. The mean of the cross-validated errors for a given λ is the mean of the partial likelihood deviance. The λ that resulted in the lowest mean cross-validated error was used as the penalty level in the final penalized Cox model fit for both ridge and lasso.

3.2.3 Penalized Casebase

The penalized Casebase method first calls on *sampleCasebase()* to get the case series and base series. The case series consists of all observations that are "cases", or all that have experienced the event. There are several steps in computing the base series. First, we specify a ratio value, the default is ratio = 10. This ratio determines the size of the base series where $b = ratio * c$, b is the base series size and c is the total number of cases in the training data. We sample b bases out of all n observations in the dataset with replacement, each observation having a unique probability of being selected. Each observation's probability of being selected is equal to the observation's event time divided by B, where B is the sum of all times from all observations in the training data. The selected observations are treated to be still at risk of the event, and their event times are all multiplied by a random number from $Unif(0, 1)$. Finally, the Casebase series dataframe combines the case series and base series, and an additional column with all equivalent offset values of $log(B/b)$ is added to it.

The resulting Casebase series will have a total of $(ratio * c) + c$ observation rows and $p+3$ columns. The 3 columns in addition to the predictor columns represent the censoring status, event time, and offset values. Next, Casebase calls on *fitSmoothHazard.fit()* to fit the model using the Casebase series data. First, it adds the $log(time)$ as an additional predictor column to $\boldsymbol{X}$. The resulting $\boldsymbol{X}$ data frame consists of all the predictors in the Casebase series plus $log(time)$. Using $log(time)$ as a predictor column yields the Weibull hazard which allows for a power dependence of the hazard on time $log(h(t; \boldsymbol{X}_i)) = \beta_0 + \beta_1 log(t) + \boldsymbol{X}_i^T \boldsymbol{\beta}$ where $t = time$. $\boldsymbol{Y}$ is a vector of 0's or 1's representing whether the observation is a case or a base in the Casebase series. *Table 2* shows what the Casebase series would look before fitting it on a model.

$\boldsymbol{y}$ (death)	$log(time)$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	...
0	-5.773	-0.019	0.373	1.498	-1.532	...
1	-3.396	-0.270	-0.633	0.283	0.706	...
1	-0.961	1.224	-0.466	0.681	1.075	...
0	-1.526	-2.946	-3.041	-3.178	-4.0398	...
1	-1.789	-0.080	1.402	-0.512	-0.561	...

Table 2: Casebase Series

Finally, we fit a logistic regression model from the binomial family on the $\boldsymbol{X}$ and $\boldsymbol{Y}$ Casebase series data. Logistic regression uses the logit link function $log(\frac{p(\boldsymbol{x}_i)}{(1-p(\boldsymbol{x}_i)}) = \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}$ to model it's log likelihood. Elastic net penalized logistic regression finds the $\boldsymbol{\beta}$ coefficients that minimize $-[\sum_{i=1}^{N} y_i * (\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}) - log(1+e^{\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}})] + \lambda(\alpha \sum_{j=1}^{p} |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2)$, where α is the relative weight of the l_1 lasso and l_2 ridge penalty, and λ controls the level of penalization. The penalty turns into a lasso penalty when $\alpha = 1$ or a ridge penalty when $\alpha = 0$ (Simon et al., 2011).

We used 10-fold cross-validation to search for the λ value out of 100 λ's that corresponded to the lowest mean cross-validation error similar to the cross-validation methodology in penalized Cox. For the penalized Casebase, the cross-validation error was defined as the binomial deviance. Finally, for each intercept corresponding to a λ value, we fix each intercept value by subtracting it with the offset value $log(B/b)$. Each λ represents a different value of penalty strength and thus fits different logistic regression models that yield different β_0 intercepts.

3.3 Evaluating the Metrics

3.3.1 Variable Selection

Several variable selection measures were used to comparison the quality of variable selection in a lasso Cox model and a lasso Casebase model. True positive (TP) is when the model estimates β_j to be non-zero and it is actually non-zero. True negative (TN) is when the model estimates β_j to be zero and it is actually zero. False positive (FP) is when the model estimates β_j to be non-zero and it is actually zero. False negative (FN) is when the model estimates β_j to be zero and it is actually non-zero. This is illustrated with a confusion matrix in *Figure 1.*

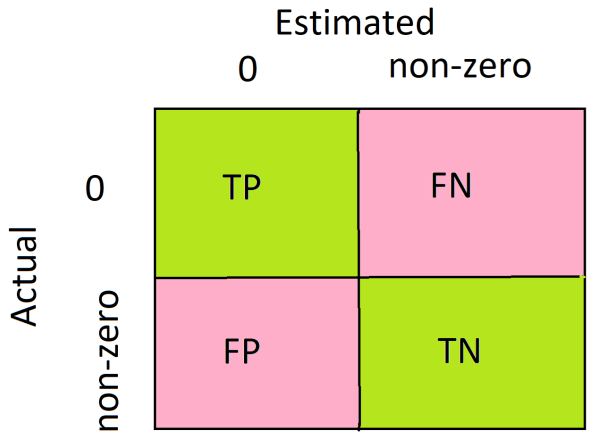

Figure 1: Confusion Matrix

Matthew's correlation coefficient (MCC) is defined as $MCC = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$. True positive rate, or sensitivity, is defined as $\frac{TP}{(TP+FN)}$. True negative rate, or specificity, is defined as $\frac{TN}{(TN+FP)}$. False positive rate is defined as $1-TNR$ and false negative rate is defined as $1-TPR$. A higher MCC, TPR, or TNR indicates better variable selection performance and a higher FPR or FNR indicates worse performance.

3.3.2 Cumulative Incidence Curves

To compute cumulative incidence curves for Cox proportional hazard models, we need to first estimate survival curves. The survival curve of the Cox proportional hazard model for an observation is defined as $S(t|\boldsymbol{X}_i) = exp\{-\int_0^t h_0(t)e^{\boldsymbol{X}_i^T\boldsymbol{\beta}}dt\}$. In order to estimate the survival function, we must estimate the hazard function

$h(t|\boldsymbol{X}_i) = h_0(t)e^{\{\boldsymbol{X}_i^T\boldsymbol{\beta}\}}$, which requires us to separately estimate the baseline hazard $h_0(t)$ for each time t. The baseline hazard is estimated using the Breslow estimator.

The Breslow estimator is defined as $\hat{\Lambda}_0(t) = \sum_{i=1}^{n} \frac{I(T_i \leq t)\Delta_i}{\sum_{j \in R_i} e^{\boldsymbol{x}_j^T\boldsymbol{\beta}}}$, where $\Delta_i = I(Y_i \leq C_i)$ and Y_i is the failure time. $I(T_i \leq t)\Delta_i$ allows only uncensored event times before t to be evaluated in the summation, and the denominator is a summation of those still at risk at time t. With the Breslow estimator, as time changes, the survival probability doesn't change until another event occurs, which leads to piece-wise estimates in the cumulative incidence function. The cumulative incidence function for an observation is defined here as $1 - S(t|\boldsymbol{X}_i)$.

For penalized Casebase models, we use the power dependence of the hazard on time to yield the Weibull hazard with $log(t)$ such that $log(h(t; \boldsymbol{X}_i)) = \beta_0 + \beta_1 log(t) + \boldsymbol{X}_i^T\boldsymbol{\beta}$. The cumulative incidence function is defined as $1 - S(t|\boldsymbol{X}_i)$, where $S(t|\boldsymbol{X}_i) = exp(-\int_0^t h(u; \boldsymbol{X}_i)du)$. The integral was computed using numerical integration.

Cumulative incidence curves were plotted as the mean probability of failure for all observations at each time t. A cumulative incidence curve for each method was plotted for comparison.

3.3.3 Concordance

Concordance compares pairs of observations to see if the patient that the model predicts to have higher risk is actually the patient that dies first.

Consider *Figure 2*. Suppose we compare 2 uncensored cases, patient 1 and patient 2. In this case, it is clear that P1 dies before P2. Thus, this pair is evaluable. But when we introduce censoring, that is not always the case. Between P2 and P3, even though P3 is right-censored, we can still see that P2 dies before P3. But let's say we compare P2 and P4. Because P4 is right-censored, we have no way of knowing whether P2 died first or P4 died first. The same goes for comparing P3 and P4, both observations are censored and thus the order of failure is unclear. Concordance only compares pairs where we can know the true survival order, and does not consider pairs where the survival order is unclear.

Concordance is the fraction of all possible evaluable pairs of observations that are concordant, over all the other evaluable pairs. A pair is concordant if the model correctly predicts which observation in the pair is going to "fail" first. And a pair is discordant if the model incorrectly predicts which observation in the pair is going to "fail" first. The observation with the higher risk score is predicted to fail first. A higher Concordance value implies better prediction performance.

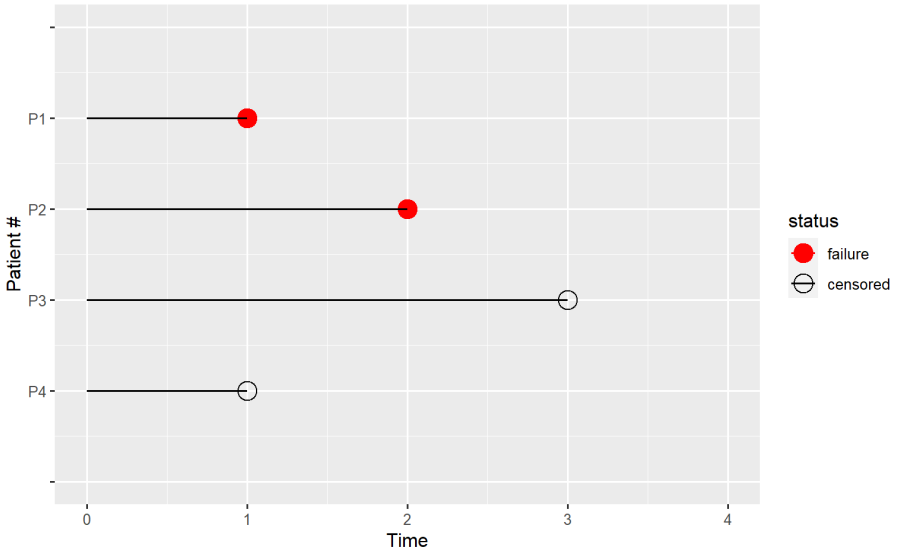

Figure 2: The Problem of Censoring in Concordance

For the unpenalized and penalized Cox models, the risk score was defined as the linear predictor $\boldsymbol{x}_i^T\boldsymbol{\beta}$ for each observation i. For the penalized Casebase models, the risk score was defined as the probability of failure $p(Y_i = 1) = \frac{1}{1+exp(-\boldsymbol{x}_i^T\boldsymbol{\beta})}$ for each observation i.

Let R_1 and R_2 represent the risk scores for a given pair, and T_1 and T_2 the true event times for a given pair.

For each evaluable pair, there are only 4 possible cases:

1. $R_1 > R_2$ and $T_1 < T_2$ OR $R_1 < R_2$ and $T_1 > T_2$: The pair is concordant (C).
2. $R_1 > R_2$ and $T_1 > T_2$ OR $R_1 < R_2$ and $T_1 < T_2$: The pair is discordant (D).
3. $R_1 == R_2$: The risk scores are equal (R).
4. $T_1 == T_2$: The times are equal (T).

The last case where the times are equal is not taken into account to estimate the concordance.

The concordance index is:

$$Concordance = \frac{C + \frac{R}{2}}{C + D + R}$$

3.3.4 Time-Dependent Brier Scores

The method of acquiring Time-Dependent Brier Scores was based on Regularized Regression for Two Phase Failure Time Studies (Soave, 2021).

Similar to the concordance, not all observations are evaluable. *Figure 3* and *Figure 4* show cases in which observations are evaluable. Consider Patient 1 (P1) in *Figure 3*. The red dot represents an uncensored observation, and in this case it is clear that the event occurs before time t. In *Figure 4*, patient 2 (P2) is also uncensored and thus it is clear that it occurs after t. Patient 3 (P3) also occurs after t, but because it is right-censored, we know that the event has to occur sometime after t.

Figure 5 shows the case where an observation is not evaluable. In this case, patient 4 (P4) is a censored observation that occurs before t. Because the observation is right-censored, we have no way of knowing whether or not the true event occurred before or after time t.

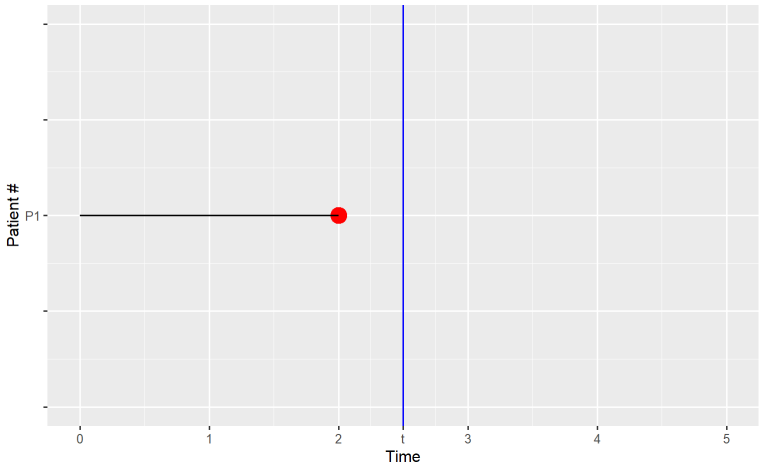

Figure 3: The Problem of Censoring in Brier Scores (Evaluable Case)

The Time-Dependent Brier Scores are computed as:

$$BS(t) = \sum_i \{\frac{I(C_i \geq min(Y_i, t))}{\hat{G}(min(Y_i, t))}\}\{I(Y_i \leq t) - \hat{F}(t|\boldsymbol{x}_i)\}^2$$

where Y_i represents the failure time for observation i, t represents the specified time t being evaluated, and C_i represents the censoring time.

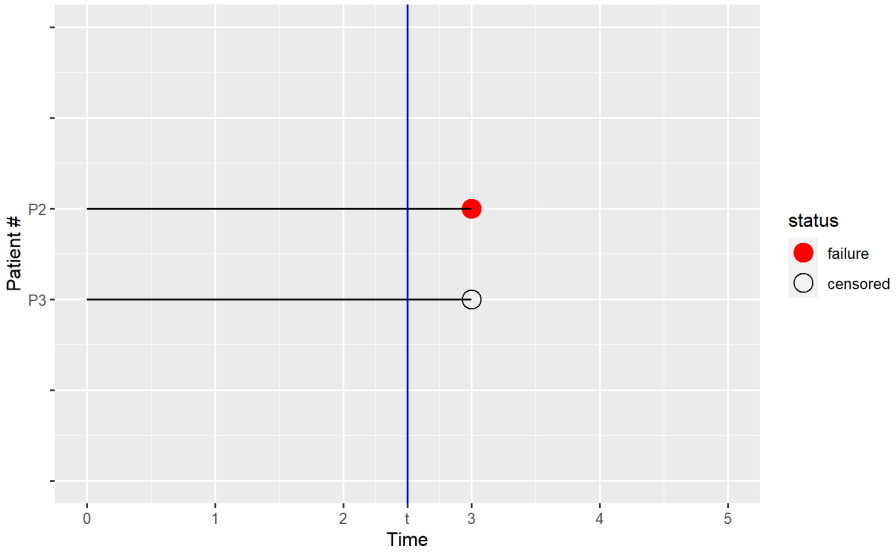

Figure 4: The Problem of Censoring in Brier Scores (Evaluable Case)

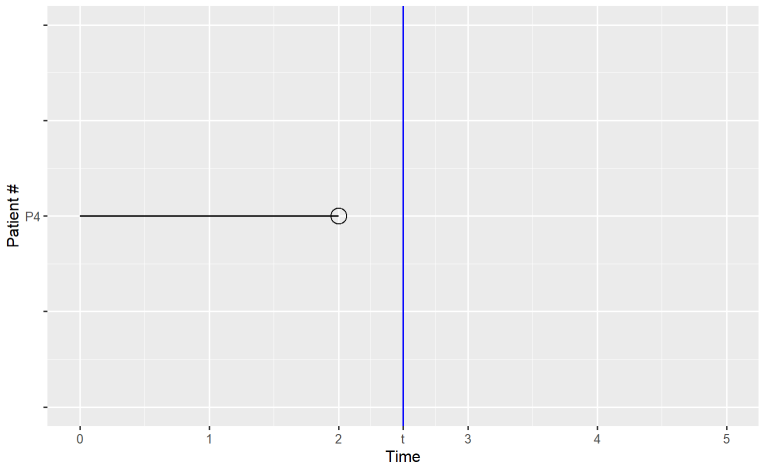

Figure 5: The Problem of Censoring in Brier Scores (Non-evaluable Case)

$I(C_i \geq min(Y_i, t))$ is an indicator function that allows us to include all uncensored observations and only censored observations with event time $\geq t$.

Cases where the observation is censored before t are not explicitly included in the Brier Score computation because there is no way of determining whether the failure for that event comes before or after t. For an uncensored observation, we assume that $C_i > Y_i$, so $I(C_i \geq min(Y_i, t)) = 1$ even if $Y_i > t$.

$\hat{G}(min(Y_i, t))$ gives the Kaplan-Meier estimate of the probability of survival for the minimum of the failure time and t. As t increases, this function evaluates to values that grow larger, putting more weight on observations at later time points. This is because as time goes on, more and more censored observations are unaccounted for, so this weighting compensates for that.

$I(Y_i \leq t)$ indicates the status of whether or not the observation has occurred at t. And $\hat{F}(t|\boldsymbol{x}_i)$ gives the probability of failure given observation i with covariates $\boldsymbol{x}_i$ at time t. For Cox models, this is $1 - S(t|\boldsymbol{X}_i)$, where $S(t|\boldsymbol{X}_i) = exp\{-\int_0^t h_0(t)e^{\boldsymbol{X}_i^T\boldsymbol{\beta}}dt\}$ models the survival probability at each time t. For Casebase models, this is $1 - S(t|\boldsymbol{X}_i)$, where $S(t|\boldsymbol{X}_i) = exp(-\int_0^t h(u; \boldsymbol{X}_i)du)$. And $h(u; \boldsymbol{X}_i)$ is the Weibull hazard where $h(t; \boldsymbol{X}_i) = e^{\beta_0 + \beta_1 log(t) + \boldsymbol{X}_i^T\boldsymbol{\beta}}$.

The time-dependent brier scores are a measure of prediction accuracy across time. It measures the squared difference between the actual status of the event and the predicted probability of the event happening. It does not explicitly include censored observations before t, but implicitly accounts for it by weighting observations at later t times.

4 Results

4.1 Simulation Setting 1: $n > p$

$$n = 500, p = 120, z = 100, snr = 3, \rho = 0.5$$

For the first simulation setting where we set the number of observations to be 500, number of predictors to be 120, number of 0's in $\boldsymbol{\beta}$ coefficients to be 100, a signal-to-noise ratio of 3, and ρ value of 0.5, we found the cumulative incidence curves to be smoother-in-time for Casebase models compared to Cox models *(see Figure 6)*.

In terms of variable selection, we found lasso Cox to perform a little better than lasso Casebase in all 3 measures of MCC, TPR, and TNR *(see Table 3)*.

But in terms of prediction performance, lasso and ridge Casebase perform better than lasso and ridge Cox, as shown in both the concordance values and brier score curves *(see Table 4 and Figure 7)*.

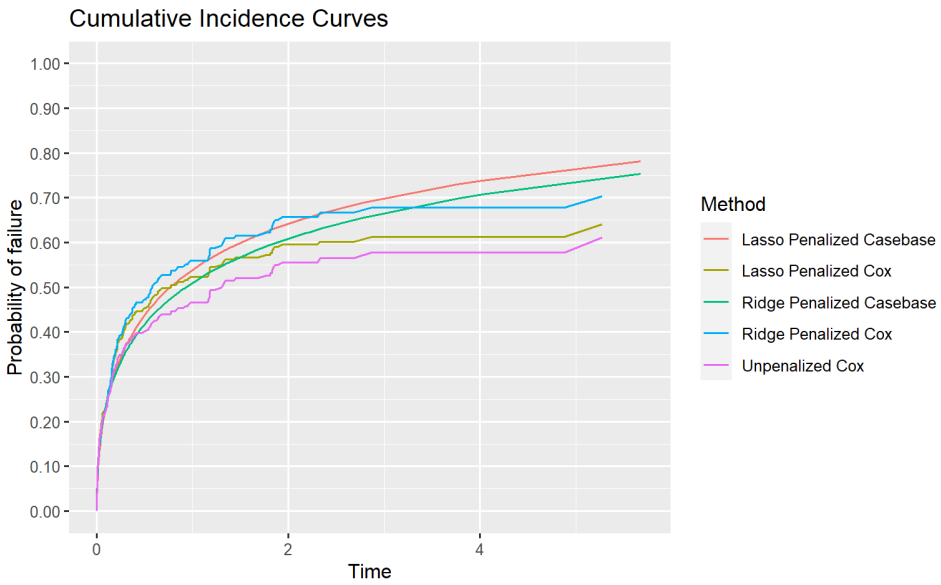

Figure 6: Cumulative Incidence Curves (Setting 1)

Measure	Lasso Cox	Lasso Casebase
MCC	0.525	0.369
TPR	0.850	0.700
TNR	0.800	0.760
FPR	0.200	0.240
FNR	0.150	0.300

Table 3: Variable Selection (Setting 1)

Method	Concordance
Unpenalized Cox	0.760
Lasso Penalized Cox	0.830
Ridge Penalized Cox	0.804
Lasso Penalized Casebase	0.873
Ridge Penalized Casebase	0.888

Table 4: Concordance (Setting 1)

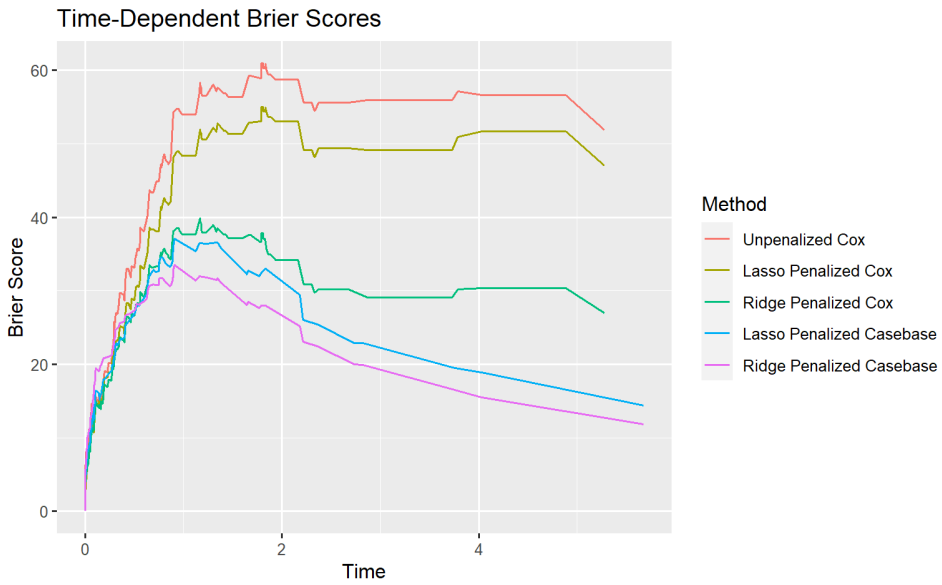

Figure 7: Time-Dependent Brier Scores (Setting 1)

4.2 Simulation Setting 2: $p > n$

$$n = 100, p = 200, z = 150, snr = 3, \rho = 0.5$$

For the second simulation setting, we tweaked the parameters where the number of predictors is now greater than the number of observations. And once again, the cumulative incidence curves are smoother for Casebase models *(see Figure 8)*.

In this case where $p > n$, the TPR is very low for both models but the TNR is very high for both models. This is because both lasso models are estimating a lot of the $\boldsymbol{\beta}$ coefficients to be 0 to be safe, even the actual nonzero ones. Thus, this leads to less true positives and more true negatives. There were 150 actual 0's in this setting, so it's not surprising that they both have a higher TNR *(see Table 5 and Table 6)*.

It is expected and consistent with past literature that in $p > n$ situations the unpenalized Cox doesn't perform well, but what is more interesting is that it seems the ridge penalized Cox is doing just as bad in terms of prediction performance *(see Table 7)*.

We looked into the estimated $\boldsymbol{\beta}$'s of the models to see how they would differ from the actual $\boldsymbol{\beta}$'s. The unpenalized Cox behaved as we thought it would, to

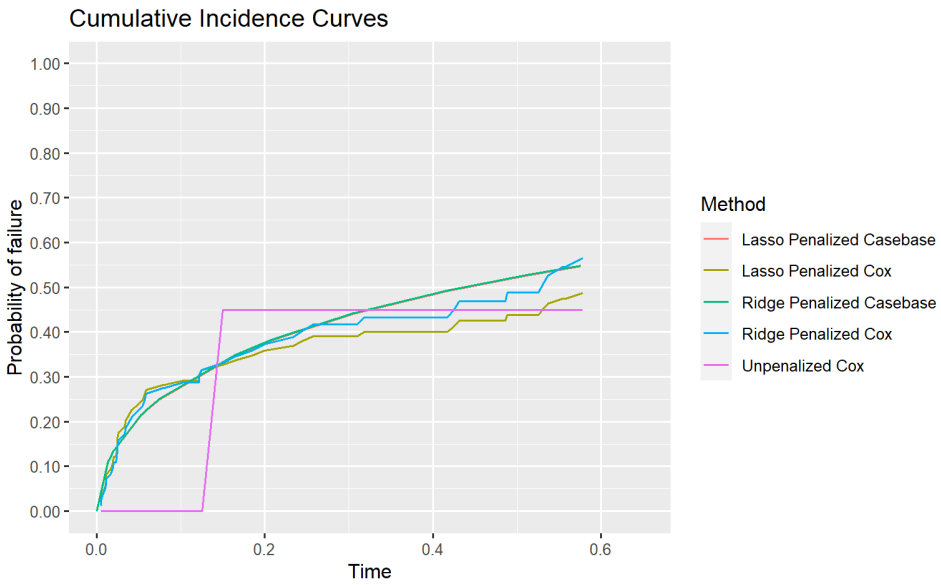

Figure 8: Cumulative Incidence Curves (Setting 2)

Measure	Lasso Cox	Lasso Casebase
MCC	0.243	0.176
TPR	0.160	0.140
TNR	0.973	0.960
FPR	0.027	0.040
FNR	0.840	0.860

Table 5: Variable Selection (Setting 2)

have really large positive or negative $\boldsymbol{\beta}$ values. But the ridge Cox seemed to estimate larger $\boldsymbol{\beta}$ values in comparison to ridge Casebase which estimated $\boldsymbol{\beta}$ coefficients that were essentially 0. It seems that because 150 of the $\boldsymbol{\beta}$'s were actually 0, ridge Casebase yielded better estimations *(see Table 8)*.

Brier Score curves showed the unpenalized Cox to perform the worst, penalized Cox to perform slightly better, and penalized Casebase to perform a bit better than that *(see Figure 9)*.

β_j	Actual $\boldsymbol{\beta}$'s	Lasso Cox $\boldsymbol{\beta}$'s	Lasso Casebase $\boldsymbol{\beta}$'s
β_1	-1.000	0.000	-0.350
β_2	0.905	-0.167	-0.399
β_3	-0.819	0.079	0.000
β_4	0.741	-0.176	-0.161
β_5	-0.670	0.068	0.000
β_6	0.606	0.000	-0.232
β_7	-0.549	0.028	0.044
β_8	0.497	0.000	0.000
β_9	-0.449	0.000	0.000
β_{10}	0.407	0.000	0.000

Table 6: $\boldsymbol{\beta}$ Comparison (Setting 2)

Method	Concordance
Unpenalized Cox	0.342
Lasso Penalized Cox	0.623
Ridge Penalized Cox	0.342
Lasso Penalized Casebase	0.894
Ridge Penalized Casebase	1.000

Table 7: Concordance (Setting 2)

β_j	Actual $\boldsymbol{\beta}$'s	Unpen Cox $\boldsymbol{\beta}$'s	Ridge Cox $\boldsymbol{\beta}$'s	Ridge Casebase $\boldsymbol{\beta}$'s
β_1	-1.000	20.837	0.006	-0.688
β_2	0.905	198.217	-0.017	-0.526
β_3	-0.819	406.150	0.012	5.160e-39
β_4	0.741	613.539	-0.022	-1.227e-38
β_5	-0.670	1000.745	0.015	7.100e-39
β_6	0.606	-1408.957	-0.009	-1.563e-38
β_7	-0.549	123.732	0.012	9.142e-39
β_8	0.497	214.724	-0.008	-1.225e-38
β_9	-0.449	999.303	0.011	1.025e-38
β_{10}	0.407	-652.457	-0.009	-6.516e-39

Table 8: $\boldsymbol{\beta}$ Comparison (Setting 2)

4.3 Simulation Setting 3: Correlated Predictors

$$n = 500, p = 120, z = 100, snr = 3, \rho = 0.9$$

In this setting we explored the effect of having correlated predictors, where

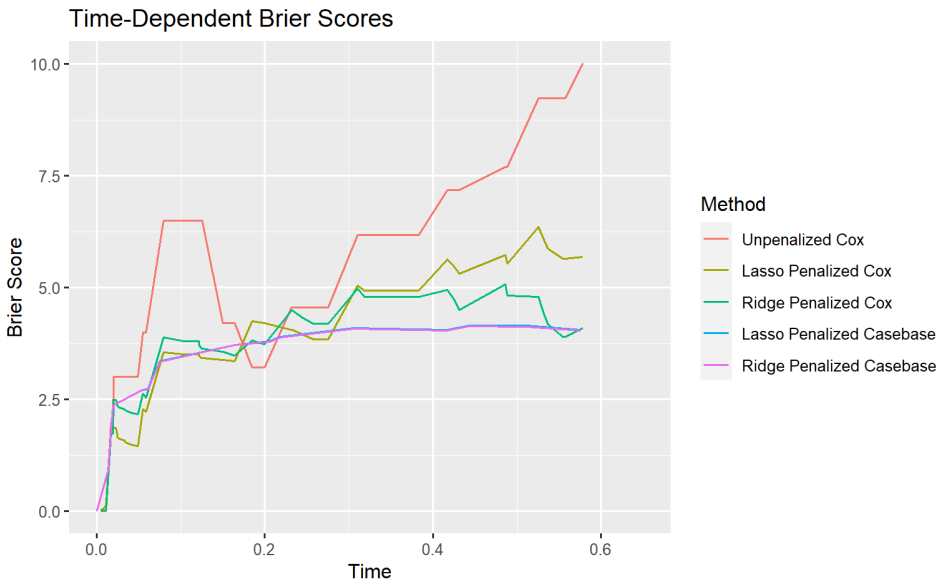

Figure 9: Time-Dependent Brier Scores (Setting 2)

rho = 0.9.

Lasso Cox and lasso Casebase were found to be quite similar in variable selection performance *(see Table 9)*.

Measure	Lasso Cox	Lasso Casebase
MCC	0.454	0.376
TPR	0.800	0.850
TNR	0.770	0.650
FPR	0.230	0.350
FNR	0.200	0.150

Table 9: Variable Selection (Setting 3)

In a correlated predictors setting, we expected ridge penalization models to perform better than the unpenalized model, and the concordance and brier score results are consistent with that, with Casebase models performing the best *(see Table 10 and Figure 11)*.

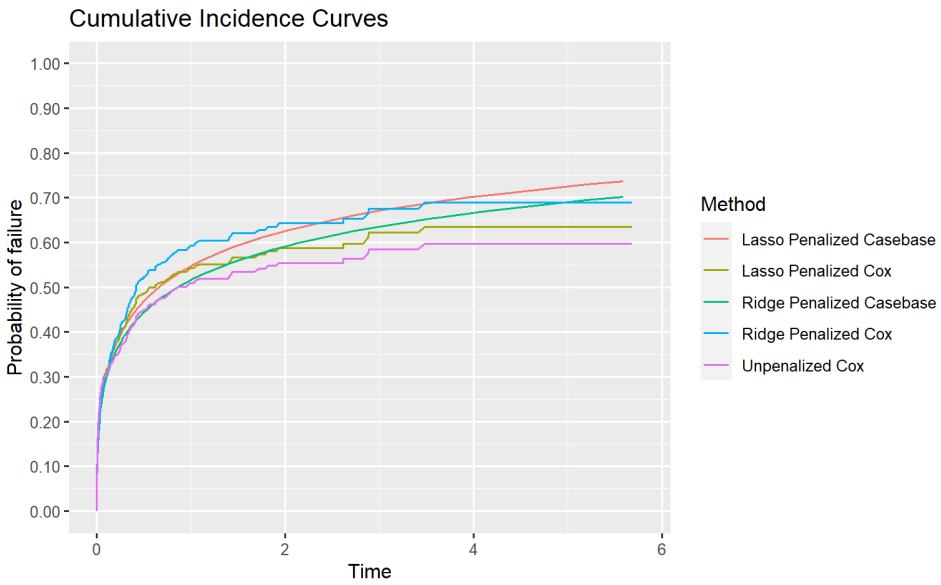

Figure 10: Cumulative Incidence Curves (Setting 3)

Method	Concordance
Unpenalized Cox	0.755
Lasso Penalized Cox	0.808
Ridge Penalized Cox	0.794
Lasso Penalized Casebase	0.887
Ridge Penalized Casebase	0.935

Table 10: Concordance (Setting 3)

4.4 Simulation Setting 4: Uncorrelated Predictors

$$n = 500, p = 120, z = 100, snr = 3, \rho = 0.1$$

In the next setting we fixed ρ, the correlation between any two predictors to be 0.1.

Lasso Cox performed slightly better in terms of variable selection performance *(see Table 11)*.

In this setting the predictors are uncorrelated, so the difference in concordance performance between unpenalized and ridge penalization models was not

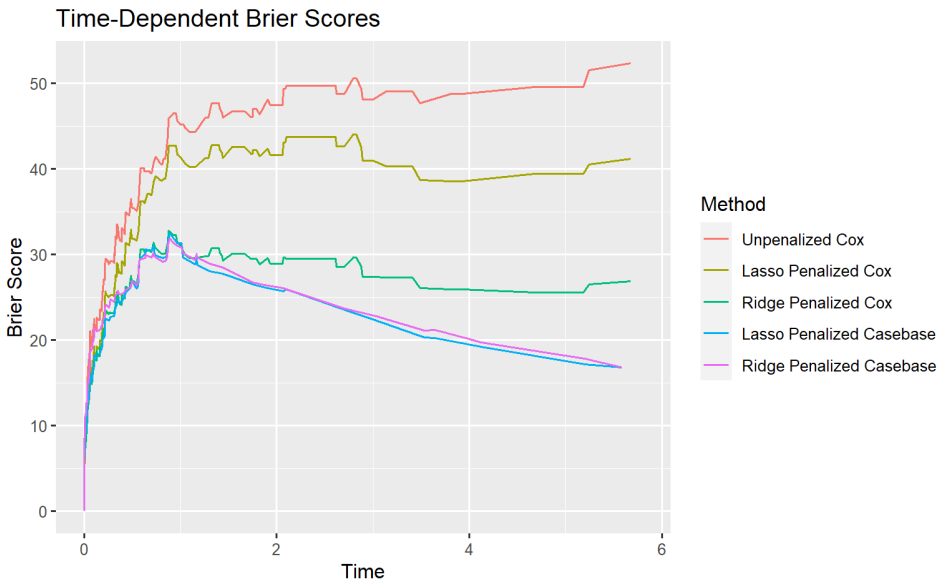

Figure 11: Time-Dependent Brier Scores (Setting 3)

Measure	Lasso Cox	Lasso Casebase
MCC	0.538	0.307
TPR	0.900	0.800
TNR	0.780	0.610
FPR	0.220	0.390
FNR	0.100	0.200

Table 11: Variable Selection (Setting 4)

as large *(see Table 12)*.

But for both concordance and brier score, penalized models again performed better than the unpenalized model, with Casebase models performing slightly better *(see Table 12 and Figure 13)*.

4.5 Simulation Setting 5: High Signal-to-noise Ratio

$$n = 500, p = 120, z = 100, snr = 7, \rho = 0.5$$

In this setting we set the signal-to-noise ratio to be 7, which is rather high.

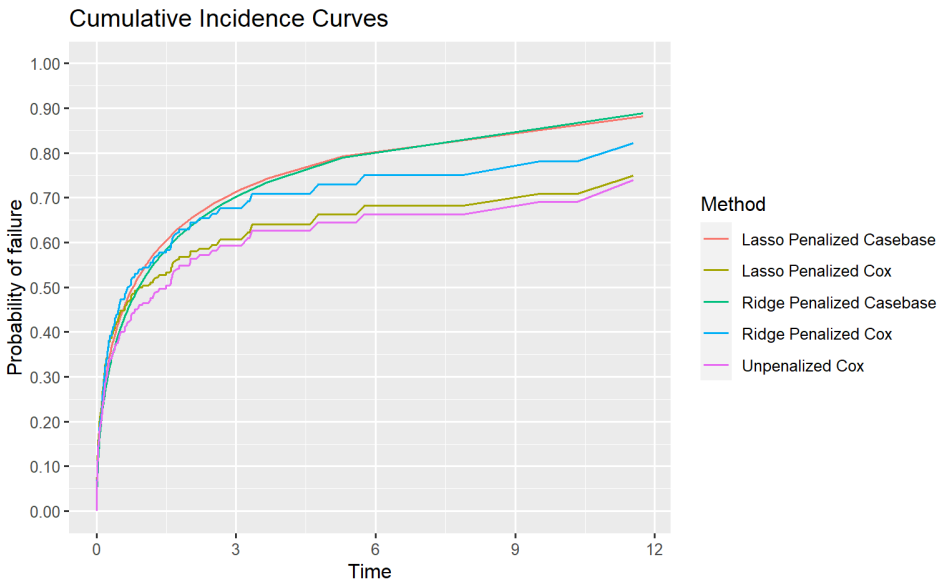

Figure 12: Cumulative Incidence Curves (Setting 4)

Method	Concordance
Unpenalized Cox	0.762
Lasso Penalized Cox	0.829
Ridge Penalized Cox	0.807
Lasso Penalized Casebase	0.878
Ridge Penalized Casebase	0.866

Table 12: Concordance (Setting 4)

Lasso Cox performed slightly better than lasso Casebase for variable selection *(see Table 13)*.

And as expected with a high signal-to-noise ratio, all the models performed well and were close to each other in terms of concordance performance *(see Table 14)*.

Brier Scores show Casebase to perform better than other models in this setting *(see Figure 15)*.

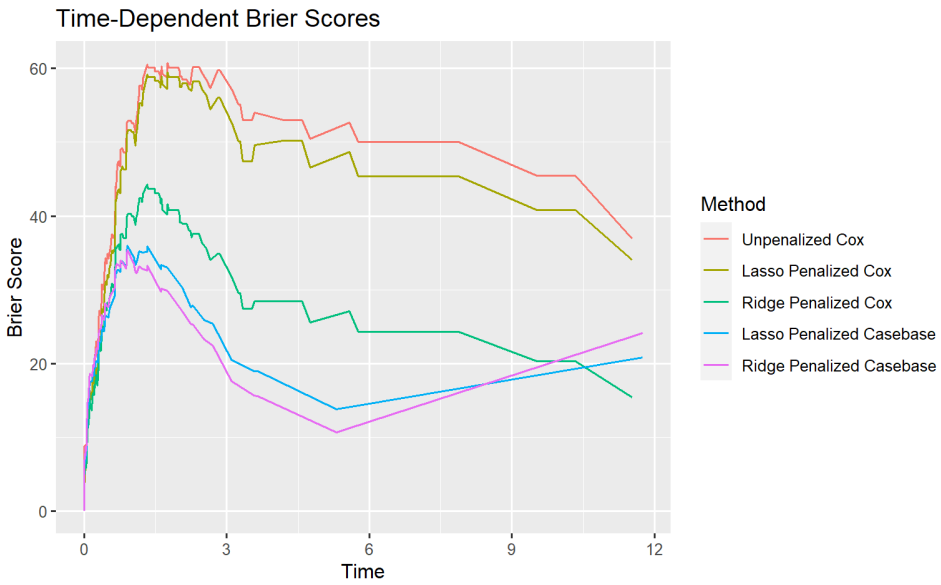

Figure 13: Time-Dependent Brier Scores (Setting 4)

Measure	Lasso Cox	Lasso Casebase
MCC	0.502	0.313
TPR	0.950	0.950
TNR	0.710	0.460
FPR	0.290	0.540
FNR	0.050	0.050

Table 13: Variable Selection (Setting 5)

Method	Concordance
Unpenalized Cox	0.832
Lasso Penalized Cox	0.881
Ridge Penalized Cox	0.865
Lasso Penalized Casebase	0.842
Ridge Penalized Casebase	0.870

Table 14: Concordance (Setting 5)

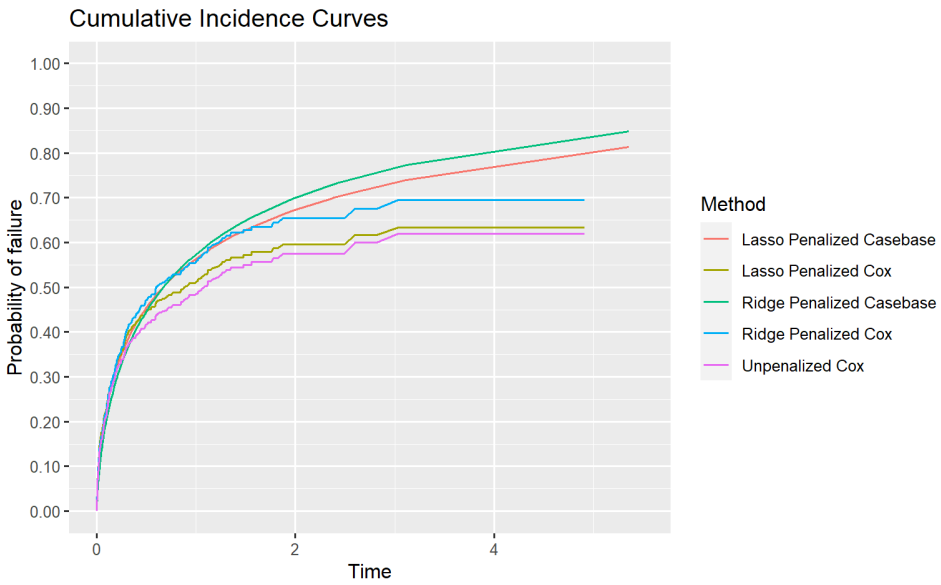

Figure 14: Cumulative Incidence Curves (Setting 5)

4.6 Simulation Setting 6: Low Signal-to-noise Ratio

$$n = 500, p = 120, z = 100, snr = \frac{1}{7}, \rho = 0.5$$

For the last setting we explored a low signal-to-noise ratio of $1/7$.

A similar phenomenon is happening here in the variable selection measures, similar to the setting of $p > n$ where the TPR is quite low and the TNR is quite high. In the low signal-to-noise ratio setting, the models are not selecting the true non-zeros but are good at finding all the true zeros *(see Table 15)*.

Measure	Lasso Cox	Lasso Casebase
MCC	0.297	0.212
TPR	0.250	0.200
TNR	0.960	0.950
FPR	0.040	0.050
FNR	0.750	0.800

Table 15: Variable Selection (Setting 6)

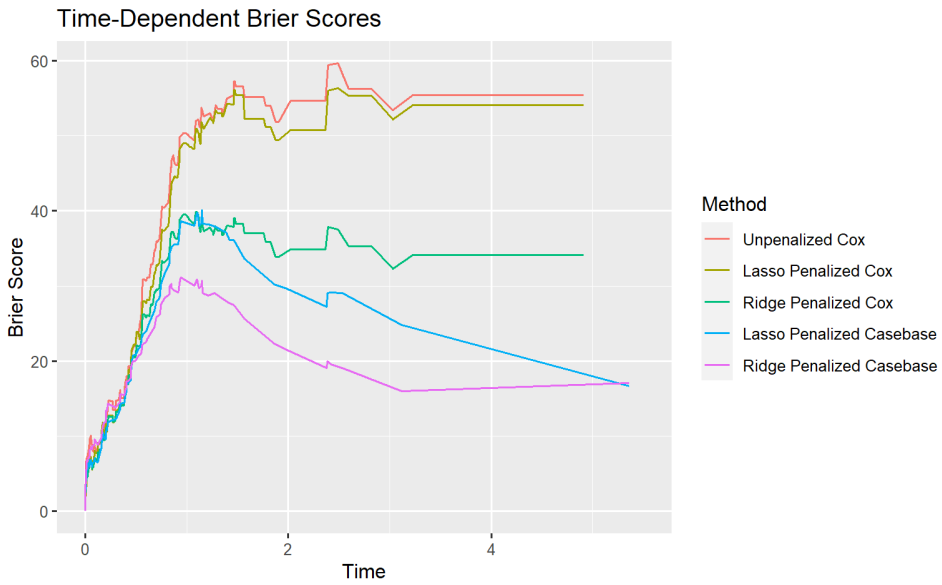

Figure 15: Time-Dependent Brier Scores (Setting 5)

We also found a clear difference in concordance performance between Casebase and other Cox methods. But Brier Score results showed the opposite for Lasso Casebase performance, although Ridge Casebase performed the best *(see Table 16 and Figure 17)*.

Method	Concordance
Unpenalized Cox	0.558
Lasso Penalized Cox	0.592
Ridge Penalized Cox	0.558
Lasso Penalized Casebase	0.943
Ridge Penalized Casebase	0.998

Table 16: Concordance (Setting 6)

4.7 Results: Summary

Penalized Casebase models performed better on average in any setting compared to unpenalized and penalized Cox models in terms of prediction accuracy. How-

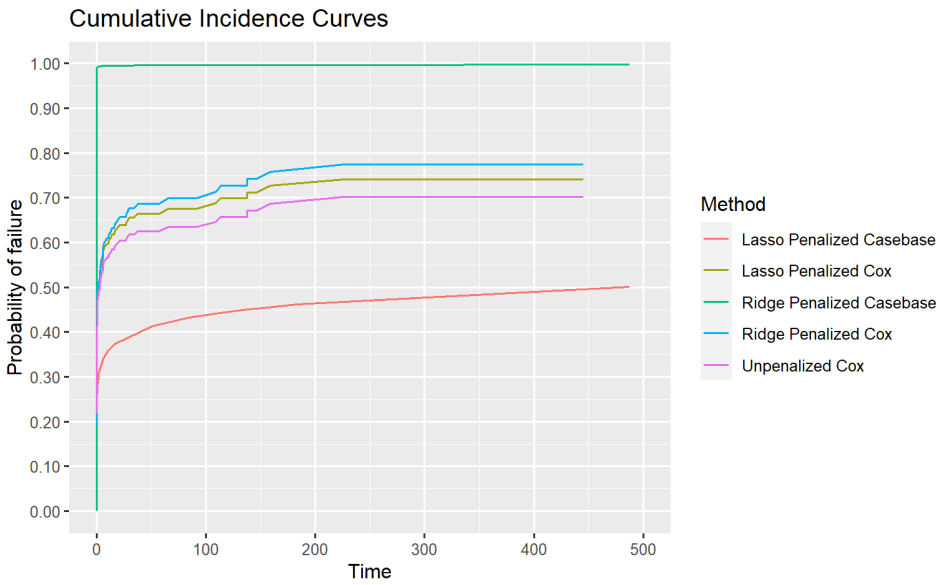

Figure 16: Cumulative Incidence Curves (Setting 6)

ever, we found lasso Cox to be slightly better at variable selection compared to lasso Casebase in all settings. And in $p > n$ settings, penalized models showed more accurate predictions, especially penalized Casebase models.

Correlation between predictors had little effect on prediction performance between models. Furthermore, all models suffered in the low signal-to-noise ratio setting except in the Concordance metric for Casebase models. And in settings where the signal-to-noise ratio was low or when $p > n$, lasso models yielded low TPR and high TNR.

5 References

1. Bhatnagar, S.R., Turgeon, M., Islam, J., Hanley, J.A., Saarela, O. (2020). Casebase: An Alternative Framework For Survival Analysis and Comparison of Event Rates. *Journal of Statistical Software*. VV(II), doi:10.18637/jss.v000.i00

2. Breslow N (1972). "Discussion of the paper by DR Cox cited below." Journal of the Royal Statistical Society, Series B, 34, 187–220.

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

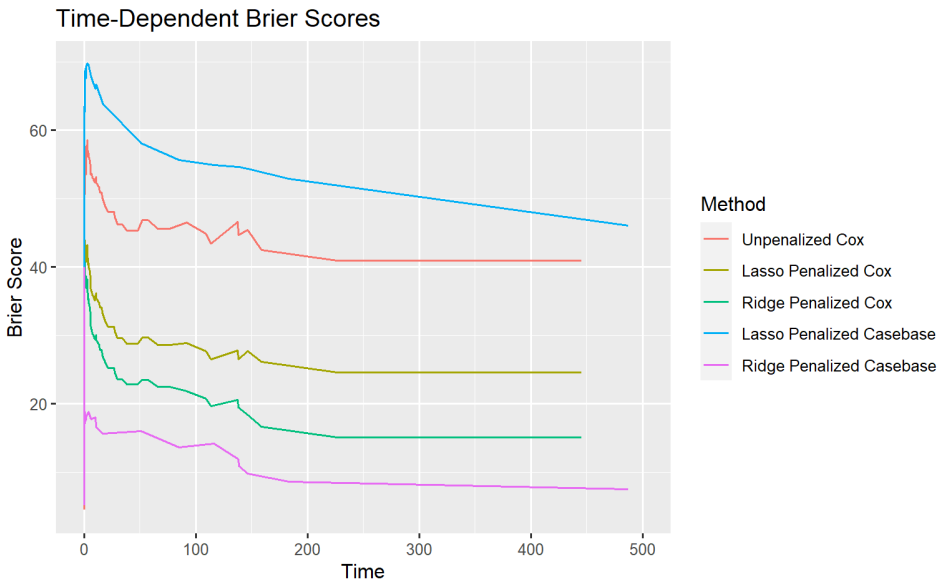

Figure 17: Time-Dependent Brier Scores (Setting 6)

4. Simon, N., Friedman, J., Hastie, T., Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5) 1-13 https://www.jstatsoft.org/v39/i05/

5. Hanley JA, Miettinen OS (2009). "Fitting smooth-in-time prognostic risk functions via logistic regression." The International Journal of Biostatistics, 5(1).

6. Bhatnagar S, Turgeon M, Islam J, Saarela O, Hanley J (2020). Casebase: Fitting Flexible Smooth-in-Time Hazards and Risk Functions via Logistic and Multinomial Regression. R package version 0.9.0, URL https://CRAN.R-project.org/package=Casebase.

7. Friedman, J., Hastie, T. and Tibshirani, R. (2008) Regularization Paths for Generalized Linear Models via Coordinate Descent, https://web.stanford.edu/ hastie/-Papers/glmnet.pdf Journal of Statistical Software, Vol. 33(1), 1-22 Feb 2010 https://www.jstatsoft.org/v33/i01/

8. Soave, D., Lawless, J.F. (2021). Regularized Regression for Two Phase Failure Time Studies. DOI: 10.1002

9. R Core Team (2020). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.