Received: 29 January 2022 | Revised: 25 June 2023 | Accepted: 29 July 2023

DOI: 10.1002/sim.9873

RESEARCH ARTICLE

Statistics in Medicine WILEY

Confidence intervals for the Cox model test error from cross-validation

Min Woo Sun[1] | **Robert Tibshirani**[1,2]

[1]Department of Biomedical Data Science, Stanford University, Stanford, California, USA

[2]Department of Statistics, Stanford University, Stanford, California, USA

Correspondence
Min Woo Sun, Department of Biomedical Data Science, Stanford University, 735 Campus Drive, Apt 888A, Stanford, CA 94305, USA.
Email: minwoos@stanford.edu

Funding information
National Institutes of Health, Grant/Award Number: 5R01 EB001988-16; National Science Foundation, Grant/Award Number: 19 DMS1208164; U.S. National Library of Medicine, Grant/Award Number: LM007033

Summary

Cross-validation (CV) is one of the most widely used techniques in statistical learning for estimating the test error of a model, but its behavior is not yet fully understood. It has been shown that standard confidence intervals for test error using estimates from CV may have coverage below nominal levels. This phenomenon occurs because each sample is used in both the training and testing procedures during CV and as a result, the CV estimates of the errors become correlated. Without accounting for this correlation, the estimate of the variance is smaller than it should be. One way to mitigate this issue is by estimating the mean squared error of the prediction error instead using nested CV. This approach has been shown to achieve superior coverage compared to intervals derived from standard CV. In this work, we generalize the nested CV idea to the Cox proportional hazards model and explore various choices of test error for this setting.

KEYWORDS

confidence interval, Cox model, cross-validation, nested cross-validation, survival analysis

1 | INTRODUCTION

Cox models are deployed ubiquitously for survival analysis, for example for assessing the success of chemohormonal therapy in prostate cancer, or estimating customer churn.[1,2] Cox proportional hazards (PH) model is a class of survival models often used to model the relationship between time to event and a set of covariates. Survival data is typically characterized by the following properties. The response variable Y is time to occurrence of an event, often time to failure or death. Observations can be censored–dented with an indicator variable ($\delta = 0$ if censored, $\delta = 1$ if the event occurs). For instance, if a participant in a clinical trial drops out during the study resulting in an unknown survival time, then the observation is right censored. Lastly, there are predictor variables of interest $X = (x_1, x_2, \ldots, x_k)^T$. The Cox PH regression model aims to understand the relationship between the response and a given set of predictors. Due to censoring, standard approaches like ordinary least squares regression cannot be employed in this setting. The proportional hazards model was introduced by Cox for solving this particular problem.[3] The Cox PH model is defined as follows,

$$\lambda_i(t) = \lambda_0(t) e^{x_i^T \beta} \tag{1}$$

where $\lambda_i(t)$ represents the hazards for individual i at time t and $\lambda_0(t)$ is the baseline hazard (ie, the hazard when $X = (0, 0, \ldots, 0)^T$) that is shared across all the individuals in the data. x_i is the vector of predictors measured on

Abbreviations: CV, cross validation; NCV, nested cross validation; PH, proportional hazards; PL, partial likelihood.

individual i, and $\beta = (\beta_1, \beta_2, \ldots, \beta_k)$ the corresponding coefficients. The exponential function is a common choice for defining the relative risk, which can be thought of as the proportionate change in risk based on the predictors. One approach to fitting the Cox model is by maximizing the partial likelihood (PL).[4] For simplicity, we can assume distinct event times to avoid ties. The PL is defined as,

$$L(\beta) = \prod_{j=1}^{m} \frac{e^{x_j^T \beta}}{\sum_{i \in R(j)} e^{x_i^T \beta}} \tag{2}$$

where $R(j)$ is the risk set representing individuals that are at risk of the event at the time of the event occurring for individual j. Note that the baseline hazard $\lambda_0(t)$ gets canceled out in the PL formulation. As such, the baseline hazard does not need to be specified for inference, making the approach semi-parametric and the problem becomes independent of time. Furthermore, regularization techniques like l_1 lasso and l_2 ridge penalties can be incorporated into the PL. The corresponding solution for the coefficients can be achieved through optimization procedures like coordinate descent, which has been incorporated in R package **glmnet**.[5]

It is important to accurately quantify uncertainty and assess the Cox model's performance especially for its prevalent use in the biomedical field. Throughout this work, we refer to model performance objectives like the concordance index and the log partial likelihood as prediction error or test error. A common approach for fitting and evaluating statistical learning models involves cross validation (CV).[6] K-fold CV is a simple technique that randomly splits data into K equally sized folds then trains a model on $K-1$ folds and tests on the held-out fold rather than relying on a single train-test split. By repeating this procedure for all K folds, CV will generate K prediction error estimates of the model. As such, it is possible to estimate the variance based on these CV-derived estimates and construct a confidence interval around the point estimate.

Confidence intervals constructed from CV estimates of the prediction error are often much smaller than their purported nominal coverage as shown in Figure 1.[7] This phenomenon occurs due to the correlation induced between the prediction errors by repeated use of samples for both training and testing in the CV procedure. More precisely, the estimate of the variance used to compute the standard error assumes that the observed prediction errors are independent, thus overlooking the covariance terms. As a result, the estimate of the variance ends up being smaller than they should be for building confidence intervals with proper coverage. The problem of narrow confidence intervals is exacerbated when there are more predictors than samples ($p > n$), which is a common setting for biological data such as gene expression data.

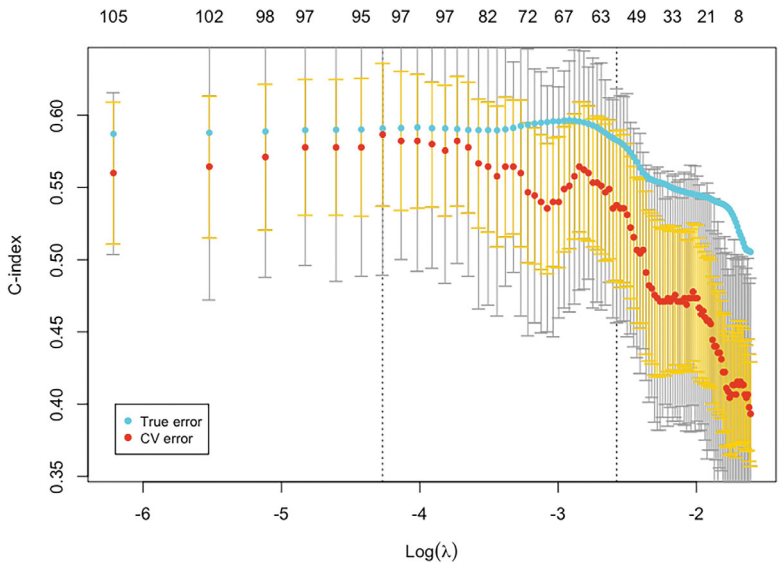

FIGURE 1 This figure demonstrates that naive confidence intervals (yellow) do not have adequate coverage and must be adjusted to reach nominal coverage. We run CV and nested CV on a train set of size $n = 100$ observations and $p = 150$ features using a ℓ_1-regularized Cox PH model to construct 90% confidence intervals. The solution path is plotted as a function of $\log(\lambda)$, where λ is weight on the penalty term. The number of selected features is shown along the top of the plot. We estimated the out-of-sample C-index using a a test set of size $n = 1000$. The models corresponding to the highest C-index and one SE rule are indicated by the vertical broken lines.

In order to address this problem, Bates et al recently proposed using the estimate of the mean squared error (MSE) of the prediction error, rather than the sample variance of the prediction error.[7] MSE is an appropriate choice as it can be decomposed into variance and bias terms, where bias is usually small, thus making MSE a conservative estimate for the squared standard error.[8] Furthermore, Bates et al introduces a convenient lemma that allows the MSE to be directly estimated from the data through nested CV without making any asymptotic or modeling assumptions as we have seen in prior related works constructing asymptotically-exact confidence intervals.[9] We delve deeper into the lemma and the nested CV procedure in Section 3. Simulation experiments and real data examples from Bates et al demonstrate that the confidence intervals constructed with MSE estimate achieves valid coverage for linear and logistic regression models in both low-dimensional and high-dimensional settings.

In this paper we demonstrate the efficacy of nested CV for constructing proper confidence intervals for test error in the survival analysis setting. We explore various choices of test error for assessing the model performance such as the concordance index and the log PL. We present a technique using nested CV to construct valid confidence intervals for the concordance index (C-index). While this work focuses on the Cox PH model, this nested CV approach can be implemented for other survival models.

2 | MEASURING MODEL PERFORMANCE

2.1 | Concordance index

Evaluating a predictive model in the presence of censored data is not simple because measures like MSE cannot be computed. We instead use Harrell's concordance index or C-index based on Kendall–Goodman-Kruskal-Somers type rank correlation as introduced by Harrell and implemented in **glmnet**.[10,11] The C-index measures a model's predictive discrimination power or more precisely, the proportion of observation pairs for which the corresponding outcomes and predictions are concordant, that is, for a pair $(i,j)_{i\neq j}, x_i > x_j, f(x_i) > f(x_j)$ or $x_i < x_j, f(x_i) < f(x_j)$, where x is an observation and $f(x)$ is the prediction from a model. In the context of the Cox model, given a pair of subjects i and j, a concordant pair would mean that i's predicted risk is greater than j's when the survival time is less than j's, or vice versa. We exclude pairs that cannot be ordered–pairs for which both samples had their event occur simultaneously or if for one of the samples the event occurs but the event status for the other sample is unknown due to right censoring (eg, patient dropped out of the study) prior to the first sample's event. The C-index ranges between 0 and 1 where a value of 1 indicates perfect separation and a value of 0.5 means no predictive discrimination, that is, randomly guessing for each sample.

The C-index presents a unique challenge for computing the variance of the measure because the C-index is defined for the set of observations not for an individual sample. The variance of the C-index is necessary for estimating the variance of the held-out error in the nested CV algorithm. In the case of MSE, one can simply compute the sample variance of the CV errors. The variance of the C-index can be estimated using the Infinitesimal Jackknife method as implemented in the "survival" package. A detailed description of the approach can be found in the package vignette: https://cran.r-project.org/web/packages/survival/vignettes/concordance.pdf

2.2 | Partial likelihood

An alternative choice for prediction error for the Cox model is the log partial likelihood, which is also used as the loss function for fitting the Cox PH model as described in Section 2,

$$\ell(\beta) = \sum_{j=1}^{m} x_j^T \beta - \log\left(\sum_{i \in R(j)} e^{x_i^T \beta}\right) \tag{3}$$

Since the PL is a product, as the number of samples increases its magnitude grows; therefore, its value needs to be scaled by the sample size. However, simply dividing the log PL by the sample size n still exhibits an increasing behavior because $\log(PL)/n$ does not converge to a finite expectation. When the sample size n is big, approximately $\mathbb{E}[\ell(\beta)/n] = c_1 + c_2 \cdot \log(n)$. This is an issue for evaluating confidence interval coverage as the the test error will increase with larger held out

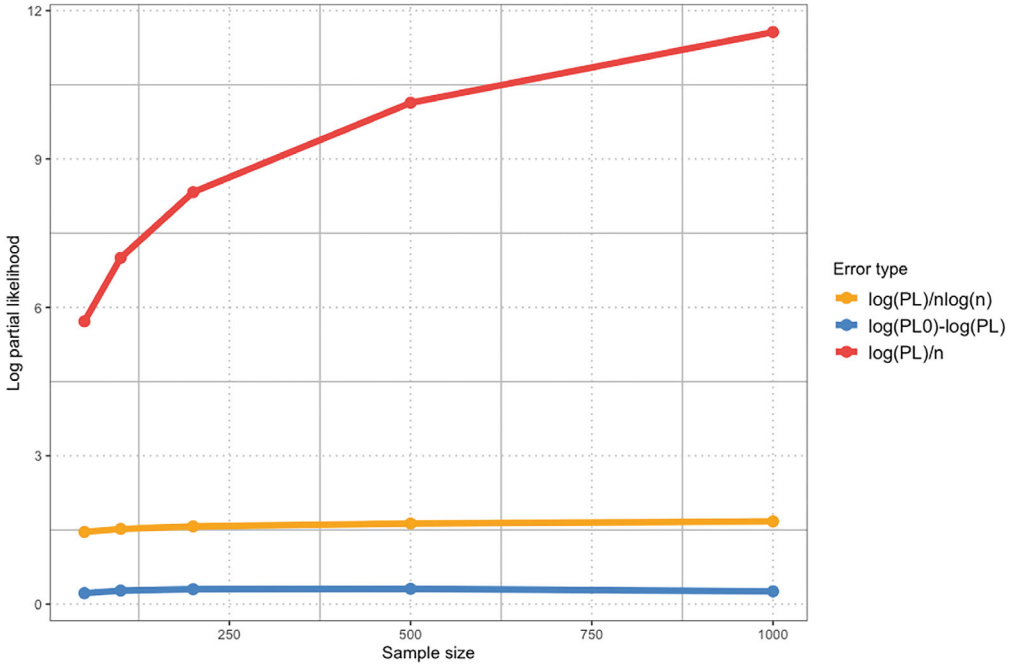

FIGURE 2 This plot shows the relationship between sample size and various test errors for the Cox model. It is clear that $\log(PL)/n$ increases with sample size despite scaling by n^{-1} while the other measures remain constant. As a result, $\log(PL)/n$ is not a viable candidate for test error.

samples. As a result, $\ell(\beta)/n$ is not a viable measure to use to evaluate model performance. In order to avoid this problem, one can instead use

$$\frac{\ell(\beta)}{n\log(n)} \tag{4}$$

or

$$\ell(\beta_{\text{null}}) - \ell(\beta) \tag{5}$$

where $\ell(\beta_{null})$ is the log PL from the Cox model trained on data with no signal (ie, covariates are zero). In Figure 2 while $\log(PL)/n$ increases with sample size, the latter two do not. However we instead focus on C-index for test set evaluation here, because it is more interpretable. As is standard, we do employ PL for cross-validation (next section).

2.3 | Cross-validating the partial likelihood

Using PL poses a challenge for traditional CV as the PL is defined for a set of observations, not just a single sample. Furthermore, due to censoring in survival analysis, a left out sample in a leave-one-out CV procedure (fold size $k = n$) can be ill-defined.[5] In this work we focus on the nested CV implementation for the C-index, but CV and nested CV procedures for the log PL can be implemented following the leave-one-out CV procedure introduced by Houwelingen et al.[12] The CV log PL is defined as,

$$\widehat{\text{CV}}_{\text{PL}}(\lambda) = \sum_{i=1}^{n} \ell(\hat{\beta}_{\lambda}^{(-i)}) - \ell_{-i}(\hat{\beta}_{\lambda}^{(-i)}) \tag{6}$$

where $\hat{\beta}_{\lambda}^{(-i)}$ refers to the optimal fit excluding sample i and $\ell_{-i}(\cdot)$ is the log PL computed excluding sample i. λ is chosen to maximize $\widehat{\text{CV}}_{\text{PL}}(\lambda)$. The CV log PL measures the difference of log PL on the full data and the log PL on the data excluding

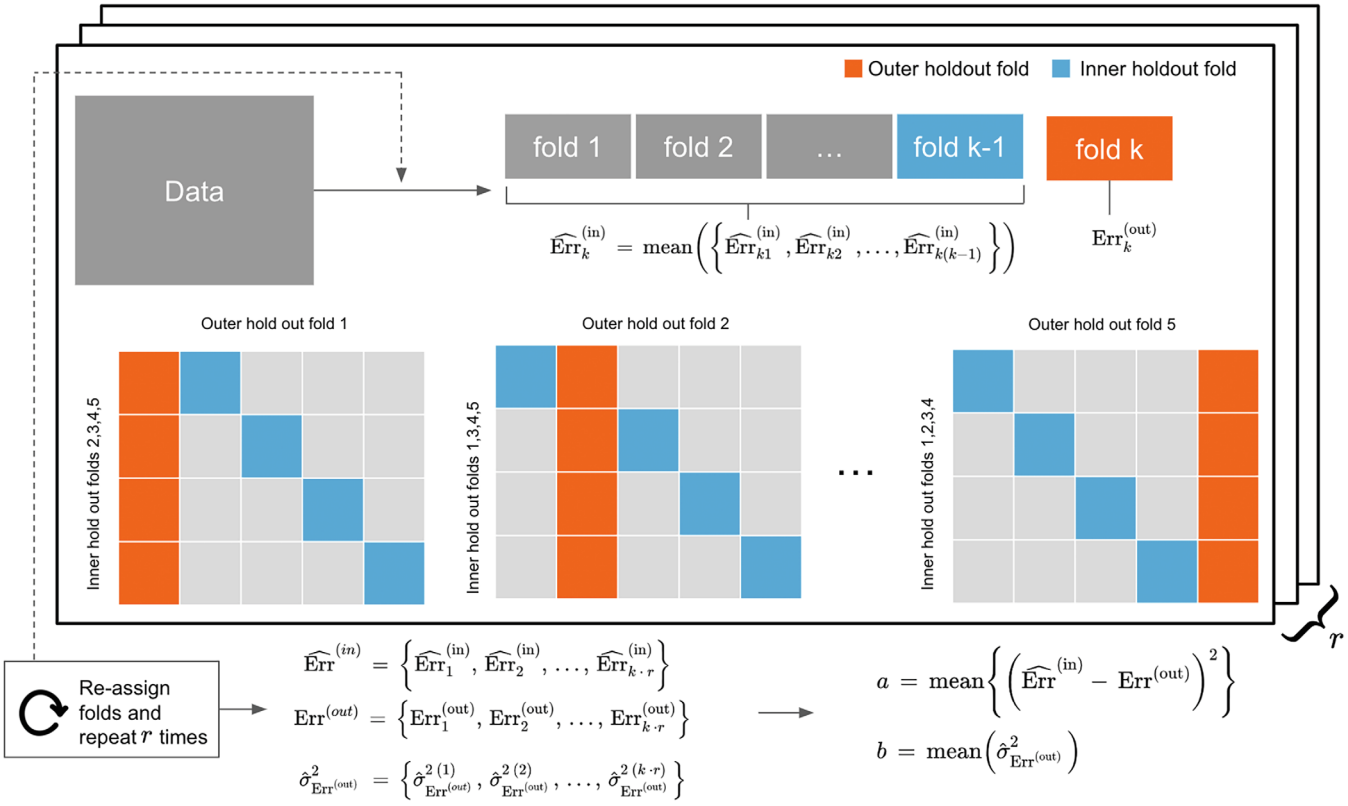

FIGURE 3 This diagram demonstrates how to compute the components a and b to estimate the MSE of the prediction error through the nested CV algorithm. To perform nested CV, the input data (typically the training data from random train-validation split) will be randomly split into K equally sized folds. At each iteration of the procedure, one fold will be held out (represented in red) and run standard K-fold CV on the remaining $K-1$ folds (represented by the blue inner holdout fold and the grey folds). We repeat this procedure for all folds. This entire process is repeated r times in order to get a reliable estimate of the MSE. Note that the 4×5 matrix is an example for when $K = 5$.

sample i, summed across all samples providing a goodness of fit estimate. Note that does not suffer from the sample size sensitivity outlined in the previous section, because it is used to compare models fit on the same sample size.

3 | NESTED CV ALGORITHM

Constructing confidence intervals for the prediction error estimate using naive CV leads to smaller intervals with poor coverage. Instead of using the empirical standard deviation of the CV estimates, we estimate the MSE of the prediction error using the MSE identity introduced by Bates et al, which allows the MSE to be directly estimated from given data as shown in Figure 3.[7] The MSE is a good choice as it can be decomposed into bias and variance, where bias tends to be small, making it a close estimate of the variance. Bates et al defines the MSE identity as follows,

$$\underbrace{\mathbb{E}\left[\left(\widehat{\mathrm{Err}}_{XY}^{(in)} - \mathrm{Err}_{XY}\right)^2\right]}_{\mathrm{MSE}} = \underbrace{\mathbb{E}\left[\left(\widehat{\mathrm{Err}}_{XY}^{(in)} - \bar{e}^{(out)}\right)^2\right]}_{\mathrm{a}} - \underbrace{\mathbb{E}\left[\left(\mathrm{Err}_{XY} - \bar{e}^{(out)}\right)^2\right]}_{\mathrm{b}} \tag{7}$$

where $\widehat{\mathrm{Err}}^{(in)}$ is the CV estimate of the errors from inner CV, $\bar{e}^{out}$ is the average error from the held out folds, and Err_{XY} is the true prediction error conditioned on the training set X, Y. To compute part a of the MSE estimate, we take the mean of $\widehat{\mathrm{Err}}^{(in)}$ and $\bar{e}^{out}$ from multiple repetition then take the squared difference. For each iteration, the observations are randomly assigned to the K folds. Part b of the MSE is simply the variance of the heldout error $\bar{e}^{out}$. For the C-index, we estimate the variance using the infinitesimal jackknife estimate. The NCV point estimate $\widehat{\mathrm{Err}}^{(\mathrm{NCV})}$ is the mean of all

$\widehat{\text{Err}}^{(\text{in})}$. The bias of $\widehat{\text{Err}}^{(\text{NCV})}$ can also be estimated through the nested CV procedure,

$$\widehat{\text{bias}} = \left(1 + \frac{K-2}{K}\right)\left[\widehat{\text{Err}}^{(\text{NCV})} - \widehat{\text{Err}}^{(\text{CV})}\right] \tag{8}$$

where the term $\widehat{\text{Err}}^{(\text{NCV})} - \widehat{\text{Err}}^{(\text{CV})}$ is the unbiased estimate of the difference in errors of sample size $n(K-2)/K$ and $n(K-1)/K$ respectively. The scaling factor $1 + (K-2)/K$ is included to account for biases going from different sample sizes. 1 is for scaling back to an estimate of sample size $n(K-1)/K$ from $n(K-2/K)$ as the point error estimate from inner CV uses a sample size of $n(K-2/K)$. The term $(K-2)/K$ is for re-scaling the point estimate back to an estimate of size n from $n(K-1/K)$.

Once we compute the point, MSE, and bias estimates through the nested CV procedure, we can construct the confidence interval using the following formula,

$$\widehat{\text{Err}}^{(\text{NCV})} - \widehat{\text{bias}} \pm q_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{K-1}{K}} \cdot \sqrt{\widehat{\text{MSE}}} \tag{9}$$

where $q_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of the standard Gaussian. As $\widehat{\text{MSE}}$ is an estimate of a $K-1$ fold CV with $n(K-1)/K$ samples, the factor $\sqrt{K-1/K}$ is included to re-scale back to an estimate of sample size of n, that is, use $(K-1/K) \cdot \widehat{\text{MSE}}$.

4 | EXPERIMENTS

We explore and demonstrate the efficacy of NCV in both numerical simulations and real survival data problems by running experiments to assess the coverage of confidence intervals constructed using the nested CV approach. Figure 4 delineates the experimental setup. The simulated or real data set is randomly split into train and test set. The train set is used to estimate both the standard CV confidence interval and the NCV confidence interval, while the test set is used to measure the test error. To evaluate whether the intervals achieve the desired $(1-\alpha)\%$ coverage, we count the number of times the test error is not contained by the interval and call this the interval's miscoverage rate. For instance, a miscoverage rate of 10% is desired for a 90% confidence interval. This would mean for a simulation of 100 iterations, we should approximately expect to see 10 of the intervals to not contain the test error. In our simulations, we primarily focus on the C-index, which is one of the most commonly employed metrics to evaluate the Cox model performance.

Throughout all our experiments, our estimand of interest is Equation (7), the mean squared error of the model prediction error, which is estimated from the training data through the nested CV procedure and used to construct the confidence intervals. We vary the number of samples n, number of features p, number of CV fold n_{fold}, and number of simulations n_{sim}. For larger $p > n$, we chose smaller values of n, n_{fold}, and n_{sim} to reduce the computational cost resulting from relatively larger p.

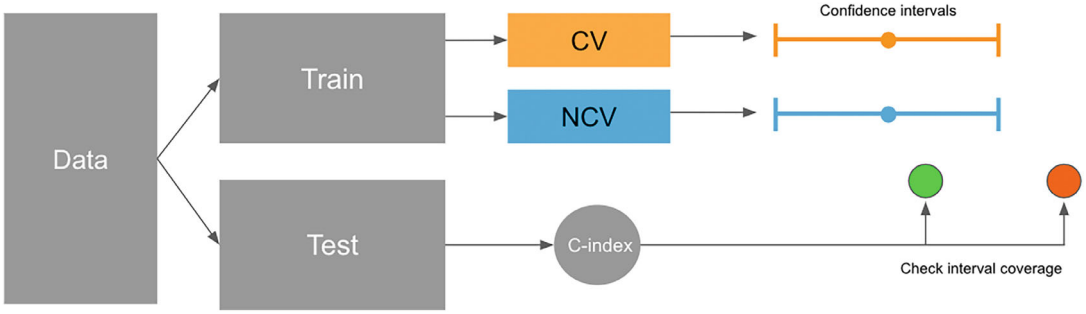

FIGURE 4 The simulation experiment process for one iteration. Data is split into train and test set randomly. We use the train set to run CV and NCV for constructing confidence intervals. We use the test set to compute the true test error (in this case C-index, but can also be PL or other measures). We repeat this procedure for a large number of times to compute the miscoverage rates of both CV and NCV based confidence intervals. Miscoverage rate is simply the count of the test set C-index falling within the CV or NCV confidence intervals. For all the experiments we aim for 90% confidence intervals, which means that a valid 90% confidence interval would have around 10% miscoverage rate.

The following experiments were implemented in **R** V4.0.2 and **glmnet** V4.1 for CV and fitting Cox models. Despite being more computationally expensive than CV, the nested CV algorithm can be implemented to compute the repetitions in parallel. These experiments were run on nodes with 64 GB memory and 20–32 cores on Sherlock, a Stanford Research Computing Center (SRCC) HPC cluster. All code for the experiments are available on: https://github.com/minwoosun/nestedcv-confint.

4.1 | Simulation study

We begin our simulation experiments with a simple base case with no censoring then extend to well-known parametric survival models and also incorporate right censoring. For each approach we cover both $n > p$ and $p > n$ settings. We describe in detail below our data-generating mechanisms for simulated data.

- **Simple base case:**

  We generate data $X \in \mathbb{R}^{n,p}$ with a signal such that time $t = x\beta + c\varepsilon$, where $\varepsilon \overset{iid}{\sim} \mathcal{N}(0,1)$ and a constant $c \in \mathbb{R}$ which allows us to vary the degree of noise. The covariates are generated such that $p \overset{iid}{\sim} \mathcal{N}(0,1)$. We choose alpha to be 0.10, that is, construct 90% confidence intervals. For the $n > p$ scenario, we generate a train data set with $n = 100, p = 10$, and a large test data set with $n = 1000, p = 10$. For the $p > n$ scenario, we generate a train data set of size $n = 100, p = 150$ and a large test data set with $n = 1000, p = 150$. We perform 10-fold CV and 10-fold nested CV fitting Cox PH model repeated 200 times for each simulation. We run a total of 1000 simulations for the $n > p$ setting and 100 simulations for $p > n$ setting.
- **Parametric models for survival time:**

  To simulate more realistic survival times for the Cox proportional hazards model, we use the inverse transform sampling method, which is a general technique for generating random variables with a given cumulative distribution function. In particular, we follow the approach described in Bender et al.[13] This method works in two stages: (1) sample from a uniform distribution (2) use the inverse of the cumulative distribution function of the target distribution to transform the uniform samples into the desired distribution like the Weibull or Gompertz distribution. Let u be the sample from a uniform distribution, $u \sim \mathcal{U}[0,1]$, x be the data matrix, and β be the vector of values representing the strength of true association between the variables and the survival time. We generate the survival times for the exponential distribution with parameter λ as,

$$T_{\text{exponential}} = -\frac{\log(u)}{\lambda e^{\beta^\top x}} \tag{10}$$

  and for the Weibull distriubtion with parameters λ and ν as,

$$T_{weibull} = \left(-\frac{\log(u)}{\lambda e^{\beta^\top x}}\right)^{1/\nu} \tag{11}$$

  This allows for more flexibility in simulating survival times under different hazard functions and baseline hazard distributions. We also incorporate fixed censoring by setting a time ceiling, that is, any time sampled past the max time will be considered right censored. For the exponential survival time simulation, each continuous variable X_j is sampled i.i.d. from a Gaussian distribution with mean $\mu = 10$ and variance $\sigma^2 = 5$. We set $\beta = \{0.1, 0.1, 0, \ldots, 0\}$. For the Weibull survival time simulation, each binary variable X_j is sampled i.i.d. as n independent Bernoulli trials with probability of success 0.5. For $n > p$ we let $n = 1000$ and $p = 10$. For $p > n$ we let $n = 200$ and $p = 250$. We set $\beta = \{0.1, 0.1, 0, \ldots, 0\}$ to induce moderate association between the features and the survival times. This set of simulation parameters leads to an average of 0.7 CV C-index.

Table 1 displays the findings for all the simulation experiments. We observe that the nested CV method attains valid coverage. In contrast, the basic CV strategy does not yield satisfactory results. In line with the regression and classification scenarios described in Bates et al, it becomes apparent that the miscoverage of the naive CV are significantly amplified when the feature count exceeds the observation count.

TABLE 1 This table summarizes the results from all the simulated data examples.

	Point estimate		Mean SE		Miscoverage			
					CV		NCV	
Setting	CV	NCV	CV	NCV	Upper	Lower	Upper	Lower
Base case ($n > p$)	0.605	0.594	0.037	0.053	0.190	0.030	0.060	0.010
Base case ($p > n$)	0.667	0.661	0.027	0.063	0.350	0.013	0.050	0.080
Exponential ($n > p$)	0.676	0.676	0.009	0.014	0.050	0.090	0.040	0.050
Exponential ($p > n$)	0.673	0.654	0.019	0.037	0.060	0.240	0.040	0.020
Weibull ($n > p$)	0.675	0.675	0.006	0.008	0.070	0.130	0.070	0.040
Weibull ($p > n$)	0.674	0.656	0.019	0.033	0.080	0.160	0.080	0.020

Note: We report the point estimate, mean standard error across multiple iterations and the confidence interval miscoverage rates.

4.2 | Real data examples

In addition to the simulated data, we compare the CV and nested CV procedures on real data sets. Similar to the simulated data setting, for each iteration we randomly sample a small number of observations from the entire data for the train set to run CV and nested CV for constructing confidence intervals. We use the remaining observations as the test set to evaluate the C-index and check whether the value falls within the confidence interval. Unlike in the simulation settings, we are no longer able to control the strength of association between the survival time and the features as the data generating mechanism is unknown. Through this repeated sampling procedure, we demonstrate the efficacy of the NCV procedure beyond the simulation setting in two real data sets representing both the $n > p$ and $p > n$ scenarios. To be comprehensive, we choose a clinical trial data set and a cancer gene expression dataset, which are both commonly found data modalities used to address modern biomedical problems.

- **Chemotherapy for stage III colon cancer data** ($n > p$)
  The colon cancer data comes from a randomized control trial that aimed to assess the efficacy of combining two adjuvant therapy for stage III colon cancer.[14] The data set comprises 929 patients and 11 predictors. The data matrix contains a total of 1858 rows as there are two observations per patient for different event types: death and recurrence. The predictors include the treatment type, clinical information like the patient age and sex, and quantitative descriptions of the tumor and colon. We filter for event type death (etype = 2) leading to a data matrix with $n = 929$ and $p = 11$. We ran a total of 200 simulations with 200 repetitions.
- **Prediction of clinical outcomes from genomic profiles (PRECOG) data** ($p > n$)
  The PRECOG data set is an amalgamation of 165 cancer gene expression data with patient clinical outcome.[15] This pan-cancer data set was curated with the goal of better understanding the prognostic landscape of genes and immune cells for humans, which can lead to novel biomarker discoveries and therapeutic targets. Gene expression data are typically characterized by a large number of genes to be used as features with a much smaller number of samples. For the experiment, we chose and preprocessed the eight data sets that were used to train the FOXM1-KLRB1 prognostic model as described in the paper resulting in $n = 1086, p = 4708$. The ID for gene expression data sets used for training are GSE32062, E-TABM-38, GSE10358, GSE4475, GSE8894, GSE1993, and GSE19234. We converted the expression values into z-scores then took the top 180 features with the highest variance to reduce the feature space leading to the final matrix of size $n = 1086, p = 180$. We randomly sampled 150 patients from the 1086 patients for constructing the confidence intervals and used the remainder for measuring the C-index. We ran a total of 150 simulations with 200 repetitions each.

The results for the real data experiments are reported in Table 2. In both the colon cancer chemotherapy data set $n > p$ and the PRECOG data set $p > n$, the nested CV confidence intervals achieve proper coverage, while the naive CV approach fails. Similar to the simulation experiments, the naive CV miscoverage is much worse when there are more features than the number of observations. These findings further validate how effective the nested CV procedure is for constructing valid confidence intervals.

TABLE 2 This table summarizes the results from all the real data examples.

	Point estimate		Mean SE		Miscoverage			
					CV		NCV	
Setting	CV	NCV	CV	NCV	Upper	Lower	Upper	Lower
Colon ($n > p$)	0.656	0.651	0.026	0.036	0.145	0.025	0.085	0.015
PRECOG ($p > n$)	0.697	0.671	0.035	0.065	0.173	0.040	0.020	0.087

Note: We report the point estimate, mean standard error across multiple iterations and the confidence interval miscoverage rates.

5 | DISCUSSION

In this study, we set out to underscore the importance and efficacy of the nested CV procedure as a reliable approach for estimating the MSE of the prediction error and constructing valid confidence intervals for the Cox model test error. Our aim was to show that this method leads to confidence intervals of the prediction error that achieve desired coverage, which the naive CV approach may fail to attain.

To validate our theory, we conducted a series of simulations and experiments with real-world data. The purpose of these experiments was twofold: First, to substantiate our claim about the nested CV procedure's efficacy in attaining valid coverage, and second, to demonstrate the limitations of naive CV based confidence intervals. The results showed that naive CV based confidence intervals indeed had poor coverage rates due to the interval width being too narrow. This shortcoming was even more pronounced in the $p > n$ scenario, where the number of features p far exceeds the number of observations n consistently throughout all the experiments. In contrast to the naive CV confidence intervals, the results of the nested CV procedure were far more satisfactory, and indeed aligned with our expectations. The confidence intervals derived from the nested CV procedure were shown to achieve the nominal coverage, a key indication of the method's reliability. This is a significant finding because it points to the robustness of the nested CV procedure in the face of increasing complexity and high-dimensional data.

While we focused on the C-index for the experiments, this approach is not limited to the C-index metric alone. Bates et al has shown that nested CV can be used for model performance metrics like the MSE in regression settings and accuracy in classification settings. Furthermore, the nested CV algorithm can be applied to other models beyond just the Cox PH model, including other supervised learning models through survival stacking.[16] However, the exact conditions for when the standard CV intervals fail even in the $n > p$ setting is still not fully understood. Generally, we observed that increasing the sample size n led to better coverage. We also saw that a sufficiently large number of repetition for the nested CV algorithm is necessary for getting a reliable estimate of the MSE for $p > n$ scenarios and for the confidence intervals to attain proper coverage. Increasing the number of repetition renders the process computationally more intensive, but these repetitions can be easily run in parallel, mitigating the computational challenges. This is an important advantage, as it allows for the process to be still viable for large-scale applications.

In conclusion, our work attests to the effectiveness of the nested CV procedure for estimating the MSE of the prediction error and for constructing confidence intervals of the Cox model test error. This method outperforms the standard CV approach, which our experiments revealed as less reliable, especially when the number of features exceeds the number of observations.

ACKNOWLEDGEMENTS

We thank Dr Lu Tian for his contribution on deriving the expectation of $\frac{\log(PL)}{n}$. Min Woo Sun was supported by the National Library of Medicine (LM007033); Robert Tibshirani was supported by the National Institutes of Health (5R01 EB001988-16) and the National Science Foundation (19 DMS1208164).

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The colon cancer dataset and its corresponding documentation are available in R package **survival** as the dataframe **colon**.[17] The PRECOG data set is available through https://precog.stanford.edu/. All the simulated data can be reproduced using the scripts in https://github.com/minwoosun/nestedcv-confint.

ORCID

Min Woo Sun https://orcid.org/0000-0003-1049-1854

REFERENCES

1. Kyriakopoulos CE, Chen YH, Carducci MA, et al. Chemohormonal therapy in metastatic hormone-sensitive prostate cancer: long-term survival analysis of the randomized phase III E3805 CHAARTED trial. *J Clin Oncol*. 2018;36(11):1080-1087. doi:10.1200/JCO.2017.75.3657
2. Van den Poel D, Larivière B. Customer attrition analysis for financial services using proportional hazard models. *Eur J Oper Res*. 2004;157(1):196-217. doi:10.1016/S0377-2217(03)00069-9
3. Cox DR. Regression models and life-tables. *J R Stat Soc B Methodol*. 1972;34(2):187-220.
4. Cox DR. Partial likelihood. *Biometrika*. 1975;62(2):269-276.
5. Simon N, Friedman J, Hastie T, Tibshirani R. Regularization paths for Cox's proportional hazards model via coordinate descent. *J Stat Softw*. 2011;39(5):1-13. doi:10.18637/jss.v039.i05
6. Stone M. Cross-Validatory choice and assessment of statistical predictions. *J R Stat Soc B Methodol*. 1974;36(2):111-133. doi:10.1111/j.2517-6161.1974.tb00994.x
7. Bates S, Hastie T, Tibshirani R. Cross-Validation: What Does it Estimate and how Well Does it Do it? 2021.
8. Efron B, Tibshirani R. Improvements on cross-validation: the .632+ bootstrap method. *J Am Stat Assoc*. 1997;92(438):548-560. doi:10.2307/2965703
9. Bayle P, Bayle A, Janson L, Mackey L. Cross-validation confidence intervals for test error. *Advances in Neural Information Processing Systems*. Vol 33. Curran Associates, Inc; 2020:16339-16350.
10. Harrell FE Jr, Lee KL, Mark DB. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Stat Med*. 1996;15(4):361-387. doi:10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4
11. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;33(1):1-22.
12. Houwelingen VHC, Bruinsma T, Hart AAM, Veer VLJ, Wessels LFA. Cross-validated Cox regression on microarray gene expression data. *Stat Med*. 2006;25(18):3201-3216. doi:10.1002/sim.2353
13. Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Stat Med*. 2005;24(11):1713-1723. doi:10.1002/sim.2059
14. Moertel CG, Fleming TR, Macdonald JS, et al. Fluorouracil plus levamisole as effective adjuvant therapy after resection of stage III colon carcinoma: a final report. *Ann Intern Med*. 1995;122(5):321-326. doi:10.7326/0003-4819-122-5-199503010-00001
15. Gentles AJ, Newman AM, Liu CL, et al. The prognostic landscape of genes and infiltrating immune cells across human cancers. *Nat Med*. 2015;21(8):938-945. doi:10.1038/nm.3909
16. Craig E, Zhong C, Tibshirani R. Survival Stacking: Casting Survival Analysis as a Classification Problem. 2021.
17. Therneau TM. A Package for Survival Analysis in R. R package version 3.2-10. 2021.

How to cite this article: Sun MW, Tibshirani R. Confidence intervals for the Cox model test error from cross-validation. *Statistics in Medicine*. 2023;42(25):4532-4541. doi: 10.1002/sim.9873