Introduction to `survivalROC`: An R Package for Survival-ROC Method

P. Saha,[1] and P. J. Heagerty[1,2]
[1] Department of Biostatistics, University of Washington
Seattle, Washington 98195, USA
[2] Division of Public Health Sciences, Fred Hutchison Cancer Research Center
Seattle, Washington 98109, USA

1 Introduction

ROC curves are a popular method for displaying sensitivity and specificity of a continuous diagnostic marker, say, X, for a binary disease variable, say, D. But many disease outcomes are time dependent, $D(t)$ rather than D, and ROC curves that vary as a function of time may be more appropriate. A common example of this time-dependent disease status would be a patient's vital status, where $D(t) = 1$ if a patient has died prior to time t and zero otherwise. A method was proposed by Heagerty et al. (2000) to summarize the discrimination potential of a marker X, measured at baseline ($t = 0$) by calculating ROC curves for cumulative disease or death incidence by time t. This ROC curve is denoted as $ROC(t)$. A typical problem with survival data is that observations may be censored. The paper (Heagerty et al. 2000) discusses two ROC curve estimators that can accommodate censored data. A simple estimator is based on using the Kaplan-Meier estimator for each possible subset $X > c$ for some arbitrary cut-off c. However, this estimator does not guarantee the necessary condition that sensitivity and specificity are monotone in X. An alternative estimator that does guarantee monotonicity is based on a nearest neighbor estimator (NNE) for the bivariate distribution function of (X, T), where T is survival time. At this point, there is no standard software to estimate or compute this time-dependent ROC curve as discussed in the paper (Heagerty et al. 2000). We here introduce an R/S-plus package `survivalROC` which is able to handle the computations as discussed in the paper. To that end, we start by briefly revisiting some necessary concepts and exhibit the utility of the functions in this package. The paper is organized as follows: in the next section we discuss some basic concepts and notations, in the Methods section, we talk briefly about the underlying model and estimation. This section concludes with a detailed discussion of the two main functions of the `survivalROC` package - `survivalROC()` and `survivalROC.C()` and an illustration of how to apply these methods to the real data using a modified version of a well known data set (Mayo PBC data).

2 Notation and Basic concepts

2.1 Notation

We introduce the following notations:

X denotes the diagnostic test or marker. By convention, higher values of X are more indicative of the disease.

T denotes the failure time.

C denotes the censoring time.

$Z = min(T, C)$ is the follow-up time.

δ is a censoring indicator with $\delta = 1$ if $T \leq C$ and $\delta = 0$ if $T > C$.

We use the counting process $D(t) = 1$ if $T \leq t$ and $D(t) = 0$ if $T > t$ to denote failure (disease) status at any time t with $D(t) = 1$ indicating that event occurred prior to time t.

We use subscript i to denote the variable(s) for a subject i.

At any given time t and a given cut-off value c, we define:

$$\begin{aligned} \text{sensitivity}(c,t) &= \Pr\{X > c \mid D(t) = 1\} = S_1(c,t) \\ \text{specificity}(c,t) &= \Pr\{X \leq c \mid D(t) = 0\} = 1 - S_0(c,t) \end{aligned}$$

Using the above definitions, we can define the corresponding ROC curve $ROC(t)$ at any time t .

2.2 The Kaplan-Meier Estimator

We use Bayes' theorem to rewrite the sensitivity and specificity as:

$$\begin{aligned} P\{X > c \mid D(t) = 1\} &= \frac{\{1 - S(t|X > c)\}\, P(X > c)}{1 - S(t)} \\ P\{X \leq c \mid D(t) = 0\} &= \frac{S(t|X \leq c)P(X \leq c)}{S(t)} \end{aligned}$$

where $S(t) = P(T > t)$ and $S(t|X > c)$ is the conditional survival function for the subset defined by $X > c$.

A widely used nonparametric estimate of $S(t)$ is given by Kaplan and Meier. This estimate of $S(t)$ uses all the information in the data, including censored observations, to estimate the survival function. A simple estimator for sensitivity and specificity at time t is given by combining the KM estimator and the empirical distribution function of the marker, X, as:

$$\begin{aligned} \widehat{P}_{KM}\{X > c \mid D(t) = 1\} &= \frac{\left\{1 - \widehat{S}_{KM}(t|X > c)\right\}\left\{1 - \widehat{F}_X(c)\right\}}{1 - \widehat{S}_{KM}(t)} \\ \widehat{P}_{KM}\{X \leq c \mid D(t) = 0\} &= \frac{\widehat{S}_{KM}(t|X \leq c)\widehat{F}_X(c)}{\widehat{S}_{KM}(t)} \end{aligned}$$

where $\widehat{F}_X(c) = \frac{1}{n}\sum \mathbb{1}(X_i \leq c)$

As discussed earlier, this method does not guarantee monotonicity. A second potential problem is that the conditional Kaplan-Meier estimator $\hat{S}_{KM}(t|X > c)$ assumes that the censoring process does not depend on X. This assumption may be violated in practice when the intensity of follow-up efforts are influenced by the baseline diagnostic marker measurements.

2.3 The Nearest Neighbor Estimation (NNE) of the Bivariate Distribution

A valid ROC at time t can be provided by using an estimator of the bivariate distribution function $F(c,t) = P(X \leq c, T \leq t)$, or equivalently, $S(c,t) = P(X > c, T > t)$, provided by Akritas (1994). This estimator is based on the representation: $S(c,t) = \int_c^\infty S(t|X = s)\, dF_X(s)$, where $F_X(s)$ is the distribution function for X. This estimator can be provided by

$$\widehat{S}_{\lambda_n}(c,t) = \frac{1}{n}\sum_i \widehat{S}_{\lambda_n}(t|X = X_i)\mathbb{1}(X_i > c)$$

where $\widehat{S}_{\lambda_n}(t|X = X_i)$ is a suitable estimator of the conditional survival function characterized by a parameter λ_n. Unless X is discrete and there are sufficient observations at each value of X_i, some smoothing is required to estimate $S(t|X = X_i)$. We define the weighted KM estimator as:

$$\widehat{S}_{\lambda_n}(t|X = X_i) = \prod_{s \in \mathcal{T}_n, s \leq t} \left\{ 1 - \frac{\sum_j K_{\lambda_n}(X_j, X_i) \mathbb{1}(Z_j = s)\delta_j}{\sum_j K_{\lambda_n}(X_j, X_i) \mathbb{1}(Z_j \geq s)} \right\}$$

where $K_{\lambda_n}(X_j, X_i)$ is a kernel function that depends on a smoothing parameter λ_n, $\mathcal{T}_n$ is the unique values of Z_i for observed events, $\delta_i = 1$. Akritas (1994) used a 0/1 nearest neighbor kernel, $K_{\lambda_n}(X_j, X_i) = \mathbb{1}\left\{-\lambda_n < \widehat{F}_X(X_i) - \widehat{F}_X(X_j) < \lambda_n\right\}$, where $2\lambda_n \in (0, 1)$ represents the percentage of observations that is included in each neighborhood (except for the boundaries in the distribution of X). This particular choice of kernel and using nearest neighbor approach means that the resulting ROC estimates are invariant to monotone transformations of the marker variable. The resulting estimates of sensitivity and specificity are given by:

$$\widehat{P}_{\lambda_n}\{X > c|D(t) = 1\} = \frac{\left\{1 - \widehat{F}_X(c)\right\} - \widehat{S}_{\lambda_n}(c, t)}{1 - \widehat{S}_{\lambda_n}(t)} \quad (1)$$

$$\widehat{P}_{\lambda_n}\{X \leq c|D(t) = 0\} = 1 - \frac{\widehat{S}_{\lambda_n}(c, t)}{\widehat{S}_{\lambda_n}(t)} \quad (2)$$

where $\widehat{S}_{\lambda_n}(t) = \widehat{S}_{\lambda_n}(-\infty, t)$. In contrast to using a KM estimator, this allows monotonicity of sensitivity and specificity and censoring process is allowed to depend on the diagnostic marker X. This results since only local KM estimators are used in each possible neighborhood of $X = x$. Since in screening studies follow-up will often be more intense for subjects with marker values that appear to be more indicative of disease, this flexibility in the censoring mechanism is likely to be important in practice.

3 Methods

In this section, we will discuss about the package `survivalROC` and the main functions: `survivalROC()` and `survivalROC.C()`. The package consists of these two functions which are intended to estimate time-dependent ROC curve $ROC(t)$ from censored survival data. `survivalROC` is able to estimate $ROC(t)$ at a given time t using either of the methods discussed above while `survivalROC.C` estimates the time-dependent ROC using NNE method. Being based on a `.C()` call, `survivalROC.C()` is comparatively faster than `survivalROC()` as we will demonstrate later.

3.1 The `survivalROC()` function

The `survivalROC()` function accepts censored survival data and is able to return a set of true positive and false positive values for construction of $ROC(t)$ for a given time t using either of KM or NNE method discussed above.

```
> library(survivalROC)
> args(survivalROC)
function (Stime, status, marker, entry = NULL, predict.time, cut.values = NULL,
    method = "NNE", lambda = NULL, span = NULL, window = "symmetric")
NULL
```

We now discuss the arguments to `survivalROC`. We first note that, given a survival data set with survival time (`Stime`), survival status (`status`), and marker value at baseline (`marker`) of the cases, we want to quantify how good the performance of the marker is to distinguish those who died or had an event by a pre-specified time, say, t, from those who did not have an event by this time. Thus the minimal argument to pass on to `survivalROC` are `Stime` denoting the survival time (or censoring time) of the cases, the `status` of the cases (with 1 indicating that the case experienced an event and 0 indicating that the case has been censored) and `marker` denoting the marker value of the case as measured at the baseline. We can additionally pass on a set of `entry` time for the cases, otherwise, time of entry for all the cases would be assigned to be 0 (the baseline). The time-point of interest is `predict.time` and its unit is the same as the unit of `Stime`, meaning that if `Stime` is given in days, then the unit of `predict.time` would also be in days. For example, if we have survival data in days on a set of subjects and we want to see how well the marker can distinguish those who died (or experienced an event) by the first year from those who did not, then the `predict.time` should be 365 (days). We can also include the `cut.values` which are essentially a set of marker values to be used as a cut-off for the calculation of sensitivity and specificity. By default, `cut.values` are not included, and a set of cut-off values are created based on unique marker values. The time-dependent ROC can be calculated by either of two methods: `NNE` (Nearest Neighbor Estimation) and `KM` (Kaplan-Meier), with NNE being the default choice. We note that for `method = "NNE"`, value of either of the two parameters (`lambda` or `span`) need to be supplied. Both of them are related to the smoothing parameter λ_n of $K_{\lambda_n}(.,.)$ as discussed earlier. If a value for `lambda` is given, then a Gaussian kernel is used with `lambda` denoting the spread or standard deviation (σ) of the Gaussian distribution. If instead, `span` is given, then a $0/1$ kernel function is used as discussed in Heagerty et al. (2000). We note here that, when an extreme marker value is used as a cut-off value for the calculation of true positive (TP) and false positive (FP), NNE method limits the number of neighbors to a great extent. To avoid this limitation, with the `span` option, one can also specify an `asymmetric` window so as to consider more marker values in a suitable direction. The default window option is `symmetric`. For `method = "KM"`, no such parameters need to be supplied. We here again emphasize the fact that using `method = "KM"` does not guarantee monotonicity whereas `method = "NNE"` guarantees monotonicity and hence the default `method` choice.

3.2 The `survivalROC.C()` function

```
> args(survivalROC.C)
function (Stime, status, marker, predict.time, span = 0.05)
NULL
```

The arguments for `survivalROC.C` are similar to `survivalROC`. Like `survivalROC` this function also returns a set of (FP,TP) values for construction of $ROC(t)$ and calculates these values by only the NNE method. This method uses a c-code to compute the set of (FP, TP) values corresponding to `predict.time` and hence is much faster than the `survivalROC` function, but limited in options.

Both of the functions returns a list of six items:

- `cut.values` : a set of cut-off values used for computation of (FP, TP), usually, the ordered unique marker values, augmented by $-\infty$.
- `FP` : a set of FP (false positive) values corresponding to the marker and time point of interest `predict.time`.
- `TP` : a set of TP (true positive) values corresponding to the marker and time point of interest `predict.time`.
- `predict.time` : as in the argument of `survivalROC`.

- `Survival` : the survival probability at `predict.time` (Kaplan-Meier survival probability).
- `AUC` : Area Under (ROC) Curve at `predict.time`, i.e., area under $ROC(t)$.

We note here that, when we use the `NNE` method with `survivalROC`, the estimated (FP, TP) and resulting AUC values are little different than those that are returned via `survivalROC.C` function due to the differences in how R and C handles the data and rounding.

3.3 An Illustration: Mayo PBC data

The `survivalROC` package contains a shorter and modified version of Mayo PBC data. This data is from a randomized placebo-controlled trial of the drug D-penicilliamine (DPCA) for the treatment of primary biliary cirrhosis (PBC) conducted at the Mayo Clinic between 1974 and 1984. Among the 312 subjects randomized to the study, 125 died by the end of the follow-up. Heagerty and Zheng (2005) used this data to construct two Cox models, one with five covariates (log(bilirubin), albumin, log(prothrombin time), edema and age) while the other one with four covariates (albumin, log(prothrombin time), edema and age). This version of the Mayo PBC data only contains the survival (or censoring) time in days (`time`), censoring status (`censor`: 1 if died, 0 if censored), and the prognostic score `mayoscore5` based on five covariates model and `mayoscore4` based on four covariates model as mentioned above.

```
> data(mayo)
> str(mayo)
'data.frame':   312 obs. of  4 variables:
 $ time      : int  41 179 334 400 130 223 51 549 216 859 ...
 $ censor    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ mayoscore5: num  11.25 10.14 10.10 10.19  9.77 ...
 $ mayoscore4: num  10.63 10.19  9.42  9.57  9.04 ...
> mayo[1:10,]
   time censor mayoscore5 mayoscore4
1    41      1  11.251850  10.629450
2   179      1  10.136070  10.185220
3   334      1  10.095740   9.422995
4   400      1  10.189150   9.567799
5   130      1   9.770148   9.039419
6   223      1   9.226429   9.033388
7    51      1  10.151430   9.427745
8   549      1   9.473292   8.665512
9   216      1   9.617877   8.154950
10  859      1   9.525200   8.379361
```

Thus, for example, the first subject died 41 days after the study began and he had a prognostic score of 11.25 from the five-covariate model and a score of 10.63 from the four-covariate model. We would now demonstrate the use of the functions `survivalROC` and `survivalROC.C` using this dataset. We will construct ROC curves to see how well each of the two prognostic score can distinguish between cases who died by the first year from those who did not.

```
> attach(mayo)

> ROC.1 = survivalROC(Stime = time, status = censor, marker = mayoscore4,
predict.time = 365, lambda = 0.05)
```

```
> str(ROC.1)
List of 6
 $ cut.values  : num [1:313] -Inf 4.58 4.90 4.93 4.93 ...
 $ TP          : num [1:313] 1.000 0.999 0.999 0.999 0.998 ...
 $ FP          : num [1:313] 1.000 0.997 0.993 0.990 0.987 ...
 $ predict.time: num 365
 $ Survival    : num 0.93
 $ AUC         : num 0.888

> ROC.2 = survivalROC(Stime = time, status = censor, marker = mayoscore4,
  predict.time = 365, method = "KM")

> str(ROC.2)
List of 6
 $ cut.values  : num [1:313] -Inf 4.58 4.90 4.93 4.93 ...
 $ TP          : num [1:313] 1 1 1 1 1 ...
 $ FP          : num [1:313] 1.000 0.997 0.993 0.990 0.986 ...
 $ predict.time: num 365
 $ Survival    : num 0.93
 $ AUC         : num 0.917
```

Given the (FP, TP), we can plot the ROC curve $ROC(t)$ at time $t = 365$ days (Figure 1).
We now compare to show that `survivalROC.C` is faster than `survivalROC`.

```
> system.time(survivalROC(Stime = time, status = censor, marker = mayoscore5,
+   predict.time = 365, span = 0.05) )
[1] 1.006 0.000 1.017 0.000 0.000

> system.time(survivalROC.C(Stime = time, status = censor, marker = mayoscore5,
+   predict.time = 365, span = 0.05))
[1] 0.004 0.000 0.004 0.000 0.000
```

Thus, `survivalROC` takes user CPU time of 1.006 units while `survivalROC.C` takes only 0.004 unit of user CPU time. We can also exploit `survivalROC` or `survivalROC.C` and plot (t, AUC(t)) for different values of t, to see the performance of the marker throughout some period of interest. For example, if we are interested to see the performance of the marker through the 10th year of the study, we can do so as follows:

```
>  AUC4 = NULL; AUC5 = NULL
>  for(t in min(mayo$time):3650)
+   {
+     AUC4 = c(AUC4, survivalROC.C(Stime = time, status = censor, marker = mayoscore4,
+   predict.time = t, span = 0.05)$AUC )
+     AUC5 = c(AUC5, survivalROC.C(Stime = time, status = censor, marker = mayoscore5,
+   predict.time = t, span = 0.05)$AUC )
+   }
+
>  plot(min(mayo$time):3650, AUC4, type = "l", xlab = "Time", ylab =
+ "AUC", ylim = c(0.7,1), col = "red")
>  lines(min(mayo$time):3650, AUC5, col = "blue", lty = 2)
>  legend(2000, 1, legend = c("4 Covariate Model", "5 Covariate Model"),
+ col = c("red", "blue"), lty = c(1, 2), bty = "n")
```

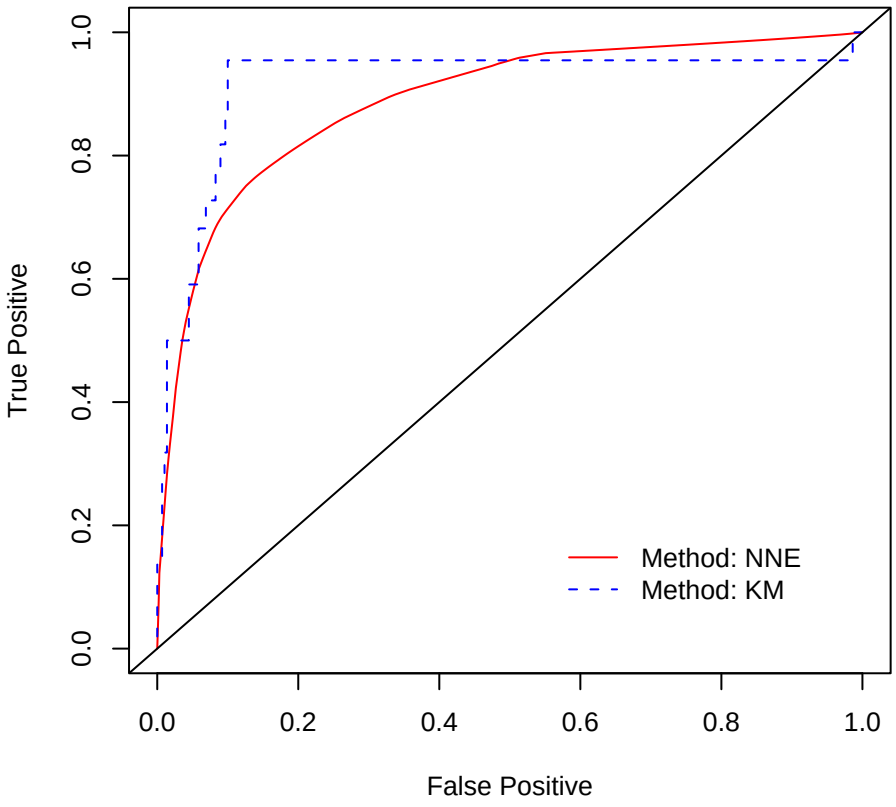

Figure 1: Estimated ROC curves $ROC(t)$ at $t = 365$ days

We can compare the prognostic performances of the two models from Figure 2 and infer that the 5-covariate model performs better than the 4-covariate model.

4 Conclusion

ROC curves are useful tool for showing diagnostic potential of continuous marker in the setting of survival data. Based on the paper by Heagerty et al. (2000), we here introduce and discuss the usage of a new R/S-plus package `survivalROC` and the functions therein, that are able to estimate the discrimination potential of a continuous marker by time-dependent ROC curve $ROC(t)$. We here show the different uses of the package and hope, it will be helpful to the scientific community.

References

Akritas, M. G. (1994). Nearest neighbor estimation of a bivariate distribution under random censoring. *Annals of Statistics*, 22 : 1299–1327.

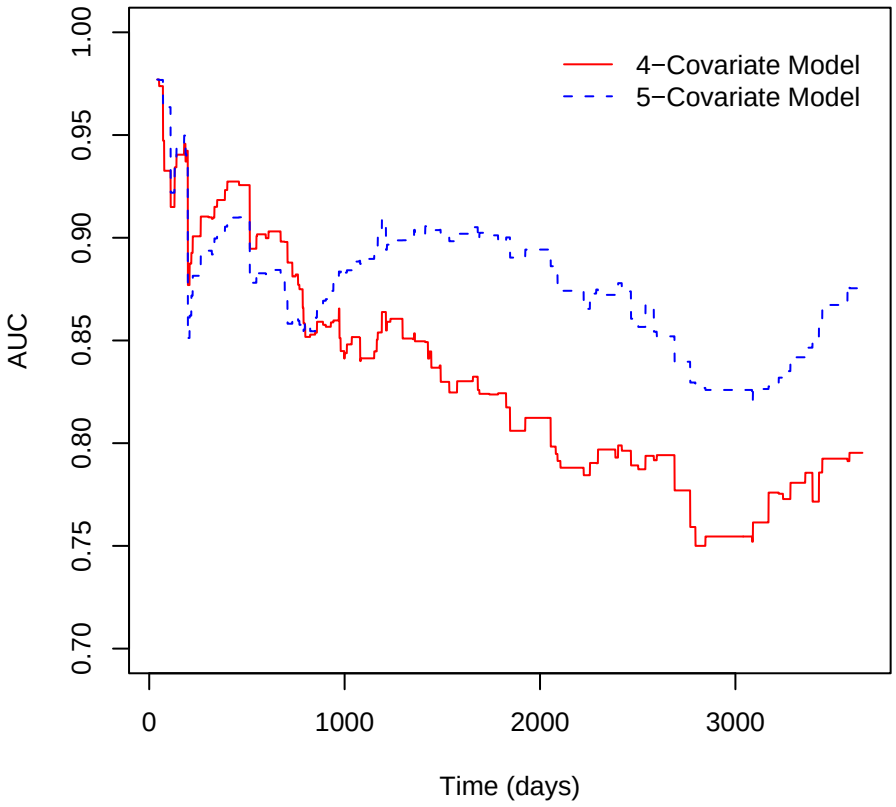

Figure 2: $AUC(t)$ based on four and five covariate model for the first 10 years

Heagerty, P. J., Lumley, T., and Pepe, M. S. (2000). Time-dependent ROC curves for censored survival data and a diagnostic marker. *Biometrics*, 56 : 337–344.

Heagerty, P. J. and Zheng, Y. (2005). Survival model predictive accuracy and ROC curves. *Biometrics*, 61 : 92–105.