Using the Time Dependent ROC Curve to Build Better Survival Model in SAS

Li Lu, [1] Chenwei Liu [2]

[1] Department of Biostatistics and Epidemiology, MedStar Research Institute, Hyattsville, MD

[2]Department of Computer Science, George Washington University, NW, DC

ABSTRACT

The ROC curve is mainly used to evaluate the discrimination power of a continuous variable for a binary outcome. Recently, time dependent ROC curves have been used to assess the predictive power of diagnostic markers for time dependent disease outcomes, thus to analyze censored survival data. Among the various methods of estimating the time dependent ROC curves, the Kaplan-Meier method is based on Bayes' theorem and Kaplan-Meier survival function. It is easy to understand, implement and use. In this paper we implement the Kaplan-Meier estimate of time dependent ROC curves in SAS. Using the data of a clinical study, we demonstrate how time dependent ROC curves and area under the curve can be used to select predictive covariates and build better survival models.

INTRODUCTION

ROC curves have become the standard tool for the purpose of assessing the predictive power of a continuous variable for a binary outcome. In SAS we can use procedures LOGISTIC, FREQ, MIXED and NLMIXED to perform various binary outcome ROC analysis, including ROC curve estimation and the calculation of the area under the ROC curve[1]. Recently, ROC curve analysis has been extended to the scenarios where outcome status can change with time, where using ROC curves that vary as a function of time is more appropriate. Heagerty et al (2000)[2] presented two methods to estimate time dependent ROC curves and showed that the time dependent ROC curves could be used to compare a standard and a modified flow cytometry measurement for predicting the survival time of patients after the detection of breast cancer. Currently, there is no SAS procedure or publicly available SAS macro to directly perform the time dependent ROC curve analysis. In this paper, we implement the Kaplan-Meier estimate of ROC(t) in SAS and demonstrate how time dependent ROC curves can be used to select covariates from a huge number of gene expression variables to build better survival model for a breast cancer study.

DEFINITION

Let T_i denote failure time and X_i the covariate value for subject i. Let C_i denote the censoring time, $Z_i=\min(T_i, C_i)$ the follow-up time, and δ_i a censoring indicator with $\delta_i=1$ if $T_i<=C_i$. $D_i(t)=1$ indicate that subject *i* has had an event prior to time t. ROC curve is a plot of sensitivity v.s 1- specificity for various cut off value c. When outcome variable is time dependent, we can plot time dependent ROC curve ROC(t) as sensitivity(t) vs. 1- specificity(t). The time dependent sensitivity and specificity can be defined as:

$$sensitivity(c,t) = P\{X > c \mid D(t) = 1\}$$

$$specificity(c,t) = P\{X \le c \mid D(t) = 0\}$$

A simple estimator of the time dependent sensitivity and specificity is given by combining the Kaplan-Meier survival estimator and the empirical distribution function of the covariate X as

$$\hat{P}_{KM}\{X > c \mid D(t) = 1\} = \frac{\{1 - \hat{S}_{KM}(t \mid X > c)\}\{1 - \hat{F}_X(c)\}}{\{1 - \hat{S}_{KM}(t)\}}$$

$$\hat{P}_{KM}\{X \le c \mid D(t) = 0\} = \frac{\hat{S}_{KM}(t \mid X \le c)\hat{F}_X(c)}{\hat{S}_{KM}(t)}$$

Where $\hat{S}_{KM}(t)$ is the Kaplan-Meier estimate of the survival of the total set, $\hat{S}_{KM}(t \mid X > c)$ is the Kaplan-Meier estimate of conditional survival of subset X>c, $\hat{S}_{KM}(t \mid X \le c)$ is the Kamplan-Meier estimate of the conditional survival of subset $X \le c$ and $\hat{F}_X(c) = \sum 1(Xi \le c)/n$ is the accumulate distribution function of X at value c.

IMPLEMENTATION OF THE ROC(t) AS A SAS MACRO KM_ROC.

In SAS we can use PROC LIFETEST to generate the Kaplan-Meier estimate of the total survival function $\hat{S}_{KM}(t)$, the conditional survival function of a subset data $\hat{S}_{KM}(t \mid X > c)$ and $\hat{S}_{KM}(t \mid X \le c)$. In order to let the subset have enough observations, we choose cut of value c as starting from the 5th percentile to 95th percentile of X by 20 uniformly increment intervals.

The code of the KM_ROC is shown in the appendix. It has six parameters: ds – the input dataset (contains survival data); v_time – name of the survival time variable in the dataset; v_censor – name of the censor indicate variable in the dataset, 0 indicating censoring; v_x – name of the covariate variable in the dataset; timepoint – value of the timepoint we are interested to generate ROC curve; outds – the output dataset which will have 20 observations, each observation has sensitivity and 1-specificity values; A dataset named &outds._area will also be created which have the area under the curve information. We use the trapezoid approximation method to calculate the area under the curve.

APPLICATION EXAMPLE

As an application example, we use this KM_ROC macro to analyze a publicly available data set of breast cancer gene expression and clinical survival reported by Van de Vijver et al. (2002)[3]. The log ratio column in the dataset represents the expression level of a gene. The dataset consists of log ratios of over 20000 genes for 295 patients who were diagnosed with breast cancer between 1984 and 1995 and treated by modified radical mastectomy or breast-conserving surgery, followed by radiotherapy at the hospital of the Netherlands Cancer Institute. The median follow up time was 7.2 years and the median survival time was 3.8 years for those 295 patients. Running the KM_ROC macro on the 20000 genes is of heavy computation burden. For the illustration purpose in this paper we restrict the KM_ROC process to a small number of genes. We first use PROC LIFETEST to identify significant genes associated with the survival time within the first 2000 genes. There were 448 genes meeting the criteria that the sum of Likelihood Ratio, Score and Wald test p-values is less than 0.15. We run the KM_ROC macro on those 448 significant genes.

```
%km_roc(genedata.gen_clin,timesurvival,eventdeath,AK001380,5,km438);
```

The above code shows how to call the KM_ROC macro. Among the parameters: genedata.gen_clin is the dataset with clinical and gene expression data for 295 patients, timesurvival is the survival time variable, eventdeath is the censor variable with 0 indicating censored observation, AK001380 is the log ratio variable of a gene labeled as 'AK001380' in the data set, 5 (years) is the time point we are interested, km438 is the output data set which will have the 20 pairs of sensitivity and (1-specificity) to plot a ROC curve, a data set named km438_area will also be created which has the area under the curve information.

The above KM_ROC macro was called 448 times to create ROC curves of the 448 significant genes at the 5-year time point. The area under the ROC curve is also calculated by the macro for each gene. Figure 1 shows the ROC curves for two genes, one has the largest area under the cure and one has the 48th largest area under the curve.

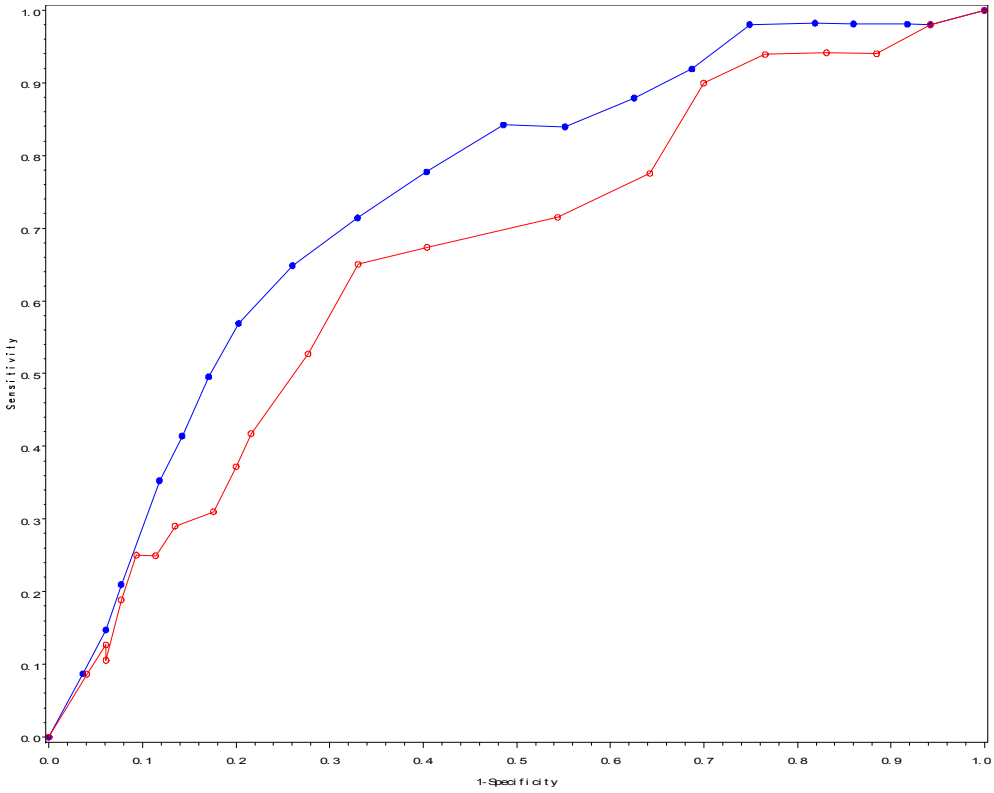

Figure 1. ROC Curves with max AUC and 48th largest AUC.

Among those 448 genes, 48 genes have all the likelihood ratio, score and Wald test p-values less than 0.0001. We run a PROC PHREG (shown below) including the 48 genes with p-values less than 0.0001 as covariates to build a final survival model using stepwise model selection method. We found ten genes entered the final model as factors affecting the survival and the fit statistics –2LOGL equals to 718.261.

```
proc phreg data=genedata.gen_clin;
          model timesurvival*eventdeath(0)=&pval_vars
                / selection=stepwise slentry=0.25
                                     slstay=0.15 details;
run;
```

As a comparison, we also run a PROC PHREG (shown below) including the 48 genes with largest area under the KM_ROC(t=5) curve as covariates to build a final survival model using the same stepwise model selection method as above. We found fifteen genes entered the final model as factors affecting the survival and the fit statistics –2LOGL equals to 702.231.

```
proc phreg data=genedata.gen_clin;
          model timesurvival*eventdeath(0)=&area_vars
                / selection=stepwise slentry=0.25
                                     slstay=0.15 details;
run;
```

CONCLUSTION

We have demonstrated the implementation of Kaplan-Meier estimate of time dependent ROC curves in SAS. The implemented KM_ROC macro can be used to provide additional information about a marker's predictive power and can help evaluate and select predictive markers from many makers such as over 20000 genes. The model built from genes based on their area under the ROC curve found more affecting

genes, it has smaller fit statistics of 702.231 than that (718.261) of model built from small p-value genes, which indicates a better fit of the survival data. One more advantage of the macro is that we can evaluate a marker's predicting power at different time point by specifying different time parameter. The above macro and method can be readily extended to other applications such as proteomic biomarker discovery when the disease outcome is time dependent and there are a large number of protein expression makers to study.

REFERENCES

[1] Gonen, M. (2006). "Receiver Operating Characteristic (ROC) Curves," Proceedings of the Thirty first Annual SAS Users Group International Conference. Cary, NC: SAS Institute Inc.
[2]. 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.
[3]. Van de Vijver, M.J., He, Y.D., et al. (2002). A Gene-expression signature as a predictor of survival in breast cancer. New England Journal of Medicine 347, 1999-2009.

ACKNOWLEDGMENTS

SAS is a Registered Trademark of the SAS Institute, Inc. of Cary, North Carolina.

CONTACT INFORMATION

Your code requests, comments and questions are valued and encouraged. Contact the author at

Li Lu
Dept of Epidemiology and Statistics
MedStar Research Institute
6495 New Hampshire Avenue
Hyattsville, MD 20783
(301) 560-7313
li.lu@medstar.net

APPENDIX

```
1    %macro km_roc(ds,v_time,v_censor,v_x,timepoint, outds);
2    proc sort data =&ds (keep= &v_time &v_censor &v_x) out=sortds;
3        by &v_x;
4    run;
5    data _null_;
6        set sortds end=last;
7        if last then call symput('total_n',_n_);
8    run;
9    proc univariate data=sortds noprint ;
10       var &v_x;
11       output out=dum p5=p5 p95=p95;
12   run;
13   data _null_;
14       set dum;
15       call symput('p5',p5);
16       call symput('p95',p95);
17   run;
18   data &outds;
19   run;
20   *get Skm(t);
21   proc lifetest data=&ds (keep= &v_time &v_censor &v_x) method=KM
22           outsurv=dum noprint;
23         time &v_time*&v_censor(0);
24   run;
25   proc sort data=dum(where=( &v_time<=&timepoint)) out=surv_all;
```

```
26      by descending &v_time;
27  run;
28  data _null_;
29      set surv_all(obs=1);
30      call symput('Skm_',survival);
31  run;
32  *choose 20 different cutoff value and get the corresponding
33  sensivity and specificity;
34  %do i=1 %to 20;
35      data sortds;
36          set sortds;
37          if &v_x >&p5 +(&i-1)*(&p95-&p5)/19 then riskc=1;
38          else riskc=0;
39      run;
40      data dum;
41          set sortds(where=( riskc=0));
42      run;
43      *get accumulate distribution function Fx(c);
44      data _null_;
45          set dum end=last;
46          if last then call symput('Fxc',_n_/&total_n);
47      run;
48      *get KM survival function by x > or < c subgroup;
49      proc lifetest data=sortds  method=KM  outsurv=surv_by;
50          time &v_time*&v_censor(0);
51          strata riskc;
52      run;
53      proc sort data=surv_by(where=( &v_time<=&timepoint))
54              out=dummy;
55          by stratum descending &v_time;
56      run;
57      *store Skm(t|x<c) in Skm_c1, Skm(t|x>c)in Skm_c2;
58      data _null_;
59          set dummy;
60          by stratum;
61          if first.stratum and stratum=1 then
62              call symput('Skm_c1',survival);
63          if first.stratum and stratum=2 then
64              call symput('Skm_c2',survival);
65      run;
66      *calculate sensivity and 1-specificity;
67      data roc1;
68          sens=(1-&Skm_c2)*(1-&Fxc)/(1-&Skm_);
69          spec=&Skm_c1*&Fxc/(&Skm_);
70          x=1-spec;
71      run;
72      data &outds;
73          set &outds roc1;
74      run;
75  %end;
76  data dum;
77      x=0; sens=0;
78  run;
79  data &outds;
80      set &outds dum;
81      if x>. and sens>.;
82  run;
```

```
83   proc sort data=&outds;
84       by x;
85   run;
86   *prepare to calculate area under the curve;
87   data &outds;
88       set &outds end=last;
89       xprev=lag(x);
90       yprev=lag(sens);
91       output;
92       if last then do;
93           xprev=x;
94           yprev=sens;
95           x=1;
96           sens=1;
97           output;
98       end;
99   run;
100  data &outds._area;
101      retain area 0;
102      set &outds(firstobs=2) end=last;
103      area=area+(sens+yprev)*(x-xprev)/2;
104      var="&v_x";
105      if last then output;
106  run;
107  data &outds(rename=(x=x_&outds sens=y_&outds));
108      set &outds;
109      if sens>.;
110      keep x sens;
111  run;
112  %mend;
```