Paper 343-2011

Compare the Effectiveness of Different Methodologies in Prediction of Dichotomous Outcome

Jimmy Fang, Thomson Reuters Healthcare & Science, Ann Arbor, MI

ABSTRACT

Dichotomous outcomes widely exist in nature including death or live, disease or no disease, success or fail, etc. Many methodologies could be developed to forecast a same dichotomous outcome. The difference among the methodologies may include explanatory variables, post-model adjustment, or even various model functions. The area under the receiver operating characteristic (ROC) curve, or its equivalent c statistic, is the most important parameter to assess the effectiveness of the prediction. In this paper we describe how to get ROC curve and c statistic from the validation data and compare the effectiveness among different methodologies. We provide two ways to make ROC curve and calculate c statistic based on the observed events and the predicted probabilities from different methodologies. One way is to transfer the predicted probability to logit and then fit the logit with the observed outcome using PROC LOGISTIC. Another way is to do the basic calculation using SAS macros. The application and advantage of each way are discussed.

INTRODUCTION

Dichotomous outcome is widely existed in nature including death or live, disease or no disease, success or fail, etc. Logistic regression is the most common model used to predict the dichotomous outcome, but many methodologies may be developed to forecast a same dichotomous outcome. The difference among the methodologies may include the definition or number of the explanatory variables, post-model adjustment, or various model functions. c statistic or its equivalent, the area under the receiver operating characteristic (ROC) curve is the most important parameter to assess the effectiveness of the prediction.

Suppose we have got the predicted probabilities for the same individuals from two or more various methodologies and we want to compare the effectiveness of those methodologies. This paper describes how to get c statistic and ROC curve for comparison. We provide two ways to calculate c statistic and make ROC curve based on the observed event and the predicted probability from various methodologies. One way is to transfer the predicted probability to logit and then fit the logit with the observed event using SAS® procedure LOGISTIC (PROC LOGISTIC). Another way is to do the basic calculation. We present the macros for both ways.

CALCULATE C STATISTICS AND CONSTRUCT ROC CURVES WITH SAS PROC LOGISTIC

This is a simple way and uses the existing SAS procedure LOGISTIC to calculate c statistic and create data for sensitivity and 1-specificity. The predicted probability form different methodologies need to be transferred to logit and then fit the logit with the observed event using PROC LOGISTIC. ROC curve can be plotted with PROC GPLOT by using the data from OUTROC statement. Below is the SAS code.

```
%MACRO mkroc1 (in_dataset, num_mthd);
*Transfer probability to logit;
%DO i=1 %TO &num_mthd;
DATA wdataset;
  SET &in_dataset;
  logit_m&i = log (pred_m&i /(1-pred_m&i));
RUN;

*Get c statistic and the data for ROC curve using proc logistic;
PROC LOGISTIC DATA= wdataset DESCENDING ;
  MODEL obs  = logit_m&i / OUTROC = roc_dataset&i;
  ODS OUTPUT association=assoc&i ;
RUN;
DATA assoc&i (KEEP=nValue2);
  set assoc&i;
```

```
  if Label2='c';
  FORMAT nValue2 6.4;
RUN;
PROC SQL NOPRINT;
  SELECT nValue2 INTO: c_stat_&i FROM assoc&i;
QUIT;

*Organize data for ROC curve;
DATA roc_dataset&i (keep=Methodology _PROB_ _sensit_ _1mspec_);
  SET roc_dataset&i;
  Methodology = &i;
RUN;
PROC APPEND BASE=dataf DATA=roc_dataset&i; RUN;
%END;

/*
DATA dataf;
  SET dataf;
  _PROB_ = ROUND (_PROB_, 0.0001);
  RUN;
PROC SORT DATA=dataf NODUPKEY;
  BY methodology _PROB_ ;
  RUN;
*/

*************Display ROC curves*****************;
AXIS1 order=(0 to 1 by .1 ) label=("1-Specificity") length=75;
AXIS2 order=(0 to 1 by .1 ) label=(angle=90 "Sensitivity");
SYMBOL1 value=dot c=blue w=2 l=1;
SYMBOL2 value=dot c=red  w=2 l=2;
PROC GPLOT DATA=dataf;
  PLOT _sensit_ * _1mspec_ = methodology /VAXIS=AXIS2 HAXIS=AXIS1;
  TITLE1 "ROC Curve in Different Methodologies";
  TITLE2 "c-statistic: Mthd1= &c_stat_1 , Mthd2= &c_stat_2 ";
RUN;
%MEND;
```

The in_dataset consists of the observed (obs) and the predicted probability (pred_m&i) from various methodologies for each observation. The probabilities are transferred to logit and then serve as the only independent variable for MODEL in PROC LOGISTIC.

PROC LOGISTIC calculates and puts the c statistic in the ASSOCIATION table, which is pulled out and passed to the macro variables "c_stat_&i". Here i can be 1,2,... or N representing the input number of methodologies to compare. PROC LOGISTIC also allows computation of sensitivity and specificity for a serial of threshold z between 0 and 1 and the output data is associated with the option OUTROC. z is correspondent to all unique values of probability in the data.

Procedure GPLOT (PROC GPLOT) creates ROC curves and we let it print the values of c statistic assigned to the corresponding macro variables under the title of chart. In this macro, two methodologies were compared. The process can be extended to compare more methodologies at same time. If more than 2 methodologies are compared, then more symbols (3,4,…N) need to be defined for PROC GPLOT.

Below is the code to call the macro and it creates a chart in PDF format. The output file name can be defined with the macro variable "outputfnm".

```
ODS PDF FILE = "&outputfnm..pdf";
OPTION CENTER;
%mkroc1 (in_dataset=all, num_mthd=2);
ODS PDF CLOSE;
RUN;
```

Figure 1 is the example chart created by running the SAS program. The chart shows c statistics and displays ROC curves of two methodologies that predict the same event in a validation sample.

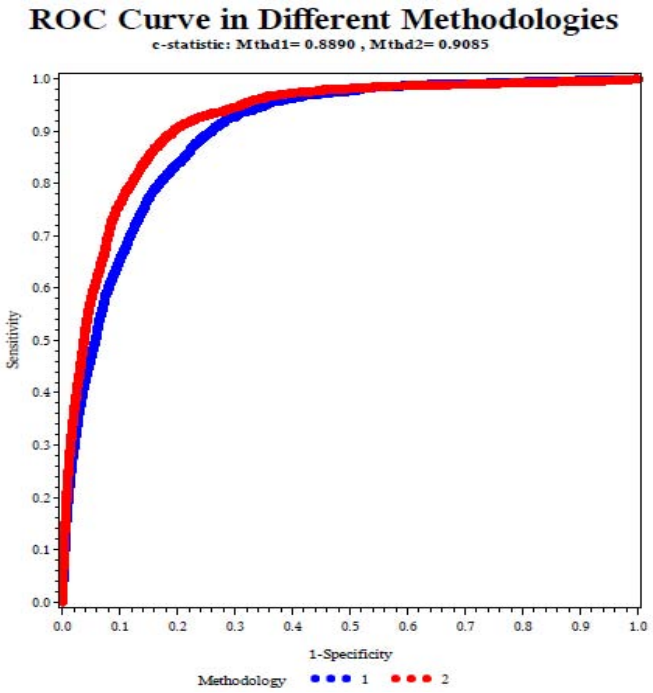

Figure 1. ROC curves made from the macro with PROC LOGISTIC

In the above macro, a paragraph of SAS code (before the code displaying chart) was commented out. The SAS code was added to run Chart 2. The function of the code is to round the cut points to 0.0001 and then delete any duplicated cut points.

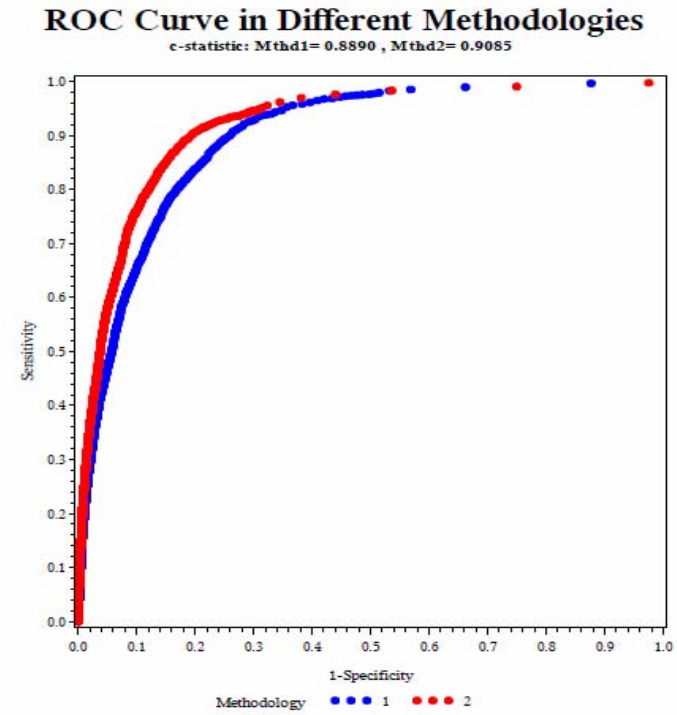

Figure 2. ROC curves made from the macro with PROC LOGISTIC and the cut points were rounded to 0.0001

CALCULATE C STATISTICS AND CONSTRUCT ROC CURVES WITH BASIC CALCULATIONS

We can also calculate c statistic and construct ROC curve by writing SAS code based on their definitions.

c statistic is calculated based on the count of concordant, discordant, ant tied pairs in the total evaluated pairs of all observations. The total evaluated pairs are formed by each observation with event to every observation without event. A pair is concordant if the observation with event has higher predicated probability than the observation without event. A pair is discordant if the predicated probabilities show reversed pattern to the concordant. A pair is tied if the observation with event has equal predicated probability to the observation without event. c statistic is related to a parameter called Somer's D.

Somer's D = (#Concordant - #Discordant) / #Total_Pairs

c statistic = 0.5 (1 + Somer's D) = (#Concordant + 0.5 x #Tied) / #Total_Pairs

When weight is in consideration, the products of the weights between the observation with event and the observation without event are used to calculate the concordant, discordant, and tied.

In the below SAS macro, we make data for ROC curves with basic calculation. Here is the algorithm: Set arbitrary cut-points (z) from 0 to 1 by 0.0001; Calculate the sensitivity and 1-specificity for each cut-point based on the predicted probability and outcome. When weight is in consideration, the weight variable is used in calculation of sensitivity and 1-specificity.

```
%MACRO mkroc2 (in_dataset, num_mthd, wght=1);
DATA &in_dataset; SET &in_dataset; wghtv=&wght; RUN;
PROC SQL NOPRINT;
  SELECT COUNT(*) INTO: allcnt    FROM &in_dataset;
  SELECT COUNT(*) INTO: eventcnt  FROM &in_dataset WHERE obs=1;
  SELECT COUNT(*) INTO: neventcnt FROM &in_dataset WHERE obs=0;
QUIT;

*************Calculate c statistic*********************;
DATA dsevent dsnevent;
  SET &in_dataset;
  IF obs=1 THEN OUTPUT dsevent ;
  ELSE IF obs=0 THEN OUTPUT dsnevent ;
RUN;

*Transpose probability of each methodology for observations with event;
%DO i = 1 %TO &num_mthd;
PROC TRANSPOSE DATA=dsevent OUT=eventprab&i (DROP=_name_)
  PREFIX=eventp;
  VAR pred_m&i ;
RUN;
%END;
*Transpose weight for observations with event;
PROC TRANSPOSE DATA=dsevent OUT=eventwght (DROP=_name_)
  PREFIX=eventw;
  VAR wghtv ;
RUN;

*For each methodology, count for concordant, discordant, and tied;
%DO i=1 %TO &num_mthd;
DATA c_stat&i (keep= methodology c_statistic pairs
                     pctconcordant pctdiscordant pcttied);
  SET dsnevent;
  IF _n_= 1 THEN SET eventprab&i;
  IF _n_= 1 THEN SET eventwght;
  methodology =  &i ;
  ARRAY eventsp {&eventcnt} eventp1 - eventp%left(&eventcnt);
  ARRAY wghts {&eventcnt} eventw1 - eventw%left(&eventcnt);
  RETAIN concordant discordant tied;
```

```
  IF _n_= 1 THEN DO;
    concordant=0;
    discordant=0;
    tied=0;
    END;
  DO k=1 TO &eventcnt;
    IF pred_m&i < eventsp[k] THEN
      concordant= concordant + (wghts[k] * wghtv);
    ELSE IF pred_m&i > eventsp[k] THEN
      discordant= discordant + (wghts[k] * wghtv);
    ELSE IF pred_m&i = eventsp[k] THEN
      tied= tied + (wghts[k] * wghtv);
      END;
    IF _n_ = &neventcnt THEN DO;
      pairs =  concordant + discordant + tied ;
      c_statistic = (concordant + 0.5 * tied) / pairs ;
      pctconcordant = (concordant / pairs) * 100 ;
      pctdiscordant = (discordant / pairs) * 100 ;
      pcttied       = (tied       / pairs) * 100 ;
      OUTPUT;
      END;
  RUN;
PROC PRINT DATA= c_stat&i ; RUN;
*Create macro variable for c statistic calculated above;
PROC SQL NOPRINT;
  SELECT c_statistic INTO: c_stat_&i FROM c_stat&i;
QUIT;
%END;

*************Make data for ROC curve******************;
%DO i=1 %TO &num_mthd;
  %DO j=1 %TO 10000;
  DATA dataj (keep= methodology z_value _sensit_ _1mspec_);
    SET &in_dataset;
    methodology= &i;
    z_value = &j /10000;
    RETAIN truepos falsepos falseneg trueneg ;
    IF _n_=1 THEN DO;
      truepos=0; falsepos=0; falseneg=0; trueneg=0;
      _sensit_=0; _1mspec_=0;
      END;
    IF obs=1 AND (pred_m&i>=z_value) THEN truepos=truepos+wghtv;
      ELSE IF obs=0 AND (pred_m&i>=z_value) THEN falsepos=falsepos+wghtv;
      ELSE IF obs=1 AND (pred_m&i< z_value) THEN falseneg=falseneg+wghtv;
      ELSE IF obs=0 AND (pred_m&i< z_value) THEN trueneg =trueneg +wghtv;
    IF _n_ = &allcnt THEN DO;
      _sensit_ = truepos / (truepos+falseneg);
      _1mspec_ = 1 - trueneg / (trueneg+falsepos);
      OUTPUT;
      END;
  RUN;
  PROC APPEND BASE=datai DATA=dataj FORCE; RUN;
  %END;
%END;

*************Display ROC curves******************;
AXIS1 order=(0 to 1 by .1) label=("1-Specificity") length=75;
AXIS2 order=(0 to 1 by .1) label=(angle=90 "Sensitivity");
SYMBOL1 value=dot c=blue w=2 l=1;
SYMBOL2 value=dot c=red  w=2 l=2;
PROC GPLOT DATA=datai;
  PLOT _sensit_ * _1mspec_ = methodology /VAXIS=AXIS2 HAXIS=AXIS1;
  TITLE1 "ROC Curve in Different Methodologies";
```

```
  TITLE2 "c-statistic: Mthd1= &c_stat_1, Mthd2= &c_stat_2";
  RUN;
PROC DATASETS; DELETE datai; RUN;
%MEND;
```

At the beginning of the macro, the weight variable is assigned. Following it, several macro variables are created for the total count, the count of observations with event and the count of observations without event. These macro variables will be used later.

The next block of code is to calculate c statistic. It pairs each observation with event to every observation without event and then count for the concordant, discordant, and tied pairs. c statistics are calculated and passed to the macro variables that can be printed in the chart of ROC curves.

Another block of code is to make data for ROC curve. Sensitivity and 1-specificity are calculated for every arbitrary cut point z. The predicted positive is defined as the predicted probability is bigger than or equal to z. The predicted negative is defined as the predicted probability is less than z. Sensitivity (_sensit_) is the portion of observations with events that were predicted positive to all observations with event. 1-specificity (_1mspec_) is the portion of observations without event that were predicted positive to all observations without event. When the value of z changes, both sensitivity and 1-specificity are calculated correspondently. ROC curve were then made with PROC GPLOT by plotting the sensitivity and 1-specificity.

With this macro the user has options to use weight or not for calculating c statistics and making ROC curves.

When run the macro without weight, just let wght=1. Here is the code to call the macro.

```
%mkroc2 (in_dataset=all, num_mthd=2, wght=1);
```

Figure 3 shows the chart displaying the ROC curves using the basic calculation macro without weight (i.e. wght=1).

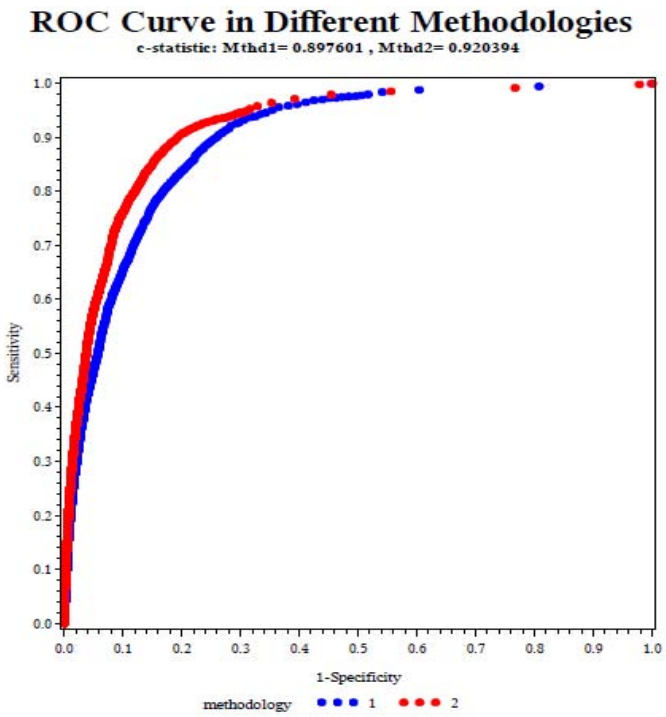

Figure 3. ROC curves made from the macro with basic calculation and weight was not used

Here is the code to call the macro with weight. pweight is the weight variable in the input dataset. We did not show the chart created with weight.

%***mkroc2*** (in_dataset=all, num_mthd=2, wght=pweight);

COMPARISON OF ROC CURVES AND C STATISTICS MADE FROM THE MACRO WITH PROC LOGISTIC AND THE MACRO WITH BASIC CALCULATION

The ROC curves made from the macro with PROC LOGISTIC are smooth and linked lines (Figure 1). To make data for ROC curve in the basic calculation, we used arbitrary cut points from 0 to 1 by 0.0001. Therefore, the ROC curves made from this macro are not totally linked lines (Figure 3). We can set the interval of the cut points by 0.00001 or even smaller to make the dots in higher density and the lines smoother. But the program will take much longer time to run. The chart in Figure 2 was made from the macro with PROC LOGISTIC, but the cut points were rounded to 0.0001 in order to compare the output ROC curves with those made from the basic calculation. As you can see, the ROC curves in Figure 2 and Figure 3 are very comparable. It is not surprise that the two charts are not identical because the ROC curves in Figure 2 were based on the recalculated probabilities from PROC LOGISTIC. The recalculated probabilities are equal to the original probabilities only if the intercept is 0 and the coefficient is 1. However, the ROC curves can be very similar with those made from the original probabilities if the intercept is small and the coefficient is close to 1. In our example, the intercept is -0.0732 and the coefficient is 0.9693 for Methodology 1. The intercept is -0.0662 and the coefficient is 0.9960 for Methodology 2.

In the below table, the c statistics are compared between the outputs from the macro with PROC LOGISTIC and the macro with basic calculation.

Table 1. c statistics and the related outputs from the two different macros that compares the effectiveness of two methodologies

Method	Output From	c statistic	Total Pairs	Concordant	Discordant	Tied
1	Proc Logistic	0.889	965604292	80.400%	8.600%	5.000%
	Basic Calculation	0.898	965604292	89.757%	10.241%	0.002%
2	Proc Logistic	0.909	965604292	88.100%	6.400%	5.500%
	Basic Calculation	0.920	965604292	92.030%	7.968%	0.002%

The small differences were noticed between the c statistics from the two macros. For Methodology 1, the macro with PROC LOGISTIC had c statistic of 0.889 while the macro with basic calculation got c statistic of 0.898. For Methodology 2, the macro with PROC LOGISTIC had c statistic of 0.909 while the macro with basic calculation got c statistic of 0.920. The difference may come from the rounding of the probabilities when counting for the concordant, discordant, and tied. As you can see from the table, the PROC LOGISTIC output had much higher percentage of tied pairs than that of the basic calculations. The PROC LOGISTIC may round the probabilities in a higher decimal position during pairing and counting.

DISCUSSION

We introduced two ways to calculate c statistic and construct ROC curves. The first one (using PROC LOGISTIC) is simple and quick (the SAS code took about 1 minute to run for the demo chart – Figure 1 or 2). The second one involves more SAS code for calculation and it is much slower to run (it took >1 hour to get the demo chart - Figure 3). All charts were made by running Window SAS v9.2.

The c statistic would be same between the two calculations. When the coefficient >0, the ranks of the re-calculated probabilities from PROC LOGISTIC should be identical to those of the originally predicted probabilities if rounding at the same level for pairing.

The ROC curves are comparable between the two ways, especially when the cut points are set at same decimal level in our example. Please keep in mind that the remodeling influences ROC curve. The values of the intercept and coefficient of the remodeling model should be reviewed when explain the results from the macro with PROC LOGISTIC. It may not happen to the meaningful data, but if the coefficient≤0 in the remodeling output, then the output results from the macro 1 are not reliable.

CONCLUSION

We introduced two SAS macros to calculate c statistics and make ROC curves in same chart to compare the effectiveness of different predictive methodologies. The first macro is good for screening models and/or methodologies because it can quickly return the results. The second macro takes much longer time to run, but it gives more accurate results. Besides, the second macro allows calculating c statistic and making ROC curve with weight in count.

REFERENCES

Allison, P.D. (2003). *Logistic Regression Using the SAS System*. Cary: SAS Institute Inc.

Zweig, M.H, Campbell, G. (1993). "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine." *Clinical Chemistry,* 39 (8), 561-577.

Hanley, J.A. and McNeil, B.J. (1982). "the Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve." *Radiology*, 143, 29-36.

Izrael, D., Battaglia, A.A., Hoaglin, D.C.,Battaglia, M.P. (2002). "Use of the ROC Curve and the Bootstrap in comparing Weighted Logistic Regression Models." *SAS SUGI Proceedings.* Paper 248-2002.

Izrael, D., Battaglia, A.A., Hoaglin, D.C.,Battaglia, M.P. (2003). "SAS Macros and Tools for Working with Weighted Logistic Regression Models That Use Survey Data." *SAS SUGI Proceedings.* Paper 275-2003.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jimmy Fang
Thomson Reuters Healthcare & Science
777 East Eisenhower Parkway
Ann Arbor, Michigan 48108
jimmy.fang@thomsonreuters.com