Paper 248-27

Use of the ROC Curve and the Bootstrap in Comparing Weighted Logistic Regression Models

David Izrael, Annabella A. Battaglia, David C. Hoaglin, and Michael P. Battaglia, Abt Associates Inc., Cambridge, MA

Abstract

In analyzing data from a survey, researchers often need to compare the effectiveness of several logistic regression models. The receiver operating characteristic curve offers one way to measure effectiveness of prediction, by calculating the area under the curve (AUC). We present a SAS® macro for calculating AUC that takes the survey weights into account. For comparing logistic regression models, one needs to assess differences in AUC against the variation in the data. We demonstrate the use of the SAS SURVEYSELECT procedure to create a set of 1,000 bootstrap samples and give some background on the calculation of separate weights for each bootstrap sample. For each sample, the AUC macro is then used to calculate the AUC for each model. We show how to use the bootstrap results to assess the significance of the difference in predictive ability of the two models.

1. **Introduction**

In analyzing data from a survey, we needed to compare the effectiveness of several logistic regression models. The receiver operating characteristic (ROC) curve offers one way to measure effectiveness of prediction, by calculating AUC. Then, for the comparisons, we needed to assess the differences in AUC against the variation in the data. We had already developed a substantial set of bootstrap samples, and those allowed us to calculate a bootstrap standard error for the difference in AUC, without making any distributional assumptions. With this brief overview we now describe these components and then discuss the application of them in our study. A key ingredient is the sampling weights associated with the survey data.

The receiver operating characteristic curve is often used to describe the accuracy of tests in diagnostic medicine, as summarized in the review by Pepe (2000). Briefly, the test yields a numerical result X, such that larger values are more indicative of disease. One can choose a threshold z and dichotomize the test by defining X $\geq$ z as a positive result. From subjects whose true disease status is known (both diseased and nondiseased), one obtains the false-positive rate and the false-negative rate for each value of z. The ROC curve is obtained by plotting 1 minus the false-negative rate against the false-positive rate for all possible choices of z. That is, each value of z yields a point on the curve, which includes the point (0,0) (if z is high enough, the test produces no positives) and the point (1,1) (if z is low enough, all outcomes are positive).

The area under the ROC curve provides a summary of the accuracy of the diagnostic test. As Pepe points out, the AUC "can be interpreted as the probability that the test result from a randomly chosen diseased individual is more indicative of disease than that from a randomly chosen nondiseased individual." This interpretation or equivalence, discussed also by Hanley and McNeil (1982), focuses attention on the distributions of the test result (for example, the concentration of a chemical in blood) in diseased and nondiseased persons. If the two distributions are clearly separated, the probability will be close to 1; but if they are centered at the same value, the probability will be ½. In the context of logistic regression we refer to event cases and non-event cases, rather than diseased and nondiseased persons. The "test result" is the predicted probability of an event, from the logistic regression model.

The bootstrap (Efron 1982) uses resampling to provide a basis for studying the behavior of estimates. For a simple random sample of size *n*, with observations $x_1, x_2, \ldots, x_n$, the main steps involve setting *B* (the number of "bootstrap samples," usually large); using sampling *with replacement* to draw a bootstrap sample of *n*, $X_1^*, X_2^*, \ldots, X_n^*$, from the set $\{ x_1, x_2, \ldots, x_n \}$ (*B* times, independently); and calculating the estimate*, t,* from each bootstrap sample to obtain $t_1^*, t_2^*, \ldots, t_B^*$. Analysis of the t_b^* then yields information on the sampling distribution of *t* when the data come from the population that underlies $x_1, \ldots, x_n$. For example, the sample standard deviation of the t_b^* is the bootstrap standard error of t.

When the data are a sizable sample from a survey with survey weights, both the calculation of the area under the ROC curve and application of the bootstrap require considerable special programming. Section 2 discusses the use of SAS to calculate AUC in the presence of survey weights. Section 3 comments on comparing the predictive value of logistic regression models. Section 4 sketches the basic framework for applying the bootstrap to a complex sample survey, and Section 5 illustrates the use of PROC SURVEYSELECT to create bootstrap samples. When the survey weights involve adjustments, the elements of a bootstrap sample cannot simply inherit the weights that they had in the original sample. Section 6 discusses the need to recalculate weights, so that each bootstrap sample has its own complete set of "replicate weights." Section 7 then reports on the use of the bootstrap to estimate the standard error of AUC for a logistic regression model and the standard error of the difference in AUC between two such models. Finally, Section 8 adds some concluding discussion.

2. Using SAS to Estimate the Area under the ROC Curve

We often use PROC LOGISTIC to fit logistic regression models to weighted survey data. In fitting a model, PROC LOGISTIC takes the survey weights into account, but it ignores them in calculating the ingredients of the ROC curve. Those ingredients are stored in the OUTROC data set, which keeps one record for each distinct predicted probability and has the following variables (whose values correspond to using that probability as the threshold): _POS_ - the number of correctly predicted event responses; _ NEG_ - the number of correctly predicted non-event responses; _FALPOS_ - the number of falsely predicted event responses; _FALNEG_ - the number of falsely predicted non-event responses; _SENSIT_ - the sensitivity, which is the proportion of event observations that were predicted to have an event response; and _1MSPEC_ - 1 minus specificity, which is the proportion of non-event observations that were predicted to have an event response.

In the presence of survey weights, the variables for the ROC curve are not computed correctly and look exactly the same as if there were no weights. The predicted probabilities, however, are correct. To calculate the AUC in the presence of survey weights, we wrote a macro, CALCAUC, which takes the weights into account when calculating the variables for the ROC curve. We give an overview of the macro below and consider its application, both as a stand-alone program and as a subroutine in a bootstrap procedure.

Macro CALCAUC

Algorithm

The macro calculates the variables for the ROC curve and then the AUC in the presence of survey weights. To formalize the algorithm, we incorporate weights in the definitions of the ROC curve variables described in Chapter 39, The Logistic Procedure, of SAS/STAT® documentation (Version 8).

Let the weighted number of individuals in a sample having a certain event be $n_{1.}$ Let this group be denoted by C_1, and let the group of the remaining n_2 (weighted) individuals who do not have the event be denoted by C_2. Let $\hat{p}$ be an estimated probability of the event in the weighted model. *W(•)* denotes the weighted indicator function. For example, if $\hat{p}_i \ge z$, $W(\hat{p}_i \ge z)$ is the sampling weight of individual i. For each cutpoint z,

$$\text{\_POS\_ (z)} = \sum_{i \in C_1} \mathrm{W}(\hat{p}_i \ge z) \qquad (1)$$

$$\text{\_FALPOS\_ (z)} = \sum_{i \in C_2} \mathrm{W}(\hat{p}_i \ge z) \qquad (2)$$

$$\text{\_SENSIT\_(z)} = \text{\_POS\_(z)}/n_1 \qquad (3)$$

$$\text{1MSPEC\_(z)} = \text{\_FALPOS\_(z)}/\ n_2 \qquad (4)$$

Note that _POS_ (z) is the weighted number of correctly predicted event responses, _FALPOS_ (z) is the weighted number of falsely predicted event responses, _SENSIT_(z) is the weighted sensitivity of the model, and _1MSPEC_(z) is one minus the weighted specificity of the model. Having calculated _SENSIT_ and _1MSPEC_ , we use them to calculate the AUC as the sum of the area of trapezoids. Formally, if S is a set of cutpoints joined with 0 as the initial one and 1 as the last one, the AUC can be expressed by the formula :

$$\text{AUC} = \sum_{i \in S} .5\ (\text{\_1MSPEC\_}_{i+1} - \text{\_1MSPEC\_}_{i})(\text{\_SENSIT\_}_{i+1} + \text{\_SENSIT\_}_{i}) \qquad (5)$$

Exhibit 1 presents the macro (with line numbers). We now describe its functions section by section and discuss such issues as computational efficiency and resource consumption.

Overview of the code

Lines 1 – 23 contain the macro's input parameters; `model` represents the string of explanatory variables, all of which must be categorical (otherwise the number of distinct predicted probabilities could be very large); `depvar` is a response variable, assumed to have the value of 1 for an event and 0 for a non-event; `round` and `acceler` control efficiency of the macro and will be described below; `replica` must be blank when running the macro as a stand-alone program; otherwise it must be assigned the name of a macro variable that serves as a replicate counter when calculation of AUC is done for each bootstrap replicate.

Lines 25 – 62 check that the variables in the model are present in the input data set. If not, the macro outputs names of absent variables into the LOG and stops (Lines 45 and 60).

Lines 64 – 70 represent PROC LOGISTIC’s statements and options. The data set specified in the OUTROC option will contain the distinct estimated probabilities (_PROB_), which will serve as the cutpoints mentioned in the Algorithm section. Also, the data set _PROBS specified in the option OUT will include all variables of the input data set, along with the predicted probability _P_HAT.

Lines 72 - 88 contain optional statements that are intended to accelerate the computational process. Computing time is especially sensitive to the number of distinct estimated probabilities. We suggest reducing computing time by rounding the predicted probabilities (lines 76 and 85). The impact of the rounding on the precision of the calculated AUC is ordinarily minimal: we observed a difference only in the fifth decimal place. Lines 79 – 81 restore the original descending order of the predicted probabilities, as rounding could change the ordering.

Lines 90 – 115 calculate the weighted variables associated with the ROC curve, _POS_ and _FALPOS_ in particular, following formulas (1) and (2). The outer DO-loop (line 92) sets sequentially the distinct predicted probability from OUTROC data set and passes it through the whole _PROBS data set , which is accessed directly in the inner DO-loop (line 98).

Lines 117 – 124 calculate _SENSIT_ and _1MSPEC_ according to formulas (3) and (4).

Lines 126 – 142 calculate the AUC by formula (5). If we are computing the AUC for each bootstrap replicate, the name of the output data set with the calculated value of AUC contains the replicate number. In this situation the name of the variable with the calculated area contains the replicate number as well (line 141).

Exhibit 1: CALCAUC Macro

```
1  %macro calcauc(dsanal = ,    /* INPUT DATA SET      */
2
3            outds  = c,         /* OUTPUT DATA SET WITH AUC*/
4
5            id     = ,          /* ID VARIABLE              */
6
7            weight = ,          /* SURVEY WEIGHT            */
8
9            model =  ,          /* ALL EXPLANATORY VAR's.  */
10                               /* MUST BE CATEGORICAL      */
11
12            depvar = ,         /* DEPENDENT VARIABLE       */
13
14           round  = .001,   /* PRECISION AT WHICH TO     */
15                            /* ROUND PRED PROBABIL       */
16
17           replica = ,      /*  COUNTER OF REPLICATES    */
18                            /* WHEN BOOTSTRAP IS USED    */
19
20           acceler = y );   /* ACCELERATE CALCULATIONS
21                               BY ROUNDING PREDICTED
22                               PROBABILITIES.  &ROUND
23                                           MUST BE PRESENT */
24
25   %let control = 1;
26
27  % macro check;
28
29    %local dsid control i nullstr rc varnum;
30
31   %let model=%upcase(&model);
32   %let depvar=%upcase(&depvar);
33   %let string=&model &depvar;
34
35    %let i=1;
36     %let  nullstr=;
37
38    %let dsid=%sysfunc(open(&dsanal));
39
40   %do %until(%scan(&string,&i)=&nullstr);
41    %let varnum=%sysfunc(varnum(&dsid,%scan(&string,&i)));
42    %if &varnum=0 %then %do;
43    %let control=0;
44    %put ;
45    %put VARIABLE %scan(&string,&i) APPEARS IN THE
46           MODEL, BUT NOT IN THE INPUT DATA SET;
47     %put ;
48    %end;
49     %let i=%eval(&i+1);
50    %end;
51
52    %let rc=%sysfunc(close(&dsid));
53    %mend check;
54
55    %if (&replica=) or (&replica=1) %then %check;
56
57    %if &control = 0 %then %do;
58    %put **** MACRO TERMINATED  BECAUSE OF ERRORS
59                     ABOVE ******;
60    %goto exit;
61     %end;
62      %else %do;
63
64  proc logistic descending data=&dsanal;
65  weight &weight /norm;
66  class &model;
67  model &depvar= &model/
68  outroc=_roc(keep=_prob_);
69  output out=_probs predicted=_p_hat;
70  run;
71
72    %if %upcase(&acceler) = Y %then %do;
73
74  data _roc;
75  set _roc;
76  _prob_=round(_prob_, &round);
77  run;
78
79  proc sort nodupkey;
80  by descending _prob_;
81  run;
82
83  data _probs;
84  set _probs;
85  _p_hat=round(_p_hat,&round);
86  run;
```

```

    %end;

 data _out1 (keep= _pos_  _neg_ _falpos_ _falneg_);

  do  i=1 to numobroc;

  set roc1 nobs=numobroc point=i ;
    retain  _pos_  _neg_  _falpos_ _falneg_ ;
  pos=0; neg=0; falpos=0; falneg=0;

   do j=1 to numobpro;

   set probs nobs=numobpro point=j;
  if _p_hat  >=_prob_ then _preddep=1;  else _preddep=0;
  if &depvar=1 and _preddep=1 then _pos_=_pos_+ &weight;
     else
  if &depvar=0 and _preddep=1 then _falpos_=_falpos_+ &weight;
  else
  if &depvar=1 and _preddep=0 then  _falneg_=_falneg_+&weight;
   else
 _neg_=_neg_+&weight;

  if j = numobpro then output;

   end;
   end;
   stop;
   run;

    data _s;
     set _out1;
     if _n_=1 then set _out1(rename=(_pos_=_n1
                          _falpos_=_n2)) nobs=numout point=numout;

  _sensit_=_pos_/_n1;
  _1mspec_=_falpos_/_n2;
    run;

  data &outds&replica(keep=area&replica);
  set _s end=fin;
  retain  _w _z area&replica 0;
  if _n_=1 then do; _w=0; _z=0;
  end;
  _x=_1mspec_-_w;
  _y=(_sensit_+_z)*0.5;
  _z=_sensit_;
  _w=_1mspec_;

  area&replica=sum(area&replica,_x*_y);

   if fin then output;
  run;

 proc print data=&outds&replica;
 run;
   %end;
   %exit:;
  %mend calcauc;
```

3. Comparing the Predictive Value of Two Models

Many analyses involve fitting two or more logistic regression models to the same data. Then, in choosing among the models, it may be useful to compare their predictive value, via the AUC. Often one of the models is the final model or full model from a stepwise logistic regression, and the other models are subsets of the full model (e.g., the first k predictors to enter – a reduced model). The difference between the AUC for the full model and the AUC for a reduced model can aid in judging whether the full model offers a real advantage over the reduced model. (This application of the difference in AUC does not require that the models be nested. It is applicable to the comparison of any two models.) To assess the size of the difference in AUC relative to the variation in the data, we need the estimated standard error of the difference.

One suitable approach is the bootstrap method, which uses replication (Wolter 1985). As mentioned in Section 1, the bootstrap involves drawing repeated independent samples (with replacement) from the original sample and then estimating the AUC for each model and the difference in AUC, for each of these bootstrap samples. The sample standard deviation of that difference (over the bootstrap samples) is the bootstrap estimate of its standard error.

4. The Bootstrap Method for Variance Estimation

Our data came from a stratified one-stage cluster sample of over 20,000 persons that incorporates several weighting adjustments. The sample design entails stratification of the U.S. into 78 geographic areas. Within each stratum, a random sample of households is drawn. The survey collects data on all eligible household members, making households the clusters in the sample design. Rust and Rao (1996) discuss the use of replication methods to obtain standard errors for complex survey designs. The application of the bootstrap procedure to our sample design involves drawing the bootstrap samples (replicates) within each stratum.

In connection with other analyses of the same data, we had previously constructed 1,000 bootstrap replicates, in order to obtain bounds for 95% confidence intervals directly from the distribution of the bootstrap estimates, as well as bootstrap standard errors. Thus, it was natural to use those 1,000 bootstrap replicates in estimating the standard error of the difference in AUC. The next section describes the use of PROC SURVEYSELECT to construct bootstrap replicates. Then Section 6 discusses the further steps required to produce sampling weights specific to each replicate.

5. Use of PROC SURVEYSELECT

The following statements show how the SURVEYSELECT procedure was used to draw the 1,000 bootstrap replicates.

```
%let dd = ourdirectory;
%let ini_iter=1;
%let max_iter=1000;
%let in_smpfile=&dd..samplefile;
%let in_nsize= geo_area_tot;

/* CREATE A DATASET WITH SAMPLE SIZES TO BE DRAWN
   FROM EACH STRATUM                                              */
```

```
proc freq data=&in_smpfile;
 table geo_area/out=&in_nsize(rename=(count=_nsize_)
                                              drop=percent);
run;

/* MACRO TO DRAW 1000 BOOTSTRAP SAMPLES */

%MACRO BOOTREP;

%do i=&ini_iter %to &max_iter;

  proc printto new print="brep_&i..lst";
  proc printto new log="brep_&i..log";
  title3 " REPLICATE =&i URS selection";
  options pageno=1;

  proc surveyselect data=&in_smpfile
     method=urs
     sampsize=&in_nsize
     out=&dd..urs_&i  outhits;
     strata geo_area;
  run;

  proc freq data=&dd..urs_&i;
    tables NumberHits;
   run;
%end;
%MEND BOOTREP;
%BOOTREP
```

To draw the 1,000 sample replicates with equal probability and with replacement, we used METHOD=URS (Unrestricted Random Sampling). SAMPSIZE = GEO_AREA_TOT identifies the SAS data set that contains _NSIZE_ , the different sample sizes for the strata. The OUT=&dd..urs_&i option outputs each of the 1,000 samples into a separate permanent SAS dataset. The OUTHITS option outputs a separate observation for each selection when an observation is selected more than once. The output dataset contains for each observation the variable NumberHits, the number of times a household was selected into the sample in a given replicate. The STRATA statement defines the variable GEO_AREA as the stratification variable.

6. Calculation of Replicate Weights

Rust and Rao (1986) give a method for adjusting the final sampling weights to obtain bootstrap weights. They also note, however, that for the variance estimators to remain close to unbiased, the weight adjustment steps applied to the original sample should be applied to each bootstrap replicate. This is an important consideration in our sample design, given the considerable number of weight adjustments. Thus, for each bootstrap replicate we repeated all of the weight calculation steps. As a result each of the 1,000 bootstrap replicates has its own set of weights.

7. Using SAS and the Bootstrap Replicates to Estimate the Variance of the AUC

Applying the macro CALCAUC to the original sample, we calculate the AUC for the weighted models with 14 explanatory variables (full model) and 6 explanatory variables (reduced model). We denote them by AUC14 and AUC6, respectively. In Exhibit 2 the macro ALLREPL uses the macro CALCAUC as a subroutine to refit a weighted logistic regression model and obtain the AUC for each of 1,000 bootstrap replicates.

Exhibit 2: ALLREPL Macro

```
%let youranal = anal;           /*ANALYTIC FILE WITH ALL DATA */
%let dsbswts = replwts;        /*  DATA SET WITH  ID AND 1,000
                                                    REPLICATE WEIGHTS       */
%let model = yourmodel;    /* STRING WITH EXPLANAT ORY
                                                            VARIABLES       */
%let depvar = yourresponse;  /*  RESPONSE VARIABLE                  */

%macro allrepl (start,end);


%do v=&start %to &end ;        /* &START and &END ARE
                                FIRST   AND LAST REPLICATE TO
                        PROCESS , 1 AND 1,000 IN  OUR EXAMPLE  */
data _anal;
merge &youranal (in=_1 )
&dsbswts (keep=ID w&v where=(w&v ne 0) in=_2);    /* RETRIEVE
                                              &V-th REPLICATE WHERE
                                   V-th REPLICATE WEIGHT NE ZERO    */
by ID;
if _2;
wgt=w&v;
drop w&v;
run;

%calcauc( dsanal = _ANAL,            /*  CALCULATE AUC FOR &V-th
                                                         REPLICATE */

       outds  = C,
       id     = ID,
       weight = WGT,
       model =  &YOURMODEL,  /*  REFIT MODEL TO
                                     DATA IN   &V-th   REPLICATE*/
       depvar =  &YOURESPONSE,
       round  = .001,
       replica = &V,
       acceler = Y );

%end;
%mend;

%allrepl(1,1000)
```

Applying the ALLREPL macro for the full model and then for the reduced model, we obtain two sets of replicate AUCs: $AUCR14_i$ (i = 1 to 1,000), and $AUCR6_i$ (i = 1 to 1,000), respectively. We denote the bootstrap sample AUC by AUCR to distinguish it from the one of the original sample. To estimate the bootstrap standard errors of AUC14 and AUC6, we simply apply PROC UNIVARIATE to the $AUCR14_i$ and the $AUCR6_i$ (i = 1 to 1,000) to obtain the standard deviations. To estimate the standard error of the difference in AUC between the two models, DIFF = AUC14 - AUC6 , we apply PROC

UNIVARIATE to the differences $AUCR14_i$ - $AUCR6_i$ (i = 1 to 1,000) to obtain the standard deviation STDDIFF. Then

t = abs(DIFF / STDDIFF)

The corresponding significance level of a two-tailed t test is given by

p = (1-PROBT(t, df))*2,

The weighted AUC is .658 for the full model and .641 for the reduced model. The difference in weighted AUC of .017 is highly significant. Table 1 gives the results.

Table 1: AUCs and Bootstrap Standard Errors

Number of Predictors	Area or Difference in Area Under the Curve				
	Area or Difference	Std. Err.	Variance	t	p
6	0.64074	0.00712	0.00005074		
14	0.65758	0.00702	0.00004930		
14 vs. 6	0.01684	0.00381	0.00001450	4.423	.0000

8. Discussion

The area under the receiver operating characteristic curve is an important and widely used measure of the predictive ability of a logistic regression model. Most survey data files have survey weights attached. The LOGISTIC procedure does not take the weights into account in its calculation of the area under the ROC curve and therefore usually does not give the correct value. The SAS macro CALCAUC uses the survey weights in the calculation of the area under the ROC curve by summing the area of trapezoids.

Hanley and McNeil (1982) indicate that the c statistic (which PROC LOGISTIC reports in the Association of Predicted Probabilities and Observed Responses table) is equivalent to the area under the ROC curve. We have also developed a SAS macro, not discussed in this paper, that calculates a weighted version of the c statistic.

We have used the area under the curve to compare the predictive ability of two logistic regression models estimated from the same survey data file. Using bootstrap samples, it is possible to test whether the two models have the same area under the ROC curve. We provide some background on how SURVEYSELECT can be used to create bootstrap samples. It is possible, however, that the survey file being analyzed already contains bootstrap samples and bootstrap replicate weights. If not, it is wise to consult with a statistician who is familiar with the bootstrap method of variance estimation before creating bootstrap samples and bootstrap replicate weights.

References

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans.* Philadelphia: Society for Industrial and Applied Mathematics.

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.

Pepe, M.S. (2000). “Receiver Operating Characteristic Methodology,” *Journal of the American Statistical Association*, 95, 308-311.

Rust, K.F. and Rao, J.N.K. (1996). “Variance Estimation for Complex Surveys Using Replication Techniques,” *Statistical Methods in Medical Research*, 5, 283-310.

SAS Institute Inc. (1995). *Logistic Regression Examples Using the SAS System*, Version 6, First Edition. Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999). *SAS/STAT, Version 8*, Chapter 39. Cary, NC: SAS Institute Inc.

Wolter, K.M. (1985). *Introduction to Variance Estimation*. New York: Springer-Verlag.

Acknowledgment

We thank Philip Primak, Principal Programmer at Genzyme Corporation, for reviewing the macros and making valuable suggestions.

Contact Information

David Izrael
Abt Associates Inc.
55 Wheeler St.
Cambridge, MA 02138
617-349-2434
david_izrael@abtassoc.com