A SAS® MACRO INCORPORATING DISCRIMINANT ANALYSIS TECHNIQUES

Sivaram Kalyandrug, Capital Technology Information Services
Antonis D. Koutsoukos, National Cancer Institute
Lawrence V. Rubinstein, National Cancer Institute

Abstract

This paper describes a SAS macro that incorporates principal component analysis, a score procedure and discriminant analysis. Using the macro, parametric and non-parametric discriminant analysis procedures are compared for varying number of principal components and for both Mahalanobis and Euclidean distance measures. The emphasis of the paper, and the need for the macro, is to overcome what SAS procedure DISCRIM does not do while estimating the variance-covariance matrix used in cross-validation. In the non-parametric case, the SAS procedure includes the observation to be classified -- along with other observations which provide information for such a classification -- in the calculation of the variance-covariance matrix. This gives misleading classifications and is seen to affect the overall results markedly. Steps are shown in the macro how principal component scores are estimated for the test data set from the principal components of the reference (training) data set, and how successive DISCRIM procedure runs are used to assign posterior probabilities for classification. An example incorporating these procedures to predict the biochemical mechanism of action of cancer drugs in a screening process is discussed. The macro also shows a comparison of the performance of parametric and non-parametric discriminant analyses.

Principal Components

Principal component analysis is a widely known dimension reduction technique in multivariate statistics.. The principal components are orthogonal linear combinations of the original variables with eigenvectors of coefficients, usually defined to be of unit length, successively constructed to account, at each step, for maximum total variation in the data. When there are m variables, m principal components can be constructed which are uncorrelated to each other and together explain all the variation in the data. It is customary to arrange them in order of decreasing variances since a measure of the amount of information conveyed by each principal component is its variance. The PRINCOMP procedure in SAS performs principal component analysis with many options. We can limit the number of principal components required, and they may be based on either the correlation matrix or the covariance matrix. When raw data is used as input, PRINCOMP can output a data set of the components and another data set containing the eigenvectors and eigenvalues of the matrix which correspond to the coefficients of the linear combinations and their variances, respectively.

The Score Procedure

PROC SCORE is a simple procedure which cross-multiplies a set of variables for each observation with a set of coefficients for the variables from a different data set. The scores so obtained are output to a data set. For example, the eigenvectors coming from a principal component procedure can be multiplied with each observation of another data set to obtain its principal component scores. If necessary the values in the target data can be standardized with respect to the mean and standard deviation of the data set that produced the scores.

Discriminant Procedure

Given one or more quantitative variables and a classification variable for each observation, the discriminant procedure arrives at a decision rule based on the quantitative variables to classify observations into one of the groups defined by the classification variable. This discriminant criterion will be based on information provided by the quantitative variables and prior probabilities of the groups. The procedure allows us to apply the criterion so derived to classify observations from a different 'test' data set whose classification may not be known. If a multivariate normal distribution is assumed to hold within each class, SAS derives linear or quadratic discriminant functions depending on whether the within-group covariance matrices are homogenous or not. In the absence of any such assumption, SAS presents two non-parametric methods, the kernel and the k-nearest neighbor methods, based on non-parametric estimates of group-specific probability densities. Description of all the choices available for these methods is extensively discussed in the Version 6 SAS/STAT® User's Guide[1], Volume 1. For the

purpose of this paper, we will consider only the linear and the k-nearest-neighbor discriminant procedures. Variations within these methods depend on either Mahalanobis distance or Euclidean distance to determine proximity and the choice of either the original set of variables or their principal components. When the k-nearest-neighbor method is used, the Mahalanobis distances are based on the pooled covariance matrix. A simple mathematical treatment of the overall methodology is presented in Koutsoukos *et al*[2] together with an example which has a direct bearing on our discussion.

Having determined the requisite discriminant criterion from either the parametric or non-parametric method using n observations from a SAS data set, its validity is determined by the cross-validation technique offered in the procedure. This essentially consists of re-computing the criterion from (n-1) observations by leaving out an observation, classifying the observation left out using the new criterion, and checking if it is misclassified. This is done for each of the n observations and the group-specific error rate is calculated taking prior probabilities of the groups as weights.

The Issue

When a non-parametric method is used, the covariance matrices used by PROC DISCRIM to compute the distances during cross-validation are based on all observations in the data set and do not exclude the observation being classified. This could seriously bias the accuracy of the cross-validation technique, rendering it invalid, particularly when the observation vectors consist of a large number of variables.

The Approach

Because of the above problem, a different cross-validation scheme is employed whenever a non-parametric discriminant procedure is used as illustrated below.

The given SAS data set is divided into m groups (m >1), approximately equal in size, consisting of randomly chosen observations. Considering (m-1) of the m groups as a reference set and the remaining one group as a test set, PROC DISCRIM is invoked with reference set as the DATA= data, test set as the TESTDATA= data, and not including the cross-validation option. Thus PROC DISCRIM uses the reference set (training set) to determine the classification rule and applies this rule to the test set for the classification of its observations. This process is repeated m times for the m possible ways of partitioning the data set into reference and test sets. All the TESTOUT= output SAS data sets produced by the procedure at each partition are collected together. This collection then contains the posterior probabilities and class into which each observation is classified which enables the calculation of misclassified observations and the resultant error rate for the discriminant procedure.

Now, since the macro incorporates principal components in place of the original quantitative variables, PROC DISCRIM at each partition above will have principal components of the reference set as its VAR variables. Also the TESTDATA= data set will consist of principal component scores which are obtained by multiplying each test set observation with the eigenvectors corresponding to the reference set, a process achieved by PROC SCORE.

The Macro

The three procedures, PRINCOMP, SCORE, and DISCRIM are incorporated into one SAS macro to implement the methodology discussed above. The macro, called PCSDISC is presented in its full length in the last few pages of this paper. In this section a step by step view of the macro is discussed followed by its application to an example.

PCSDISC macro is applicable when principal components for a set of quantitative variables are to be used to classify observations in a regular SAS data set (and not a data set of correlations or covariances) using a k-nearest neighbor and a linear discriminant procedure.

The macro starts off with the macro definition and the description of its parameters. As discussed before, it is assumed that the input data set is made up of m randomly chosen groups designated by the 1 through m values of a numeric variable called 'group'. One should not confuse this variable with the classification variable. Since each observation of the input data set will be classified, it is also assumed that the observations are identified by at least one ID variable.

Non-parametric discriminant procedure: Following macro variable definition, a FORMAT procedure and a DATASETS procedure are executed. The FORMAT procedure defines few variable descriptors and the DATASETS procedure initializes a work data set to which observations will be appended later. Next, a loop is started for repeatedly calculating principal components for the reference set, generating principal component scores for the test set, running the nearest neighbor discriminant procedure, and for accumulating output data sets. The procedure and data steps in this loop are designed to do the non-parametric discriminant procedure first. Note that the parameter 'nparts' in the macro controls this loop. Its value must equal m, the total number of random groups into which input data is divided. However, a smaller value for the parameter may be assigned for testing purposes. To begin with, the first random group of the input data is

considered as the test set and all the remaining groups together as the reference set. Such an assignment is easily done by the 'where' data set options which select the necessary input data segments. In the first PRINCOMP procedure, the principal components for the reference set and their eigenvectors are captured in the OUT= and OUTSTAT= data sets respectively. Only the specified number of principal components are computed which will be based on either the covariance matrix or the correlation matrix as opted. Now, the idea is to obtain 'principal components' for the test set which are compatible. This can be achieved by applying the eigenvectors derived from the reference set to the raw data of the test set. If the principal components in the current step are based on a correlation matrix, then the test data needs to be standardized. This is accomplished in the subsequent STANDARD procedure. The STANDARD procedure transforms the test set by subtracting each variable from its mean and dividing by its standard deviation. However, when the principal components are based on the covariance matrix, no such transformation is done to the test set. The SCORE procedure that follows cross-multiplies the eigenvectors corresponding to the reference set with the test data vectors to obtain the principal component scores for the latter. It must be noted that by virtue of the 'nostd' option in the score procedure, the test data set does not unnecessarily get centered or scaled by the mean and variance of the reference set. Next, principal component data sets for the reference and test sets are passed on to the k-nearest neighbor discriminant procedure that follows. For the chosen distance metric, the chosen number of nearest neighbors, and based on the selected number of principal components, the DISCRIM procedure classifies observations of the test set. Posterior probabilities for the test set, and the class into which they get classified, are passed on from this procedure, to be accumulated into a data set by the subsequent APPEND procedure, reaching the end of the loop. Thus when this set of procedures goes through all the different reference-test set definitions, the observations in the input data get classified by the discriminant criterion ensuring that each time the generalized distance uses the current variance covariance matrix. The data step after the loop determines if the observations have been correctly classified or not by a simple comparison, the PRINT procedure lists the misclassified observations, and the FREQ procedure obtains overall frequency distributions for the classification.

Linear discriminant procedure: Continuing with the macro, the next PRINCOMP and DISCRIM procedures are for the application of Fisher's linear discriminant function. While the distance measure and the type of matrix for principal components are the same as for the non-parametric case, the macro allows for a different number of principal components than used in the parametric case. Based on the complete input data set, the PRINCOMP procedure computes the required number of principal components and passes them on to the discriminant procedure. Using the pooled covariance matrix in calculating either the Euclidean or the Mahalanobis distance, as specified earlier, the discriminant procedure computes linear discriminant functions to classify the observations based on the principal components. Cross-validation is performed correctly, as defined by the procedure, and a data set is created containing all the input data, plus the posterior probabilities and the class into which each observation is classified by cross-validation. The data step that follows DISCRIM determines if the classification is correct or not. The print procedure lists the misclassified observations while the FREQ procedure produces the overall classification of the class variable.

Comparison of the two methods: The data sets obtained from the non-parametric and linear discriminant procedures containing each observation and its classification status may be used to compare the two procedures. The remaining part of the macro performs this comparison, if requested.

The final data sets obtained from the two procedures above are sorted by the ID variable(s) and a new data set is created by merging them. Three new variables are defined in the merged data set, two based on whether the parametric and non-parametric classifications differ from the true classification, and one based on whether the two methods differ between themselves. If the non-parametric classification is the same as the true classification, the first variable takes a value 0, otherwise 1. Similarly, if the parametric classification is the same as the true classification, the second variable is defined as 0, otherwise 1. The third variable is simply a difference between these two, subtracting the second variable from the first. Next, these variables are displayed and cross-classified by print and frequency procedures, and a UNIVARIATE procedure is used to obtain summary statistics on the third variable.

In preparation for Mantel-Haenszel association test for the agreement between the two methods, a new data set is created next. Here, for every original observation, two new observations -- one for the non-parametric method and another for the parametric -- are created along with the method's response. The response shows whether the method correctly classifies the observation or not. Thus the new data set has twice as many observations as the merged data. The final step in the comparison provides a FREQ procedure incorporating a cross tabulation between

the method and its response.. The printed output consists of the stratified analysis of the 2x2 tables producing Cochran-Mantel-Haenszel chisquare statistic with one degree of freedom to test the alternative hypothesis that the correlation between the two methods is zero.

An Example

In its attempt to identify potential anti-cancer drugs, The National Cancer Institute tests approximately 400 chemical compounds per week using an *in vitro* test screen. The screen consists of a panel of sixty tumor cell lines representing eight disease sub-panels including brain, colon, leukemia, non-small cell lung, small cell lung, melanoma, ovary, and renal. The bio-chemical mechanism of action of a compound, the reason why it behaves as it does, is usually unknown. In this example, we consider classification of the drugs into their mechanism of action by discriminant analyses on the basis of their response to the sixty-cell-line screen. The set of quantitative variables for a given drug consists of sixty values each representing drug's performance towards a cell line, computed as the dose value at which 50% cell growth inhibition (GI50) is realized. These values may be standardized by subtracting the average GI50 for the drug over all the cell lines, as described in Boyd *et al* [3].

The macro is invoked on a data set consisting of 141 compounds, each with a known mechanism of action and sixty standerdized GI50 values for the cell lines. The data set is divided into 10 randomly chosen groups of 14 or 15 compounds each, given by a 'group' variable. Using the macro, principal components ranging from 10 to 50 in number, using the covariance matrix option, are generated, and non-parametric 3-nearest neighbor discriminant procedures based on both Mahalanobis and Euclidean distances are compared with the linear discriminant procedure. Table I (taken from Koutsoukos et al) presents a summary of results from these attempts showing the number of incorrect classifications of mechanism of action.

Table I. Number of incorrect mechanism of action classifications by procedure type and principal components.

Procedure	Distance	Number of principal components 10	20	30	40	50
Linear	Mahalanobis	33	23	22	16	13
3NNeighbors	Euclidean	34	26	22	20	19
	Mahalanobis	38	22	22	18	19

For each execution of the macro, the output -- too large to present here -- gives the misclassified observations for the two methods, their classification distribution, and method comparisons if requested.

Other considerations

Variations Apart from changing the options and parameter values in the macro, the macro itself may easily be customized to suit one's purpose by incorporating simple changes to it. Statements to specify the prior probabilities of group membership may be included, kernel density estimation method may be substituted for the k-nearest neighbor method etc.

Processing large data sets: Though no bench marks are made, it may be seen that the demand on computer resources to process very large data sets will be heavy. Caution must be taken in increasing the number of principal components, the number of random groups, and in processing data sets which are very large.

REFERENCES

1. SAS/STAT® User's Guide, Version 6, Fourth Edition, SAS Institute, Inc., Cary, NC., 1990

2. Koutsoukos, A. D., Rubinstein, L. V., Faraggi, D., Simon, R. M., Kalyandrug, S., Weinstein, J. N., Kohn, K. W., and Paull, K. D. 'Discrimination techniques applied to the NCI *in vitro* anti-tumor drug screen: Predicting biochemical mechanism of action'. *Statistics in Medicine*, 13, 719-730 (1994).

3. Boyd, M. R., Paull, K. D., and Rubinstein, L. R., 'Data display and analysis strategies for the NCI disease-oriented *in vitro* antitumor drug scree'. Proceedings of the 22nd Annual Detroit Oncology Symposium on Anticancer Drug Discovery and Development, Detroit. April 25-27, 1990, Kluwer Academia Publications, Amsterdam, 1992.

For further information contact:
Sivaram Kalyandrug
Capital Technology Information Services, Inc.
1355 Piccard Drive, Suite 450
Rockville, MD 20850

The Macro Code

```
/* This program performs linear and non-parametric (k nearest-neighbor) discriminant procedures following principal
component and score procedures. It assumes that the input data set is made up of m randomly chosen groups given by the
1 thru m values of a numeric variable 'group'. */

%macro PCSDISC(indat=_last_,npc=5,k=3,dpc=5,nparts=1,opt=cor,dist=M,prnt=N,compr=Y,
        varlist=%str(),class=,id=%str());

%* The parameters for the macro are described below. Values for class and id are required. ;

%*
    indat       Input SAS data set

    npc         # principal components used in non-parametric discriminant procedure

    k           # nearest neighbors used in non-parametric discriminant procedure

    dpc         # principal components used in linear discriminant procedure

    nparts      Number of Reference-Test partitions to use. Usually this
                number is m, the number of groups input data is divided into.

    opt         cor   principal components will be based on correlation matrix
                cov   principal components will be based on covariance matrix

    dist        M     Mahalanobis distance is used in proc discrim
                E     Euclidean  distance is used in proc discrim

    prnt        Y     prints output from proc princomp
                N     suppresses printing from proc princomp

    compr       Y     compares parametric and non-parametric discriminant procedures
                N     supresses the comparison

    varlist     list of input variables for proc princomp. e.g., Var1-Var10 or  A--Z.

    class       Classification variable

    id          ID variable(s)
   ;

  proc format;
   value star 0='*';
   value rate 0='Wrong'
              1='Right';
  run;

  proc datasets library=work;
   delete np&npc;
  run;
```

```
%do indx=1 %to &nparts;                    %* ----- Start indx Loop ------ ;

    %if %upcase(&opt)=COV %then %do;
    title2 "&npc Principal Components Using Covariance Matrix";
    %end; %else %do;
    title2 "&npc Principal Components Using Correlation Matrix";
    %end;

   proc princomp
      data=&indat(where=(group ne &indx) keep=group &id &class &varlist)
      n=&npc
      %if %upcase(&opt)=COV %then %do; cov  %end;
      out=refpc
      outstat=refegn
      %if %upcase(&prnt)=N %then %do; noprint %end;;
   var  &varlist;
   run;

   data refpc;
    set refpc(drop=&varlist);
   run;

   /* Whenever Correlation option is used, the Test data set is
     Standardized to have Zero Mean and Unit Standard Deviation. */

   %if %upcase(&opt)=COR %then %do;
    proc standard data=&indat(where=(group=&indx) keep=group &id &class &varlist)
     out=tstand mean=0 std=1;
     var &varlist;
    run;
   %end;

   /* The following Score procedure calculates PC scores for
     Test Set by multiplying Test data vectors with eigenvectors
     derived from Reference Set. */

   proc score
      %if %upcase(&opt)=COR %then %do; data=tstand  %end;
      %else %do;
        data=&indat(where=(group=&indx) keep=group &id &class &varlist)
        %end;
      nostd
      out=testpc
      score=refegn;
    var &varlist;
     id &id &class;
   run;

   %if %upcase(&dist)=E %then %do;
   title3 'Euclidean distance'; %end;
   %else %do;
   title3 'Mahalanobis distance'; %end;
```

```
      proc  discrim noprint
          data=refpc
          testdata=testpc
          method=npar  k=&k  short  noclassify
          %if %upcase(&dist)=E %then metric=identity;
          testout=dout ;
      class &class;
        var prin1-prin&npc;
         id &id ;
      title4 "Prediction of &class for Group: &indx - NP &k Nearest Neighbors";
      run;

      proc append data=dout out=np&npc;
      run;
%end;                                          %* ----- End  indx Loop -----;

data np&npc;
 set np&npc(drop=prin1-prin&npc);
 if &class =_into_ then classify=1; else classify=0;
run;

proc print data=np&npc(where=(classify=0));
 id &id &class;
 format classify star.;
 title4 "Misclassified Observations - &k Nearest Neighbor Nonparametric Discriminant Proc.";
run;

proc freq data=np&npc;
 tables &class *(_into_ classify);
 format classify rate.;
 title4 "Overall Classification - &k Nearest Neighbor Nonparametric Discriminant Proc.";
run;

 proc princomp
  data=&indat(keep=group &id &class &varlist)
  n=&dpc
  %if %upcase(&opt)=COV %then %do; cov  %end;
  out=linout
  %if %upcase(&prnt)=N %then %do; noprint %end;;
 var  &varlist;
 run;

 proc  discrim noprint
      data=linout
      method=normal short  noclassify
      outcross=doutl;
 class &class;
  var  prin1--prin&dpc;
    id &id ;
 run;
```

```
data lin&dpc;
 set doutl(drop=prin1-prin&dpc &varlist);
 if &class =_into_ then classify=1; else classify=0;
run;

proc print data=lin&dpc(where=(classify=0));
 id &id &class;
 format classify star.;
 title4 'Misclassified Observations -- Linear Discriminant Procedure';
run;

proc freq data=lin&dpc;
 tables &class *(_into_ classify);
 format classify rate.;
 title4 'Overall Classification -- Linear Discriminant Procedure';
run;

 /* The following code compares performance of the two methods */

%if %upcase(%compr) ne N %then %do;    %* ----- Start compr condition ----- ;
    proc sort data=np&npc out=one&npc;
     by &id;
    run;

    proc sort data=lin&dpc out=two&npc;
     by &id;
    run;

    data both&npc;
     merge one&npc(in=in1 rename=(_into_=into1) drop= classify)
           two&npc(in=in2 rename=(_into_=into2) drop= classify);
     by &id;
     if in1 and in2;

     if &class=into1 then diff1=0; else diff1=1;
     if &class=into2 then diff2=0; else diff2=1;
     diff3=diff1-diff2;
     keep &id &class into1 into2 diff1 diff2 diff3;
    run;

    proc print data=both&npc(obs=25);
     title3 'Classification by the two mothods';
    run;

    proc freq data=both&npc;
     tables  diff1*diff2;
    run;

    proc univariate data=both&npc;
     var diff3;
     title3 'Univariate Statistics';
    run;
```

```
      data mc&npc;
       set both&npc;
       cmhnum+1;
       method='first';
       response=diff1;
       output;
       method='second';
       response=diff2;
       output;
      run;

      proc freq data=mc&npc;
       tables cmhnum*method*response/noprint cmh1;
       title3 'Test of Agreement';
      run;
  %end;                          %* ----- End   compr condition ----- ;
%mend PCSDISC;
```