MAUCROC: Stata Syntax for Plotting ROC and AUC for Univariate and Multivariate Discrimination Models Using All Possible Pairwise Class Comparisons and Leave-One-Out Cross Validation

Leif E. Peterson

February 11, 2014

The Stata code below (MAUCROC) will plot an ROC curve for each variable (univariate) for each pairwise class comparison, plus an ROC curve for a multivariate discrimination model based on all variables. The code provided below was employed for a run with 5 variables and three classes of economic conditions. Hence, it was a 3-class problem.

For a 3-class problem (Ω=3), there will be $3=\Omega(\Omega-1)/2=3(3-1)/2$ plots for the three pairwise class comparisons (1 vs. 2, 1 vs. 3, and 2 vs. 3) showing ROC curves for each variable (univariate classification) plus an ROC curve for all variables combined (called "MULTI"). The legend of each plot will include the AUC value for each ROC curve, and the AUC matrix will also be printed to the screen.

This solution was also based on the k-nearest neighbor classifier ("knn"), and as you can note when using **`help discrim`**, this can be swapped with "lda" (linear discriminant), "logistic" for logistic discrimination, and "qda" for quadratic discriminant. (Although qda is a poor classifier – so I don't recommend it).

For a strategy, recall that one of the goals during classification analysis is to determine if the problem is linearly separable. If it is, then use a simple linear model such as lda, and not a complex or more expensive method such as support vector machines, boosted decision trees, neural network, swarm intillegence, etc. However, lda is based on covariance matrices, which assume some degree of normality, so if the data are complex the lda won't perform well. Lda and logistic also may not perform well if there are scattered objects in one class near tightly clustered objects from another class, and this is where knn can come to the rescue. Knn tends to work well when other classifiers break down. Also note that k=7 was used, which is an odd number. Odd numbers are recommended for k because if k is even and there is a tie with half the nearest neighbors in one class and half in another, then it can't decide which class dominates among nearest neighbors. However, when k is odd, this won't commonly happen – but can occasionally because you can always have ties for multiclass problems.

Requirements:

1. The class variable must be named "class"
2. Substitute your variable names where the code below contains yellow highlights (this example was for 5 input variables). For example, change **`var1name`** to **`mpg`**, or **`var2name`** to **`proteinA`**.
3. For the upper case labels such as **`VAR1NAME`**, you should change these to e.g. **`MPG`** or **`PROTEINA`**.

```
//Design & Code: L.E. Peterson, HMRI, Feb 11, 2014
//MAUCROC: Plotting ROC and AUC for Univariate and Multivariate Discrimination
//         Models Using All Possible Pairwise Class Comparisons

set more off
matrix auctable=J(3,6,0)
set more off
local classlbls class1 class2 class3 //change these class names to your class names
local numclasses=3
local lbound=`numclasses'-1
local cnt=0
forv j=1(1)`lbound'{
```

```
    local lbound2=`j'+1
    forv k=`lbound2'(1)`numclasses'{
      local cnt=`cnt'+1
      cap drop binaryclass
      gen binaryclass=0 if class==`j'
      replace binaryclass=1 if class==`k'
      discrim knn var1name, k(7) group(binaryclass)
      cap drop prob1
      predict prob1, loopr group(#2)
      roctab binaryclass prob1,detail
      local auc1=string(r(area),"%9.2f")
      local myauctitle1 `auc1'
      local m1 VAR1NAME(`myauctitle1')  //leave the var 1 name in upper case

      discrim knn var2name, k(7) group(binaryclass)
      cap drop prob2
      predict prob2, loopr group(#2)
      roctab binaryclass prob2,detail
      local auc2=string(r(area),"%9.2f")
      local myauctitle2 `auc2'
      local m2 VAR2NAME(`myauctitle2')

      discrim knn var3name, k(7) group(binaryclass)
      cap drop prob3
      predict prob3, loopr group(#2)
      roctab binaryclass prob3,detail
      local auc3=string(r(area),"%9.2f")
      local myauctitle3 `auc3'
      local m3 VAR3NAME(`myauctitle3')

      discrim knn var4name, k(7) group(binaryclass)
      cap drop prob4
      predict prob4, loopr group(#2)
      roctab binaryclass prob4,detail
      local auc4=string(r(area),"%9.2f")
      local myauctitle4 `auc4'
      local m4 VAR4NAME(`myauctitle4')

      discrim knn var5name, k(7) group(binaryclass)
      cap drop prob5
      predict prob5, loopr group(#2)
      roctab binaryclass prob5,detail
      local auc5=string(r(area),"%9.2f")
      local myauctitle5 `auc5'
      local m5 VAR5NAME(`myauctitle5')

      discrim knn var1name var2name var3name var4name var5name, k(7) group(binaryclass)
      cap drop prob6
      predict prob6, loopr group(#2)
      roctab binaryclass prob6,detail
      local auc6=string(r(area),"%9.2f")
      local myauctitle6 `auc6'
      local m6 MULTI(`myauctitle6')  //don't need to change "MULTI"

      local lblclass1:word `j' of `classlbls'
      local lblclass2:word `k' of `classlbls'
      local mytitle `lblclass1' vs. `lblclass2'
      roccomp binaryclass prob1 prob2 prob3 prob4 prob5 prob6,graph norefline legend(keygap(.5)
rowgap(.3) size(medsmall) symxsize(5) region(lwidth(none)) rows(3) pos(5) ring(0) label(1 `m1')
label(2 `m2') label(3 `m3') label(4 `m4')  label(5 `m5') label(6 `m6')) plot1opts(lpattern(l)
lwidth(medthick) msymbol(i) mcolor(yellow) lcolor(yellow))  plot2opts(lpattern(l) lwidth(medthick)
msymbol(i) mcolor(green) lcolor(green))  plot3opts(lpattern(l) lwidth(medthick)  msymbol(i)
mcolor(blue) lcolor(blue))  plot4opts(lpattern(l) lwidth(medthick)  msymbol(i) mcolor(magenta)
lcolor(magenta))  plot5opts(lpattern(l) lwidth(medthick)  msymbol(i) mcolor(black) lcolor(black))
plot6opts(lpattern(l) lwidth(medthick)  msymbol(i) mcolor(red) lcolor(red))
title({bf:`mytitle'},color(black)) xtitle("FPR") ytitle("TPR") ysc(r(0 1)) xsc(r(0 1))
xlab(0.0(.5)1.0,format(%5.1f)) ylabel(0(0.5)1.0,format(%5.1f)) saving(roc`j'v`k',replace)
```

```
      mat auctable[`cnt',1]=`auc1'
      mat auctable[`cnt',2]=`auc2'
      mat auctable[`cnt',3]=`auc3'
      mat auctable[`cnt',4]=`auc4'
      mat auctable[`cnt',5]=`auc5'
      mat auctable[`cnt',6]=`auc6'
   }
}

  local gphstring=""
  local lbound=`numclasses'-1
  forv j=1(1)`lbound'{
    local lbound2=`j'+1
    forv k=`lbound2'(1)`numclasses'{
      local gphstring `gphstring' roc`j'v`k'.gph
    }
  }
  graph combine `gphstring' ,xsize(15) ysize(10)

  matrix list auctable

end
```