Multiclasstesting Vignette

Christine Nardini, Yuanhua Liu

1 The problem

This is an introduction to the `Multiclasstesting` package in R. Specificity, sensitivity, negative and positive predictive value are used in combination to quantify different aspects of the accuracy of a binary test, evaluating different proportions of correctly and incorrectly classified items, when compared to a known classification, considered the gold standard. In this context the *test* is the ensemble of all the operations performed to classify each items; *positive* and *negatives* label the items according to the two classes ($c = 0, 1$) they belong to; *true* (T) and *false* (F) represent the ability of the test to classify coherently or not a given item in the test classification with respect to the gold standard classification. These concepts are usually formalized with the relationships in the left hand-side of Equations 1.

$$\begin{array}{rcccl} PPV & = & TP/TP+FP) & = & TP/P_t \\ NPV & = & TN/(TN+FN) & = & TN/N_t \\ Se & = & TP/(TP+FN) & = & TP/P_{gs} \\ Sp & = & TN/(TN+FP) & = & TN/N_{gs} \end{array} \quad (1)$$

When the test classifies $n > 2$ categories, these definitions become more complex to apply. In fact, the meaning of *positive* and *negative* is not relevant anymore, since there are now *positives*. Then, while the definition of *true* remains straightforward, as it indicates coherence between the classification of the test and the gold standard, the definition of *false* can be cumbersome, since there are $n - 1$ ways to misclassify an item. To avoid confusion and ambiguities the actual values of all false can be identified by rewriting the problem in terms of a system of equation based on the relationships indicated in Table 1. Here P_t, N_t represent the total number of positive and negative items that can be found in the test (t) categorization, and P_{gs}, Ngs in the gold standard (gs) classification. The definitions can be generalized to $n > 2$ classes changing the term negative and positive with the indices of the corresponding classes $c = 0, 1, ..., n$, and having C_c to design the total number of positives for each given class. The system of equations obtained from the relationships in the rows and columns of Table 1 contains $2 \cdot n$ equations (i.e. $TP + FP = P_t$) and $2 \cdot n$ unknown (x_{ij}), thus it is completely specified. It is worth noticing, that with these general definitions, in case of 2-classes test, Se and Sp appear to be dual scores. Thus, when generalizing to n-classes it is possible to define the predictive ability of the test for each given class $c \in 0, 1, .., n$ as $PV_c = T_c/C_t$ and the Sensitivity/Specificity (now called S) for the same class c as $S_c = T_c/C_{gs}$. To clarify the situation it is extremely useful to rewrite the definitions as they are written on the right hand-side of Equation 1, namely:

For n classes this gives:

$$\begin{array}{rcl} PPV & = & \sum_c T_c/\sum_c C_{c,t}, c = 1, .., n \\ NPV & = & T_0/N_t = T_0/C_{0,t} \\ Se & = & \sum_c T_c/\sum_c C_{c,gs}, c = 1, .., n \\ Sp & = & T_0/N_{gs} = T_0/C_{0,gs} \end{array} \quad (2)$$

This package is developed to estimate the performance of both binary and multiple test (n-ary is used to include both cases). The statistical scores described above are finally calculated as output of the function in this package.

2 MultiClassTesting usage

`Nclasstest` is the only function in `Multiclasstesting`. It serves for the computation of the performance of n-class test. In binary case, the output includes the statistical scores, PPV, NPV, Se and Sp. In multiple

Table 1. Classical definition and generalization to 3 classes for *true, false, negatives, positives.*

(a) Classical Definition

Test		Gold Standard T	Gold Standard F	
	P	TP	FP	$\rightarrow P_t$
	N	FN	TN	$\rightarrow N_t$
		$\downarrow P_{gs}$	$\downarrow N_{gs}$	

(b) 3-Classes Definition

Test	Gold Standard 2	Gold Standard 1	Gold Standard 0	
2	T_2	x_{12}	x_{13}	$\rightarrow C_{2,t}$
1	x_{21}	T_1	x_{23}	$\rightarrow C_{1,t}$
0	x_{31}	x_{32}	T_0	$\rightarrow C_{0,t}$
	$\downarrow C_{2,gs}$	$\downarrow C_{1,gs}$	$\downarrow C_{0,gs}$	

classes case, the output consists of two parts. One is called `multi.performance`, indicating the details of the predictive value (PV) and Sensitivity/Specificity (S) for each class. The other, called `binary.performance`, is to summarize the PPV, NPV, Se and Sp for the classification operations, as described in Equation 2. This is useful when interested in the ability of the n-ary test to identify positives and negative, globally. A common example of application is the computation of the performances of a gene network algorithm reconstruction: when interested in the directed network each edge can belong to class 1 (direct interaction), class -1 (inverse interaction) or class 0 (no interaction). However, it may be interesting to know how the algorithm performs simply in terms of recognizing the existing connections (edges 1 AND -1) in this case the summary binary performances of the 3 class test are the correct way to compute PPV, NPV, Se, Sp.
For the binary case, R statement is ,

```
> library(Multiclasstesting)
> GS <- cbind(c(0, 1), c(0, 0), c(1, 1))
> T <- cbind(c(1, 1), c(1, 0), c(1, 1))
> Nclasstest(T, GS)

$binary.performance
     PPV NPV Se        Sp
[1,] 0.6   1  1 0.3333333
```

where, GS and T are the arguments to the function `Nclasstest`, representing the results from Gold Standard and test, respectively. They can be matrices or vectors with the elements labeling the category type, as `0` and `1` in the example above.

For multiple-class test,

```
> library(Multiclasstesting)
> GS <- cbind(c(0, -1, 1), c(0, 1, 0), c(1, 0, 1))
> T <- cbind(c(1, -1, 1), c(0, 1, -1), c(0, 1, 1))
> Nclasstest(T, GS)

$multi.performance
  class.type   PV   S
1         -1 1.00 0.5
2          0 0.25 0.5
3          1 0.75 0.6

$binary.performance
           PPV NPV  Se   Sp
[1,] 0.5714286 0.5 0.8 0.25
```