An R Function for the Multi-classification Fisher Discriminant Method

ZHANG Ying-ying[1, a *] and ZHANG Xiu-ting[2,b]

[1] Department of Statistics and Actuarial Science, Chongqing University, Chongqing, China

[2] Department of Statistics and Actuarial Science, Chongqing University, Chongqing, China

[a] robertzhangyying@qq.com, [b] 1950579894@qq.com

Keywords: Multi-classification; Fisher discriminant analysis; R function; discriminant analysis.

Abstract. We first introduce an R function distinguish.fisher() of the multi-classification Fisher discriminant analysis and two other useful R functions: discriminant_table() (used to calculate the discriminant table) and plot_fisher() (used for the scatterplot of the first two discriminant scores). Then we do two case studies, the first case study shows that the R function distinguish.fisher() written by us is correct; the second case study shows that the function distinguish.fisher() can solve two-classification problems.

Introduction

Discriminant analysis [1,2,3,4,5,6,7,8,9] is used to determine the belonging of the group of the new observation under the condition that the study objects are divided into a number of known groups. Commonly used discriminant analysis methods are distance discriminant method, Bayes discriminant method, and Fisher discriminant method [10,11,12]. From the point of view of the number of categories, it is also divided into two classification problems and multi-classification (classification number is greater than or equal to 3) problems. In practical applications, softwares (such as R and SAS) are used to calculate the two classification and multi-classification problems. Reference [6] gives the two classification and multi-classification distance discriminant method R functions (discriminant.distance () and distinguish.distance ()), and the two classification and multi-classification Bayes discriminant method R functions (discriminant.bayes () and distinguish.bayes ()), but it only gives two classification Fisher discriminant method R function (discriminant.fisher ()), and it does not give the multi-classification Fisher discriminant method R function. We have written an R function distinguish.fisher() for the multi-classification Fisher discriminant method, by referencing the theoretical part of the multi-classification Fisher discriminant method in [5] and the programming structure of discriminant.fisher(). So people using R software for discriminant analysis can compare three kinds of multi-classification discrimination methods.

The rest of the paper is organized as follows. Section 2 introduces the R function distinguish.fisher() of the multi-classification Fisher discriminant method and two other useful R functions: discriminant_table() (used to calculate the discriminant table) and plot_fisher() (for scatter plot of the first two discriminant scores). Section 3 is case analysis, case 1 is used to verify the correctness of distinguish.fisher(), case 2 is used to explain that the distinguish.fisher() function can solve two classification problem. Section 4 summaries.

The R Function for the Multi-classification Fisher Discriminant Method

See [5] for the theoretical part of the multi-classification Fisher discriminant method.

Note that the R functions of the paper can be downloaded from ZHANG Ying-ying's academic homepage: http://user.qzone.qq.com/93347989/blog/1308306747. The R function distinguish.fisher() for the multi-classification Fisher discriminant method is given as follows.

```
distinguish.fisher = function(TrnX, TrnG, TstX = NULL, r = 0, delta = 1e-7, CVCCR = 0.75){
[function body omitted!]
```



```
res = list(p = p, n = n, k = k, N = N, mu = mu, xb = xb, B = B, E = E, EinvB = EinvB,
  orderedEigValues = orderedEigValues, orderedEigVectors = orderedEigVectors,
  s = s, r = r, T = T, T_tilda = T_tilda, Sp = Sp, yb_tilda = yb_tilda, sumT_tilda = sumT_tilda,
  D_tilda = D_tilda, belong_tilda = belong_tilda)
}
```

The input variables of the function are explained below. TrnX is the training sample, and the input format is a data frame or matrix (the sample is inputted rowly). TrnG is a factor variable, indicating the classification of the input training samples. TstX is the test sample, the input format is a data frame or matrix (the sample is inputted rowly) or a vector (one test sample). If you do not enter a TstX (the default), then the test sample is the training sample. r is the number of discriminants (the default is 0). delta is a critical value (the default is 1e-7). It is used to determine whether a number x is 0, if abs(x) < delta, then x is considered to be 0. CVCCR is short for Critial Value of Cumulative Contribution Rate, and the default value is 0.75.

The output variables of the function are explained below. p is the number of variables. n is the number of samples. k is the number of groups. N is a vector of the number of samples of each group. mu is the matrix of sample mean vector, and the ith row represents the sample mean vector of the ith group. xb is the sample mean vector of all samples. B is the matrix $\boldsymbol{B}$ of sum of squares between groups. E is the matrix $\boldsymbol{E}$ of sum of squares within groups. EinvB is the matrix $\boldsymbol{E}^{-1}\boldsymbol{B}$. OrderedEigValues is the decending vector of the eigenvalues of the matrix $\boldsymbol{E}^{-1}\boldsymbol{B}$. OrderedEigVectors is the matrix of the corresponding eigenvectors of orderedEigValues of the matrix $\boldsymbol{E}^{-1}\boldsymbol{B}$. s is the number of nonzero eigenvalues of the matrix $\boldsymbol{E}^{-1}\boldsymbol{B}$. r is the number of discriminants. T is a matrix of the first r (decending according to the eigenvalues) eigenvectors of the matrix $\boldsymbol{E}^{-1}\boldsymbol{B}$. Each column of T_tilda ($\tilde{\boldsymbol{t}}_i$) is the standardized eigenvector (the standardization is $\tilde{\boldsymbol{t}}_i^T \boldsymbol{S}_p \tilde{\boldsymbol{t}}_i = 1, i = 1,2,\ldots,r$) of the column vector of T ($\boldsymbol{t}_i$). Sp is the pooled sample covariance matrix $\boldsymbol{S}_p$. yb_tilda is the matrix of the group means of the discriminants, and its (i, j) element is $\bar{\tilde{y}}_{ij} = \tilde{\boldsymbol{t}}_j^T \left(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}} \right)$. SumT_tilda is the matrix $\tilde{\boldsymbol{\Sigma}}_T = \sum_{q=1}^{r} \left(\tilde{\boldsymbol{t}}_q \tilde{\boldsymbol{t}}_q^T \right)$. D_tilda is the squared Euclidean distance matrix $\tilde{\boldsymbol{D}}$ composed of $\tilde{\boldsymbol{D}}_{(i)}, i = 1,2,\ldots,k$. belong_tilda is a one dimension matrix of numbers, and the number represents the corresponding class.

In addition, we write two useful R functions: discriminant_table() which is used to calculate the discrimination table and plot_fisher() which is used to draw the scatter plot of the first two discriminant scores.

Case Studies

Case 1. The Iris data. See Table 5.4.1 of [5].

Input the data, then call the function distinguish.fisher() to distinguish. The outputs of the program agree with the results of Example 5.4.2 of [5]. Therefore, the R function distinguish.fisher() written by us is correct. See Table 1 for the discrimination table.

Table 1. The discrimination table of the iris data.

To / From	I	II	III
I	50	0	0
II	0	48	2
III	0	1	49

Case 2. Two classification of sand based liquefaction. The data is from Table 8.1 of [6].

This case is used to illustrate that the distinguish.fisher() function can be used to solve two classification problems. Input the data, then call the distinguish.fisher() function to distinguish. And call the existing discriminant.fisher() function from [6] to distinguish. See Table 2 for the discrimination table. From Table 2 we find that the discriminant results of the two functions are different, and the discriminant result of discriminant.fisher() function is slightly better than that of the distinguish.fisher() function.

By a careful study of the corresponding theoretical part of discriminant.fisher() (see [6] Section 8.1.3) and the corresponding theoretical part of distinguish.fisher() (see [5] Section 5.4), we summarize the comparison results of the two classification Fisher discriminant methods in Table 3. For the sake of unity, we are in the mark of [5]. In Table 3, the discriminant $w(\boldsymbol{x})$ depends on $\boldsymbol{d}$, $\boldsymbol{S}$ and $\boldsymbol{c}$. We find that the two discriminant methods are consistent concerning the rule of discriminantion and the definition of $\boldsymbol{d}$. There is a positive scale constant $1/(n-2)$ between $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, and the constant will not affect the property of positive or negative of $w(\boldsymbol{x})$, and consequently it will not affect the result of discriminantion. Therefore, the difference between the two discriminantion methods can only lie in the difference of the definition of $\boldsymbol{c}$. From the discriminantion result of Table 2, we find that the definition of $\boldsymbol{c}_1$ is better than that of $\boldsymbol{c}_2$. From the further analysis, we find that the difference between the two methods lie in the difference of the definition of $\bar{y}$. Similarly, from the discriminantion result of Table 2, we find that the definition of $\bar{y}^{(1)}$ is better than that of $\bar{y}^{(2)}$. The discriminantion results become consistent after replacing $\boldsymbol{c}_1$ by $\boldsymbol{c}_2$ in discriminant.fisher()

Table 2. The discrimination table of two classification of sand based liquefaction.

	discriminant.fisher()		distinguish.fisher()	
To / From	I	II	I	II
I	12	0	11	1
II	2	21	2	21

Table 3. The comparison results of the two classification Fisher discriminant methods.

	discriminant.fisher()	distinguish.fisher()
Discriminant	$w_1(\boldsymbol{x}) = \boldsymbol{d}_1^T \boldsymbol{S}_1^{-1}(\boldsymbol{x} - \boldsymbol{c}_1)$ [6] (8.55)	$w_2(\boldsymbol{x}) = \boldsymbol{d}_2^T \boldsymbol{S}_2^{-1}(\boldsymbol{x} - \boldsymbol{c}_2)$ [5] P184
Rule	$R_1 = \{\boldsymbol{x} \mid w_1(\boldsymbol{x}) \le 0\}$	$R_1 = \{\boldsymbol{x} \mid w_2(\boldsymbol{x}) \ge 0\}$
$\boldsymbol{d}$	$\boldsymbol{d}_1 = \bar{\boldsymbol{x}}_2 - \bar{\boldsymbol{x}}_1$	$\boldsymbol{d}_2 = \bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2$
$\boldsymbol{S}$	$\boldsymbol{S}_1 = \boldsymbol{E} = \sum_{i=1}^{2}\sum_{j=1}^{n_i}(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_i)(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_i)^T$	$\boldsymbol{S}_2 = \boldsymbol{S}_p = \frac{1}{n-2}\boldsymbol{E} = \frac{1}{n-2}\sum_{i=1}^{2}\sum_{j=1}^{n_i}(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_i)(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_i)^T$
$\boldsymbol{c}$	$\boldsymbol{c}_1 = \bar{\boldsymbol{x}} = \frac{n_1}{n}\bar{\boldsymbol{x}}_1 + \frac{n_2}{n}\bar{\boldsymbol{x}}_2 = \frac{1}{n}\sum_{i=1}^{2}\sum_{j=1}^{n_i}\boldsymbol{x}_{ij}$	$\boldsymbol{c}_2 = \frac{1}{2}(\bar{\boldsymbol{x}}_1 + \bar{\boldsymbol{x}}_2)$
$\boldsymbol{a}$	$\boldsymbol{a}_1 = \boldsymbol{S}_1^{-1}\boldsymbol{d}_1$	$\boldsymbol{a}_2 = \boldsymbol{S}_2^{-1}\boldsymbol{d}_2$

$\overline{y}$	$\overline{y}^{(1)} = \boldsymbol{a}_1^T \boldsymbol{c}_1 = \frac{n_1}{n}\overline{y}_1^{(1)} + \frac{n_2}{n}\overline{y}_2^{(1)}$ $\overline{y}_i^{(1)} = \left(\overline{\boldsymbol{x}}_2 - \overline{\boldsymbol{x}}_1\right)^T \boldsymbol{S}_1^{-1}\overline{\boldsymbol{x}}_i, i = 1,2$	$\overline{y}^{(2)} = \boldsymbol{a}_2^T \boldsymbol{c}_2 = \frac{1}{2}\left(\overline{y}_1^{(2)} + \overline{y}_2^{(2)}\right)$ $\overline{y}_i^{(2)} = \left(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2\right)^T \boldsymbol{S}_2^{-1}\overline{\boldsymbol{x}}_i, i = 1,2$

Summary

We first introduce an R function distinguish.fisher() of the multi-classification Fisher discriminant analysis and two other useful R functions: discriminant_table() (used to calculate the discriminant table) and plot_fisher() (used for the scatterplot of the first two discriminant scores). Then we do two case studies. The first case study compares the result of R function distinguish.fisher() and the SAS result of Example 5.4.2 of [5]. The results are consistent, which indicates that the R function distinguish.fisher() written by us is correct. The second case study is used to illustrate that the distinguish.fisher() function can be used to solve two classification problems. We also compare the results of distinguish.fisher() and the existing discriminant.fisher() from [6] in Table 2. Finally, we analyze the reasons for the difference of the two Fisher discriminant R functions in Table 3.

Acknowledgement

The research was supported by the Fundamental Research Funds for the Central Universities (CQDXWL-2012-004 and CDJRC10100010) and the Natural Science Foundation Project of CQ CSTC (CSTC2011BB0058).

References

[1] R.A. Johnson, D.W. Wichern (authors), X. Lu (translator), Applied Multivariate Statistical Analysis, 4th ed., Tsinghua University Press, Beijing, 2001.

[2] H. Yang, Q.S. Liu, B. Zhong, Mathematical Statistics, Higher Education Press, Beijing, 2004.

[3] H.X. Gao, Applied Multivariate Statistical Analysis, Peking University Press, Beijing, 2005.

[4] Y.C. Tang, R Language and Statistical Analysis, Higher Education Press, Beijing, 2008.

[5] X.M. Wang, Applied Multivariate Analysis, 3rd ed., Shanghai University of Finance and Economics Press, Shanghai, 2009.

[6] Y. Xue and L.P. Chen, Statistical Modeling and R Software, Tsinghua University Press, Beijing, 2009.

[7] X.S. Ren and X.L. Yu, Multivariate Statistical Analysis, 2nd ed., China Statistics Press, Beijing, 2011.

[8] X.Q. He, Multivariate Statistical Analysis, 3rd ed., China Renmin University Press, Beijing, 2012.

[9] S.L. Li, Data Analysis and R Software, Science Press, Beijing, 2013.

[10] H.J. Chen, X.B. Li, A.H. Liu, and S.Q. Peng, Identifying of mine water inrush sources by Fisher discriminant analysis method, Journal of Central South University (Science and Technology), 40 (2009) 1114–1120.

[11] L.N. Zhao, The study on introduction weighting factor of Fisher discriminant, Natural Science Journal of Harbin Normal University, 28 (2012) 24–26.

[12] M.Q. Chen, The relationship between canonical correlation analysis and Fisher discriminant method, Statistics and Decision, no. 2 (2013) 21–24.