Songklanakarin J. Sci. Technol.
45 (1), 156–164, Jan. – Feb. 2023

http://www.sjst.psu.ac.th

Original Article

Discriminant analysis involving count data

George Chinanu Mbaeyi* and Chijioke Joel Nweke

Department of Mathematics and Statistics, Faculty of Physical Sciences,
Alex Ekwueme Federal University Ndufu-Alike, 482131 Nigeria

Received: 19 October 2022; Revised: 21 December 2022; Accepted: 18 January 2023

Abstract

A situation giving rise to a violation of the normality assumption in discriminant analysis is that which involves count observations. For a two-variable case involving count observations, this paper presents a new discriminant analysis approach when one variable is observed conditional on the other. Two cases involving Poisson-Binomial and Poisson-Poisson distributions were considered. The derived allocation rules are based on the resulting joint distribution of the two count variables. Applicability of the suggested allocation rules in discriminant analysis involving count data and its performance in comparison with Fisher linear discriminant rule was studied under different conditions. Results obtained show promising applicability of the suggested allocation rules when compared with the Fisher linear discriminant method.

Keywords: count data, discriminant analysis, error rate, allocation rules, poisson distribution, binomial distribution

1. Introduction

Normality assumption is fundamental for useful inference in linear discriminant analysis. Unfortunately, most real-life situations are generated by mechanisms that violates this assumption. Such real-life phenomena include those involving discrete distributions that are amenable to count data. The most common probability distributions often used in analyzing count data are the Poisson and the negative Binomial distribution (Witten, 2011). In analyzing count data, Poisson variates are mostly faced with the problem of under- (or over) dispersion (Inouye, Yang, Allen, & Ravikumar, 2017). With Poisson variates in discriminant analysis, the authors are of the opinion that it may no longer be appropriate to use the linear discriminant analysis approach whose allocation rule is derived based on normality assumption. An optimal approach for discriminant analysis in this situation would be to derive the allocation rule based on the originating distribution (Mbaeyi & Nweke, 2021), and not the assumed normal distribution. Some studies concerning classification analysis involving count data are available in prior literature. Examples are Poisson linear discriminant analysis (Witten 2011), Negative Binomial linear discriminant analysis (Dong, Zhao, Tong, & Wan, 2016), zero inflated Poisson logistic discriminant analysis (Zhou, Wan, Zhang, & Tong, 2018), zero inflated negative Binomial logistic discriminant analysis (Zhu, Yuan, Shu, Liao, Zhao, & Zhou, 2021), and a decision tree model for count data (Wah, Nasaruddin, Voon, & Lazim, 2012). These methods are either based on no (or very weak) assumptions, or some inconsistent transformations. In addition, conditional relationships between variables were not adequately considered.

A typical scenario which this work attempts to consider is, for example, road accident occurrences that may lead to one or more casualties. The casualties are characterized by the extent of physical injury or fatality. Let X_i be the number of accidents in a given location for a given interval. X_i is assumed to follow Poisson distribution with parameter θ. Suppose that the variable Y_i assumes the value 1 with probability p if the ith accident is fatal, and the value 0 with probability $q = 1 - p$ if the ith accident is not fatal, then $Y = Y_1 + Y_2 + \cdots + Y_X$ represents the number of fatal accidents out of a total of X accidents. In this case, Y_i is better represented by a Binomial distribution with parameter p. It is expected that $Y \leq X$ and the bivariate distribution $f(x, y)$ represents the joint distribution of number of accidents and cases of fatal accident. One may also consider jointly the number of accidents X_i and the corresponding number of casualties Z_i in which case both X_i and Z_i follow Poisson distribution. The number of accidents

*Corresponding author
Email address: george.chinanu@funai.edu.ng

together with the corresponding casualty indices can be classified as having resulted from one of either a clear or a cloudy weather.

2. Methodology

Based on the typical scenario described in section I which this work attempts to focus on, the Poisson and Binomial discrete distributions shall be considered. Joint distribution of Poisson-Binomial and Poisson-Poisson types shall be considered, and, on the basis of the derived joint distribution, allocation rule for classifying observation $w = (x, y)$ shall then be obtained and applied consequently.

2.1 Poisson-binomial

Let X be a Poisson random variable with parameter λ, and Y be a Binomial random variable with parameters n, p and q. That is,

$$f(x) = \frac{e^{-\lambda}\lambda^x}{x!};\ \ \lambda > 0, x = 0,1,2,\ldots \tag{1}$$

and

$$f(y) = \binom{n}{y} p^y q^{n-y}; y = 0,1,2,\ldots,n \tag{2}$$

where n is the number of times the event of interest was observed, p is the probability of success in each of the n observed events, q is the probability of failure in each of the n observed events, and λ is the mean number of events. y is a realization of observations with defined attribute when observing x such that $P(y_i = 1) = p$ whenever the attribute is present in x and $P(y_i = 0) = q = 1 - p$ whenever the attribute is not present in x. It follows that y observation is made given that x is observed already. Hence, we can now define the conditional distribution of y given x as (Ramachandran & Tsokos, 2009).

$$f(y|x) = \binom{x}{y} p^y q^{x-y}; y = 0,1,2,\ldots,x \tag{3}$$

Combining (1) and (3), the joint distribution of x and y is given as

$$f(x,y) = \frac{e^{-\lambda}\lambda^x p^y q^{x-y}}{(x-y)!y!}; x = 0,1,2,\ldots; y = 0,1,2,\ldots,x \tag{4}$$

The marginal of (4) is a discrete Poisson probability mass function (See Appendix for proof).

Following Kendall & Stuart (1967), it can easily be shown that for (3) and (4),

$$E(y|x) = xp \tag{5}$$

$$Var(y|x) = xpq \tag{6}$$

$$E(x,y) = p\lambda(\lambda + 1) \tag{7}$$

$$Cov(x,y) = p\lambda \tag{8}$$

$$\rho(x,y) = +\sqrt{p} \tag{9}$$

The maximum likelihood estimates of the parameters of (4) are $\lambda = \bar{x}$ and $p = \frac{\bar{y}}{\bar{x}}$.

For the purpose of classification, optimal allocation rule for classifying the observation $\boldsymbol{w} = (x, y)$ into group D_1 or group D_2 can be derived using (4). Let $L_1(x, y, \lambda_1, p_1)$ and $L_2(x, y, \lambda_2, p_2)$ be the likelihood functions of (4) for group D_1 and group D_2, respectively. According to Anderson (1958), observation $\boldsymbol{w} = (x, y)$ can be classified as belonging to group D_1 if $L_1(x, y, \lambda_1, p_1) \geq L_2(x, y, \lambda_2, p_2)$. That is, if

$$R: \left(\frac{\lambda_1}{\lambda_2}\right)^x \left(\frac{p_1}{p_2}\right)^y \exp(\lambda_2 - \lambda_1) \geq 1 \tag{10}$$

By taking logarithm of (10) and simplifying, we have that

$$R: x\ln\left(\frac{\lambda_1}{\lambda_2}\right) + y\ln\left(\frac{p_1}{p_2}\right) \geq (\lambda_1 - \lambda_2) \tag{11}$$

Let the prior probabilities of an observation falling into group D_1 or group D_2 be π_1 and π_2 $(\pi_1 + \pi_2 = 1)$ respectively, then the Bayes' rule can obtained by comparing $\pi_i L_i(x, y, \lambda_i, p_i), i = 1,2$ in which case we are to allocate observation $\boldsymbol{w} = (x, y)$ to group D_1 if

$$R: x\ln\left(\frac{\lambda_1}{\lambda_2}\right) + y\ln\left(\frac{p_1}{p_2}\right) \geq \ln\left(\frac{\pi_2}{\pi_1}\right) + (\lambda_1 - \lambda_2) \tag{12}$$

Otherwise, observation $\boldsymbol{w} = (x, y)$ is allocated to group D_2. Where equal prior probability is assumed for group D_1 and D_2, (12) remains as in (11).

2.2 Poisson-poisson

Let k and r be two available variables for consideration in a discriminant analysis. Both k and r are independent Poisson variates with parameters λ and θ respectively.

$$g(k) = \frac{e^{-\lambda}\lambda^k}{k!};\ \ \lambda > 0, k = 0,1,2,\ldots \tag{13}$$

and

$$g(r) = \frac{e^{-\theta}\theta^r}{r!};\ \ \theta > 0, r = 0,1,2,\ldots \tag{14}$$

Observations r are made given that k has been observed. Thus, the conditional distribution of r given that k has been observed is given as

$$g(r|k) = \frac{e^{-\theta k}(\theta k)^r}{r!};\ r = 0,1,2,\ldots \tag{15}$$

As above, combining (13) and (15), the joint distribution of r and k is given in (16) as

$$g(k,r) = \frac{\lambda^k(\theta k)^r \exp\{-(\lambda + \theta k)\}}{k!r!}; \lambda, \theta > 0, k = 0,1,2,\ldots; r = 0,1,2,\ldots \tag{16}$$

As in (4), the marginal of (16) is also a univariate Poisson probability mass function (See Appendix for proof).

Some properties of (15) and (16) are readily given as

$$E(r|k) = \theta k \tag{17}$$

$$Var(r|k) = \theta k \tag{18}$$

$$E(k,r) = \theta\lambda(\lambda+1) \tag{19}$$

$$Cov(k,r) = \theta\lambda \tag{20}$$

$$\rho(k,r) = \sqrt{\tfrac{\theta}{\theta+1}} \tag{21}$$

The maximum likelihood estimates of the parameters of (16) are also given as $\lambda = \bar{k}$ and $\theta = \frac{\bar{r}}{\bar{k}}$. To perform discriminant analysis using the values of the variables r and k, the optimal allocation rule is best derived using the joint distribution in (16). By defining $L_1(k,r,\lambda_1,\theta_1)$ and $L_2(k,r,\lambda_2,\theta_2)$ to be the likelihood functions of (16) for group D_1 and group D_2 respectively, the allocation rule will be to allocate observation $\boldsymbol{w} = (k,r)$ to group D_1 (be classified as belonging to group D_1) if $L_1(k,r,\lambda_1,\theta_1)/L_2(k,r,\lambda_2,\theta_2) \geq 1$. That is, if

$$R\colon \left(\tfrac{\lambda_1}{\lambda_2}\right)^k \left(\tfrac{\theta_1}{\theta_2}\right)^r \exp(\lambda_2 - \lambda_1) \geq 1 \tag{22}$$

By also taking logarithm of (22) and simplifying, we have that

$$R\colon k\ln\left(\tfrac{\lambda_1}{\lambda_2}\right) + r\ln\left(\tfrac{\theta_1}{\theta_2}\right) \geq (\lambda_1 - \lambda_2) \tag{23}$$

Let the prior probabilities of observation falling into group D_1 and group D_2 be π_1 and π_2 $(\pi_1 + \pi_2 = 1)$ respectively, then the Bayes' rule can obtained by comparing $\pi_i L_i(k,r,\lambda_i,\theta_i), i = 1{,}2$ in which case we are to allocate observation $\boldsymbol{w} = (k,r)$ to group D_1 if

$$R\colon k\ln\left(\tfrac{\lambda_1}{\lambda_2}\right) + r\ln\left(\tfrac{\theta_1}{\theta_2}\right) \geq \ln\left(\tfrac{\pi_2}{\pi_1}\right) + (\lambda_1 - \lambda_2) \tag{24}$$

Otherwise, observation $\boldsymbol{w} = (k,r)$ is allocated to group D_2. Where equal prior probability is assumed for group D_1 and D_2, (24) remains as in (23).

2.3 Fisher linear discriminant analysis

Fisher linear discriminant (FLD) analysis is a generalization of linear discriminant analysis, a method commonly used to find a linear combination of attributes that characterizes or separates two or more groups of objects. Given a set of independent multivariate observations, the FLD assumes that observations are distributed multivariate normal with vector of means which varies in each group but a covariance matrix that is common across all groups. FLD rule maximizes the ratio between sum of squares between and sum of squares within and then finds a linear combination of the predictors to predict group membership. Typically, given a vector $\boldsymbol{x}$ of observations assumed to be multivariate normal with mean $\boldsymbol{\mu_i}(i = 1{,}2)$ and covariance matrix $\boldsymbol{\Sigma}$ coming from one of two groups and assuming equal prior probabilities of group membership, the FLD allocation rule is to classify observation $\boldsymbol{x}$ as belonging to group G_1 if

$$\boldsymbol{x}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \geq \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

It has been argued that discriminant analysis is relatively robust to violations of normality and homoscedasticity assumptions (Hardle & Simar, 2007). Hence, in line with the focus of this work, this study shall consider an application of FLD to count data.

3. Analysis, Results and Discussion

In other to demonstrate the applicability of the allocation rules presented in section (2), both artificial and real-life data were considered. For the artificial data, random samples of Poisson and Binomial variates were generated under various sample sizes (n_1=n_2=20, 50, 100, 150, 200, 300, 500, 750, 1000, 1500, 2000) for a two-group discriminant analysis. In other to evaluate the allocation rules in (12) and (24), $\lambda_1 = 1.2, \lambda_2 = 0.8, \theta_1 = 0.42, \theta_2 = 0.31, p_1 = 0.42$ and $p_2 = 0.31$ were used in generating Poisson and Binomial data. λ_1 and λ_2 are Poisson parameters for variable x_1 in groups D_1 and D_2. Similarly, θ_1 and θ_2 are Poisson parameters for variable x_2 in groups D_1 and D_2, while p_1 and p_2 are Binomial parameters for variable x_2 in groups D_1 and D_2 respectively. As a way of introducing over-dispersion and under-dispersion into the dataset, 20% of each of n_1 and n_2 was generated by replacing λ_1 and λ_2 by $9.5\lambda_1$ and $9.5\lambda_2$ respectively to obtain over-dispersed data while under-dispersion was introduced by replacing λ_1 and λ_2 by $0.125\lambda_1$ and $0.125\lambda_2$ respectively. The Poisson-Binomial (P-B) and Poisson-Poisson (P-P) based allocation rules were then applied to the generated data for a two-group discriminant analysis. For comparison purpose, the Fisher linear discriminant (FLD) analysis procedure was also applied to the generated data. Error rates resulting from the P-B, P-P and FLD allocation rules were obtained and are presented when the data are (i) under-dispersed, (ii) undispersed, and (iii) over-dispersed. Error rate is a measure of misclassifications made by any given allocation rule. It is simply obtained by dividing the sum of misclassified observations in groups D_1 and D_2 by the total number of observations.

For the real-life data, record of accidents in the UK was extracted from https://www.kaggle.com/datasets/benoit72/uk-accidents-10-years-history-with-many-variables?select=Vehicles0514.csv and analyzed as in two-group discriminant analysis. The data contain many variables related to road accidents in the UK between 2005 and 2014, some of which include weather, location, latitude/longitude, accident severity, casualty severity, casualty type, age of casualty, age of driver, age of victim, etc. Number of accidents (x) and number of casualties (y) were used for the P-P approach while number of casualties (k) and incidences of fatal casualties (r) were used for the P-B approach. After extraction, $n = 20$ observed cases of accidents classified under "Clear" and "Cloudy" weather for the period 2005 – 2014 were available for analysis. All analysis was performed in R (2020) programming environment and the results are presented in Table 1 and Table 2 below.

Table 1. Error rates of the P-P and FLD under various types of dispersion and sample sizes

Sample size (n)	Over-dispersed		Under-dispersed		Undispersed	
	P-P	FLD	P-P	FLD	P-P	FLD
20	0.3750	0.4000	0.2750	0.4000	0.2750	0.4250
50	0.5000	0.4800	0.2800	0.4900	0.3600	0.4200
100	0.5150	0.4900	0.2450	0.4550	0.3600	0.4200
150	0.4567	0.4867	0.3133	0.4533	0.3800	0.4267
200	0.4225	0.4400	0.2950	0.4350	0.3500	0.4150
300	0.4367	0.4433	0.2900	0.4217	0.3600	0.4200
500	0.4390	0.4790	0.3020	0.4350	0.3610	0.4200
750	0.4327	0.4500	0.3307	0.4333	0.3760	0.4387
1000	0.4380	0.4555	0.3320	0.4400	0.3895	0.4305
1500	0.4347	0.4590	0.3203	0.4350	0.3813	0.4253
2000	0.4303	0.4435	0.3215	0.4235	0.3840	0.4100

Table 2. Error rates of P-B and FLD under various types of dispersion and sample sizes

Sample size (n)	Over-dispersed		Under-dispersed		Undispersed	
	P-B	FLD	P-B	FLD	P-B	FLD
20	0.5000	0.3000	0.5000	0.3250	0.4750	0.325
50	0.5000	0.3333	0.5100	0.3600	0.4100	0.3700
100	0.4950	0.3800	0.4050	0.3300	0.5050	0.3300
150	0.4667	0.3567	0.4500	0.3433	0.4967	0.3533
200	0.4450	0.3100	0.4300	0.3275	0.4950	0.3200
300	0.4833	0.3167	0.4200	0.3333	0.4883	0.3217
500	0.4300	0.3530	0.4340	0.3500	0.4950	0.3370
750	0.4540	0.3440	0.4773	0.3460	0.4693	0.3447
1000	0.4615	0.3320	0.4210	0.3465	0.4900	0.3355
1500	0.4373	0.3393	0.3937	0.3440	0.4893	0.3380
2000	0.4228	0.3545	0.4290	0.3493	0.4935	0.3448

The results show the applicability of the suggested allocation rules. Since the problem of dispersion is common with Poisson variates, it suffices to consider various forms of dispersion in the application. Generally, in terms of error rate as presented in Table 1, P-P performed better than FLD both when there is under-dispersion and no dispersion in the dataset. However, with over-dispersed data, FLD fairly outperformed P-P. Error rates resulting from FLD in Table 1 appear to show slight evidence of stability regardless of the form of dispersion present in the data, as the error rates were between approximately 0.4100 and 0.4900. From the results presented in Table 2, FLD has an overall better performance than the P-B approach. Again, a fair case-wise stability in error rate was noted for both P-B and FLD. Hence, the effect of changing sample size appear not to noticeably affect the error rate of P-B or FLD, but this appears not to be the case with the various forms of dispersion considered. Moreover, introducing a binomial variate for the P-B analysis appears to have slightly improved the error rates of FLD in Table 2 when compared with those of FLD in Table 1, but such was not the case for the P-B approach. In Tables 1 and 2, the error rates for P-P, P-B and FLD show no trend with respect to increasing/decreasing sample size, hence, it may not be inferred that increasing/ decreasing sample size improves error rate of any of the approaches considered in this work. On a general note, the lower error rates from FLD may not be valid enough to justify its usage in discriminant analysis when available data are not normally distributed. In case the argument persists, the optimal answer would depend on a choice between having a lower error rate using an incorrect methodology or having a moderate error rate using a correct methodology.

For the real-life data, all the approaches considered performed almost equally in terms of their respective error rates. When the data involving number of accidents (x) and number of casualties (y) were analyzed, the error rates were 0.4880 and 0.5000 for P-P and FLD respectively. Similarly, with the data involving number of casualties (k) and incidences of fatal casualties (r), corresponding error rates were 0.4920 and 0.5000 for P-B and FLD respectively. Unlike in the artificial datasets, error rates from FLD were not better than P-P and P-B. However, it can easily be observed that the former looks similar to the result in Table 1 when $n = 20$ while the latter is somewhat dissimilar with the $n = 20$ case in Table 2. Hence, whereas applicability of the various approaches has been demonstrated with a real-life dataset, more cases representing real-life scenarios need to be analyzed in other to make valid inference regarding these approaches and their application in real-world data analysis.

4. Conclusions

This paper has presented allocation rules based on discrete distributions amenable to count data. The allocation rules suggested in this paper demonstrated straightforward applicability and ability to handle cases of under-dispersion and over-dispersion in Poisson variates. The suggested allocation rules are amenable to simple error rate estimation procedures, cope with the problem of small available samples, and their implementation is easy in any user-defined program package. The performance of the suggested allocation rules in comparison with the Fisher linear discriminant analysis indicates that an appreciable level of accuracy can be gained when allocation rules are derived based on the originating distribution of the data.

References

Dong, K., Zhao, H., Tong, T., & Wan, X. (2016). Negative binomial linear discriminant analysis for RNA-seq data. *BMC Bioinformatics.* doi:10.1186/s12859-016-1208-1

Hardle, W, & Simar L. (2007). *Applied multivariate statistical analysis*. Berlin Heidelberg, Germany: Springer.

Inouye, D. I., Yang, E., Allen, G. I., & Ravikumar, P. (2017). A review of multivariate distributions for count data derived from the Poisson distribution. *Computational Statistics, 9*(3), e1398. Retrieved from https://doi.org/10/1002/wics.1398

Kendall, M. G., & Stuart, A. (1967). *The advanced theory of statistics.* London, England: Griffin.

Mbaeyi, G. C. & Nweke, C. J. (2021). Discriminant analysis with non normal variables. *Communication in Statistics – Theory and Methods.* doi:10.1080/03610926.2021.1908563

Ramachandran, K. M. & Tsokos, C. P. (2009). *Mathematical statistics with applications.* London, England: Elsevier Academic Press.

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org/

Wah, Y. B., Nasaruddin, N., Voon, W. S. & Lazim, M. A. (2012). Decision tree model for count data. *Proceedings of the World congress in Engineering, 1,* 330-335. Retrieved from http://www.iaeng.org/publication/WCE2012/WCE2012_pp330-335.pdf

Witten, D. (2011). Classification and clustering of sequencing data using a Poisson model. *Annals Applied Statistics, 5,* 2493-2518. doi:10.1214/11-AOAS493.

Zhou, Y., Wan, X., Zhang, B. & Tong, T. (2018). Classifying next-generation sequencing data using a zero-inflated Poisson model. *Bionformatics, 34,* 1329-1335. doi: 10.1093/bioinformatics/btx768

Zhu, J., Yuan, Z., Shu, L., Liao, W., Zhao, M. & Zhou, Y. (2021). Selecting classification methods for small samples of next-generation sequencing data. *Frontiers in Genetics, 12,* doi:10.3389/fgene.2021.6422

Appendix

Marginal of P-B

$$f(x,y)=\frac{e^{-\lambda}\lambda^{x}p^{y}q^{x-y}}{(x-y)!\,y!}=\left(\frac{x!}{x!}\right)\left(\frac{e^{-\lambda}\lambda^{x}p^{y}q^{x-y}}{(x-y)!\,y!}\right)=\frac{\lambda^{x}e^{-\lambda}}{x!}\binom{x}{y}p^{y}q^{x-y}$$

$$f_X(x)=\frac{f(x,y)}{f(y|x)}=\frac{\frac{\lambda^{x}e^{-\lambda}}{x!}\binom{x}{y}p^{y}q^{x-y}}{\binom{x}{y}p^{y}q^{x-y}}=\frac{\lambda^{x}e^{-\lambda}}{x!};x=0,1,2,\ldots$$

Marginal of P-P

$$g_K(k)=\frac{g(k,r)}{g(r|k)}=\frac{\frac{\lambda^{k}(\theta k)^{r}\exp\{-(\lambda+\theta k)\}}{k!\,r!}}{\frac{e^{-\theta k}(\theta k)^{r}}{r!}}=\frac{\lambda^{k}\exp^{\{-\lambda\}}}{k!};k=0,1,2$$

R Source Code

```
library(MASS)
## Over-Dispersed P-P and FLD
Overdispersed<-function(n1,n2){
set.seed(200)
lambda1<-1.2
lambda2<-0.8
theta1<-0.42
theta2<-0.31
xx<-rpois(0.8*n1,lambda1)
x<-rpois(0.2*n1,lambda1*9.5)
x11<-c(x,xx)
x21<-rpois(n1,theta1)
XX<-rpois(0.8*n2,lambda2)
X<-rpois(0.2*n2,lambda2*9.5)
x12<-c(XX,X)
x22<-rpois(n2,theta2)
g1<-rep(1,n1)
g2<-rep(2,n2)
d1<-cbind(x11,x21,g1)
d2<-cbind(x12,x22,g2)
Grp1<-cbind(x11,x21)
Grp2<-cbind(x12,x22)
c<-Grp1[,1]
cc<-Grp1[,2]
C<-Grp2[,1]
CC<-Grp2[,2]
Data<-rbind(d1,d2)
g<-Data[,3]
x1<-Data[,1]
x2<-Data[,2]
Dt<-data.frame(g,x1,x2)
fit<-lda(fomula=g~.,data=Dt)
tabel<-table(actual=Dt$g,predicted=predict(fit,Dt)$class)
pi1<-0.5
pi2<-0.5
D<-log(pi2/pi1)+(lambda1-lambda2)
a <- 0
```

```
for(i in 1:length(Grp1[,1])){
A<-cc[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-c[i]*log(theta1/theta2)
w1 <- A+B
if(w1>=D) {
a <- a+1
}
}
b <- 0
for(i in 1:length(Grp1[,1])){
A<-cc[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-c[i]*log(theta1/theta2)
w1 <- A+B
if(w1<D) {
b <- b+1
}
}
c <- 0
for(i in 1:length(Grp2[,1])){
A<-CC[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-C[i]*log(theta1/theta2)
w2 <- A+B
if(w2>=D) {
c <- c+1
}
}
d <- 0
for(i in 1:length(Grp2[,1])){
A<-CC[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-C[i]*log(theta1/theta2)
w2 <- A+B
if(w2<D) {
d <- d+1
}
}
tr <- ftable(tabel)
ErFLD <- (tr[1,2]+tr[2,1])/(n1+n2)
ErPP <- (b + c)/(n1+n2)
list(c(ErPP,ErFLD))
#END
}

## Under-Dispersed P-P and FLD
Underdispersed<-function(n1,n2){
set.seed(200)
lambda1<-1.2
lambda2<-0.8
theta1<-0.42
theta2<-0.31
xx<-rpois(0.8*n1,lambda1)
x<-rpois(0.2*n1,lambda1/8)
x11<-c(x,xx)
x21<-rpois(n1,theta1)
XX<-rpois(0.8*n2,lambda2)
X<-rpois(0.2*n2,lambda2/8)
x12<-c(XX,X)
x22<-rpois(n2,theta2)
g1<-rep(1,n1)
g2<-rep(2,n2)
d1<-cbind(x11,x21,g1)
d2<-cbind(x12,x22,g2)
Grp1<-cbind(x11,x21)
Grp2<-cbind(x12,x22)
c<-Grp1[,1]
cc<-Grp1[,2]
C<-Grp2[,1]
CC<-Grp2[,2]
Data<-rbind(d1,d2)
g<-Data[,3]
x1<-Data[,1]
x2<-Data[,2]
Dt<-data.frame(g,x1,x2)
fit<-lda(fomula=g~.,data=Dt)
tabel<-table(actual=Dt$g,predicted=predict(fit,Dt)$class)
pi1<-0.5
pi2<-0.5
D<-log(pi2/pi1)+(lambda1-lambda2)
a <- 0
for(i in 1:length(Grp1[,1])){
A<-cc[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-c[i]*log(theta1/theta2)
w1 <- A+B
if(w1<D) {
a <- a+1
}
}
b <- 0
for(i in 1:length(Grp1[,1])){
A<-cc[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-c[i]*log(theta1/theta2)
w1 <- A+B
if(w1>=D) {
b <- b+1
}
}
c <- 0
for(i in 1:length(Grp2[,1])){
A<-CC[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-C[i]*log(theta1/theta2)
w2 <- A+B
if(w2>=D) {
c <- c+1
}
}
d <- 0
for(i in 1:length(Grp2[,1])){
A<-CC[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-C[i]*log(theta1/theta2)
w2 <- A+B
if(w2<D) {
d <- d+1
}
}
tr <- ftable(tabel)
ErFLD <- (tr[1,2]+tr[2,1])/(n1+n2)
ErPP <- (b + c)/(n1+n2)
list(c(ErPP,ErFLD))
#END
}

## Undispersed P-P and FLD
Undispersed<-function(n1,n2){
set.seed(200)
lambda1<-1.2
```

```
lambda2<-0.8
theta1<-0.42
theta2<-0.31
x11<-rpois(n1,lambda1)
x21<-rpois(n1,theta1)
x12<-rpois(n2,lambda2)
x22<-rpois(n2,theta2)
g1<-rep(1,n1)
g2<-rep(2,n2)
d1<-cbind(x11,x21,g1)
d2<-cbind(x12,x22,g2)
Grp1<-cbind(x11,x21)
Grp2<-cbind(x12,x22)
c<-Grp1[,1]
cc<-Grp1[,2]
C<-Grp2[,1]
CC<-Grp2[,2]
Data<-rbind(d1,d2)
g<-Data[,3]
x1<-Data[,1]
x2<-Data[,2]
Dt<-data.frame(g,x1,x2)
fit<-lda(fomula=g~.,data=Dt)
tabel<-table(actual=Dt$g,predicted=predict(fit,Dt)$class)
pi1<-0.5
pi2<-0.5
D<-log(pi2/pi1)+(lambda1-lambda2)
a <- 0
for(i in 1:length(Grp1[,1])){
A<-cc[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-c[i]*log(theta1/theta2)
w1 <- A+B
if(w1<D) {
a <- a+1
}
}
b <- 0
for(i in 1:length(Grp1[,1])){
A<-cc[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-c[i]*log(theta1/theta2)
w1 <- A+B
if(w1>=D) {
b <- b+1
}
}
c <- 0
for(i in 1:length(Grp2[,1])){
A<-CC[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-C[i]*log(theta1/theta2)
w2 <- A+B
if(w2>=D) {
c <- c+1
}
}
d <- 0
for(i in 1:length(Grp2[,1])){
A<-CC[i]*(log(lambda1/lambda2)+theta2-theta1)
B<-C[i]*log(theta1/theta2)
w2 <- A+B
if(w2<D) {
d <- d+1
}
}
tr <- ftable(tabel)
ErFLD <- (tr[1,2]+tr[2,1])/(n1+n2)
ErPP <- (b + c)/(n1+n2)
list(c(ErPP,ErFLD))
#END
}

## Overdispersed P-B and FLD
Overdispersed<-function(n1,n2){
set.seed(200)
N<-10
lambda11<-1.2
lambda22<-0.8
p11<-0.42
p22<-0.31
xx<-rpois(0.8*n1,lambda11)
x<-rpois(0.2*n1,lambda11*9.5)
x11<-c(xx,x)
x21<-rbinom(n1,N,p11)
XX<-rpois(0.8*n2,lambda22)
X<-rpois(0.2*n2,lambda22*9.5)
x12<-c(XX,X)
x22<-rbinom(n2,N,p22)
g1<-rep(1,n1)
g2<-rep(2,n2)
d1<-cbind(x11,x21,g1)
d2<-cbind(x12,x22,g2)
C1<-cbind(x11,x21)
C2<-cbind(x12,x22)
c<-C1[,1]
cc<-C1[,2]
C<-C2[,1]
CC<-C2[,2]
Data<-rbind(d1,d2)
g<-Data[,3]
x1<-Data[,1]
x2<-Data[,2]
Dt<-data.frame(g,x1,x2)
fit<-lda(fomula=g~.,data=Dt)
tabel<-table(actual=Dt$g,predicted=predict(fit,Dt)$class)
pi1<-0.5
pi2<-0.5
lambda1<-mean(x11)
lambda2<-mean(x12)
p1<-mean(x11)/mean(x21)
p2<-mean(x12)/mean(x22)
D<-log(pi2/pi1)+(lambda1-lambda2)
a <- 0
for(i in 1:length(C1[,1])){
A<-c[i]*log(lambda1/lambda2)
B<-cc[i]*log(p1/p2)
w1 <- A+B
if(w1>=D) {
a <- a+1
}
}
b <- 0
for(i in 1:length(C1[,1])){
A<-c[i]*log(lambda1/lambda2)
B<-cc[i]*log(p1/p2)
w1 <- A+B
if(w1<D) {
b <- b+1
}
}
c <- 0
for(i in 1:length(C2[,1])){
A<-C[i]*log(lambda1/lambda2)
B<-CC[i]*log(p1/p2)
w2 <- A+B
if(w2>=D) {
c <- c+1
}
}
d <- 0
for(i in 1:length(C2[,1])){
A<-C[i]*log(lambda1/lambda2)
B<-CC[i]*log(p1/p2)
w2 <- A+B
if(w2<D) {
d <- d+1
}
}
tr <- ftable(tabel)
ErFLD <- (tr[1,2]+tr[2,1])/(n1+n2)
ErPB <- (b + c)/(n1+n2)
list(c(ErPB,ErFLD))
#END
}

## Underdispersed P-B and FLD
Underdispersed<-function(n1,n2){
set.seed(200)
N<-10
lambda11<-1.2
lambda22<-0.8
p11<-0.42
p22<-0.31
xx<-rpois(0.8*n1,lambda11)
x<-rpois(0.2*n1,lambda11/8)
x11<-c(xx,x)
x21<-rbinom(n1,N,p11)
XX<-rpois(0.8*n2,lambda22)
X<-rpois(0.2*n2,lambda22/8)
x12<-c(XX,X)
x22<-rbinom(n2,N,p22)
g1<-rep(1,n1)
g2<-rep(2,n2)
d1<-cbind(x11,x21,g1)
d2<-cbind(x12,x22,g2)
C1<-cbind(x11,x21)
C2<-cbind(x12,x22)
c<-C1[,1]
cc<-C1[,2]
C<-C2[,1]
CC<-C2[,2]
Data<-rbind(d1,d2)
g<-Data[,3]
x1<-Data[,1]
x2<-Data[,2]
Dt<-data.frame(g,x1,x2)
fit<-lda(fomula=g~.,data=Dt)
tabel<-table(actual=Dt$g,predicted=predict(fit,Dt)$class)
pi1<-0.5
pi2<-0.5
lambda1<-mean(x11)
lambda2<-mean(x12)
p1<-mean(x11)/mean(x21)
p2<-mean(x12)/mean(x22)
D<-log(pi2/pi1)+(lambda1-lambda2)
a <- 0
for(i in 1:length(C1[,1])){
A<-c[i]*log(lambda1/lambda2)
B<-cc[i]*log(p1/p2)
w1 <- A+B
if(w1>=D) {
a <- a+1
}
}
b <- 0
for(i in 1:length(C1[,1])){
A<-c[i]*log(lambda1/lambda2)
B<-cc[i]*log(p1/p2)
w1 <- A+B
if(w1<D) {
b <- b+1
}
}
c <- 0
for(i in 1:length(C2[,1])){
A<-C[i]*log(lambda1/lambda2)
B<-CC[i]*log(p1/p2)
w2 <- A+B
if(w2>=D) {
c <- c+1
}
}
d <- 0
for(i in 1:length(C2[,1])){
A<-C[i]*log(lambda1/lambda2)
B<-CC[i]*log(p1/p2)
w2 <- A+B
if(w2<D) {
d <- d+1
}
}
tr <- ftable(tabel)
ErFLD <- (tr[1,2]+tr[2,1])/(n1+n2)
ErPB <- (b + c)/(n1+n2)
list(c(ErPB,ErFLD))
#END
}

## Undispersed P-B and FLD
Undispersed<-function(n1,n2){
set.seed(200)
N<-10
lambda11<-1.2
lambda22<-0.8
p11<-0.42
p22<-0.31
x11<-rpois(n1,lambda11)
x21<-rbinom(n1,N,p11)
x12<-rpois(n2,lambda22)
x22<-rbinom(n2,N,p22)
```

```
g1<-rep(1,n1)
g2<-rep(2,n2)
d1<-cbind(x11,x21,g1)
d2<-cbind(x12,x22,g2)
C1<-cbind(x11,x21)
C2<-cbind(x12,x22)
c<-C1[,1]
cc<-C1[,2]
C<-C2[,1]
CC<-C2[,2]
Data<-rbind(d1,d2)
g<-Data[,3]
x1<-Data[,1]
x2<-Data[,2]
Dt<-data.frame(g,x1,x2)
fit<-lda(fomula=g~.,data=Dt)
tabel<-table(actual=Dt$g,predicted=predict(fit,Dt)$class)
pi1<-0.5
pi2<-0.5
lambda1<-mean(x11)
lambda2<-mean(x12)
p1<-mean(x11)/mean(x21)
p2<-mean(x12)/mean(x22)
D<-log(pi2/pi1)+(lambda1-lambda2)
a <- 0
for(i in 1:length(C1[,1])){
A<-cc[i]*log(lambda1/lambda2)
B<-c[i]*log(p1/p2)
w1 <- A+B
if(w1>=D) {
a <- a+1
}
}
b <- 0
for(i in 1:length(C1[,1])){
A<-cc[i]*log(lambda1/lambda2)
B<-c[i]*log(p1/p2)
w1 <- A+B
if(w1<D) {
b <- b+1
}
}
c <- 0
for(i in 1:length(C2[,1])){
A<-CC[i]*log(lambda1/lambda2)
B<-C[i]*log(p1/p2)
w2 <- A+B
if(w2>=D) {
c <- c+1
}
}
d <- 0
for(i in 1:length(C2[,1])){
A<-CC[i]*log(lambda1/lambda2)
B<-C[i]*log(p1/p2)
w2 <- A+B
if(w2<D) {
d <- d+1
}
}
tr <- ftable(tabel)
ErFLD <- (tr[1,2]+tr[2,1])/(n1+n2)
ErPB <- (b + c)/(n1+n2)
list(c(ErPB,ErFLD))
#END
}
```