Journal of Statistical Software

September 2005, Volume 14, Issue 10. *http://www.jstatsoft.org/*

Bivariate Poisson and Diagonal Inflated Bivariate Poisson Regression Models in R

Dimitris Karlis
Athens Univ. of Economics and Business

Ioannis Ntzoufras
Athens Univ. of Economics and Business

Abstract

In this paper we present an R package called **bivpois** for maximum likelihood estimation of the parameters of bivariate and diagonal inflated bivariate Poisson regression models. An Expectation-Maximization (EM) algorithm is implemented. Inflated models allow for modelling both over-dispersion (or under-dispersion) and negative correlation and thus they are appropriate for a wide range of applications. Extensions of the algorithms for several other models are also discussed. Detailed guidance and implementation on simulated and real data sets using **bivpois** package is provided.

Keywords: bivariate Poisson distribution, EM algorithm, zero and diagonal inflated models, R functions, multivariate count data.

1. Introduction

Bivariate Poisson models are appropriate for modeling paired count data exhibiting correlation. Paired count data arise in a wide context including marketing (number of purchases of different products), epidemiology (incidents of different diseases in a series of districts), accident analysis (number of accidents in a site before and after infrastructure changes), medical research (the number of seizures before and after treatment), sports (the number of goals scored by each one of the two opponent teams in soccer), econometrics (number of voluntary and involuntary job changes), just to name a few. Unfortunately the literature on such models is sparse due to computational problems involved in their implementation.

Bivariate Poisson models can be expanded to allow for covariates, extending naturally the univariate Poisson regression setting. Due to the complicated nature of the probability function of the bivariate Poisson distribution, applications are limited. The aim of this paper is to introduce and construct efficient Expectation-Maximization (EM) algorithms for such models including easy-to-use R functions for their implementation. We further extend our methodology to construct inflated versions of the bivariate Poisson model. We propose a model that allows inflation in the diagonal elements of the probability table. Such models are quite useful when, for some reasons, we expect diagonal combinations with higher probabilities than the ones fitted under a bivariate Poisson model. For example, in pre and post treatment studies, the treatment may not have an effect on some specific patients for unknown reasons. Another example arises in sports where, for specific cases, it has been found that the number of draws in a game is larger than those predicted by a simple bivariate Poisson model (Karlis and Ntzoufras 2003).

In addition, an interesting property of inflated models is their ability to allow for modeling both correlation between two variables and over-dispersion (or alternatively under-dispersion) of the corresponding marginal distributions. Given their simplicity, such models are quite interesting for practical purposes.

The remaining of the paper proceeds as follows: in Section 2 we introduce briefly the bivariate Poisson and the diagonal inflated bivariate Poisson regression models. In Section 3 we provide a detailed description of the R functions. Several illustrative examples (simulated and real) including guidance concerning the fitting of the models can be found in Section 4. Finally, we end up with some concluding remarks in Section 5. Detailed description and presentation of the EM algorithms for maximum likelihood (ML) estimation is provided at the appendix.

2. Models for bivariate Poisson data

2.1. Bivariate Poisson regression models

Consider random variables X_κ, $\kappa = 1, 2, 3$ which follow independent Poisson distributions with parameters λ_κ, respectively. Then the random variables $X = X_1 + X_3$ and $Y = X_2 + X_3$ follow jointly a bivariate Poisson distribution, $BP(\lambda_1, \lambda_2, \lambda_3)$, with joint probability function

$$f_{BP}(x, y \mid \lambda_1, \lambda_2, \lambda_3) = e^{-(\lambda_1+\lambda_2+\lambda_3)} \frac{\lambda_1^x}{x!} \frac{\lambda_2^y}{y!} \sum_{i=0}^{\min(x,y)} \binom{x}{i} \binom{y}{i} i! \left(\frac{\lambda_3}{\lambda_1 \lambda_2} \right)^i . \tag{1}$$

The above bivariate distribution allows for positive dependence between the two random variables. Marginally each random variable follows a Poisson distribution with $\mathsf{E}(X) = \lambda_1 + \lambda_3$ and $\mathsf{E}(Y) = \lambda_2 + \lambda_3$. Moreover, $\mathsf{COV}(X, Y) = \lambda_3$, and hence λ_3 is a measure of dependence between the two random variables. If $\lambda_3 = 0$ then the two variables are independent and the bivariate Poisson distribution reduces to the product of two independent Poisson distributions (referred as double Poisson distribution). For a comprehensive treatment of the bivariate Poisson distribution and its multivariate extensions the reader can refer to Kocherlakota and Kocherlakota (1992) and Johnson, Kotz, and Balakrishnan (1997).

More realistic models can be considered if we model λ_1, λ_2 and λ_3 using covariates as regressors. In such case, the Bivariate Poisson regression model takes the form

$$\begin{array}{rcl} (X_i, Y_i) & \sim & BP(\lambda_{1i}, \lambda_{2i}, \lambda_{3i}), \\ \log(\lambda_{1i}) & = & \boldsymbol{w}_{1i}^\top \boldsymbol{\beta}_1, \\ \log(\lambda_{2i}) & = & \boldsymbol{w}_{2i}^\top \boldsymbol{\beta}_2, \\ \log(\lambda_{3i}) & = & \boldsymbol{w}_{3i}^\top \boldsymbol{\beta}_3, \end{array} \tag{2}$$

where $i = 1, \ldots, n$, denotes the observation number, $\boldsymbol{w}_{\kappa i}$ denotes a vector of explanatory variables for the i-th observation used to model $\lambda_{\kappa i}$ and $\boldsymbol{\beta}_\kappa$ denotes the corresponding vector of regression coefficients, $\kappa = 1, 2, 3$. The explanatory variables used to model each parameter $\lambda_{\kappa i}$ may not be the same. Usually, we consider models with constant λ_3 (no covariates on λ_3) because such models are easier to interpret. Although assuming a constant covariance term results to models that are easy to interpret, using covariates on λ_3 helps us to have more insight regarding the type of influence that a covariate has on each pair of variables. To make this understood recall that the marginal mean for X_i is equal (from (2)) to $\mathsf{E}(X_i) = \exp(\boldsymbol{w}_{1i}^\top \boldsymbol{\beta}_1) + \exp(\boldsymbol{w}_{3i}^\top \boldsymbol{\beta}_3)$. If a covariate is present in both $\boldsymbol{w}_1$ and $\boldsymbol{w}_3$, then a considerable part of the influence of this covariate is through the covariance parameter that is common for both X and Y variables. Moreover, such an effect is no longer multiplicative on the marginal mean (additive on the logarithm) but much more complicated (multiplicative on λ_1 and λ_3 and additive on the marginal mean).

Jung and Winkelmann (1993) introduced and implemented bivariate Poisson regression model using a Newton-Raphson procedure. Ho and Singer (2001) and Kocherlakota and Kocherlakota (2001) proposed a generalized least squares and Newton Raphson algorithm for maximizing the log-likelihood respectively. Here we construct an EM algorithm to remedy convergence problems encountered with the Newton Raphson procedure. The algorithm is easily coded to any statistical package offering algorithms fitting generalized linear models (GLM). Here, we provide R functions for implementing the algorithm. Standard errors for the parameters can be calculated using the information matrix provided in Jung and Winkelmann (1993) or using standard bootstrap methods. The latter is quite easy since good initial values are available and the algorithm converges fairly quickly. Finally, Bayesian inference has been implemented by Tsionas (2001) and Karlis and Meligkotsidou (2005).

2.2. Diagonal inflated bivariate Poisson regression models

A major drawback of the bivariate Poisson model is its property to model data with positive correlation only. Moreover, since its marginal distributions are Poisson they cannot model over-dispersion/under-dispersion. As a remedy to the above problems, we may consider mixtures of bivariate Poisson models like those of Munkin and Trivedi (1999) and Chib and Winkelmann (2001). However, such models involve difficult computations regarding estimation and can not handle under-dispersion. In this section, we propose diagonal inflated models that are computationally tractable and allow for over-dispersion, (under-dispersion) and negative correlation. The results reported are new apart from a quick comment in Karlis and Ntzoufras (2003) and some special cases treated in Li, Lu, Park, Kim, and Peterson (1999), Dixon and Coles (1997) and Wang, Lee, Yau, and Carrivick (2003).

In the univariate setting, inflated models can be constructed by inflating the probabilities of certain values of variable under consideration, X. Among them, zero-inflated models are very popular; see, for example, Lambert (1992), Bohning, Dietz, Schlattmann, Mendonca, and Kirchner (1999). Moreover, package **zicounts** is available via CRAN which can be used to fit zero inflated models. In the multivariate setting, there are few papers discussing inflated model in bivariate discrete distributions. Such models have been proposed by Dixon and Coles (1997) for modeling soccer games, Li *et al.* (1999) and Wang *et al.* (2003) who considered inflation only for the (0,0) cell, Wahlin (2001) who discussed zero-inflated bivariate Poisson models and Gan (2000).

We propose a more general model formulation which inflates the probabilities in the diagonal of the probability table. This model is an extension of the simple zero-inflated model which allows only for an excess in $(0, 0)$ cell. We consider, for generality, that the starting model is the bivariate Poisson model. Under this approach a diagonal inflated model is specified by

$$f_{IBP}(x, y) = \begin{cases} (1-p)f_{BP}(x, y \mid \lambda_1, \lambda_2, \lambda_3), & x \neq y \\ (1-p)f_{BP}(x, y \mid \lambda_1, \lambda_2, \lambda_3) + p\, f_D(x \mid \boldsymbol{\theta}), & x = y, \end{cases} \tag{3}$$

where $f_D(x \mid \boldsymbol{\theta})$ is the probability function of a discrete distribution $D(x; \boldsymbol{\theta})$ defined on the set $\{0, 1, 2, \ldots\}$ with parameter vector $\boldsymbol{\theta}$. Note that for $p = 0$ we have the simple bivariate Poisson model defined in the previous section. Diagonal inflated models (3) can be fitted using the EM algorithm provided at the appendix.

Useful choices for $D(x; \boldsymbol{\theta})$ can be the Poisson, the geometric or simple discrete distributions denoted by *Discrete*(J). The geometric distribution might be of great interest since it has mode at zero and decays quickly as one moves away from zero. As *Discrete*(J) we consider the distribution with probability function

$$f(x|\boldsymbol{\theta}, J) = \begin{cases} \theta_x & \text{for } x = 0, 1, \ldots, J \\ 0 & \text{for } x \neq 0, 1, \ldots, J \end{cases} \tag{4}$$

where $\sum_{x=0}^{J} \theta_x = 1$. If $J = 0$ then we end up with the zero-inflated model.

Two are the most important and distinctive properties of such models. Firstly, the marginal distributions of a diagonal inflated model are not Poisson distributions, but mixtures of distributions with one Poisson component. Namely the marginal for X is given by

$$f_{IBP}(x) = (1-p)\, f_{Po}(x \mid \lambda_1 + \lambda_3) + p\, f_D(x \mid \boldsymbol{\theta}), \tag{5}$$

where $f_{Po}(x \mid \lambda)$ is the probability function of the Poisson distribution with parameter λ. This can be easily recognized as a 2-finite mixture distribution with two components, the one having a Poisson distribution and the other a f_D distribution. For example, if we consider a geometric inflation then the resulting marginal distribution is a 2-finite mixture distribution with one Poisson and one geometric component. Thus the marginal mean is given by

$$\mathsf{E}(X) = (1-p)\,(\lambda_1 + \lambda_3) + p\, \mathsf{E}_D(X)$$

where $\mathsf{E}_D(X)$ denotes the expectation of the distribution $D(x; \boldsymbol{\theta})$. The variance is much more complicated and is given by

$$\mathsf{VAR}(X) = (1-p)\left\{(\lambda_1+\lambda_3)^2 + (\lambda_1+\lambda_3)\right\} + p\,\mathsf{E}_D(X^2) - \{(1-p)\,(\lambda_1+\lambda_3) + p\,\mathsf{E}_D(X)\}^2 \,.$$

Since the marginals are not Poisson distributions, they can be either under-dispersed or over-dispersed depending on the choices of $D(x; \boldsymbol{\theta})$. For example, if $D(x; \boldsymbol{\theta})$ is a degenerate at one (that is, *Discrete*(1) with $\boldsymbol{\theta}^\top = (0, 1)$) implying inflation only on the $(1, 1)$ cell, then, for $\lambda_1 + \lambda_3 = 1$ and $p = 0.5$, the resulting distribution is under-dispersed (variance equal to 0.5 and mean equal to 1). On the other hand, if the inflation distribution has positive probability on more points, for example a geometric or a Poisson distribution, the resulting marginal distribution will be over-dispersed. In the simplest case of zero-inflated models, the marginal distributions are also over-dispersed relative to the simple Poisson distribution.

Another important characteristic is that, even if $\lambda_3 = 0$ (double Poisson distribution), the resulting inflated distribution introduces a degree of dependence between the two variables under consideration. In general, the simple bivariate Poisson models has $E_{BP}(XY) = \lambda_3 + (\lambda_1+\lambda_3)(\lambda_2+\lambda_3)$. Thus for an inflated model we obtain

$$\begin{aligned}\mathsf{COV}_{IBP}(X,Y) &= (1-p)\,\{\lambda_3+(\lambda_1+\lambda_3)(\lambda_2+\lambda_3)\} + p\,\mathsf{E}_D(X^2)\\ &\quad -(1-p)^2(\lambda_1+\lambda_3)(\lambda_2+\lambda_3)\\ &\quad -(1-p)\,p\,\mathsf{E}_D(X)(\lambda_1+\lambda_2+2\lambda_3) - p^2\{\mathsf{E}_D(X)\}^2.\end{aligned}$$

The above formulas are a generalization of simpler case where inflation is imposed only on the $(0,0)$ cell given by Wang *et al.* (2003). If the data before introducing inflation are independent, that is if $\lambda_3 = 0$, the covariance is given by

$$\mathsf{COV}_{IBP}(X,Y) = p\,(1-p)\,\lambda_1\lambda_2 + p\,\mathsf{E}_D(X^2) - p\,(1-p)\,\mathsf{E}_D(X)(\lambda_1+\lambda_2) - p^2\{\mathsf{E}_D(X)\}^2$$

which implies non-zero correlation between X and Y. Note that for certain combinations of $D(x;\boldsymbol{\theta})$, the covariance can be negative as well. For example, if $p = 0.5$, $\lambda_1 = 0.5$, $\lambda_2 = 2$ and the inflation is a degenerate at one distribution then the covariance equals -0.125. When inflation is added only on the cell $(0,0)$, we obtain that $\mathsf{E}_D(X) = \mathsf{E}_D(X^2) = 0$ and $\mathsf{COV}_{IBP}(X,Y) = p(1-p)\lambda_1\lambda_2$ which is always positive. For this reason, diagonal inflation can possibly correct both over/under-dispersion and correlation problems encountered in modeling count data.

3. R functions for bivariate Poisson models

3.1. Short description of functions and installation

In order to run in R the EM algorithm for the bivariate Poisson models presented in the previous sections, you have to install the **bivpois** package which is available from Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/` or from the authors' web page at `http://www.stat-athens.aueb.gr/~jbn/papers/paper14.htm`.

The following functions are available for direct use in R:

`bivpois.table` Bivariate Poisson probability function (in tabular form) using recursive relationships.

`pbivpois` Probability function (and its logarithm) of bivariate Poisson.

`simple.bp` EM for fitting a simple bivariate Poisson model with constant λ_1, λ_2 and λ_3 (no covariates are used).

`lm.bp` EM for fitting a general linear bivariate Poisson model with covariates on λ_1, λ_2 and λ_3.

`lm.dibp` EM for fitting a diagonal inflated bivariate Poisson model with covariates on λ_1, λ_2 and λ_3.

Two additional functions (`newnamesbeta` and `splitbeta`) are used internally in `lm.bp` and `lm.dibp` in order to identify which parameters are associated to λ_1 and λ_2.

Finally the following four datasets (used for illustration in Section 4) are also included in **bivpois** package:

`ex1.sim` Simulated data of example one.

`ex2.sim` Simulated data of example two.

`ex3.health` Health care data from the book of Cameron and Trivedi (1998) used as example three.

`ex4.ita91` Football (soccer) data of the Italian League for season 1991-92. Data were originally analysed in Karlis and Ntzoufras (2003) and used in this paper as example 4.

All the above datasets can be loaded (after attaching the **bivpois** package) using the command `data`; for example `data("ex1.sim")` loads the first dataset included in the package.

Finally demos are available for all examples. All demos illustrate and recall the commands used in the examples of this paper. It can help the user to reproduce exactly the same results as in section 4. Five demos are available: `main`, `ex1`, `ex2`, `ex3` and `ex4`. The first one just produces a menu which prompts you to select the example you wish to use while the rest reproduce directly the results of each example. Each demo can be called using the command `demo(avdemo, package="bivpois")`; where `avdemo` is one of the available demos described above. For example `demo("ex1", package="bivpois")` will provide the demo of the first example.

3.2. The function `pbivpois`

Function `pbivpois` evaluates the probability function (or its logarithm) of $BP(\lambda_1, \lambda_2, \lambda_3)$ for x and y values. The function can be called using the following syntax:

```
pbivpois(x, y = NULL, lambda = c(1, 1, 1), log = FALSE)
```

Required Arguments

`x` Matrix or vector containing the data. If this argument is a matrix then `pbivpois` evaluates the distribution function of the bivariate Poisson for all the pairs provided by the first two columns of `x`. Additional columns of `x` and `y` are ignored.

Optional Arguments

`y` Vector containing the data for the second value of each pair (x_i, y_i) for which we calculate the distribution function of Bivariate Poisson. It is used only if `x` is also a vector. Vectors `x` and `y` should be of equal length.

`lambda` Vector of length three containing values of the parameters λ_1, λ_2 and λ_3 of the bivariate Poisson distribution.

`log` Logical argument controlling the calculation of the logarithm of the probability or the probability function itself. If the argument is not used, the function returns the probability value of (x_i, y_i) since the default value is `FALSE`.

3.3. The function `simple.bp`

Function `simple.bp` implements the EM algorithm for fitting the simple bivariate Poisson model of the form $(x_i, y_i) \sim BP(\lambda_1, \lambda_2, \lambda_3)$ for $i = 1, \ldots, n$. It produces a 'list' object which gives various details regarding the fit of such a model. The function can be called using the following syntax:

```
simple.bp(x, y, ini3 = 1.0, maxit = 300, pres = 1e-8)
```

Required Arguments

`x` Matrix or vector containing the data. If this argument is a matrix then `simple.bp` uses as input data the first two columns of `x`. Additional columns of `x` and `y` are ignored.

Optional Arguments

`y` Vector containing the data for the second variable in a bivariate Poisson model. It is used only if `x` is also a vector. Vectors `x` and `y` should be of equal length.

`ini3` Initial value for λ_3.

`maxit` Maximum number of EM steps. EM algorithm stops when the number of iterations exceeds `maxit` and returns as result the values obtained by the last iteration.

`pres` Precision used in log-likelihood improvement. If the relative log-likelihood difference between two subsequent EM steps is lower than `pres` then the algorithm stops. Note that the algorithm stops if one of the arguments `maxit` or `pres` is satisfied.

Output Components

A list object is returned with the following output components:

`lambda` Parameters $\lambda_1, \lambda_2, \lambda_3$ of the model.

`loglikelihood` Log-likelihood of the fitted model. This argument is given in a vector form of length equal to `iterations` with one value per iteration. This vector can be used to monitor the log-likelihood improvement and the convergence of the algorithm.

`parameters` Number of estimated parameters of the fitted model.

`AIC`, `BIC` AIC and BIC values of the fitted model. Values of AIC and BIC are also given for the double Poisson model and the saturated model.

`iterations` Number of iterations of the EM algorithm.

During the execution of the algorithm the following details are printed: the iteration number, λ_1, λ_2, λ_3, the log-likelihood and the relative difference of the log-likelihood. For an illustration of using this function see example 1 in Section 4.1.

3.4. The function `lm.bp`

Function `lm.bp` implements the EM algorithm for fitting the bivariate Poisson regression model (2) with

$$\begin{aligned}(x_i, y_i) &\sim BP(\lambda_{1i}, \lambda_{2i}, \lambda_{3i}) \text{ for } i = 1, \dots, n,\\ \boldsymbol{l}_\kappa &= \boldsymbol{w}_\kappa \boldsymbol{\beta}_\kappa = \beta_{\kappa,0} + \beta_{\kappa,1}\boldsymbol{w}_{\kappa,1} + \cdots + \beta_{\kappa,p_\kappa}\boldsymbol{w}_{\kappa,p_\kappa},\end{aligned} \tag{6}$$

for $\kappa = 1, 2, 3$; where $\boldsymbol{l}_\kappa$ is a $n \times 1$ vector containing the $\log \lambda_\kappa$ for each observation, that is $\boldsymbol{l}_\kappa = (\log \lambda_{\kappa 1}, \dots, \log \lambda_{\kappa n})^\top$; n is the sample size; p_1, p_2 and p_3 are the number of explanatory variables used for each parameter λ_κ; $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ and $\boldsymbol{w}_3$ are the design/data matrices used for each parameter λ_κ; $\boldsymbol{w}_{\kappa,j}$ are $n \times 1$ vectors corresponding to j column of $\boldsymbol{w}_\kappa$ matrix; $\boldsymbol{\beta}_\kappa$ is the vector of length $p_\kappa + 1$ with the model coefficients for λ_κ and $\beta_{\kappa,j}$ is the coefficient which corresponds to j explanatory variable used in the linear predictor of λ_κ (or j column of $\boldsymbol{w}_\kappa$ design matrix).

This function produces a ‘list’ object which gives various details regarding the fit of the estimated model. The function can be called using the following syntax:

```
lm.bp(l1, l2, l1l2 = NULL, l3 = ~1, data, common.intercept = FALSE,
      zeroL3 = FALSE, maxit = 300, pres = 1e-8, verbose = getOption("verbose"))
```

Required Arguments

`l1` Formula of the form “`x ~ X1 +...+ Xp`” for parameters of $\log \lambda_1$.

`l2` Formula of the form “`y ~ X1 +...+ Xp`” for parameters of $\log \lambda_2$.

`data` data frame containing the variables in the model.

Optional Arguments

`l1l2=NULL` Formula of the form “`~ X1 +...+ Xp`” for the common parameters of $\log \lambda_1$ and $\log \lambda_2$. If the explanatory variable is also found on `l1` and/or `l2` then a model using interaction type parameters is fitted (one parameter common for both predictors [main effect] and differences from this for the other predictor [interaction type effect]). Special terms of the form “`c(X1,X2)`” can be also used here. These terms imply common parameters on λ_1 and λ_2 for different variables. For example if `c(x1,x2)` is used then use the same beta for the effect of x_1 on $\log \lambda_1$ and the effect of x_2 on $\log \lambda_2$. For details see example 4.

`l3 = ~ 1` Formula of the form “`~ X1 +...+ Xp`” for the parameters of $\log \lambda_3$.

`common.intercept=FALSE` Logical function specifying whether a common intercept on $\log \lambda_1$ and $\log \lambda_2$ should be used.

`zeroL3=FALSE` Logical argument controlling whether λ_3 should be set equal to zero (therefore whether a double Poisson model is fitted).

`maxit=300` Maximum number of EM steps.

`pres=1e-8` Precision used to terminate the EM algorithm. The algorithm stops if the relative log-likelihood difference is lower than the value of `pres`.

`verbose=getOption("verbose")` Logical argument for controlling the printing details during the iterations of the EM algorithm. Default value is taken equal to the value of `options()$verbose`. If `verbose=FALSE` then only the iteration number, the log-likelihood and its relative difference from the previous iteration are printed. If `print.details=TRUE` then the model parameters for λ_1, λ_2 and λ_3 are additionally printed.

Output Components

A list object is returned with the following output components:

`coefficients` Vector $\boldsymbol{\beta}$ containing the estimates of the model parameters. When a factor is used then its default set of contrasts is used.

`fitted.values` Data frame of size $n \times 2$ containing the fitted values for the two responses x and y. For the bivariate Poisson model are simply given as $\lambda_1 + \lambda_3$ and $\lambda_2 + \lambda_3$ respectively.

`residuals` Data frame of size $n \times 2$ containing the residual values for the two responses x and y. The residuals of x and y are given by $x - \hat{x}$ and $y - \hat{y}$; where $\hat{x}$ and $\hat{y}$ are the fitted values for x and y respectively given by the two columns of the component `fitted.values`.

`beta1`, `beta2`, `beta3` Vectors $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_3$ containing the coefficients involved in the linear predictor of λ_1, λ_2 and λ_3 respectively. When `zeroL3=TRUE` then `beta3` is not calculated.

`lambda1`, `lambda2`, `lambda3` Vectors of length n containing the estimated λ_1, λ_2 and λ_3, respectively.

`loglikelihood` Log-likelihood of the fitted model given in a vector form of length equal to `iterations` (one value per iteration).

`parameters` Number of estimated parameters of the fitted model.

`AIC`, `BIC` AIC and BIC values of the model. Values are also given for the saturated model.

`iterations` Number of iterations of the EM algorithm.

`call` Argument providing the exact calling details of the `lm.bp` function.

The resulted object of the `lm.bp` function is a list of class `lm.bp`, `lm` for which for which `coef`, `residuals` and `fitted` methods are provided.

3.5. The function `lm.dibp`

Function `lm.dibp` implements the EM algorithm for fitting the simple diagonal inflated bivariate Poisson model of the form

$$(x_i, y_i) \sim DIBP(\, x_i, y_i \mid \lambda_{1i}, \lambda_{2i}, \lambda_{3i}, p, D(\boldsymbol{\theta})\,) \text{ for } i = 1, \ldots, n,$$

where $\lambda_{\kappa i}$ are specified as in (6), $DIBP(\, x, y \mid \lambda_1, \lambda_2, \lambda_3, p, D(\boldsymbol{\theta})\,)$ is the density of the diagonal inflated bivariate Poisson distribution with $\lambda_1, \lambda_2, \lambda_3$ parameters of the bivariate Poisson component, inflated distribution D with parameter vector $\boldsymbol{\theta}$ and mixing (inflation) proportion p evaluated at (x, y); see also equation (3).

This function produces a 'list' object which gives various details regarding the fit of such a model. The function can be called using the following syntax:

```
lm.dibp(l1, l2, l1l2 = NULL, l3 = ~1, data, common.intercept = FALSE,
        zeroL3 = FALSE, distribution = "discrete", jmax = 2, maxit = 300,
        pres = 1e-08, verbose = getOption("verbose"))
```

Required Arguments

Arguments `l1`, `l2` and `data` are exactly the same as in `lm.bp` function.

Optional Arguments

Arguments `l1l2`, `l3`, `common.intercept`, `zeroL3`, `maxit`, `pres` and `verbose` are the same as in `lm.bp` function.

`distribution="discrete"` Specifies the type of inflated distribution; `"discrete"` $= Discrete(J = jmax)$, `"poisson"` $= Poisson(\theta)$, `"geometric"`$=$ $Geometric(\theta)$.

`jmax=2` Number of parameters used in *Discrete* distribution. This argument is not used when the diagonal is inflated with the geometric or the Poisson distribution.

Output Components

A list object is returned with the following output components:

`coefficients` Vector containing the estimates of the model parameters ($\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_3, p$ and θ).

`fitted.values` Data frame with n rows and 2 columns containing the fitted values for x and y. For the diagonal inflated bivariate Poisson model are given by

$$\begin{array}{lll} \hat{x} = (1-p)(\lambda_1+\lambda_3) & \hat{y} = (1-p)(\lambda_2+\lambda_3) & \text{if } x \neq y \text{ and} \\ \hat{x} = (1-p)(\lambda_1+\lambda_3) + p\,\mathsf{E}_D(X) & \hat{y} = (1-p)(\lambda_2+\lambda_3) + p\,\mathsf{E}_D(X) & \text{if } x = y \end{array}$$

where $\mathsf{E}_D(X)$ is the mean of the distribution used to inflate the diagonal.

Variables `beta1`, `beta2`, `beta3`, `lambda1`, `lambda2`, `lambda3`, `residuals`, `loglikelihood`, `AIC`, `BIC`, `parameters`, `iterations`
are similar as in `lm.bp` function. The `residual` components is given by $x - \hat{x}$ and $y - \hat{y}$; where $\hat{x}$ and $\hat{y}$ are given by the component `fitted.values` and are calculated as described above.

`diagonal.distribution` Label stating which distribution was used for the inflation of the diagonal.

`p` mixing proportion.

`theta` Estimated parameters of the diagonal distribution. If `distribution="discrete"` then the variable is a vector of length `jmax` with θ_j=`theta[j]` for $j = 1, \ldots, jmax$ and $\theta_0 = 1 - \sum_{j=1}^{jmax} \theta_j$; if `distribution="poisson"` then θ is the mean of the Poisson; if `distribution="geometric"` then θ is the success probability of the geometric distribution.

`call` Argument providing the exact calling details of the `lm.dibp` function (same as in `lm.bp`).

The resulted object of the `lm.dibp` function is a list of class `lm.dibp`, `lm` for which for which `coef`, `residuals` and `fitted` methods are provided. For an illustration of using this function see examples in Sections 4.2, 4.3.2 and 4.4.

3.6. More details on formula arguments

In this subsection we provide details regarding the formula arguments used in functions `lm.bp` and `lm.dibp`. The formulas `l1` and `l2` are required in both functions. They are of the form "`x ~ X1 + X2 +...+ Xp`" and "`y ~ X1 + X2 + ...+ Xp`" respectively; where `x` and `y` are the two response variables. Variables used as explanatory variables in `l1` and `l2` specify unique and not equal terms on the linear predictors of λ_1 and λ_2 parameters respectively. Therefore, if `X1` is used as independent variable in `l1` but not in `l2` then X_1 will be used in the linear predictor of λ_1 but not on the linear predictor of λ_2.

Formula arguments `l1l2` and `l3` are not required. The first is used to specify variables that have common effect on the linear predictors of both λ_1 and λ_2. The second argument, `l3`, is used to model the linear predictor of λ_3 which controls the covariance between the two response variables. Both of these formulas have the form "`~ X1 +...+ Xp`", hence no response variable should be provided. Note that if a term is present as an explanatory variable in `l1` and/or `l2` and `l1l2` then non-common effect will be fitted. In more detail, the following combinations can be used as terms:

1. **Effect only on λ_1**: `lm.bp(x ~ z1, y ~ 1 , l1l2 = NULL, ...)`.
2. **Effect only on λ_2**: `lm.bp(x ~ 1 , y ~ z1, l1l2 = NULL, ...)`.
3. **Common effect on both λ_1 and λ_2**: `lm.bp(x ~ 1, y ~ 1 , l1l2 = ~ z1, ...)`.
4. **Different effects on λ_1 and λ_2**: `lm.bp(x ~ z1, y ~ z1, l1l2 = NULL, ...)`. Exactly the same result will be provided if we use the combination `lm.bp(x ~ z1, y ~ z1, l1l2 = ~ z1, ...)`.
5. **Different effect on λ_1 and λ_2** will be also fitted using the combination `lm.bp(x ~ z1, y ~ 1, l1l2 = ~ z1, ...)` with the difference that two parameters will be provided for effect of `z1` on λ_1. The first one will be equal to the effect of `z1` on λ_2 (common effect) and the second will be the difference effect of `z1` on λ_1. Similarly the combination `lm.bp(x ~ 1, y ~ z1, l1l2 = ~ z1, ...)` will provide a common effect for both λ_1 and λ_2 and an additional difference effect of `z1` for λ_2.

A special term can be also used in both `lm.bp` and `lm.dibp` when we wish to specify a common effect on λ_1 and λ_2 for different variables. This can be achieved using terms of type `c(z1,z2)` in the `l1l2` formula. Such as a term results to a common parameter for both λ_1 and λ_2 for the variable z_1 and z_2 respectively. For example, the combination of the following formulas `l1 = x ~ z4, l2 = y ~ z4, l1l2 = ~ c(z1,z2) + z5` will result to the following model

$$\begin{aligned} \log \lambda_1 &= \beta_{1,0} + \beta_{12} z_1 + \beta_{1,4} z_4 + \beta_5 z_5 \\ \log \lambda_2 &= \beta_{2,0} + \beta_{12} z_2 + \beta_{2,4} z_4 + \beta_5 z_5. \end{aligned}$$

The parameters $\beta_{\kappa,j}$ denote the effect of variable z_j on the linear predictor of λ_κ. Parameters denoting the common effect of a variable z_j on both λ_1 and λ_2 can be denoted by β_j (omitting the comma separator and the indicator of λ). Finally, if a parameter denotes the common effect of z_{j_1} and z_{j_2} on λ_1 and λ_2 respectively then it can be denoted by $\beta_{j_1 j_2}$.

Term `c(z1,z2)` will be denoted with the label `z1..z2`. Such kind of terms are useful for fitting the models of Karlis and Ntzoufras (2003) for sports outcomes; see for an illustration in Section 4.4.1. Note that β_{12} is common to both regressions, this is why we did not use the comma separator to show that this is a common coefficients for regressors z_1 and z_2 in the two lines.

Some commonly used models can be specified by the following syntax:

- `lm.bp(x ~ 1, y ~ 1, common.intercept=TRUE, ...)` : Common constant for λ_1 and λ_2 that is $\log \lambda_{1i} = \beta_0$ and $\log \lambda_{2i} = \beta_0$.
- `lm.bp(x ~ 1, y ~ 1, common.intercept=FALSE, ...)` : Constant but not equal λ_1 and λ_2 that is $\log \lambda_{1i} = \beta_{1,0}$ and $\log \lambda_{2i} = \beta_{2,0}$ with $\beta_{1,0} \neq \beta_{2,0}$. This syntax of `lm.bp` fits the same model as `simple.bp` function.
- `lm.bp(x ~ . , y ~ . , common.intercept=FALSE, ... )` : Full model with different parameters for each λ_1 and λ_2.

Operators `+`, `-`, `:`, `*` can be used as in normal formula objects to define additional main effects, interaction terms, or both.

4. Examples

4.1. Example 1: Simulated data

In order to illustrate our algorithm, we have simulated 100 data points (x_i, y_i) from a bivariate Poisson regression model of type (2) with $\lambda_{1i}, \lambda_{2i}, \lambda_{3i}$ given by

$$\begin{aligned} \lambda_{1i} &= \exp(1.8 + 2Z_{1i} - 3Z_{3i}) \\ \lambda_{2i} &= \exp(0.7 - Z_{1i} - 3Z_{3i} + 3Z_{5i}) \\ \lambda_{3i} &= \exp(1.7 + Z_{1i} - Z_{2i} + 2Z_{3i} - 2Z_{4i}) \end{aligned}$$

for $i = 1, \ldots, 100$; where Z_{ki} ($k = 1, \ldots, 5$ and $i = 1, \ldots, 100$) have been generated from a normal distribution with mean zero and standard deviation equal to 0.1 . The sample means were found equal to 11.8 and 7.9 for X and Y respectively. The correlation and the covariance were found equal to 0.623 and 6.75 respectively indicating that a bivariate Poisson model should be fitted.

We have fitted various models presented in Table 1 with their BIC and AIC values. Estimated parameters are presented in Table 2.

Both AIC and BIC indicate that the best fitted model (among the ones we have tried) is model 11 which is the actual model we have used to generate our data. Using asymptotic χ^2 statistics based on the log-likelihood, we may also test the significance of specific parameters and identify which model should be selected.

All commands used in the illustration of example one (which follows) can be called using the command `demo("ex1", package="bivpois")`.

Simple (no regressors) bivariate Poisson model

In order to fit the simple bivariate Poisson model and store the results in an object called `ex1.simple` we firstly load the **bivpois** package and the data of the first example using the following commands:

```
  library("bivpois")
  data("ex1.sim")
```

and then fit the model by:

```
  ex1.simple <- simple.bp(ex1.sim$x, ex1.sim$y)
```

where `x, y` are vectors of length 100 containing our data included in `ex1.sim` data frame. If we wish to monitor the calculated arguments then we type `names(ex1.simple)` resulting to:

```
[1] "lambda"        "loglikelihood" "parameters"    "AIC" "BIC"
[6] "iterations"
```

We may further monitor any of the above values by typing the name of the stored object (here `ex1.simple` followed by the dollar character (`$`) and the name of the output component we wish to monitor. For example `ex1.simple$lambda` will print the values of λ_1, λ_2 and λ_3:

```
> ex1.simple$lambda

[1] 6.574695 2.894695 5.125305
```

Similarly the command `ex1.simple$BIC` returns:

```
> ex1.simple$BIC

Saturated   DblPois   BivPois
 1869.759  1091.842  1049.359
```

From the above output, the BIC of our fitted model is equal to 1049.36 . For comparison, the values of the simple double Poisson model and the saturated model are also given (1091.84 ad 1869.76 respectively). As saturated model we consider the double Poisson model with perfect fit, that is the expected values are equal to the data. Here BIC indicates that the bivariate Poisson model is better than both the simple double Poisson model and the saturated one. Finally, all variables of `ex1.simple` can be printed by simple typing its name:

```
> ex1.simple

$lambda [1] 6.574695 2.894695 5.125305

$loglikelihood
  [1] -532.8476 -532.4675 -532.0811 -531.6893 -531.2928 -530.8926 -530.4895
 ...    .......   ........  .......   .......   .......   .......   .......
[162] -516.7321 -516.7321 -516.7321

$parameters
[1] 3

$AIC Saturated   DblPois   BivPois
 1210.095  1085.246  1039.464

$BIC Saturated   DblPois   BivPois
 1869.759  1091.842  1049.359

$iterations
[1] 164
```

We may monitor the evolution of the log-likelihood by producing Figure 1 by typing

```
plot(1:ex1.simple$iterations, ex1.simple$loglikelihood, xlab="Iterations",
     ylab="Log-likelihood", type="l")
```

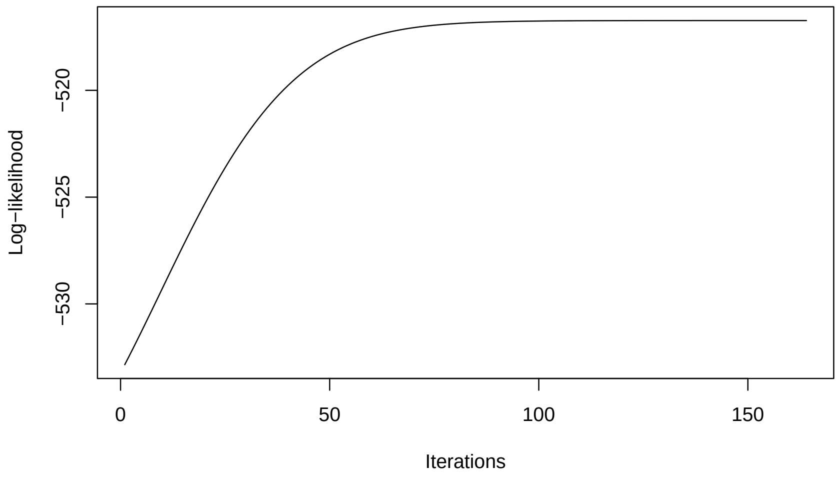

Figure 1: Log-likelihood evolution for the simple bivariate Poisson model fitted on data of the simulated example 1.

Bivariate Poisson regression models

In this section, we illustrate how we can fit models 2 – 11 of Table 1 for the data of the simulated example 1. All data were stored in a data frame called `ex1.sim`. Response variables were stored in vectors `x` and `y` while the explanatory variables in vector `z1, z2, z3, z4` and `z5` within the above data frame.

Firstly, the simple model fitted in the previous section using the `simple.bp` function can be also fitted using the `lm.bp` function (see model `ex1.m3` in the code given below). The major difference is that the second function provides a wider variety of output components. The code for fitting models 2 – 11 presented in Table 1 is given below:

```
ex1.m2 <- lm.bp(x~1, y~1, data=ex1.sim, zeroL3=TRUE)
ex1.m3 <- lm.bp(x~1, y~1, data=ex1.sim)
ex1.m4 <- lm.bp(x~., y~.,data=ex1.sim, zeroL3=TRUE)
ex1.m5 <- lm.bp(x~., y~., data=ex1.sim)
ex1.m6 <- lm.bp(x~z1, y~z1+z5, l1l2=~z3, data=ex1.sim, zeroL3=TRUE)
ex1.m7 <- lm.bp(x~z1, y~z1+z5, l1l2=~z3, data=ex1.sim)
ex1.m8 <- lm.bp(x~., y~., l3=~., data=ex1.sim)
ex1.m9 <- lm.bp(x~.,y~., l3=~.-z5, data=ex1.sim)
ex1.m10 <- lm.bp(x~z1, y~z1+z5, l1l2=~z3, l3=~., data=ex1.sim)
ex1.m11 <- lm.bp(x~z1, y~z1+z5,l1l2=~z3, l3=~.-z5, data=ex1.sim)
```

The estimates for model 11 are given by

```
> ex1.m11$coef  # monitor all beta parameters of model 11

(l1):(Intercept)          (l1):z1          (l1):z3 (l2):(Intercept)
       1.7469235        2.6168303       -2.4265136        0.6145637
         (l2):z1          (l2):z3          (l2):z5 (l3):(Intercept)
       0.4946940       -2.4265136        3.7776722        1.7163345
         (l3):z1          (l3):z2          (l3):z3          (l3):z4
       0.1471195       -1.2234633        1.9514209       -2.5064762
```

In the above results, within parentheses the parameter λ_i is indicated for which each parameter is referred. Separate estimates for the effects on λ_1, λ_2 and λ_3 can be obtained using the commands `ex1.m11$beta1`, `ex1.m11$beta2` and `ex1.m11$beta3`, respectively.

From the above results, the model can be summarized by the following equation

$$
\begin{aligned}
\log(\lambda_{1i}) &= 1.75 + 2.62 Z_{1i} - 2.43 Z_{3i} \\
\log(\lambda_{2i}) &= 0.61 + 0.50 Z_{1i} - 2.43 Z_{3i} + 3.78 Z_{5i} \\
\log(\lambda_{3i}) &= 1.72 + 0.15 Z_{1i} - 1.22 Z_{2i} + 1.95 Z_{3i} - 2.51 Z_{4i}
\end{aligned}
$$

4.2. Example 2: Simulated data

In this second example we have considered the data of example 1 contaminated in the diagonal (for values of $x = y$) with values generated from a *Poisson*(2) distribution and mixing

		Model Details						
		λ_1	λ_2	λ_3	Par.	Log-lik.	AIC	BIC
1	DP	Saturated			200	-405.05	1210.10	1869.76
2	DP	Constant	Constant	$-$	2	-540.62	1085.25	1091.84
3	BP	Constant	Constant	Constant	3	-516.73	1039.46	1049.36
4	DP	Full	Full	$-$	12	-494.98	1013.96	1053.54
5	BP	Full	Full	Constant	13	-478.25	982.50	1025.38
6	DP*	$Z_1 + Z_3$	$Z_1 + Z_3 + Z_5$	$-$	6	-527.05	1066.10	1085.89
7	BP*	$Z_1 + Z_3$	$Z_1 + Z_3 + Z_5$	Constant	7	-500.41	1014.82	1037.90
8	BP	Full	Full	Full	18	-471.51	979.03	1038.40
9	BP	Full	Full	$Z_1 + Z_2 + Z_3 + Z_4$	17	-472.51	979.02	1035.09
10	BP*	$Z_1 + Z_3$	$Z_1 + Z_3 + Z_5$	Full	12	-476.50	976.99	1016.57
11	BP*	$Z_1 + Z_3$	$Z_1 + Z_3 + Z_5$	$Z_1 + Z_2 + Z_3 + Z_4$	11	-476.81	975.61	1011.89

Table 1: Details for fitted models of the simulated example 1 (constant terms are included in all models; *Par.*: number of parameters; *Log-Lik.*: log-likelihood; *Constant*: no covariates were used; *Full*: all covariates Z_k, $k = 1, \ldots, 5$ were used; (*): parameter of Z_3 is common for both λ_1 and λ_2); ($-$): the parameter is set equal to zero.

	Actual	3	4	5	6	7	8	9	10	11
λ_1		6.55								
Constant	1.80	1.88	2.46	1.87	2.46	1.71	1.75	1.73	1.75	1.75
Z_1	2.00		1.42	2.65	1.23	3.10	2.64	2.73	2.65	2.62
Z_2	0.00		-0.57	-0.83			-0.56	-0.59		
Z_3	-3.00		-0.25	-0.83	0.18	-1.40	-2.33	-2.45	-2.43	-2.43
Z_4	0.00		-1.15	-1.70			-0.46	-0.81		
Z_5	0.00		0.02	0.15			1.14	0.52		
λ_2		2.89								
Constant	0.70	1.06	2.04	0.82	2.07	0.45	0.49	0.45	0.59	0.61
Z_1	-1.00		0.20	0.70	0.46	2.47	0.36	0.33	0.57	0.49
Z_2	0.00		-1.11	-2.82			-2.83	-2.99		
Z_3	-3.00		0.90	1.68	0.18	-1.40	-1.77	-1.98	-2.43	-2.43
Z_4	0.00		-2.03	-5.27			-2.62	-3.91		
Z_5	3.00		0.65	2.11	1.03	5.10	6.25	4.58	4.41	3.78
λ_3		5.10	0.00	5.10	0.00	6.17				
Constant	1.70	1.63	$-\infty^*$	1.63	$-\infty^*$	1.82	1.72	1.75	1.72	1.72
Z_1	1.00						0.19	0.10	0.17	0.15
Z_2	-1.00						-0.56	-0.58	-1.19	-1.22
Z_3	2.00						1.83	1.85	1.94	1.95
Z_4	-2.00						-1.87	-1.37	-2.58	-2.51
Z_5	0.00						-0.96		-0.45	

Table 2: Estimated parameters for fitted models of the simulated example 1 (models 6, 7, 10, 11: parameter of Z_3 is common for both λ_1 and λ_2; blank cells correspond to zero coefficients (the corresponding covariate was not used); (*): corresponds to $\lambda_3 = 0$).

Model Details			Sim.Example 2				Mix.Prop.
		Diagonal distribution	Par.	Log-Like	AIC	BIC	(p)
1	BP	No Diagonal Inflation	11	-555.47	1132.94	1169.22	0.000
2	DIBP	*Discrete*(0)	12	-551.26	1126.53	1166.11	0.019
3	DIBP	*Discrete*(1)	13	-513.50	1053.01	1095.89	0.100
4	DIBP	*Discrete*(2)	14	-495.93	1019.86	1066.03	0.139
5	DIBP	*Discrete*(3)	15	-472.23	974.46	1023.93	0.198
6	DIBP	*Discrete*(4)	16	-462.48	956.96	1009.74	0.237
7	DIBP	*Discrete*(5)	17	-458.06	950.11	1006.19	0.265
8	DIBP	*Discrete*(6)	18	-458.06	952.11	1011.48	0.265
9	DIBP	*Poisson*	13	-460.00	945.99	988.87	0.268
10	DIBP	*Geometric*	13	-465.69	957.39	1000.27	0.274

Table 3: Details for fitted models of the simulated example 2 (*Par.*: number of parameters; *Log-like*: log-likelihood; *Mix.Prop.*: mixing proportion).

proportion equal to 0.30. The contamination was completed by generating a binary vector $\boldsymbol{\gamma}$ (of length 100) with success probability 0.30 and a Poisson vector $\boldsymbol{d}$ (of length 100) with mean equal to 2. The new data (x'_i, y'_i) were constructed by setting $x'_i = x_i(1-\gamma_i)+\gamma_i d_i$ and $y'_i = y_i(1-\gamma_i)+\gamma_i d_i$ for $i = 1, \ldots, n$. Finally, 27 observations were contaminated with sample mean equal to 2.4. This means that 27 observations of the data generated in example 1 were randomly substituted by equal X and Y values generated from a Poisson distribution with mean equal to 2. Data are available using the command `data("ex2.sim")` after loading the **bivpois** package.

To illustrate our method we have implemented diagonal inflated models on both data of simulated example 1 and 2. For the data of the previous section no improvement was evident (estimated mixing proportion for all models was found equal to zero). For the data of example 2, both BIC and AIC values indicate the Poisson distribution is the most suitable for the diagonal inflation. Moreover, for the discrete distribution, we need to set at least $J = 4$ in order to get values of BIC lower than the corresponding values of the bivariate Poisson model with no inflation (see Table 3). Estimated parameters for the diagonal inflated model with the best discrete distribution, Poisson and geometric distributions are provided in Table 4. In all models we have used the actual underlying covariate set-up as given for model 11 in Section 4.1.

The code of example two which follows can be called using the command `demo("ex2", package="bivpois")`.

Diagonal inflated bivariate Poisson regression models

Here we illustrate how we can fit diagonal inflated models presented in Table 3 using `lm.dibp` function. The following commands have been used to fit each model

```
ex2.m1 <- lm.bp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim)
ex2.m2 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim, jmax=0)
ex2.m3 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim, jmax=1)
ex2.m4 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim, jmax=2)
```

	λ_1			λ_2				λ_3				
Model	Const.	Z_1	Z_3	Const.	Z_1	Z_3	Z_5	Const.	Z_1	Z_2	Z_3	Z_4
1	1.34	3.25	-2.92	-0.09	1.41	-2.92	5.87	1.63	-0.24	-0.92	0.86	-2.27
7	1.78	2.43	-2.06	0.67	-0.39	-2.06	3.03	1.67	0.17	-0.92	1.92	-3.18
9	1.78	2.42	-2.02	0.68	-0.35	-2.02	2.97	1.67	0.13	-0.96	1.91	-3.25
10	1.78	2.38	-1.96	0.67	-0.33	-1.96	3.15	1.67	0.15	-1.01	1.82	-3.35

Model	p	θ
7	0.265	$\hat{\boldsymbol{\theta}} = (0.08, 0.30, 0.15, 0.22, 0.15, 0.10)$
9	0.268	$\hat{\theta} = 2.43$ (Poisson parameter)
10	0.274	$\hat{\theta} = 0.27$ (geometric parameter)

Table 4: Estimated parameters for fitted models of the simulated example 2 (Parameter vector for model 7 is given by $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^\top$).

```
ex2.m5 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim, jmax=3)
ex2.m6 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim, jmax=4)
ex2.m7 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim, jmax=5)
ex2.m8 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim, jmax=6)
ex2.m9 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim,
                 distribution="poisson")
ex2.m10 <- lm.dibp(x~z1, y~z1+z5, l1l2=~z3, l3=~.-z5, data=ex2.sim,
                 distribution="geometric")
```

For model 7 we have the following estimates

```
# # printing parameters of model 7
> ex2.m7$beta1

(Intercept)          z1          z3
   1.779195    2.432678   -2.055471

> ex2.m7$beta2

(Intercept)          z1          z3          z5
  0.6694631  -0.3865983  -2.0554712   3.0293632

> ex2.m7$beta3

(Intercept)          z1          z2          z3          z4
  1.6685754   0.1668765  -0.9175397   1.9242793  -3.1779283

> ex2.m7$p

[1] 0.2648003

> ex2.m7$theta
```

```
[1] 0.3018780 0.1494183 0.2237955 0.1469629 0.1026404
```

The above results can be summarized by the following model

$$\begin{aligned}
f_{IBP}(x_i, y_i) &= \begin{cases} 0.735 f_{BP}(x_i, y_i \mid \lambda_{1i}, \lambda_{2i}, \lambda_{3i}), & x_i \neq y_i \\ 0.735 f_{BP}(x_i, y_i \mid \lambda_{1i}, \lambda_{2i}, \lambda_{3i}) + 0.265\theta_{x_i}, & x_i = y_i, \end{cases} \\
(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) &= (0.302, 0.149, 0.224, 0.147, 0.103) \\
\theta_j &= 0 \text{ for } j > 5 \\
\theta_0 &= 1 - \sum_{j=1}^{\infty} \theta_j = 0.075 \\
\log(\lambda_{1i}) &= 1.78 + 2.43 Z_{1i} - 2.06 Z_{3i} \\
\log(\lambda_{2i}) &= 0.67 - 0.39 Z_{1i} - 2.06 Z_{3i} + 3.03 Z_{5i} \\
\log(\lambda_{3i}) &= 1.67 + 0.17 Z_{1i} - 0.92 Z_{2i} + 1.92 Z_{3i} - 3.18 Z_{4i}.
\end{aligned}$$

Similarly model 9 produces the following results

```
> ex2.m9$beta1

(Intercept)          z1          z3
   1.783636    2.421730   -2.019664

> ex2.m9$beta2

(Intercept)          z1          z3          z5
  0.6796153  -0.3503677  -2.0196635   2.9746267

> ex2.m9$beta3

(Intercept)          z1          z2          z3          z4
  1.6674937   0.1250575  -0.9612863   1.9131331  -3.2475727

> ex2.m9$p

[1] 0.2677972

> ex2.m9$theta

[1] 2.433654
```

which can be summarized by the following model

$$\begin{aligned}
f_{IBP}(x_i, y_i) &= \begin{cases} 0.732 f_{BP}(x_i, y_i \mid \lambda_{1i}, \lambda_{2i}, \lambda_{3i}), & x_i \neq y_i \\ 0.732 f_{BP}(x_i, y_i \mid \lambda_{1i}, \lambda_{2i}, \lambda_{3i}) + 0.268 e^{-2.434} \frac{(2.434)^{x_i}}{x_i!}, & x_i = y_i, \end{cases} \\
\log(\lambda_{1i}) &= 1.78 + 2.42 Z_{1i} - 2.02 Z_{3i} \\
\log(\lambda_{2i}) &= 0.68 - 0.35 Z_{1i} - 2.02 Z_{3i} + 2.97 Z_{5i} \\
\log(\lambda_{3i}) &= 1.67 + 0.12 Z_{1i} - 0.96 Z_{2i} + 1.91 Z_{3i} - 3.25 Z_{4i}.
\end{aligned}$$

4.3. Example 3: Health care data

In this example, we have fitted bivariate Poisson models on data concerning the demand for Health Care in Australia, reported by Cameron and Trivedi (1998). The data refer to the Australian Health survey for 1977-1978. The sample size is quite large ($n = 5190$) although they are only a subsample of the collected data. We will use two variables, namely the number of consultations with a doctor or a specialist (X) and the total number of prescribed medications used in past 2 days (Y) as the responses; see Table 5 for a cross-tabulation of the data. The data are correlated (Pearson correlation equal to 0.308) indicating that a bivariate Poisson model should be used. It is also interesting to examine the effect of the correlation in the estimates.

Three variables have been used as covariates: namely the gender (1 female, 0 male), the age in years divided by 100 (measured as midpoints of age groups) and the annual income in Australian dollars divided by 1000 (measured as midpoint of coded ranges). More details on the data and the study can be found in Cameron and Trivedi (1998).

Three competing models were fitted to the data: a) a model with constant covariance term (no covariates on λ_3), b) a model with covariates on the covariance term λ_3 (only gender was used which induces different covariance for each gender) and c) a model without any covariance (double Poisson model); detailed results are given at Table 6.

Standard errors for the bivariate regression models have been calculated using 200 bootstrap replications. This is easily implemented since, in this case, the convergence of the algorithm is fast due to the use of good initial values.

Comparing models (a) and (c) we can conclude that the covariance term is significant (p-value< 0.01). The effects of all covariates are statistically significant using asymptotic t-tests. Furthermore, the effect of gender on the covariance term is significant. Similar results can be obtained using AIC and BIC values.

Let us now examine the estimated parameters. Concerning models (a) and (c), we observe that covariate effects for the two models are quite different. This can be attributed to the covariance λ_3 which is present. Using a bivariate Poisson model we take into account the covariance

Number of Doctor	Number of Prescribed medications (Y)								
Consultations (X)	0	1	2	3	4	5	6	7	8
0	2789	726	307	171	76	32	16	15	9
1	224	212	149	85	50	35	13	5	9
2	49	34	38	11	23	7	5	3	4
3	8	10	6	2	1	1	2	0	0
4	8	8	2	2	3	1	0	0	0
5	3	3	2	0	1	0	0	0	0
6	2	0	1	3	1	2	2	0	1
7	1	0	3	2	1	2	1	0	2
8	1	1	1	0	1	0	1	0	0
9	0	0	0	0	0	0	0	0	1

Table 5: Cross-tabulation of data from the Australian health survey (Cameron and Trivedi 1998).

		Model (a)		Model (b)		Model (c)	
		constant λ_3		covariates on λ_3		$\lambda_3 = 0$	
	Covariate	Coef.	St.Er.	Coef.	St.Er.	Coef.	St.Er.
λ_1	Constant	-2.11	0.13	-2.08	0.13	-1.71	0.09
	Gender (female)	0.22	0.08	0.05	0.08	0.22	0.06
	Age	1.37	0.18	1.44	0.19	1.24	0.13
	Income	-0.34	0.11	-0.33	0.10	-0.28	0.08
λ_2	Constant	-2.19	0.08	-2.19	0.08	-1.87	0.07
	Gender (female)	0.63	0.04	0.58	0.04	0.58	0.04
	Age	3.25	0.10	3.29	0.10	2.96	0.09
	Income	-0.12	0.06	-0.11	0.06	-0.13	0.05
λ_3		0.0922	0.0064			0.00	
	constant	-2.38	0.12	-2.72	0.11		
	Gender (female)			0.69	0.14		
	Parameters		9		10		8
	Log-likelihood		-10030.26		-10015.48		-10233.21
	AIC		20078.51		20050.96		20482.41
	BIC		20143.74		20123.44		20540.39

Table 6: Results from the fitted bivariate Poisson models for the data of example 3.

between the two variables and hence the effect of each variable on the other including the effect of the covariates. This may indicates that a Double Poisson would estimate incorrectly the true effect of each covariate on the marginal mean.

When comparing models (a) and (b), the covariate 'gender' in the covariance parameter is significant indicating that males and females have different covariance term. Note that, the gender effect on λ_1 (mean of variable X: number of doctor consultations) has changed dramatically, while this is not true for the rest parameters. This is due to the fact that the marginal mean for X is now $\lambda_1 + \lambda_3$ (instead of λ_1) and, since they share the same variate (gender), we observe different estimates concerning the gender effect on the number of doctor consultations. A plausible explanation might be that gender influences the number of doctor consultations mainly through the covariance term. A final important comment is that the gender effect in the bivariate Poisson model is no longer multiplicative on the mean (additive on the logarithm) since the marginal mean is equal to $\lambda_1 + \lambda_3$.

Bivariate Poisson regression models

The data were downloaded from the web page of the book of Cameron and Trivedi (1998, `http://www.econ.ucdavis.edu/faculty/cameron/racd/racddata.html`). The dataset is provided with the **bivpois** package and can be restored using the command `data("ex3.health")`. The available variables are the following:

```
> names(ex3.health)

 [1] "sex"      "age"      "agesq"    "income"   "levyplus" "freepoor"
 [7] "freepera" "illness"  "actdays"  "hscore"   "chcond1"  "chcond2"
```

```
[13] "doctorco" "nondocco" "hospadmi" "hospdays" "medicine""prescrib"
[19] "nonpresc" "constant"
```

Variables `doctorco` and `prescrib` represent the two response vectors (number of doctor consultations and the total number of prescribed medications used in past 2 days) while the variables `sex, age` and `income` are the regressors used. Variable `sex` here is used as a 0-1 dummy variable. We fit model (a), (b) and (c) using the following commands:

```
library("bivpois")
data("ex3.health")
# Bivariate Poisson models
ex3.model.a <- lm.bp(doctorco~sex+age+income, prescrib~sex+age+income,
                    data=ex3.health)
ex3.model.b <- lm.bp(doctorco~sex+age+income, prescrib~sex+age+income,
                    l3=~sex,data=ex3.health)
# Double Poisson model
ex3.model.c <- lm.bp(doctorco~sex+age+income, prescrib~sex+age+income,
                    data=ex3.health,zeroL3=TRUE)
```

The objects `ex3.model.a, ex3.model.b, ex3.model.c` contain all the results for models (a), (b) and (c), respectively. For the best fitted model (b) we can monitor the estimated parameters using the command `ex3.model.b$beta` resulting to

```
> ex3.model.b$coef

(l1):(Intercept)        (l1):age     (l1):income        (l1):sex  (l2):(Intercept)
     -2.08426577      1.44174252     -0.33472529      0.05461013       -2.18707854
        (l2):age     (l2):income        (l2):sex (l3):(Intercept)         (l3):sex
      3.29130692     -0.11112314      0.57959417     -2.72484380        0.68927119
```

Bootstrap standard errors can be obtained easily using the following script for model (b) and similarly for the rest models.

```
n <- length(ex3.health$doctorco)
bootrep <- 200
results <- matrix(NA,bootrep,10) for (i in 1:bootrep) {
       bootx1 <- rpois(n,ex3.model.b$lambda1)
       bootx2 <- rpois(n,ex3.model.b$lambda2)
       bootx3 <- rpois(n,ex3.model.b$lambda3)
       bootx <- bootx1+bootx3
       booty <- bootx2+bootx3
#      Model (a)
       testtemp <- lm.bp(doctorco~sex+age+income, prescrib~sex+age+income,
                        data=ex3.health)
#      Model (b)
#      testtemp <- lm.bp(doctorco~sex+age+income, prescrib~sex+age+income,
#                    l3=~sex,data=ex3.health)
#      Double Poisson model
#      testtemp <- lm.bp(doctorco~sex+age+income, prescrib~sex+age+income,
#                    data=ex3.health, zeroL3=TRUE)
       betafound <- c(testtemp$beta,testtemp$beta3)
       results[i,] <- betafound
}
```

The commented part of the code reflects the appropriate changes in order to obtain bootstrap standard errors for the cases of models (b) and (c). At the end matrix `results` contains the bootstrap values of the parameters and thus bootstrap standard errors can be obtained merely by calculating the standard errors of the columns. Note that objects `bootx1, bootx2, bootx3` are used to simulate from the bivariate Poisson model using the trivariate reduction scheme.

Diagonal inflated models

Looking at the entries of Table 5 we can clearly see that the proportion of $(0, 0)$ is quite larger than the other frequencies. Hence it is reasonable to fit a model with diagonal inflation described in Section 2.2. Three diagonal inflated models have been fitted to our data using the same covariates for comparison purposes. As inflation distributions we have used the *Discrete*(2), the Poisson and the geometric distribution. All fitted models led to zero-inflated model since we obtained: $\hat{\theta}_1$ and $\hat{\theta}_2 < 10^{-6}$ for the *Discrete*(2) distribution, $\hat{\theta} < 10^{-6}$ for the Poisson and $\hat{\theta} = 0.9999$ for the geometric model. Therefore, only the estimated parameters of the zero-inflated model are presented as model (a) in Table 7.

Following the above analysis, two additional models have been fitted (models (b) and (c) of Table 7). In model (b) we facilitate gender as a covariate of the covariance term (λ_3) while in model (c) we additionally introduce covariates at the mixing proportion p. The latter can be achieved by using a logit link function for p, namely $logit(p_i) = \boldsymbol{w}_{4i}\boldsymbol{\beta}_4$; where $\boldsymbol{w}_{4i}$ is a vector the values of covariates corresponding to i observation and $\boldsymbol{\beta}_4$ denotes the corresponding vector of coefficients. In order to fit such a model, at the M-step described in the appendix (Section A.3) we replace the estimation of p by fitting a logistic regression model assuming the Bernoulli distribution with v_i's as response variable. Here, as a covariate on the modeling of p, we have used only 'gender' for illustration.

From the results in Table 7, it is obvious that the inflation proportion is quite high ($p > 0.30$) which has large effect on most of the estimated parameters. The model with different mixing proportion for each gender [model (c)] exhibits much better values of the log-likelihood, AIC and BIC. Moreover, females appear to have significantly lower number of $(0, 0)$ cells than males which indicates that the men either avoid to visit their doctor and take any kind of medication or simply do not have the physical need to take such actions. The rest of the parameters are also influenced by introducing gender as a covariate on the mixing proportion, with most evident, the large change on the effect of gender on λ_1.

Assuming a diagonal inflated model, we account for over-dispersion of both variables (visits to a doctor and number of medications) since, according to our selected model, the marginal distributions are zero-inflated Poisson distributions. Hence, we have that the expectation is given by $\mathsf{E}(X_i) = (1 - p_i)(\lambda_{1i} + \lambda_{2i})$. The reduced effect of gender on λ_1 is compensated with the increase in the term $(1 - p_i)$. Concluding, the model helps us to clarify the type of the

		Model (a)		Model (b)		Model (c)	
		constant λ_3		covariates on λ_3		covariates on p	
	Covariate	Coef.	St.Er.	Coef.	St.Er.	Coef.	St.Er.
λ_1	Constant	-1.47	0.13	-1.48	0.13	-0.92	0.10
	Gender (female)	0.11	0.07	0.02	0.08	-0.32	0.08
	Age	1.18	0.19	1.26	0.17	0.86	0.15
	Income	-0.29	0.10	-0.29	0.10	-0.29	0.09
λ_2	Constant	-1.60	0.08	-1.61	0.09	-1.16	0.10
	Gender (female)	0.52	0.04	0.49	0.04	0.20	0.05
	Age	2.96	0.10	3.01	0.13	2.72	0.11
	Income	-0.08	0.06	-0.07	0.06	-0.08	0.06
λ_3		0.09	0.01			0.09	0.01
	constant	-2.46	0.11	-2.71	0.19	-2.45	0.15
	Gender (female)			0.52	0.22		
p		0.32	0.01	0.31	0.01		
	constant	-1.15	0.03	-1.16	0.03	0.27	0.07
	Gender (female)					-1.43	0.10
	Parameters	10		11		11	
	Log-likelihood	-9623.08		-9619.88		-9508.72	
	AIC	19266.15		19261.77		19039.44	
	BIC	19338.63		19341.49		19119.16	

Table 7: Results from the fitted diagonal inflated bivariate Poisson models for the data of example 3; the number of parameters, AIC and BIC refer to the zero-inflated version of models (a), (b) and (c).

effect of each variable in the assumed model. Hence the increase in the marginal mean for X_i for the females is due to the decreased frequency of $(0, 0)$ cell which corresponds to lower rate of visits to the doctor and medication taken.

Diagonal inflated bivariate Poisson regression models

Again the diagonal inflated models were fitted using the functions described in Section 3. The models (a) and (b) were fitted using the following syntax:

```
ex3.dibp.a <- lm.dibp(doctorco~sex+age+income, prescrib~sex+age+income,
                      data=ex3.health, jmax=0)
ex3.dibp.b <- lm.dibp(doctorco~sex+age+income, prescrib~sex+age+income, l3=~sex,
                      data=ex3.health, jmax=0)
```

The argument `form1` is the same as in the previous Section (4.3.1). The above models are DIBP with diagonal inflation the *Discrete*(2) distribution. Since $\hat{\theta}_1 = \hat{\theta}_2 = 0$, it is sensible to fit the reduced zero inflated model using *Discrete*(0) as inflation distribution. This can be done if we add the argument `jmax=0` in the above commands. For model (c) we must intervene slightly in the function `lm.dibp` in order to allow for covariate effects on the mixing proportion p.

4.4. Example 4: Sports data

In this section we briefly present applications of bivariate Poisson models on athletic data. We have used the data of Italian football championship (Serie A) for season 1991-92 presented in Karlis and Ntzoufras (2003) to illustrate how we can handle such data using our algorithms and R functions. The data consist of pairs of counts indicating the number of goals scored by each of the two competing teams. As covariates we have used dummy variables to model the team strength. In modeling outcomes of football games, it has been observed excess of draws and small over-dispersion; see for example in Karlis and Ntzoufras (2000, 2003). Introducing diagonal inflated models we correct for both the over-dispersion and the excess of draws.

We have fitted the same models and reproduced the results presented in Karlis and Ntzoufras (2003). Note a misprint on the number of parameters concerning models 3, 4 and 5 presented in Table 1 of the original paper. The actual number of parameters are 54, 54 and 71 instead of 55, 55 and 72 and hence AIC and BIC measures are slightly different.

```
model params loglike   AIC     BIC
 3      54    -758.91 1625.83  1864.33
 4      54    -755.60 1619.17  1857.67
 5      71    -745.85 1633.70  1947.28
```

The above changes do not affect the selection of the best model which is the diagonal inflated Poisson model with *Discrete*(1) distribution as inflation. Note that λ_3 was found equal to 0.23 which is relatively low but statistically significant (p-value<0.001) and the mixing proportion p was found equal to 0.09 (p-value<0.001). The discrete distribution degenerates at one since $\theta_1 = 1$. This indicates an excess of $1-1$ score which was very popular score in Italian football during that period. Detailed results of this dataset can be found in Karlis and Ntzoufras (2003).

Bivariate Poisson regression models

Here we illustrate how we can use our R functions to fit models implemented by Karlis and Ntzoufras (2003). Data concerning the football data of Italian Serie A league for season 1991-92 are stored in a data frame object called `ex4.ita91` and can loaded using the command `data("ex4.ita91")`. Four variables are included in the dataset: `g1, g2, team1, team2` corresponding to the goals scored by the home and away team and the coded level of the home and the away team respectively. Sample of the data frame is given below:

```
> data("ex4.ita91")
> ex4.ita91

    g1 g2    team1    team2
  2  1  1 Atalanta   Ascoli
  3  2  1 Bari       Ascoli
 .........................
322  2  0 Sampdoria  Verona
323  0  0 Torino     Verona
```

Note that the team levels are given in the following alphabetical order:

```
>levels(ita91[,3])

 [1] "Ascoli"     "Atalanta" "Bari"     "Cagliari"   "Cremonese"
 [6] "Fiorentina" "Foggia"   "Genoa"    "Inter"      "Juventus"
[11] "Lazio"      "Milan"    "Napoli"   "Parma"      "Roma"
[16] "Sampdoria"  "Torino"   "Verona"
```

In this example `team1` indicates the attacking teams for variable `g1` while `team2` indicates the attacking teams for variable `g2`. Similarly `team1` and `team2` indicate the defensive teams for variables `g2` and `g1` respectively. Since we want to estimate common attacking and defensive effects we should use a term of type `c(team1,team2)` for estimating the common attacking parameters and a term of type `c(team2,team1)` for estimating the common defensive parameters. Therefore we fit the models presented in Karlis and Ntzoufras (2003) using the following code:

```
#
# formula for modeling of lambda1 and lambda2
form1 <- ~c(team1,team2)+c(team2,team1)
#
# Model 1: Double Poisson
ex4.m1 <- lm.bp(g1~1, g2~1, l1l2=form1, zeroL3=TRUE, data=ex4.ita91)
#
# Models 2-5: bivariate Poisson models
ex4.m2 <- lm.bp(g1~1,g2~1, l1l2=form1, data=ex4.ita91)
ex4.m3 <- lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1, data=ex4.ita91)
ex4.m4 <- lm.bp(g1~1,g2~1, l1l2=form1, l3=~team2, data=ex4.ita91)
ex4.m5 <- lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1+team2, data=ex4.ita91)
#
# Model 6: Zero Inflated Model
ex4.m6 <- lm.dibp(g1~1,g2~1, l1l2=form1, data=ex4.ita91, jmax=0)
#
# Models 7-11: Diagonal Inflated Bivariate Poisson Models
ex4.m7  <- lm.dibp(g1~1,g2~1, l1l2=form1, data=ex4.ita91, distribution="geometric")
ex4.m8  <- lm.dibp(g1~1,g2~1, l1l2=form1, data=ex4.ita91, jmax=1)
ex4.m9  <- lm.dibp(g1~1,g2~1, l1l2=form1, data=ex4.ita91, jmax=2)
ex4.m10 <- lm.dibp(g1~1,g2~1, l1l2=form1, data=ex4.ita91, jmax=3)
ex4.m11 <- lm.dibp(g1~1,g2~1, l1l2=form1, data=ex4.ita91, distribution="poisson")
#
# Models 12: Diagonal Inflated Double Poisson Model
ex4.m12 <- lm.dibp(g1~1,g2~1, l1l2=form1, data=ex4.ita91, distribution="poisson",
                   zeroL3=TRUE)
```

Parameters of the best fitted diagonal inflated model (given in Table 3 of Karlis and Ntzoufras (2003) follow:

```
> # printing details for the diagonal distribution
> ex4.m8$diagonal.distribution
```

```
[1] "Inflation Distribution: Discrete with J= 1"

> round(ex4.m8$beta1,2)            # model parameters for lambda1

   (Intercept)  team1..team21 team1..team210 team1..team211 team1..team212
         -0.07          -0.64           0.22           0.28           0.84
team1..team213 team1..team214 team1..team215 team1..team216 team1..team217
          0.51          -0.14           0.02           0.10           0.18
 team1..team22  team1..team23  team1..team24  team1..team25  team1..team26
         -0.21          -0.50          -0.21          -0.36           0.29
 team1..team27  team1..team28  team1..team29  team2..team11 team2..team110
          0.57          -0.09          -0.37           0.75          -0.70
team2..team111 team2..team112 team2..team113 team2..team114 team2..team115
          0.21          -1.17           0.19          -0.34          -0.17
team2..team116 team2..team117  team2..team12  team2..team13 team2..team14
         -0.16          -0.86          -0.11           0.33          -0.01
 team2..team15  team2..team16  team2..team17  team2..team18  team2..team19
          0.45           0.28           0.63           0.40          -0.29

> round(ex4.m8$beta2[1]-ex4.m8$beta2[2],2)# estimated home effect

(Intercept)
       0.07

> # estimating the effect for 18th level of attack (team1..team2) [Verona]
> round(-sum(ex4.m8$coef[ 2:18]),2)

[1] -0.51

> # estimating the effect for 18th level of defence(team2..team1) [Verona]
> round(-sum(ex4.m8$coef[19:35]),2)

[1] 0.57

> ex4.m8$beta3     # parameters for lambda3 (here the intercept)

(Intercept)
  -1.466041

> exp(ex4.m8$beta3)# lambda3 (here constant)

(Intercept)
  0.2308375

> ex4.m8$p         # mixing proportion
```

```
[1] 0.09033468

> ex4.m8$theta     # printing theta parameters

[1] 0.9999988
```

In the above fitted model `ex4.m8$beta1` and `ex4.m8$beta2` have the same attacking and defensive parameters. The only difference is in the estimated intercepts. The intercept of λ_2 (`ex4.m8$beta2[1]`) indicates the intercept μ reported by Karlis and Ntzoufras (2003) while the home effect is given by the difference of the two constant terms, hence `ex4.m8$beta1[1] - ex4.m8$beta2[1]`. Moreover, the attacking and defensive parameters are given with the names `team1..team2` and `team2..team1`, respectively. In both cases, the last one or two numbers indicate the factor code. Finally, when the discrete distribution is used to inflate the diagonal (draws) then θ_i for $i = 1, \ldots, jmax$ stands for the probability of $P(X = Y = i)$; the probability of $P(X = Y = 0) = \sum_{i=1}^{jmax} P(X = Y = i) = \sum_{i=1}^{jmax} \theta_i$.

All the above models can be directly fitted using the command `demo("ex4",package="bivpois")`. The full reproduced results of Table 1 of Karlis and Ntzoufras (2003) using the `lm.bp` and `lm.dibp` functions (with precision 10^{-8} and maximum number iterations equal to 300) follows:

```
   params loglike     AIC     BIC
1      36 -771.48 1614.95 1773.95
2      37 -764.95 1603.90 1767.32
3      54 -758.91 1625.83 1864.33
4      54 -755.58 1619.17 1857.67
5      71 -745.85 1633.70 1947.28
6      38 -764.95 1605.90 1773.73
7      39 -764.95 1607.90 1780.15
8      39 -756.57 1591.14 1763.39
9      40 -756.57 1593.14 1769.81
10     41 -756.38 1594.76 1775.85
11     39 -763.56 1605.13 1777.38
12     38 -767.01 1610.02 1777.86
```

Finally, in Table 8 we present the parameters of models 1, 2 and 8 in connection with the R output; see also Table 3 of Karlis and Ntzoufras (2003) and Figure 2 for a graphical representation of defensive versus attacking parameters. In this figure, defensive parameters have multiplied by minus one. Thus, teams with good offensive and defensive performance will be concentrated on the upper right area of the plot.

5. Concluding remarks

In this article we have presented R functions implementing maximum likelihood estimation for bivariate Poisson regression models and their diagonal inflated variations. Diagonal inflated models, also presented here, are useful in cases where excess of combinations of pairs with equal x and y values appear (for example in sports data; see Karlis and Ntzoufras (2003). All functions are based on EM algorithms constructed for such models; see appendix for details.

			m_1	m_2	m_8
	R name or syntax	Comments	DblPois	Bivpois	DIBP
	Attack Parameters	team/level			
1	`team1..team212`	Milan	0.68	0.79	0.84
2	`team1..team210`	Juventus	0.18	0.24	0.22
3	`team1..team217`	Torino	0.11	0.18	0.18
4	`team1..team213`	Napoli	0.43	0.53	0.51
5	`team1..team215`	Roma	0.00	0.00	0.02
6	`team1..team216`	Sampdoria	0.02	0.08	0.10
7	`team1..team214`	Parma	-0.15	-0.17	-0.14
8	`team1..team29`	Inter	-0.29	-0.35	-0.37
9	`team1..team27`	Foggia	0.49	0.59	0.57
10	`team1..team211`	Lazio	0.16	0.22	0.28
11	`team1..team22`	Atalanta	-0.18	-0.22	-0.21
12	`team1..team26`	Fiorentina	0.18	0.27	0.29
13	`team1..team28`	Genoa	-0.04	-0.07	-0.09
14	`team1..team24`	Cagliari	-0.21	-0.19	-0.21
15	`-sum(ex4.m1$beta1[2:18])`	Verona	-0.40	-0.49	-0.51
16	`team1..team23`	Bari	-0.33	-0.47	-0.50
17	`team1..team25`	Cremonese	-0.29	-0.32	-0.36
18	`team1..team21`	Ascoli	-0.34	-0.61	-0.64
	Defense Parameters	team/level			
1	`team2..team112`	Milan	-0.50	-0.92	-1.18
2	`team2..team110`	Juventus	-0.50	-0.71	-0.70
3	`team2..team117`	Torino	-0.60	-0.80	-0.86
4	`team2..team113`	Napoli	0.12	0.16	0.19
5	`team2..team115`	Roma	-0.16	-0.20	-0.17
6	`team2..team116`	Sampdoria	-0.16	-0.15	-0.16
7	`team2..team114`	Parma	-0.27	-0.33	-0.34
8	`team2..team19`	Inter	-0.28	-0.32	-0.29
9	`team2..team17`	Foggia	0.50	0.61	0.63
10	`team2..team111`	Lazio	0.10	0.17	0.21
11	`team2..team12`	Atalanta	-0.11	-0.11	-0.11
12	`team2..team16`	Fiorentina	0.13	0.22	0.28
13	`team2..team18`	Genoa	0.25	0.32	0.40
14	`team2..team14`	Cagliari	-0.08	-0.01	-0.01
15	`-sum(ex4.m1$beta1[19:35])`	Verona	0.43	0.58	0.57
16	`team2..team13`	Bari	0.24	0.32	0.33
17	`team2..team15`	Cremonese	0.28	0.42	0.45
18	`team2..team11`	Ascoli	0.61	0.73	0.75
	Other Parameters				
	Syntax	Parameter			
	`ex4.m8$beta1[1]`	Intercept for λ_1	0.19	-0.10	-0.07
	`ex4.m8$beta2[1]`	Intercept for λ_2	-0.18	-0.58	-0.57
	`ex4.m8$beta3[1]`	Intercept for λ_3	$-\infty$	-1.35	-1.47
	`exp(ex4.m8$beta3[1])`	λ_3	0.00	0.26	0.23
	Difference of two constants	Home Effect	0.36	0.47	0.50
	`ex4.m8$p`	Mixing Proportion (p)			0.09
	`ex4.m8$theta`	$\theta_1 = P(X = Y = 1)$			1.00

Table 8: Parameters of models 1, 2 and 8 (see also Table 3 of Karlis and Ntzoufras (2003); all parameters can be obtained using the commands `ex4.m8$beta1` or `ex4.m8$beta2`).

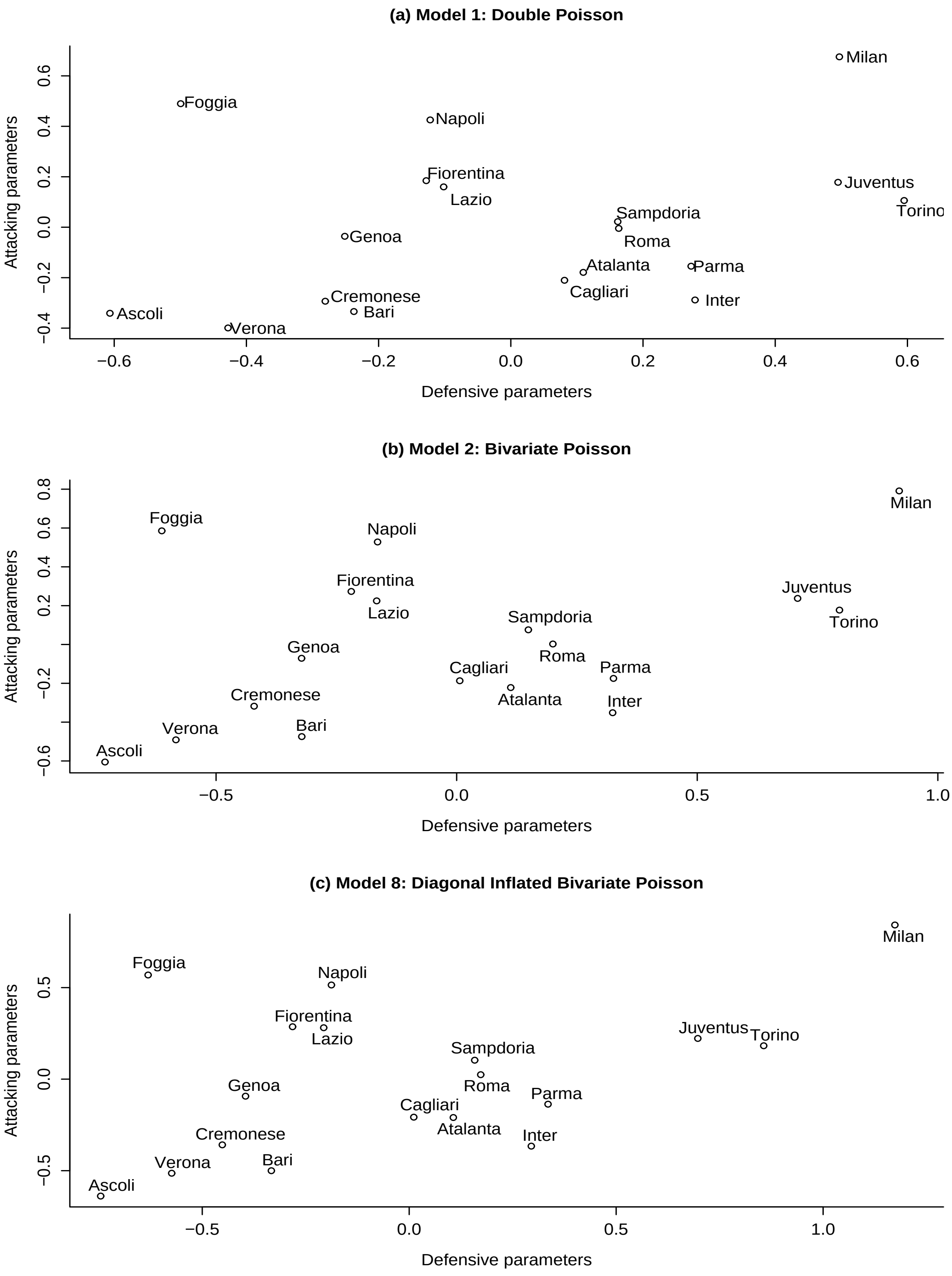

Figure 2: Graphical representation of defensive versus attacking parameters of models 1, 2 and 8 for example 4; defensive parameters were multiplied by minus one so as greater values to indicate teams with better defense.

The software presented in this paper implements methodology which can be easily extended and implemented in several variations of models discussed in this article. For example, the extension of the EM algorithm to the multivariate Poisson models is straightforward since such models are obtained through similar multivariate reduction techniques and the same data augmentation approach can be easily applied. Similarly, an EM algorithm can be easily modified to cover the case of finite mixtures of bivariate Poisson regressions. Such a model is a generalization of the approach presented by Wang, Cockburn, and Puterman (1996). The inflated models can be seen as a special case of finite mixtures of bivariate Poisson distributions.

Another generalization of the above algorithm can be constructed by considering a bivariate inflation distribution. Such a model is given by Dixon and Coles (1997) where the cells (0,0), (0,1), (1,0) and (1,1) were inflated.

The algorithms can be even extended to cover the case of models with random effects. For example, assuming gamma random effects, we obtain a bivariate negative binomial regression model, as in Munkin and Trivedi (1999).

Finally, the trivariate reduction technique (used in the data augmentation approach here) is useful for constructing multivariate models from simpler ones; see for example the bivariate generalized Poisson model of Vernic (1997). Clearly, EM algorithms, identical to the ones presented here, can be used to cover several other models arising by similar trivariate technique.

References

Bohning D, Dietz E, Schlattmann P, Mendonca L, Kirchner U (1999). "The Zero-inflated Poisson Model and the Decayed, Missing and Filled Teeth Index in Dental Epidemiology." *Journal of the Royal Statistical Society A*, **162**, 195–209.

Cameron A, Trivedi P (1998). *Regression Analysis of Count Data*. Oxford University Press.

Chib S, Winkelmann R (2001). "Markov Chain Monte Carlo Analysis of Correlated Count Data." *Journal of Business and Economic Statistics*, **19**, 428–435.

Dempster A, Laird N, Rubin D (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society B*, **39**, 1–38.

Dixon M, Coles S (1997). "Modelling Association Football Scores and Inefficiencies in Football Betting Market." *Applied Statistics*, **46**, 265–280.

Gan N (2000). *General Zero-inflated Models and their Applications*. Ph.D. thesis, North Carolina State University.

Ho L, Singer J (2001). "Generalized Least Squares Methods for Bivariate Poisson Regression." *Communications in Statistics - Theory and Methods*, **30**, 263–277.

Johnson N, Kotz S, Balakrishnan N (1997). *Discrete Multivariate Distributions*. Wiley - New York.

Jung R, Winkelmann R (1993). "Two Aspects of Labor Mobility: A Bivariate Poisson Regression Approach." *Empirical Economics*, **18**, 543–556.

Karlis D, Meligkotsidou L (2005). "Multivariate Poisson Regression with Full Covariance Structure." *Statistics and Computing (to appear)*.

Karlis D, Ntzoufras I (2000). "On Modelling Soccer Data." *Student*, **3**, 229–245.

Karlis D, Ntzoufras I (2003). "Analysis of Sports Data by Using Bivariate Poisson Models." *Journal of the Royal Statistical Society D (The Statistician)*, **52**, 381 – 393.

Kocherlakota S, Kocherlakota K (1992). *Bivariate Discrete Distributions*. New York: Marcel Dekker.

Kocherlakota S, Kocherlakota K (2001). "Regression in the Bivariate Poisson Distribution." *Communications in Statistics - Theory and Methods*, **30**, 815–827.

Lambert D (1992). "Zero-inflated Poisson Regression, with an Application to Defects in Manufacturing." *Technometrics*, **34**, 1–14.

Li C, Lu J, Park J, Kim K, Peterson J (1999). "Multivariate Zero-inflated Poisson Models and their Applications." *Technometrics*, **41**, 29–38.

McLachlan G, Krishnan N (1997). *The EM Algorithm and its Extensions*. Wiley, New York.

Meng X, Van Dyk D (1997). "The EM Algorithm: An Old Folk Song Sung to a Fast New Tune (with discussion)." *Journal of the Royal Statistical Society B*, **59**, 511–567.

Munkin M, Trivedi P (1999). "Simulated Maximum Likelihood Estimation of Multivariate Mixed-Poisson Regression Models, with Application." *Econometrics Journal*, **2**, 29–48.

Tsionas E (2001). "Bayesian Multivariate Poisson Regression." *Communications in Statistics - Theory and Method*, **30**, 243–255.

Vernic R (1997). "On the Bivariate Generalized Poisson Distribution." *ASTIN Bulletin*, **27**, 23–31.

Wahlin J (2001). "Bivariate ZIP models." *Biometrical Journal*, **43**, 147–160.

Wang K, Lee A, Yau K, Carrivick P (2003). "A Bivariate Zero Inflated Poisson Regression Model to Analyze Occupational Injuries." *Accident Analysis and Prevention*, **35**, 625–629.

Wang P, Cockburn I, Puterman M (1996). "Analysis of Patent Data - A mixed Poisson Regression Model Approach." *Journal of Business and Economic Statistics*, **16**, 27–41.

A. EM algorithms

A.1. Data augmentation

The EM algorithm (Dempster, Laird, and Rubin 1977) is a powerful algorithm for maximum likelihood (ML) estimation for data containing missing values or they can be considered as containing missing values. The EM algorithm is not only a numerical technique but also offers useful statistical insight (Meng and Van Dyk 1997). The key idea is to augment the observed data with some unobserved data so as the maximization of the complete likelihood is easier. More details on the algorithm can be found in McLachlan and Krishnan (1997).

Here we facilitate the trivariate reduction of the bivariate Poisson distribution. Suppose that for the i-th observation X_{1i}, X_{2i}, X_{3i} represent the non-observable data, while $X_i = X_{1i}+X_{3i}$ and $Y_i = X_{2i}+X_{3i}$ are the observed data. If the unobserved data were available the estimation would have been straightforward: we just had to fit Poisson regression models on X_1, X_2 and X_3 variables. Hence, in order to construct our EM-algorithm we need to estimate the unobserved data by their conditional expectations and then fit Poisson regression models to the pseudovalues obtained by the E-step. Denoting as $\boldsymbol{\phi}$ the entire vector of parameters, that is $\boldsymbol{\phi} = (\boldsymbol{\beta}'_1, \boldsymbol{\beta}'_2, \boldsymbol{\beta}'_3)$, the complete data log-likelihood is given by

$$L(\boldsymbol{\phi}) = -\sum_{i=1}^{n}\sum_{\kappa=1}^{3}\lambda_{\kappa i} + \sum_{i=1}^{n}\sum_{\kappa=1}^{3} x_{\kappa i}\log(\lambda_{\kappa i}) - \sum_{i=1}^{n}\sum_{\kappa=1}^{3}\log(x_{\kappa i}!),$$

where λ's are given by (2).

In the inflated case we need to introduce additional latent variables. Inflated models are in fact mixtures of two distributions which in our case are the bivariate Poisson, $BP(\lambda_1, \lambda_2, \lambda_3)$, and the distribution used to inflate the diagonal. Thus the standard EM approach for finite mixture applies. We introduce further latent variables V_i, $i = 1, \ldots, n$ which take the values 1 or 0 according to whether the observation comes from the inflation or the original component respectively. Now the complete data log-likelihood takes the form

$$\begin{aligned} L(\boldsymbol{\phi}, p, \boldsymbol{\theta}) &= \sum_{i=1}^{n} v_i \left\{\log(p) + \log f_D(x_i; \boldsymbol{\theta})\right\} \\ &+ \sum_{i=1}^{n}(1-v_i)\left\{\log(1-p) - \sum_{i=1}^{n}\sum_{\kappa=1}^{3}\lambda_{\kappa i} + \sum_{i=1}^{n}\sum_{\kappa=1}^{3} x_{\kappa i}\log(\lambda_{\kappa i}) - \sum_{i=1}^{n}\sum_{\kappa=1}^{3}\log(x_{\kappa i}!)\right\} \end{aligned}$$

Thus, at the E-step we also have to estimate V_i for $i = 1, \ldots, n$ using their conditional expectations. Full details concerning the algorithm follow in the next sub-sections.

A.2. The bivariate Poisson model

The EM-algorithm for the bivariate Poisson model (2) is given by:

E-step: Using the current parameter values of k iteration noted by $\boldsymbol{\phi}^{(k)}$, $\lambda_{1i}^{(k)}$, $\lambda_{2i}^{(k)}$ and $\lambda_{3i}^{(k)}$, calculate the conditional expected values of X_{3i}, for $i = 1, \ldots, n$, by

$$s_i = E\left(X_{3i} \mid X_i, Y_i, \boldsymbol{\phi}^{(k)}\right)$$

$$= \begin{cases} \lambda_{3i}^{(k)} \frac{f_{BP}\left(x_i-1, y_i-1 | \lambda_{1i}^{(k)}, \lambda_{2i}^{(k)}, \lambda_{3i}^{(k)}\right)}{f_{BP}\left(x_i, y_i | \lambda_{1i}^{(k)}, \lambda_{2i}^{(k)}, \lambda_{3i}^{(k)}\right)} & \text{if } \min(x_i, y_i) > 0 \\ 0 & \text{if } \min(x_i, y_i) = 0 \end{cases} \tag{7}$$

where $f_{BP}(x, y \mid \lambda_1, \lambda_2, \lambda_3)$ is given in (1).

M-step: Update the estimates by

$$\begin{aligned} \boldsymbol{\beta}_1^{(k+1)} &= \hat{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{s}, \boldsymbol{W}_1), \\ \boldsymbol{\beta}_2^{(k+1)} &= \hat{\boldsymbol{\beta}}(\boldsymbol{y} - \boldsymbol{s}, \boldsymbol{W}_2), \\ \boldsymbol{\beta}_3^{(k+1)} &= \hat{\boldsymbol{\beta}}(\boldsymbol{s}, \boldsymbol{W}_3), \\ \lambda_{\kappa i}^{(k+1)} &= \exp\left(\boldsymbol{W}_{\kappa i}^{\top} \hat{\boldsymbol{\beta}}_{\kappa}^{(k+1)}\right) \; for \quad \kappa = 1, 2, 3; \end{aligned}$$

where $\boldsymbol{s} = (s_1, \ldots, s_n)^{\top}$ is the $n \times 1$ vector calculated in the E-step, $\hat{\boldsymbol{\beta}}(\boldsymbol{x}, \boldsymbol{W})$ are the maximum likelihood estimates of a Poisson model with response the vector $\boldsymbol{x}$ and design or data matrix given by $\boldsymbol{W}$. Each data matrix $\boldsymbol{W}_{\kappa}$ is a $n \times p_{\kappa}$ matrix and $\boldsymbol{W}_{\kappa i}^{\top}$ is its corresponding i row (for $i = 1, \ldots, n$). If we wish to have common (or equal) parameters among different λ_{κ} then we should construct a common design matrix $\boldsymbol{W}$ and the corresponding parameter vector $\boldsymbol{\beta}$ will be estimated as $\boldsymbol{\beta}^{(k+1)} = \hat{\boldsymbol{\beta}}(\boldsymbol{u}, \boldsymbol{W})$, with $\boldsymbol{u}^{\top} = (\boldsymbol{x}^{\top} - \boldsymbol{s}^{\top}, \boldsymbol{y}^{\top} - \boldsymbol{s}^{\top}, \boldsymbol{s}^{\top})$. In the functions provided, we have consider the possibility to have common parameters only between λ_1 and λ_2. Note also that standard GLM procedures can be used for the M-step despite the fact that the responses are not any more integers. The latter does not cause any numerical problems.

A.3. The inflated bivariate Poisson model

For the EM-algorithm of inflated models, we introduced a binary latent indicator V_i for each $i = 1, \ldots, n$ indicating the inflation when $V_i = 1$. The EM algorithm for the diagonal inflated model (3) is now given by

E-step :

(a) Using the current parameter values of k iteration noted by $\boldsymbol{\phi}^{(k)}$, $\lambda_{1i}^{(k)}$, $\lambda_{2i}^{(k)}$, $\lambda_{3i}^{(k)}$, $p^{(k)}$ and $\boldsymbol{\theta}^{(k)}$, for $i = 1, \ldots, n$ calculate

$$\begin{aligned} v_i &= E\left(V_i \mid X = x_i, Y = y_i, \boldsymbol{\phi}^{(k)}, p^{(k)}, \boldsymbol{\theta}^{(k)}\right) \\ &= \begin{cases} \frac{p^{(k)} f_D\left(x_i | \boldsymbol{\theta}^{(k)}\right)}{p^{(k)} f_D\left(x_i | \boldsymbol{\theta}^{(k)}\right) + \left(1 - p^{(k)}\right) f_{BP}\left(x_i, x_i | \lambda_{1i}^{(k)}, \lambda_{2i}^{(k)}, \lambda_{3i}^{(k)}\right)}, & \text{if } \ x_i = y_i \\ 0 & \text{if } \ x_i \neq y_i \end{cases} \end{aligned} \tag{8}$$

where $f_D\left(x \mid \boldsymbol{\theta}\right)$ is the probability function of the inflation distribution with parameter vector $\boldsymbol{\theta}$ evaluated at the value of x.

(b) For $i = 1, \ldots, n$, calculate s_i using (7).

M-step: Update the parameters by

$$
\begin{aligned}
p^{(k+1)} &= \frac{1}{n}\sum_{i=1}^{n} v_i \\
\boldsymbol{\beta}_1^{(k+1)} &= \hat{\boldsymbol{\beta}}_{\tilde{\boldsymbol{v}}}(\boldsymbol{x}-\boldsymbol{s}, \boldsymbol{W}_1), \\
\boldsymbol{\beta}_2^{(k+1)} &= \hat{\boldsymbol{\beta}}_{\tilde{\boldsymbol{v}}}(\boldsymbol{y}-\boldsymbol{s}, \boldsymbol{W}_2), \\
\boldsymbol{\beta}_3^{(k+1)} &= \hat{\boldsymbol{\beta}}_{\tilde{\boldsymbol{v}}}(\boldsymbol{s}, \boldsymbol{W}_3), \\
\boldsymbol{\theta}^{(k+1)} &= \hat{\boldsymbol{\theta}}_{\boldsymbol{v},D}, \\
\lambda_{\kappa i}^{(k+1)} &= \exp\left(\boldsymbol{W}_{\kappa i}^{\top}\hat{\boldsymbol{\beta}}_{\kappa}^{(k+1)}\right) \quad for \quad \kappa = 1,2,3;
\end{aligned}
$$

where $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{s}$, $\boldsymbol{v}$ and $\tilde{\boldsymbol{v}}$ are $n \times 1$ vectors with elements x_i, y_i, s_i, v_i and $\tilde{v}_i = 1 - v_i$ for $i = 1, \ldots, n$, $\hat{\boldsymbol{\beta}}_{\boldsymbol{v}}(\boldsymbol{y}, \boldsymbol{W})$ is the weighted maximum likelihood estimates of $\boldsymbol{\beta}$ of a Poisson regression model with response $\boldsymbol{y}$, data matrix $\boldsymbol{W}$ and weight vector $\boldsymbol{v}$, and $\hat{\boldsymbol{\theta}}_{\boldsymbol{v},D}$ is the weighted maximum likelihood estimates of $\boldsymbol{\theta}$ for the distribution $D(x;\boldsymbol{\theta})$ and weights given by vector $\boldsymbol{v}$. The design matrices $\boldsymbol{W}_\kappa$, $\kappa = 1,2,3$ are defined as above.

For specific choices of the inflation distribution we obtain the following estimates:

- Geometric distribution: For the geometric distribution, with probability function $f(x|\theta) = (1-\theta)^x\theta$, $0 \le \theta \le 1$, $\ x = 0, 1, \ldots$, θ is updated by

  $$\theta^{(k+1)} = \frac{\sum_{i=1}^{n} v_i}{\sum_{i=1}^{n} v_i x_i + \sum_{i=1}^{n} v_i}.$$

  Note that, if $\theta = 1$ the zero-inflated model is deduced.

- Poisson distribution: For the Poisson distribution with probability function $f(x|\theta) = e^{-\theta}\theta^x/x!$, $\theta \ge 0$, $\ x = 0, 1, \ldots$, θ is updated by $\theta^{(k+1)} = \left(\sum_{i=1}^{n} v_i\right)^{-1}\sum_{i=1}^{n} v_i x_i$. Note that if $\theta = 0$ the zero-inflated model is deduced.

- Discrete distribution: For any discrete distribution, *Discrete*(J), with probability function (4) then the model parameters are given by $\theta_j = \left(\sum_{i=1}^{n} v_i\right)^{-1}\sum_{i=1}^{n} I(X_i = Y_i = j)v_i$ for $j = 1, \ldots, J$ and $\theta_0 = 1 - \sum_{j=1}^{J}\theta_j$; where $I(x)$ is the indicator function taking value equal to one if x is true and zero otherwise.

- Zero-inflated model: The zero inflated model is a special case of *Discrete*(J) with $J = 0$ and $\theta_0 = 1$ which results to the inflation of cell $(0,0)$. Hence, there is no need to estimate additional parameters except p which is the mixing proportion of the inflation component. Further note that the zero-inflated model is a limiting case when either the Poisson (with $\theta \to 0$) or the geometric (with $\theta \to 1$) inflation is used.

In fact the M-step consists of several iterations of the iterated reweighted algorithm used for GLM. Hence the algorithm is an Expectation Conditional Maximization (ECM) algorithm. Usually the number of iterations needed to fit the GLM within each M-step can be considerably reduced if we use as starting values the values obtained by the previous EM step.

Alternatively, we may constrain the number of iterations for fitting the GLM to a small number. This will be still sufficient to improve the log-likelihood, despite the fact that the fitted model may not be the best within each iteration of the EM algorithm.

Further complexity can be added to the model by imposing a additional covariate structure on parameters θ or p. EM-algorithms need to be slightly modified in order to incorporate such extensions. Similarly, the model of Dixon and Coles (1997) can be fitted using an EM algorithm identical to the one proposed here. The algorithm presented here, can be considered as a generalization of the algorithm described in Wang *et al.* (2003). Finally, generalizations of the models for multivariate versions can easily be derived.

Affiliation:

Dimitris Karlis
Department of Statistics
Athens University of Economics and Business
76, Patission Str., 10434,
Athens, Greece.
Telephone: +30/210/8203920
E-mail: karlis@aueb.gr
URL: http://www.stat-athens.aueb.gr/~karlis/

Ioannis Ntzoufras
Department of Statistics
Athens University of Economics and Business
76, Patission Str., 10434
Athens, Greece.
Telephone: +30/210/8203968
E-mail: ntzoufra@aueb.gr
URL: http://www.stat-athens.aueb.gr/~jbn/

Journal of Statistical Software *Submitted:* 2003-11-14
September 2005, Volume 14, Issue 10. *Accepted:* 2005-09-05
http://www.jstatsoft.org/