Bivariate Poisson and Diagonal Inflated Bivariate Poisson Regression Models in R/SPLUS

Dimitris Karlis and Ioannis Ntzoufras

Department of Statistics, Athens University of Economics and Business,

76, Patission Str., 10434, Athens, Greece,

e-mails: {karlis,ntzoufras}@aueb.gr

Abstract

In this paper we present R/SPLUS functions for the maximum likelihood estimation of the parameters of bivariate and diagonal inflated bivariate Poisson regression models. An Expectation - Maximization (EM) algorithm is facilitated. 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 R/SPLUS functions is provided.

Keywords: Bivariate Poisson distribution; EM algorithm; Zero and Diagonal Inflated Models; R/SPLUS functions; Multivariate Count Data.

1 Introduction

Bivariate Poisson models are appropriate for modelling 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/SPLUS 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 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 modelling 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/SPLUS 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

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

The above bivariate distribution allows for positive dependence between the two random variables. Marginally each random variable follows a Poisson distribution with $E(X) = \lambda_1 + \lambda_3$ and $E(Y) = \lambda_2 + \lambda_3$. Moreover, $Cov(X, Y) = \lambda_3$, 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 *et al.* (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

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

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. 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 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 equation 2) to $E(X_i) = \exp(\boldsymbol{W}_{1i}^T \boldsymbol{\beta}_1) + \exp(\boldsymbol{W}_{3i}^T \boldsymbol{\beta}_3)$. 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 loglikelihood 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/SPLUS 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).

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.

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 *et al.* , 1999). 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 modelling 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 Poison 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

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

where $f_{Po}(x \mid \lambda)$ is the probability function of the Poisson distribution. For example, if we consider a Geometric inflation then the resulting marginal distribution is a 2-finite mixture with one Poisson and one geometric component. Thus the marginal mean is given by

$$E(X) = (1-p)\ (\lambda_1 + \lambda_3) + p\ E_D(X)$$

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

$$Var(X) = (1-p)\left\{(\lambda_1+\lambda_3)^2 + (\lambda_1+\lambda_3)\right\} + pE_D(X^2) - \{(1-p)(\lambda_1+\lambda_3) + pE_D(X)\}^2 .$$

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 $\boldsymbol{\theta}^T = (0, 1)$) 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

$$\begin{aligned} COV_{IBP}(X,Y) \quad = \quad & (1-p)\left\{\lambda_3 + (\lambda_1+\lambda_3)(\lambda_2+\lambda_3)\right\} + pE_D(X^2) \\ & -(1-p)^2(\lambda_1+\lambda_3)(\lambda_2+\lambda_3) \\ & -(1-p)pE_D(X)(\lambda_1+\lambda_2+2\lambda_3) - p^2\{E_D(X)\}^2. \end{aligned}$$

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 $\lambda_3 = 0$, the covariance is given by

$$COV_{IBP}(X,Y) \quad = \quad p(1-p)\lambda_1\lambda_2 + pE_D(X^2) - p(1-p)E_D(X)(\lambda_1+\lambda_2) - p^2\{E_D(X)\}^2$$

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 $E_D(X)=E_D(X^2)=0$ and $COV_{IBP}(X,Y)=p(1-p)\lambda_1\lambda_2$ which is always positive. For this reason, diagonal inflation can possibly correct both over/under-dispersion and correlation problems encountered in modelling count data.

3 R/SPLUS Functions for Bivariate Poisson Models

3.1 Short description of Functions and Installation

In order to run the EM algorithm in R/SPLUS, you have to download the associated zipped file: `bivpois.zip` from `http://www.stat-athens.aueb.gr/~jbn/papers/paper14.htm` . The zipped file contains the source files `bivpois.r` for R and `bivpois.s` for SPLUS (including all the relevant functions) and some short help files. When you extract all files in a directory, then you install the bivariate Poisson functions by typing in the R environment:

`source('c:/directory/bivpois.r')` [ENTER]

or `source('c:/directory/bivpois.s')` in SPLUS. Note that the differences between the two source files are minor. The original version of the functions was written in SPLUS but the authors strongly recommend using R mainly because the latter is much faster in computations. Although the presentation below is focused and implemented in R, the same analysis can be also implemented in SPLUS with minor differences.

The above procedure installs the following functions:

1. `bivpois.table`: Bivariate Poisson probability function (in tabular form) using recursive relationships.
2. `bivpois`: Probability function of bivariate Poisson.
3. `logbp`: Logarithm of the probability function of bivariate Poisson.
4. `simple.bp`: EM for fitting a simple bivariate Poisson model with constant λ_1, λ_2 and λ_3 (no covariates are used).
5. `glm.bp`: EM for fitting a general linear bivariate Poisson model with covariates on λ_1, λ_2 and λ_3.

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

3.2 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, nmax = 300, pres = 1e-008, ini3 = 1)
```

- REQUIRED ARGUMENTS:
  - `x, y` : vectors containing the data.
- OPTIONAL ARGUMENTS
  - `nmax`: Maximum number of EM steps. The program terminates if the number of iterations exceed this number 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.
  - `ini3`: Initial value for λ_3.

A list object is returned with the following output variables:

- `parameters`: Number of estimated parameters of the fitted model.
- `iterations`: Number of iterations of the EM algorithm.
- `lambda1, lambda2, lambda3`: Parameters 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.
- `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.

During the run 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.

3.3 The Function `glm.bp`

Function `glm.bp` implements the EM algorithm for fitting the bivariate Poisson regression model (2) with $\boldsymbol{W}_1 = [\mathbf{1}_n, \mathbf{0}_n, \boldsymbol{w}_1]$, $\boldsymbol{W}_2 = [\mathbf{0}_n, \mathbf{1}_n, \boldsymbol{w}_2]$, $\boldsymbol{W}_3 = [\mathbf{1}_n, \boldsymbol{w}_3]$ and $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \boldsymbol{\beta}$. Hence the final form of the model is given by

$$\begin{aligned}
(x_i, y_i) &\sim BP(\lambda_{1i}, \lambda_{2i}, \lambda_{3i}) \text{ for } i = 1, \ldots, n, \\
\boldsymbol{l}_1 &= [\mathbf{1}_n, \mathbf{0}_n, \boldsymbol{w}_1]\boldsymbol{\beta} = \beta_1 + \beta_3 \boldsymbol{w}_{11} + \ldots + \beta_{p+2}\boldsymbol{w}_{1p} \\
\boldsymbol{l}_2 &= [\mathbf{0}_n, \mathbf{1}_n, \boldsymbol{w}_2]\boldsymbol{\beta} = \beta_2 + \beta_3 \boldsymbol{w}_{21} + \ldots + \beta_{p+2}\boldsymbol{w}_{2p} \qquad (6) \\
\boldsymbol{l}_3 &= [\mathbf{1}_n, \boldsymbol{w}_3]\boldsymbol{\beta}_3 \quad = \beta_{31} + \beta_{32} \boldsymbol{w}_{31} + \ldots + \beta_{3p_2+1}\boldsymbol{w}_{3p_2} \\
\boldsymbol{l}_\kappa &= (\log \lambda_{\kappa 1}, \ldots, \log \lambda_{\kappa n})^T \text{ for } \kappa = 1, 2, 3;
\end{aligned}$$

where n is the sample size, $\mathbf{1}_n$ and $\mathbf{0}_n$ are $n \times 1$ vectors of one's and zero's, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are $n \times p$ matrices containing the explanatory variables used for λ_1 and λ_2, respectively, $\boldsymbol{w}_3$ is a $n \times p_2$ matrix containing the explanatory variables used for λ_3, $\boldsymbol{w}_{\kappa j}$ are $n \times 1$ vectors corresponding to j column of $\boldsymbol{w}_\kappa$ matrix, $\boldsymbol{\beta}$ is the vector of length $p+2$ with the model coefficients for λ_1 and λ_2 (with elements β_j) while $\boldsymbol{\beta}_3$ is the vector of length $p_2 + 1$ with the model coefficients for λ_3 (with elements β_{3j}). Here we use the same parameter vector $\boldsymbol{\beta}$ for λ_1 and λ_2 in order to be able to fit models with common parameters between λ_1 and λ_2.

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:

```
glm.bp(x, y, w1, w2, w3=0, l3=1, nmax=300, constant=c(1,1), pres=0.00000001)
```

- REQUIRED ARGUMENTS:
  - `x, y`: vectors containing the response data.
  - `w1, w2`: data frames (or matrices) with n rows and p columns. Names of columns should be the same for both matrices and should correspond to the same variables. Generally we have the following cases of effects on λ_1 and λ_2:
    1. Common effect of a variable z on both λ_1 and λ_2: Set the same column j of `w1` and `w2` equal to z.
    2. Effect only on λ_1: Set one column of `w1` equal to z and exactly the same column of `w2` equal to a vector of zero's.
    3. Effect only on λ_2: Set one column of `w2` equal to z and exactly the same column of `w1` equal to a vector of zero's.
    4. Different effect on λ_1 and λ_2: Set the a column j of `w1` equal to z and exactly the same column j of `w2` equal to a vector of zero's (to define the effect of z

on λ_1). Further, set a different column $k \neq j$ of `w2` equal to z and exactly the same column k of `w1` equal to a vector of zero's (to define the effect of z on λ_2).

You should be careful on the definition of the names of each column. Different names should be attributed to each column that corresponds to the effect of the same variable `z` on λ_1 and λ_2. We propose to use the name of the variable and in brackets the number indicating the linear predictor which is associated with the corresponding parameter. For example in fourth case above we may use `'Z(1)'` and `'Z(2)'`. For a detailed example of defining such matrices see example 1 in section 4.1. Two common cases of `w1` and `w2` matrices are the following:

1. Use common effects for both x and y by setting `w1=W.matrix, w2=W.matrix`. The first two components of the estimated parameters $\boldsymbol{\beta}$ (`beta` in R) are the constant terms. Then each of the remaining components of `beta` vector corresponds to the common effect on λ_1 and λ_2 of each column of `W.matrix`.
2. Use different effects for x and y on the same covariates by setting `w1=cbind(W.matrix, W.matrix*0), w2=cbind(W.matrix*0, W.matrix)`. In this case, assuming that the dimension of the original `W.matrix` is $n \times p$, the dimension of `w1` and `w2` will be $n \times 2p$. Hence the first two components of the estimated parameters $\boldsymbol{\beta}$ (`beta` in R) are the constant terms (again), while the components `beta[j]`for $j = 3, \ldots, p+2$ correspond to the effects of `W.matrix` on λ_1 while for $j = p+3, \ldots, 2p+2$ correspond to the effects of `W.matrix` on λ_2.

- OPTIONAL ARGUMENTS

– `l3=1`: Indicator variable specifying which model we wish to fit. If `l3=0` then we fit a double Poisson model ($\lambda_3 = 0$). If `l3=1` then we fit a bivariate Poisson model with λ_3 constant (no covariates on λ_3). In both cases `w3` is not used. If `l3=2` then we fit a bivariate Poisson model with covariates on λ_3 (given by `w3`).

– `w3`: data frame (or matrix) with n rows and p_2 columns. This argument is required only if l3=2.

– `nmax=300`: Maximum number of EM steps.

– `constant=c(1,1)`: Includes constant terms for λ_1, λ_2 [1st value] and λ_3 [2nd value].

– `pres=0.00000001`: Precision used in relative log-likelihood difference.

A list object is returned with the following output variables:

- `parameters`: Number of estimated parameters of the fitted model.
- `iterations`: Number of iterations of the EM algorithm.
- `beta`: Vector $\boldsymbol{\beta}$ containing the estimates of the model parameters for modelling λ_1 and λ_2. When a column of `w1` and `w2` is factor then sum-to-zero constraints are used by default. The first two parameters correspond to the constant terms of λ_1 and λ_2 respectively, while the rest of the parameters correspond to each column of `w1` and `w2` matrices.
- `beta3`: Vector $\boldsymbol{\beta}_3$ containing the estimates of the model parameters for modelling λ_3. If `l1=0` then this vector is not calculated. If `l3=1` then this vector contains only one value (equal to the log of λ_3). When a column of `w3` is factor then sum-to-zero constraints are used by default.
- `lambda1, lambda2`: vectors of length n containing the estimated λ_1 and λ_2.
- `lambda3`: vector containing the values of λ_3. If `l3=0` then λ_3 is equal to zero. If `l3=1` then λ_3 contains only one value. If `l3=2` then λ_3 is a vector of length n.
- `loglikelihood`: Log-likelihood of the fitted model given in a vector form of length equal to `iterations` (one value per iteration).
- `AIC, BIC`: AIC and BIC values of the model. Values are also given for the saturated model.

During the run of the algorithm the following details are printed: the iteration number, vector $\boldsymbol{\beta}$, vector $\boldsymbol{\beta}_3$, the log-likelihood and the relative difference of the log-likelihood. For an illustration using this function see examples in sections 4.1, 4.3 and 4.5.

3.4 The Function `glm.dibp`

Function `glm.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 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:

```
glm.dibp(x, y, w1, w2, w3=0, l3=1, nmax=300, constant=c(1,1),
pres=0.00000001, distribution=1, jmax=2) .
```

- REQUIRED ARGUMENTS: Arguments `x,y,w1,w2,w3` are exactly the same as in `glm.bp` function.
- OPTIONAL ARGUMENTS
  - Arguments `l3, w3, nmax, constant, press`: are exactly the same as in `glm.bp` function.
  - `distribution=1`: Specifies the type of inflated distribution; 1= $Discrete(J = jmax)$, 2= $Poisson(\theta)$, 3= $Geometric(\theta)$.
  - `jmax=2`: Number of parameters used in *Discrete* distribution. This argument is not used if `distribution`=2 or 3.

A list object is returned with the following output variables:

- Variables `beta, beta3, lambda1, lambda2, lambda3` are the same as in `glm.bp` function and correspond to the parameters of the bivariate Poisson component.
- Variables `loglikelihood, AIC, BIC, parameters, iterations` are also the same as in `glm.bp` function.
- `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`=1 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`=2 then θ is the mean of the Poisson; if `distribution`=3 then θ is the success probability of the Geometric distribution.

During the run of the algorithm the following details are printed: the iteration number, vector $\boldsymbol{\beta}$, vector $\boldsymbol{\beta}_3$, the mixing proportion, parameter vector $\boldsymbol{\theta}$, the log-likelihood and the relative difference of the log-likelihood. For an illustration of using this function see examples in sections 4.2, 4.4 and 4.5.

4 Examples

4.1 Simulated Example 1

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.

4.1.1 Fitting the Constant Bivariate Poisson Model in R

In order to fit the simple bivariate Poisson model and store the results in an object called `simple.ex1` we type the following:

```
simple.ex1<-simple.bp( v1, v2 )
```

where `v1,v2` are vectors of length 100 containing our data. If we wish to monitor the calculated arguments then we type `names(simple.ex1)` resulting to:

```
[1] "lambda1"       "lambda2"       "lambda3"       "loglikelihood"
[5] "parameters"    "AIC"           "BIC"           "iterations"
```

We may further monitor any of the above values by typing the name of the stored object (here `simple.ex1`) followed by the dollar character (`$`) and the name of the output variable we wish to monitor. For example `simple.ex1$lambda1` will print the value of λ_1:

```
> simple.ex1$lambda1
[1] 6.574695
```

		Model Details						
		λ_1	λ_2	λ_3	Par.	Log-like	AIC	BIC
1	DP	Saturated			200	-405.05	1210.09	1869.76
2	DP	Constant	Constant	-	2	-540.62	1085.25	1091.84
3	BP	Constant	Constant	Constant	3	-516.73	1039.47	1049.36
4	DP	Full	Full	-	12	-494.98	1013.96	1053.54
5	BP	Full	Full	Constant	13	-478.26	982.52	1025.39
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	1014.83	1037.91
8	BP	Full	Full	Full	18	-471.52	979.04	1038.41
9	BP	Full	Full	$Z_1 + Z_2 + Z_3 + Z_4$	17	-472.52	979.03	1035.10
10	BP*	$Z_1 + Z_3$	$Z_1 + Z_3 + Z_5$	Full	12	-476.50	977.00	1016.58
11	BP*	$Z_1 + Z_3$	$Z_1 + Z_3 + Z_5$	$Z_1 + Z_2 + Z_3 + Z_4$	11	-476.81	975.62	1011.90

Table 1: Details for Fitted Models for Simulated Example 1 (Constant terms are included in all models; Par.: Number of Parameters; Log-Like: 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).

	Actual	3	4	5	6	7	8	9	10	11
λ_1		6.55								
Constant	1.80	1.88	2.46	1.87	2.45	1.71	1.75	1.73	1.75	1.74
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	0.50
Z_2	0.00		-1.11	-2.82			-2.83	-3.00		
Z_3	-3.00		0.90	1.68	0.18	-1.40	-1.76	-1.98	-2.43	3.78
Z_4	0.00		-2.03	-5.27			-2.62	-3.91		
Z_5	3.00		0.65	2.12	1.03	5.11	6.26	4.59	4.42	1.13
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 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 $\lambda_3 = 0$).

Similarly the command `simple$BIC` returns:

```
> simple.ex1$BIC
     Saturated Double Poisson  Bivar.Poisson
      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 `simple.ex1` can be printed by simple typing its name:

```
> simple.ex1
$lambda1
[1] 6.574695

$lambda2
[1] 2.894695

$lambda3
[1] 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 Double Poisson  Bivar.Poisson
      1210.095       1085.246       1039.464

$BIC
     Saturated Double Poisson  Bivar.Poisson
      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:simple.ex1$iterations, simple.ex1$loglikelihood, xlab='Iterations',
ylab='Log-likelihood', type='l' )
```

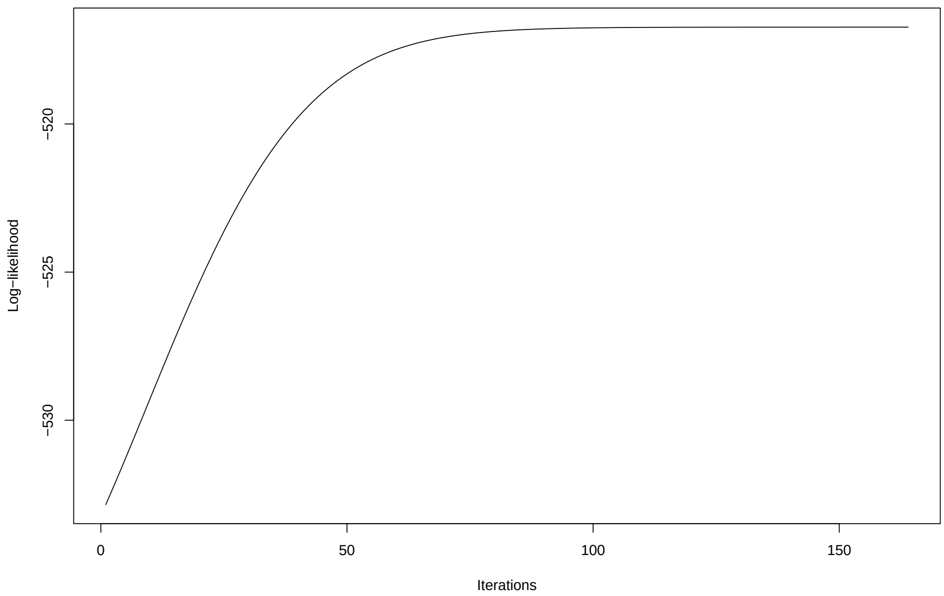

Figure 1: Log-likelihood Evolution for the Simple Bivariate Poisson Model Fitted on Data of Simulated Example 1.

4.1.2 Fitting Bivariate Poisson Models Regression Models in R for Simulated Example 1

Here we illustrate how we can fit models 3-11 for simulated example 1. All covariates were stored in a `z` vector of dimension 100×5. Response variables are stored in vectors `v1` and `v2`. Each column indicates one variable. Before we proceed we calculate the following `w` matrices:

```
# Matrices used when full model with non-common parameters is fitted
# (models 4,5,8,9)
w1full<- cbind( z  , z*0 )
w2full<- cbind( z*0, z )
names(w1full)_c( paste( 'Z',1:5,'(1)',sep=''), paste('Z',1:5,'(2)',sep='') )
names(w2full)_c( paste( 'Z',1:5,'(1)',sep=''), paste('Z',1:5,'(2)',sep='') )
# Matrices used when the actual model with common and non-common parameters
# is fitted (models 6,7,10,11)
#
w1act<- as.data.frame(cbind( z[,1]   , z[,1]*0, z[,3], z[,5]*0))
w2act<- as.data.frame(cbind( z[,1]*0, z[,1]   , z[,3], z[,5]  ))
names(w1act)_c(  'Z1(1)', 'Z1(2)', 'Z3', 'Z5(2)' )
names(w2act)_c(  'Z1(1)', 'Z1(2)', 'Z3', 'Z5(2)' )
```

Data matrices `w1act` and `w2act` and their names have been defined following the logic described in section 3.3. Column labels used have the form `'Zj(k)'`; where j is the variable index and k denotes if the corresponding effect is linked with λ_1 or λ_2. If the effect is the same on both λ_1 and λ_2 then we use simply the name of the variable. Hence the first column of `w1act` and `w2act` corresponds to the effect of Z_1 on λ_1 (since `w1act[,1]=z[,1]` and `w2act[,1]=0`) and is denoted as '`Z1(1)`'. Similarly the 2nd and the 4th column correspond to the effect of Z_1 and Z_5 respectively on λ_2 hence they are denoted by '`Z1(2)`' and '`Z5(2)`'. Finally, the 3rd column is the common effect of Z_3 on both λ_1 and λ_2 (since `w1act[,3]=z[,3]` and `w2act[,3]=z[,3]`) and is denoted by '`Z3`'. Using similar logic, we can specify the names for models using `w1full` and `w2full` matrices. Columns 1-5 correspond to the effects of Z_j on λ_1 (for $j = 1, \ldots, 5$) and are denoted by '`Zj(1)`' while columns 6-10 correspond to the effects of Z_j on λ_2 (for $j = 1, \ldots, 5$) and are denoted by '`Zj(2)`'.

After defining the data matrices we need we use the following commands to fit each model presented in Table 1:

```
model3 <- simple.bp(v1, v2)
model4 <- glm.bp(v1,v2, w1full, w2full, l3=0)
model5 <- glm.bp(v1,v2, w1full, w2full)
model6 <- glm.bp(v1,v2, w1act , w2act , l3=0)
model7 <- glm.bp(v1,v2, w1act , w2act)
model8 <- glm.bp(v1,v2, w1full, w2full, z, l3=2)
model9 <- glm.bp(v1,v2, w1full, w2full, z[,1:4], l3=2)
model10<- glm.bp(v1,v2, w1act, w2act, z, l3=2)
model11<- glm.bp(v1,v2, w1act, w2act, z[,1:4], l3=2)
```

For model 11 we have the following estimates

```
> model11$beta
Intercept_1 Intercept_2       Z1(1)       Z1(2)          Z3       Z5(2)
  1.7469566   0.6146818   2.6168688   0.4952161  -2.4263279   3.7773169
>
> model11$beta3
(Intercept)          Z1          Z2          Z3          Z4
  1.7163097   0.1470043  -1.2234779   1.9514278  -2.5065180
```

In the above results `Intercept1, Intercept2` correspond to constant terms for λ_1 and λ_2 while the rest of the parameters are identified according to their labels. From the above results, the model can be summarized by the following equation

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

Model Details			Sim.Example 2				Mix.Prop.
		Diagonal Distribution	Par.	Log-Like	AIC	BIC	(p)
1	BP	No Diagonal Inflation	11	-476.81	975.61	1011.89	0.000
2	DIBP	*Discrete*(0)	12	-551.27	1126.53	1166.11	0.019
3	DIBP	*Discrete*(1)	13	-513.51	1053.01	1095.89	0.100
4	DIBP	*Discrete*(2)	14	-495.93	1019.86	1066.03	0.139
5	DIBP	*Discrete*(3)	15	-472.47	974.96	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	-465.70	957.39	1000.27	0.274

Table 3: Details for Fitted Models for Simulated Example 2 (Par.: Number of Parameters; Log-like: Log-likelihood; Mix.Prop.: Mixing Proportion).

4.2 Simulated Example 2: Diagonal Inflated models

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

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.

	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	1.75	2.62	-2.43	0.61	0.50	-2.43	3.78	1.72	0.15	-1.22	1.95	-2.51
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	0.12	-0.96	1.91	-3.25
10	1.78	2.38	-1.96	0.67	-0.33	-1.96	3.15	1.66	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 Simulated Example 2. Parameter vector for model 7 is given by $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$.

4.2.1 Fitting Diagonal Inflated Bivariate Poisson Models Regression Models in R for Simulated Example 2

Here we illustrate how we can fit models 2-10 for the data of the simulated example 2 using the R/SPLUS function `glm.bibp` (see Table 3). Matrices `w1act` and `w2act` are defined as in section 4.1.2. The following commands have been used to fit each model

```
sim2.model2 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=1, jmax=0)
sim2.model3 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=1, jmax=1)
sim2.model4 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=1, jmax=2)
sim2.model5 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=1, jmax=3)
sim2.model6 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=1, jmax=4)
sim2.model7 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=1, jmax=5)
sim2.model8 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=1, jmax=6)
sim2.model9 <-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=2 )
sim2.model10<-glm.dibp(d1,d2, w1act,w2act, z[,1:4], l3=2, distribution=3 )
```

For model 7 we have the following estimates

```
> sim2.model7$beta
Intercept_1 Intercept_2       Z1(1)       Z1(2)          Z3       Z5(2)
  1.7792597   0.6696941   2.4328104  -0.3855074  -2.0553295   3.0287995
> sim2.model7$beta3
(Intercept)          Z1          Z2          Z3          Z4
  1.6685013   0.1665392  -0.9176112   1.9244447  -3.1781414
> sim2.model7$p
[1] 0.2648003
> sim2.model7$theta
[1] 0.3018781 0.1494186 0.2237955 0.1469628 0.1026400
```

The logic for identifying $\boldsymbol{\beta}$ parameters is the same as in `glm.bp` function. The above results can be summarized by the following model

$$
\begin{aligned}
f_{IBP}(x_i, y_i) &= \begin{cases} 0.735 \times f_{BP}(x_i, y_i \mid \lambda_{1i}, \lambda_{2i}, \lambda_{3i}), & x_i \neq y_i \\ 0.735 \times f_{BP}(x_i, y_i \mid \lambda_{1i}, \lambda_{2i}, \lambda_{3i}) + 0.265 \times \theta_{x_i}, & x_i = y_i, \end{cases} \\
(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5) &= (0.302, 0.149, 0.224, 0.147, 0.103) \\
\theta_j &= 0 \text{ for } j > 5 \\
\theta_0 &= 1 - \sum_{j=1}^{\infty} \theta_j = 0.075 \\
\log(\lambda_{1i}) &= 1.78 + 2.43 Z_{1i} - 2.06 Z_{3i} \\
\log(\lambda_{2i}) &= 0.67 - 0.39 Z_{1i} - 2.06 Z_{3i} + 3.03 Z_{5i} \\
\log(\lambda_{3i}) &= 1.67 + 0.17 Z_{1i} - 0.92 Z_{2i} + 1.92 Z_{3i} - 3.17 Z_{4i}.
\end{aligned}
$$

Similarly model 9 produces the following results

```
> sim2.model9$beta
Intercept_1 Intercept_2       Z1(1)       Z1(2)          Z3       Z5(2)
  1.7836971   0.6798381   2.4219333  -0.3491147  -2.0195228   2.9741247
> sim2.model9$beta3
(Intercept)          Z1          Z2          Z3          Z4
  1.6674222   0.1246431  -0.9613666   1.9132802  -3.2477915
> sim2.model9$p
[1] 0.2677973
> sim2.model9$theta
[1] 2.433656
```

which can be summarized by the following model

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

4.3 Example 3: Real Application

In this example, we have fitted bivariate Poisson models on data concerning the demand for Health Care in Australia, reported by Cameron and Trivedi (1986). 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 Doctor Consultations (X)	Number of Prescribed medications (Y) 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, 1986)

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 (1986).

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

		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	1.36	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 Fitting Bivariate Poisson Models for the Data of Example 3.

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 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 $\lambda_1 + \lambda_3$.

4.3.1 Fitting Bivariate Poisson Models Regression Models in R for Example 3

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`). All the data are stored in data frame named `ex3`. The available variables are the following:

```
> names(ex3)
 [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. First of all we create the design matrices.

```
w1<-as.data.frame(cbind(ex3$sex,ex3$age,ex3$income,0,0,0))
w2<-as.data.frame(cbind(0,0,0,ex3$sex,ex3$age,ex3$income))
w3<-as.data.frame(cbind(ex3$sex))
```

```
names(w1)<-c('Gender(1)','Age(1)','Income(1)','Gender(2)','Age(2)','Income(2)')
names(w2)<-c('Gender(1)','Age(1)','Income(1)','Gender(2)','Age(2)','Income(2)')
names(w3)<-c('Gender')
```

Note that for each design matrix `w1,w2` we have added columns with 0, according to the guidelines provided in section 3.3. Hence the three first columns specify the effect of sex, age and income on λ_1 while the remaining three columns define the effect of the same variables on λ_2.

We fit model (a), (b) and (c) using the following commands:

```
ex3.model.a<-glm.bp(ex3$doctorco,ex3$medicine,w1,w2)          # for model (a)
ex3.model.b<-glm.bp(ex3$doctorco,ex3$medicine,w1,w2,w3,l3=2) # for model (b)
ex3.model.c<-glm.bp(ex3$doctorco,ex3$medicine,w1,w2,l3=0)    # for model (c)
```

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$beta
Intercept_1 Intercept_2   Gender(1)      Age(1)   Income(1)   Gender(2)
-2.08426391 -2.18707826  0.05460993  1.44174038 -0.33472517  0.57959410
Age(2)   Income(2)
3.29130665 -0.11112314
```

The coefficients for the regression in λ_3 can be seen using the command

```
> ex3.model.b$beta3
(Intercept)      Gender
 -2.7248441   0.6892711
```

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

```
n<-length(x)
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$lambda3)
bootx<-bootx1+bootx3
booty<-bootx2+bootx3
testtemp<-glm.bp(bootx,booty,w1,w2,w3,l3=2)
betafound<-c(testtemp$beta,testtemp$beta3)
results[i,]<-betafound
}
```

At the end matrix `results` contains the bootstrap values of the parameters and thus bootstrap standard errors can be obtained merely be taking the standard errors of the columns. Note that objects `bootx1, bootx2, bootx3` are used to simulate from the bivariate Poisson model through the trivariate reduction scheme.

4.4 Example 3 continued: 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 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 modelling 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 $E(X_i) = (1-p_i)(\lambda_{1i}+\lambda_{2i})$. 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 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.

		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	-1.59	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	-2.44	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		-9623.07		-9619.88		-9508.72
	AIC		19266.15		19261.77		19039.44
	BIC		19331.68		19333.86		19111.54

Table 7: Results from Fitting Diagonal Inflated Bivariate Poisson Models for the Data of Example 3; The number of parameters, AIC and BIC measures refer to the zero inflated version of models (a), (b) and (c).

4.4.1 Fitting Diagonal Inflated Bivariate Poisson Models Regression Models in R for Example 3

Again the diagonal inflated models were fitted using the functions described in section 3. The three different models were fitted using:

```
ex3.diagmodel.a<-glm.dibp(ex3$doctorco,ex3$prescrib,w1,w2)
ex3.diagmodel.b<-glm.dibp(ex3$doctorco,ex3$prescrib,w1,w2,w3,l3=2)
```

The design matrices are the same to those used in section 4.3.1. With the above commands, we can estimate diagonal models using $Discrete(2)$ 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 `glm.dibp` in order to allow for covariate effects on the mixing proportion p.

4.5 Application to 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/SPLUS 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 modelling outcomes of football games, it has been observed an excess of draws and small over-dispersion. 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 lower given by

```
model loglike AIC    BIC
 3    -758.92 1625.8 1864.3
 4    -755.61 1619.2 1857.7
 5    -745.87 1633.7 1947.3
```

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).

4.5.1 Fitting Bivariate Poisson Models Regression Models in R for Italian Football Data of Season 1991/92

Here we illustrate we can use our R/SPLUS functions to fit models implemented by Karlis and Ntzoufras (2003). Data concerning the football data of Italian Serie A league for season 1991-92 were stored in a data frame object called `ita91` with four variables: `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:

```
> 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 order to fit the models presented in Karlis and Ntzoufras we use the following code:

```
# w matrices
w1ita91_ita91[,3:4]
w2ita91_ita91[,c(4,3)]
# names of w matrices
names(w1ita91)_c( 'att', 'def' )
names(w2ita91)_c( 'att', 'def' )
# Models
# Model 1: Double Poisson
ita91model1_glm.bp( ita91$g1, ita91$g2, w1ita91, w2ita91, l3=0 )
#
# Models 2-5: bivariate Poisson models
ita91model2_glm.bp(ita91$g1, ita91$g2, w1ita91, w2ita91 )
```

```
ita91model3_glm.bp(ita91$g1, ita91$g2, w1ita91, w2ita91,
                                        as.data.frame(ita91[,3]), l3=2)
ita91model4_glm.bp(ita91$g1, ita91$g2, w1ita91, w2ita91,
                                        as.data.frame(ita91[,4]), l3=2)
ita91model5_glm.bp(ita91$g1, ita91$g2, w1ita91, w2ita91, ita91[,3:4], l3=2 )
#
# Model 6: Zero Inflated Model
ita91model6_glm.dibp(ita91$g1, ita91$g2, w1ita91, w2ita91, jmax=0 )
#
# Models 7-11: Diagonal Inflated Bivariate Poisson Models
ita91model7_glm.dibp(ita91$g1, ita91$g2, w1ita91, w2ita91, distribution=3 )
ita91model8_glm.dibp(ita91$g1, ita91$g2, w1ita91, w2ita91, jmax=1 )
ita91model9_glm.dibp(ita91$g1, ita91$g2, w1ita91, w2ita91, jmax=2 )
ita91model10_glm.dibp(ita91$g1, ita91$g2, w1ita91, w2ita91, jmax=3 )
ita91model11_glm.dibp(ita91$g1, ita91$g2, w1ita91, w2ita91, distribution=2 )
#
# Models 12: Diagonal Inflated Double Poisson Model
ita91model12_glm.dibp(ita91$g1,ita91$g2,w1ita91,w2ita91,distribution=2,l3=0)
```

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

```
> ita91model8$diagonal.distribution
[1] "Inflation Distribution: Discrete with J= 1"
>
> round(ita91model8$beta,2)
Intercept_1 Intercept_2        att1        att2        att3        att4
      -0.07       -0.57       -0.64       -0.21       -0.50       -0.21
       att5        att6        att7        att8        att9       att10
      -0.36        0.29        0.57       -0.09       -0.37        0.22
      att11       att12       att13       att14       att15       att16
       0.28        0.84        0.51       -0.14        0.02        0.10
      att17        def1        def2        def3        def4        def5
       0.18        0.75       -0.11        0.33       -0.01        0.45
       def6        def7        def8        def9       def10       def11
       0.28        0.63        0.40       -0.29       -0.70        0.21
      def12       def13       def14       def15       def16       def17
      -1.17        0.19       -0.34       -0.17       -0.16       -0.86
>
> round(ita91model8$beta3,2)
(Intercept)
      -1.47
> round(ita91model8$lambda3,2)
   1
0.23
> ita91model8$p
[1] 0.09033478
```

```
> ita91model8$theta
[1] 0.9999988
```

Note that `Intercept2` indicates intercept μ reported in Karlis and Ntzoufras (2003) while the home effect is given by the difference `Intecept1 - Intercept2`.

5 Concluding Remarks

In this article we have presented R/SPLUS 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.

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 *et al*. (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.

A APPENDIX: EM algorithms

A.1 Data Augmentation

The EM algorithm (Dempster *et al.* , 1977) is a powerful algorithm for maximum likelihood (ML) estimation for data containing missing values or they can be considered as containing missing values. 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 loglikelihood 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 loglikelihood 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

$$\begin{aligned} s_i &= E\left(X_{3i} \mid X_i, Y_i, \boldsymbol{\phi}^{(k)}\right) \\ &= \begin{cases} \lambda_{3i}^{(k)} \frac{f_{BP}\left(x_i-1, y_i-1 | \lambda_{1i}^{(k)}, \lambda_{2i}^{(k)}, \lambda_{3i}^{(k)}\right)}{f_{BP}\left(x_i, y_i | \lambda_{1i}^{(k)}, \lambda_{2i}^{(k)}, \lambda_{3i}^{(k)}\right)} & \text{if } min(x_i, y_i) > 0 \\ 0 & \text{if } min(x_i, y_i) = 0 \end{cases} \end{aligned} \tag{7}$$

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

M-step: Update the estimates by

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

where $\boldsymbol{s} = (s_1, \ldots, s_n)^T$ 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 $\boldsymbol{W}_{\kappa i}^T$ 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 $\boldsymbol{u}^T = (\boldsymbol{x}^T - \boldsymbol{s}^T, \boldsymbol{y}^T - \boldsymbol{s}^T, \boldsymbol{s}^T)$. 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

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

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 $f(x|\theta) = (1-\theta)^x \theta$, $0 \leq \theta \leq 1$, $\;x = 0, 1, \ldots$, 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 $f(x|\theta) = e^{-\theta}\theta^x/x!$, $\theta \geq 0$, $\;x = 0, 1, \ldots$, 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.

References

[1] Bohning, D., Dietz, E., Schlattmann, P., Mendonca, L. and 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

[2] Cameron, A.C. and Trivedi, P.K. (1998), *Regression analysis of count data.* Oxford University Press.

[3] Chib, S and Winkelmann, R. (2001). Markov Chain Monte Carlo Analysis of Correlated Count data. *Journal of Business and Economic Statistics*, **19**, 428-435.

[4] Dempster, A.P., Laird, N.M. and Rubin, D., (1977). Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society,* **B 39,** 1-38.

[5] Dixon, M.J. and Coles, S.G. (1997). Modelling association football scores and inefficiencies in football betting market. *Applied Statistics*, **46**, 265–280.

[6] Gan, N. (2000). General zero-inflated models and their applications. *PhD Thesis, North Carolina State University.*

[7] Ho, L.L. and Singer, J.M. (2001). Generalized least squares methods for bivariate Poisson regression. *Communications in Statistics, Theory and Methods*, **30**, 263–277.

[8] Johnson, N., Kotz, S. and Balakrishnan, N. (1997). *Discrete multivariate distributions.* Wiley - New York.

[9] Jung, R.C. and Winkelmann, R. (1993). Two aspects of Labor mobility: A bivariate Poisson regression approach. *Empirical Economics*, **18**, 543–556.

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

[11] Kocherlakota, S. and Kocherlakota, K. (1992). *Bivariate discrete distributions.* New York: Marcel Dekker.

[12] Kocherlakota, S. and Kocherlakota, K. (2001). Regression in the bivariate Poisson distribution. *Communications in Statistics - Theory and Methods*, **30**, 815–827.

[13] Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, **34**, 1–14.

[14] Li, C.S, Lu, J.C., Park, J., Kim, K. and Peterson, J. (1999). Multivariate zero-inflated Poisson models and their applications. *Technometrics*, **41**, 29–38.

[15] McLachlan, G. and Krishnan, N. (1997). *The EM algorithm and its extensions.* Wiley, New York.

[16] Meng, X.L. and 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.

[17] Munkin, M.K. and Trivedi, P.K (1999). Simulated maximum likelihood estimation of multivariate mixed-Poisson regression models, with application. *Econometrics Journal*, **2**, 29–48.

[18] Tsionas, E.G. (2001). Bayesian Multivariate Poisson Regression. *Communications in Statistics - Theory and Methods*, **30**, 243–255.

[19] Vernic, R. (1997). On the bivariate generalized Poisson distribution. *ASTIN Bulletin*, **27**, 23–31.

[20] Wahlin, J.F. (2001). Bivariate ZIP models. *Biometrical Journal*, **43**, 147–160.

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

[22] Wang, K., Lee, A.H., Yau, K.K.W. and Carrivick, P.J.W. (2003). A bivariate zero inflated Poisson regression model to analyze Occupational Injuries. *Accident Analysis and Prevention*, **35**, 625-629