DGLMExtPois: Advances in Dealing with Over and Under-dispersion in a Double GLM Framework

by Antonio J. Sáez-Castillo, Antonio Conde-Sánchez and Francisco Martínez

Abstract In recent years the use of regression models for under-dispersed count data, such as COM-Poisson or hyper-Poisson models, has increased. In this paper the DGLMExtPois package is presented. DGLMExtPois includes a new procedure to estimate the coefficients of a hyper-Poisson regression model within a GLM framework. The estimation process uses a gradient-based algorithm to solve a nonlinear constrained optimization problem. The package also provides an implementation of the COM-Poisson model, proposed by Huang (2017), to make it easy to compare both models. The functionality of the package is illustrated by fitting a model to a real dataset. Furthermore, an experimental comparison is made with other related packages, although none of these packages allow you to fit a hyper-Poisson model.

1 Introduction

The Poisson regression model is the most common framework for modeling count data, such as the number of accidents on a stretch of road, the number of visits to the doctor, the number of recreational or shopping trips, etc (Cameron and Trivedi, 2013; Hilbe, 2011; Winkelmann, 2008). This model is constrained by its equi-dispersion assumption, that is, the mean is equal to the variance. However, this constraint is not usually met in real applications due to the existence of over-dispersion (the variance is greater than the mean) or under-dispersion (the variance is less than the mean). As over-dispersion is more common than under-dispersion, the former has received more attention from the research community.

Although the number of probabilistic models for count data affected by under-dispersion is not as high as that for over-dispersion, in recent years a significant effort has been made to provide suitable alternatives for applied researches to fit the data when the variance is below the mean. Generalized Poisson (Consul and Famoye, 1992; Wang and Famoye, 1997; Famoye et al., 2004; Zamani and Ismail, 2012; Grover et al., 2015; Sellers and Morris, 2017), Double Poisson (Efron, 1986; Zou et al., 2013; Sellers and Morris, 2017), Gamma count (Winkelmann, 1995; Oh et al., 2006; Lord et al., 2010; Daniels et al., 2010, 2011; Zeviani et al., 2014; Sellers and Morris, 2017), discrete Weibull (McShane et al., 2008; Ong et al., 2015; Klakattawi et al., 2018; Yoo, 2019) or Extended Poisson Process (Faddy and Smith, 2011; Smith and Faddy, 2016) are examples of models that may facilitate, in a regression context, a flexible dispersion structure wherein the Poisson assumption of equi-dispersion cannot be admitted because of the existence of factors that determine an excess or a shortfall of variability. In any case, Conway-Maxwell-Poisson, from now on CMP, is probably the most widely used model to fit data in the presence of under-dispersion. Sellers and Shmueli (2010) presented a formulation of CMP in a regression context and demonstrated that the model could provide good fits to data not only affected by over-dispersion but also by under-dispersion, in the presence of covariates. The number of subsequent papers applying the model in different contexts is noteworthy (Sellers et al., 2012; Sellers and Morris, 2017; Chatla and Shmueli, 2018; Abdella et al., 2019).

Nevertheless, the use of this CMP model has also been subjected to criticism because of some limitations. First, there are some convergence problems in the calculation of the normalizing constant, the mean and the variance (since there are no closed expressions for them) (Francis et al., 2012). Second, and this is in our opinion its most important drawback, the original implementation of Sellers and Shmueli (2010) and, therefore, all other applications based on it, consider a log-linear relationship between the covariates and the location parameter instead of the mean, so that the regression coefficients cannot be interpreted in terms of the effect size. In fact, some authors have considered a reparameterization looking for a central tendency parameter (Guikema and Coffelt, 2008; Lord et al., 2010; Francis et al., 2012; Chanialidis et al., 2018), although it is not really the mean.

Fortunately, Huang (2017) has recently implemented a new CMP parametrization where covariates are log-linearly related to the mean and the dispersion parameter, which has been continued in subsequent works (Huang and Kim, 2019; Ribeiro et al., 2019; Forthmann et al., 2019). The mpcmp package is based on the work of Huang (2017). The hyper-Poisson regression model (Saez-Castillo and Conde-Sanchez, 2013; Khazraee et al., 2015; Saez-Castillo and Conde-Sanchez, 2017), hereafter hP, provided an alternative that also presented both characteristics: on the one hand, covariates are introduced in the equation of the mean, commonly using a log-linear link, with regression
coefficients measuring the effect size; on the other hand, the dispersion parameter may also be related to the covariates to facilitate a better fit of datasets which may be affected by over- or under-dispersion. Unfortunately, although the authors provided the necessary R code for a complete fit, the computational time needed to fit real data with medium or high sample size was so huge that its usefulness in real applications was very limited.

That is why our main aim in this paper is to improve the estimation procedure of the hP model. For this purpose, we have developed an R package, called DGLMExtPois, in which the maximum likelihood optimization procedure has been accelerated. Of course, the package includes the common methods and functions that allow for a complete analysis, including model diagnostics. The package also provides the Huang (2017) implementation of the CMP model that admits covariates in the dispersion parameter, mainly to facilitate comparisons between hP and CMP fits. In conclusion, what we want to demonstrate is the usefulness of CMP and hP models in a genuine GLM framework to test the significance and to assess the size effect of any explanatory variable on the mean value. At the same time, other explanatory variables may take part in the explanation of the dispersion structure, providing over- or under-dispersion depending on their different values.

Comparison with other packages

There is currently no other package in R that estimates the hyper-Poisson model. There are two other packages that fit a COM-Poisson model: the mpcmp package that estimates the Huang’s GLM implementation (Huang, 2017), named CMPp, and the COMPoissonReg package that estimates the (Sellers and Shmueli, 2010) model, denoted as CMPp. The DGLMExtPois package also fits the CMPp model applying an optimization process similar to the one used in fitting hP model. The CMPp model was included in the DGLMExtPois package because, at the time the package was developed, no other package estimated this model with covariates in the dispersion parameter. However, recently the mpcmp package has included this feature. Table 1 summarizes this comparison along with some additional features included in these packages for the analysis of the previously mentioned models.

<table>
<thead>
<tr>
<th>Feature</th>
<th>DGLMExtPois</th>
<th>mpcmp</th>
<th>COMPoissonReg</th>
</tr>
</thead>
<tbody>
<tr>
<td>hP model (μ, γ)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CMP model (μ, ν)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CMP model (θ, ν)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Covariates in dispersion</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>parameter</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Zero inflated models</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1: Feature comparison between packages DGLMExtPois (version 0.2.1), mpcmp (version 0.3.6) and COMPoissonReg (version 0.7.0).

The remainder of this paper is structured as follows. First, we theoretically describe the hP model and the main estimation results for fitting data. Second, we do the same for the CMP model. After that, we describe the DGLMExtPois package, illustrating the fit of a real dataset. Next, we include a comparison of fits of hP, CMP, and CMP models with different well-known datasets. Finally, we carry out a simulation to assess the performance of the estimates and standard errors of the hP model and discuss the optimization process.

2 The hP model

Let us consider Y as count variable of a hyper-Poisson distribution, whose probability mass function is (Saez-Castillo and Conde-Sanchez, 2013)

\[
P(Y = y) = \frac{1}{\Gamma(1 + \gamma)} \frac{\lambda^y}{(\gamma)^y}, \quad \lambda, \gamma > 0, \tag{1}
\]

where

\[
\Gamma(1 + \gamma) = \sum_{y=0}^{\infty} \frac{\lambda^y}{(\gamma)^y}.
\]

and \((\gamma)^y = \Gamma(\gamma + y)/\Gamma(\gamma)\). The parameter \(\gamma\) is known as the dispersion parameter because for \(\gamma > 1\) the distribution is over-dispersed and for \(\gamma < 1\) it is under-dispersed (for \(\gamma = 1\) we have the Poisson distribution).
Let \((X_{1,1}, \ldots, X_{n_1}) \in \mathbb{R}^{n_1}\) and \((Z_{1,1}, \ldots, Z_{n_2}) \in \mathbb{R}^{n_2}\) be two sets of explanatory variables or covariates. These two sets can be disjoint or overlapping (they can even be equal). In an \(hP_{\mu, \gamma}\) GLM model, \(Y\) follows a hyper-Poisson distribution whose mean, \(\mu\), and dispersion parameter, \(\gamma\), are functions of the covariates. In our case, log-linear relations are always considered, so \(\log \mu = X\beta\) and \(\log \gamma = Z\delta\), such that \(X = (1, X_{1,1}, \ldots, X_{n_1})\) and \(Z = (1, Z_{1,1}, \ldots, Z_{n_2})\) the design matrices, and \(\beta = (\beta_0, \beta_1, \ldots, \beta_{n_1})\) and \(\delta = (\delta_0, \delta_1, \ldots, \delta_{n_2})\) the regression coefficients. Moreover, the parameter \(\lambda\) is the solution of
\[
\mu = \exp(X\beta) = \sum_{y=0}^{\infty} y \times P[Y = y].
\]

Saez-Castillo and Conde-Sanchez (2013) include a study of the relationship between \(\lambda(\mu, \gamma)\), which can be interpreted as a location parameter, and the mean, \(\mu\), for a wide range of \(\gamma\) values.

The \(hP\) distribution belongs to the exponential family only for a given \(\gamma\) value. This justifies that the \(hP_{\mu, \gamma}\) model can be considered as a member of the GLM family. Moreover, since the model also allows you to introduce covariates in the dispersion parameter, we refer to it as double GLM, although we highlight that the \(hP\) distribution does not properly belong to the exponential family for unknown \(\gamma\).

The previous definition of the \(hP_{\mu, \gamma}\) model (Saez-Castillo and Conde-Sanchez, 2013) was based on an alternative equation, different from (2). Concretely, the authors proposed a closed equation that involves the mean, the variance and the normalizing constant. The substitution of the previous equation by equation (2) and the change of the optimization algorithm have been the key points in the great reduction of the computational time needed to fit real data provided by the \texttt{DGLMEExtPois} package.

To this end, we consider maximum likelihood for the estimation of the regression coefficients. Letting \((Y_i, X_{i,1}, \ldots, X_{i,n_1}, Z_{i,1}, \ldots, Z_{i,n_2})\), \(i = 1, \ldots, n\), be a random sample, the log-likelihood is given by
\[
l_{hP}(\beta, \delta) = \sum_{i=1}^{n} \{Y_i \log (\lambda(\mu_i, \gamma_i)) - \log (\Gamma (\gamma_i + Y_i)) + \log (\Gamma (\gamma_i)) - \log (\Gamma (1, 1; \lambda(\mu_i, \gamma_i)))\}
\]
where
\[
\log \mu_i = x_i^T \beta
\]
and
\[
\log \gamma_i = z_i^T \delta,
\]
being \(x_i^T\) and \(z_i^T\) the \(i\)th row of the design matrices and each \(\lambda_i = \lambda(\mu_i, \gamma_i)\) is a solution of (2) for a given \(\mu_i\) and \(\gamma_i\). This converts the maximum likelihood procedure into a nonlinear constrained optimization problem where the objective function is \((n + q_1 + q_2 + 2)\)-dimensional, depending on \(n\) nuisance \(\lambda\) parameters, \(q_1 + 1\) regression coefficients for the mean, \(\beta\), and \(q_2 + 1\) regression coefficients for the dispersion parameter, \(\delta\), and it is constrained by \(n\) non-linear equations from (2). We have implemented it using a sequential quadratic programming (SQP) gradient-based algorithm (Kraft, 1994); concretely, the \texttt{NLOPT_LD_SLSQP} algorithm included in the \texttt{nloptr} package (Ypma et al., 2022).

In the implementation, it was necessary to compute the gradient of the objective function and the gradient of restriction (2), in which the reparameterization of \(\lambda' = \log \lambda\) has been considered to ensure the positivity of \(\lambda\). These expressions have been included in the appendix.

Moreover, the following results are verified:

**Result 1** For a sample \(Y_1, \ldots, Y_n\) from a hyper-Poisson the score function with respect to \(\beta\) is
\[
S_\beta = \frac{\partial l_{hP}}{\partial \beta} = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{V(\mu_i, \gamma_i)} x_i^T
\]
where \(V(\mu_i, \gamma_i)\) is the variance of an \(hP\) with a probability mass function given by (1), which depends on \(\mu_i\) and \(\gamma_i\). This way, the MLE of \(\beta\) can be characterized as the solution to the usual GLM score equations.

**Result 2** For a sample of size \(n\) from a hyper-Poisson it is verified that
\[
S_\delta = \frac{\partial l_{hP}}{\partial \delta} = \sum_{i=1}^{n} \left\{ (Y_i - \mu_i) \frac{\text{Cov}(Y_i \phi(\gamma_i + Y_i))}{V(\mu_i, \gamma_i)} - \psi(\gamma_i + Y_i) + E[\psi(\gamma_i + Y_i)] \right\} \gamma_i x_i^T
\]
where \(\psi()\) is the digamma function.

The proof is included in the appendix. We have obtained estimates of the standard errors of the maximum likelihood estimators of \(\beta\) and \(\delta\) based on Cramer-Rao bound, which requires a closed expression of the Fisher information matrix.
Result 3 The Fisher information matrix in the sample can be approximated by

\[
I_n (\hat{\beta}, \hat{\delta}) = \begin{pmatrix} \sum_{i=1}^{n} E \left[ S_\beta S_\beta^t \right] & 0 \\ 0 & \sum_{i=1}^{n} E \left[ S_\delta S_\delta^t \right] \end{pmatrix}
\]

where

\[
\sum_{i=1}^{n} E \left[ S_\beta S_\beta^t \right] = \sum_{i=1}^{n} \frac{\mu_i^2}{V(\mu_i, \gamma_i)} x_i x_i^t
\]

\[
\sum_{i=1}^{n} E \left[ S_\delta S_\delta^t \right] = \sum_{i=1}^{n} \left[ \text{Var} (\psi (\gamma_i + Y_i)) - \frac{\text{Cov} (Y_i, \psi (\gamma_i + Y_i))}{V(\mu_i, \gamma_i)} \right] \gamma_i^2 z_i z_i^t.
\]

are evaluated in the maximum likelihood estimates, \( \hat{\beta} \) and \( \hat{\delta} \). The proof is included in the appendix.

Standard error estimates would appear in the diagonal of the inverse of \( I_n (\hat{\beta}, \hat{\delta}) \). We have not experienced any difficulties in this computation, nor in the inversion of \( \sum_{i=1}^{n} E \left[ S_\beta S_\beta^t \right] \). In contrast, \( \sum_{i=1}^{n} E \left[ S_\delta S_\delta^t \right] \) is sometimes nearly singular, so the standard errors of \( \hat{\delta} \) are not always estimated and, sometimes, are quite large. That is the reason why we recommend assessing the significance of covariates included in \( Z \) using the likelihood ratio test (LRT) instead of the Wald test, since it requires a convenient estimation of \( \hat{\delta} \) standard errors.

3 The \( CMP_{\theta, \nu} \) and \( CMP_{\mu, \nu} \) models

The probabilities of the CMP distribution are given by

\[
P \left[ Y = y \right] = \frac{1}{Z(\theta, \nu)} \frac{\theta^y}{y^!}, \quad \theta, \nu > 0,
\]

with

\[
Z(\theta, \nu) = \sum_{y=0}^{\infty} \frac{\theta^y}{y^!}.
\]

Here, \( \nu < 1 \) corresponds to over-dispersion and \( \nu > 1 \) to under-dispersion. The distribution belongs to the bi-parametric exponential family (Huang, 2017). In this distribution, \( \theta \) is a location parameter with a similar role to \( \lambda \) for the \( hP \) distribution. It is worth emphasizing that \( \theta \) is not a mean, see for example Francis et al. (2012) for the exact relationship between \( \theta \) and \( \mu \).

If \( Y \) is again the response count variable and \( X \) and \( Z \) are defined as above, then we denote as \( CMP_{\theta, \nu} \) the model wherein

\[
\log \theta = X \beta
\]

and

\[
\log \nu = Z \delta.
\]

This is the model with constant \( \nu \) defined by Sellers and Shmueli (2010), later extended to consider covariates in the \( \nu \) equation (Sellers et al., 2012; Chatla and Shmueli, 2018). In contrast, the model defined by Huang (2017) is given as

\[
\log \mu = X \beta
\]

and

\[
\log \nu = Z \delta,
\]

where additionally equation (2) must be verified for each observation, and the probabilities in (6). We denote this model as \( CMP_{\mu, \nu} \).

Estimation and inference for the \( CMP_{\theta, \nu} \) is described, for example, in Sellers and Shmueli (2010). The \texttt{COMPoissonReg} package (Sellers et al., 2018) provides adequate methods and functions to estimate such a model. Huang (2017) also provides theoretical results describing the procedure to obtain estimates of \( \beta \) and \( \delta \) using maximum-likelihood and related inference results for \( CMP_{\mu, \nu} \), which are included in the \texttt{mpcmp} package.
Function | Description
---|---
glm.hP | Fits an \( hP_{\mu, \gamma} \) model
glm.CMP | Fits an \( CMP_{\mu, \nu} \) model
lrtest | Likelihood ratio chi-squared test
residuals | Extracts model residuals (Pearson, response and quantile) and produces a simulated envelope
fitted | Fitted values of the model
plot | Plot of residuals against fitted values and a Normal Q-Q plot
predict | Predictions
confint | Confidence intervals for \( \beta \) regression coefficients
dhP | Probability mass function for an \( hP \) distribution
phP | Distribution function for an \( hP \) distribution
rhP | Random generation for an \( hP \) distribution
hP_expected | Expected frequencies for \( hP_{\mu, \gamma} \) model
CMP_expected | Expected frequencies for \( CMP_{\mu, \nu} \) model

Table 2: Main functions in the DGLMExtPois package.

4 The DGLMExtPois package

At present, packages such as COMPoissonReg provide what a practitioner commonly requires to fit real data. Our main criticism, as we have mentioned in the introduction section, is that \( \beta \) regression coefficients cannot be interpreted in terms of the effect size. In this sense, mpcmp is, in our opinion, a better proposal for practitioners looking for a clearer interpretation of the \( \beta \) regression coefficients in relation to the effect of each explanatory variable on the mean of the response variable.

In this context, what the DGLMExtPois package provides is, first, a much faster implementation of the \( hP_{\mu, \gamma} \) model than the previous existing R code and, second, our own implementation of \( CMP_{\mu, \nu} \) to facilitate the comparison of the models.

DGLMExtPois has been created by trying to reproduce the syntax of GLM fits. Thus, functions glm.hP and glm.CMP provide the corresponding fits for \( hP_{\mu, \gamma} \) and \( CMP_{\mu, \nu} \), respectively. There are also print and summary methods to visualize the fitted models and AIC, confint, predict, residuals and plot methods to evaluate the goodness of fit, obtain confidence intervals of \( \beta \) regression coefficients, predictions, residuals and some associated plots, such as QQ-plots and simulated envelopes. The package additionally incorporates expected, a function which calculates the marginal probabilities of the \( Y \) variable, and lrtest to get the likelihood ratio test in nested models. Other functions are provided, for example, to work with probabilities of CMP and \( hP \) distributions. A brief description of the main functions in the package is given in Table 2.

Installation

As usual, you can install the DGLMExtPois package from CRAN with the code:

```
install.packages("DGLMExtPois")
```

You may encounter problems on Windows operating systems if you are not logged in with administrator rights due to the strong dependency on the nloptr package.

5 Numerical examples

In this section the main methods and functions in the package are illustrated by fitting a real dataset. Also, several well-known datasets in regression on count variables have been used to compare COMPoissonReg and DGLMExtPois in terms of goodness of fit, computational times and reliability. Finally, some guidelines are given on the optimization process for estimating the models.

Example of application

The dataset originates from research on customer profiles for a household supplies company. An observation corresponds to census tracts within 10-mile radius around a certain store. The response
Estimate Std. Error Estimate Std. Error Estimate Std. Error
(Intercept) 2.9424 0.2072 2.9465 0.2155 2.9426 0.2051
nhu 0.0606 0.0142 0.0600 0.0147 0.0606 0.0141
aid -0.0117 0.0021 -0.0115 0.0022 -0.0117 0.0021
aha -0.0037 0.0018 -0.0038 0.0018 -0.0037 0.0018
dnc 0.1684 0.0258 0.1662 0.0269 0.1683 0.0255
ds -0.1288 0.0162 -0.1285 0.0168 -0.1288 0.0160
(Intercept) 0.6229 0.6574 0.0219 0.1407

Table 3: Estimation of coefficients and standard errors for fitted models with a constant dispersion parameter. The last row is the estimation of $\delta_0$ and its standard error.

variable is the number of customers of each census tract who visit the store and the explanatory variables are related to relevant demographic information for each tract. The dataset was analyzed in Neter et al. (1996) as an example of a Poisson regression model, since it is a classic instance of equidispersed count data. They consider the number of housing units (nhu), the average income in dollars (aid), the average housing unit age in years (aha), the distance to the nearest competitor in miles (dnc) and the distance to store in miles (ds) as covariates. Here, we consider this dataset to illustrate the performance of DGLMExtPois fitting models with all the explanatory variables in the mean equation and one of them, dnc, in the dispersion parameter equation. The dataset is also analyzed in (Bonat et al., 2018) considering the Extended Poisson-Tweedie distribution, but they also obtain an equidispersed distribution.

First, we fit a Poisson regression model, an $hP_{\mu,\gamma}$ model and a $CMP_{\mu,\nu}$ model without explanatory variables in the dispersion parameter:

```r
library(DGLMExtPois)
formula.loc <- ncust ~ nhu + aid + aha + dnc + ds # Mean formula
formula.disp0 <- ncust ~ 1 # Dispersion parameter formula
pois <- glm(formula.loc, data = CustomerProfile, family = 'poisson')
hp0 <- glm.hP(formula.loc, formula.disp0, data = CustomerProfile)
cmp0 <- glm.CMP(formula.loc, formula.disp0, data = CustomerProfile)
c(AIC(pois), AIC(hp0), AIC(cmp0))
[1] 571.0243 571.7983 573.0006
```

The functions `glm.hP` and `glm.CMP` return S3 objects of classes "glm_hP" and "glm_CMP", respectively, with information on the fitted models. The most suitable model according to AIC is Poisson. The fits for these models, shown in Table 3, indicate that there is equidispersion. In fact, if the LRT is carried out to determine whether the dispersion parameter is equal to 1 (Poisson model), the hypothesis is not rejected:

```r
lrt_hp <- -2 * (hp0$logver + logLik(pois)[1])
pchisq(lrt_hp, 1, lower.tail = FALSE)
[1] 0.2681826
lrt_cmp <- -2 * (cmp0$logver + logLik(pois)[1])
pchisq(lrt_cmp, 1, lower.tail = FALSE)
[1] 0.8777006
```

However, if we introduce covariates in the dispersion parameter we find that the equidispersion hypothesis is rejected, hence the importance of being able to include variables that determine a different variability in the model. We have checked that none of the explanatory variables lead to a statistically significant improvement in the likelihood of the $CMP_{\mu,\nu}$ model when they are introduced as covariates for the dispersion parameter. That is why, from now on, we concentrate on how to use the package with the $hP_{\mu,\gamma}$ model.

Let us now consider a new $hP_{\mu,\gamma}$ model with dnc in the dispersion parameter equation:

```r
formula.disp1 <- ncust ~ dnc # New dispersion parameter formula
hp1 <- glm.hP(formula.loc, formula.disp1, data = CustomerProfile)
```

The R Journal Vol. 14/4, December 2022 ISSN 2073-4859
The summary method provides, as usual, the regression coefficient estimates, their standard errors, the corresponding t-statistics and p-values and other details about the model:

```r
summary(hp1)
```

**Call:**

```r
glm.hP(formula.mu = formula.loc, formula.gamma = formula.disp1, data = CustomerProfile)
```

**Mean model coefficients (with log link):**

| Term   | Estimate | Std. Error | z value | Pr(>|z|) |
|--------|----------|------------|---------|----------|
| (Intercept) | 2.996083 | 0.204212   | 14.671  | < 2e-16 *** |
| nhu    | 0.054473 | 0.013960   | 3.902   | 9.54e-05 *** |
| aid    | -0.010930| 0.002068   | -5.285  | 1.26e-07 *** |
| aha    | -0.003631| 0.001751   | -2.074  | 0.0381 *    |
| dnc    | 0.156249 | 0.026233   | 5.956   | 2.58e-09 *** |
| ds     | -0.128101| 0.015636   | -8.193  | 2.55e-16 *** |

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**Dispersion model coefficients (with logit link):**

| Term   | Estimate | Std. Error | z value | Pr(>|z|) |
|--------|----------|------------|---------|----------|
| (Intercept) | 4.749    | 2.416      | 1.965   | 0.0494 * |
| dnc    | -2.782   | 2.272      | -1.224  | 0.2208   |

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

AIC: 568.1

The Wald test assesses whether the dnc variable is significant or not for the dispersion parameter. In any case, as previously mentioned, we also used the LRT to confirm the adequacy of hp1 versus the hp0 model, with a significance level of 0.05, resulting in the rejection of the constant dispersion hypothesis:

```r
lrtest(hp0, hp1)
```

**Statistics:** 5.668851

**Degrees of freedom:** 1

**p-value:** 0.01726877

Since the dispersion parameter in the hp1 model depends on dnc, its values may determine over- or under-dispersion. It is interesting to note that low values of the dnc variable show over-dispersion while high values show under-dispersion. Specifically, the point at which the change occurs is 1.7069. There are 24 out of the 110 cases with \( \gamma_i > 1 \), and thus over-dispersed, underlying \( hP(\mu_i, \gamma_i) \) distributions, see Figure 1. The plot suggests a certain correlation between under-dispersion and high response variable values.

![Figure 1: Value of the dispersion parameter according to the response variable. The points above the dashed line are greater than 1, so the distribution is over-dispersed. These points occur for low values of the response variable.](image-url)
Regression diagnostics. The plots do not show evidence against the adequacy of the model.

Figure 2: Regression diagnostics. The plots do not show evidence against the adequacy of the model.

```r
plot(CustomerProfile$ncust, hp1$gamma,
     xlab = expression(ncust[i]),
     ylab = expression(gamma[i]),
     pch = 19
)
abline(h = 1, lty = "dashed", col = "blue")
```

The plot method provides two classical plots for the model diagnostics: residuals against fitted values and a normal QQ-Plot. In relation to the residuals, Pearson, deviance and response residuals are offered, although here we only calculate the Pearson type (top row of Figure 2). The package also provides the possibility to compare in a barplot the observed marginal frequencies of the response variable with the \( \hat{h}_P \) expected and the CMP_expected functions (Figure 2c). Finally, the diagnostic can be completed by a simulated envelope for the residuals to assess the accuracy of the model on the entire dataset (Atkinson, 1981; Garay et al., 2011). The residuals method provides this diagnostic tool, and in Figure 2d the cases for which the model assumptions are rejected are highlighted in red. In this case there are 5 observations that fall outside the envelope, therefore there is no evidence against the adequacy of the model.

```r
par(mfrow = c(2, 2))
set.seed(21)
plot(hp1) # Residuals against fitted values and normal QQ plot

# Observed and expected marginal frequencies
expec_hp <- hP_expected(hp1)
barplot(expec_hp$obsvfreq[0:33], xlab = "ncust", ylab = "Frequencies")
lines(expec_hp$freq[1:33], x = c(0:32), col = "blue")
text(x = c(25, 25), y = c(10, 9), c(paste("Dif = ", round(expec_hp$dif, 2)),

chi^2 = 44.27
```

Figure 2: Regression diagnostics. The plots do not show evidence against the adequacy of the model.
# Envelope
r1 <- residuals(hp1, envelope = TRUE)
envelope_down <- apply(r1$sim_residuals, 1, min)
envelope_up <- apply(r1$sim_residuals, 1, max)
weirds <- which(r1$residuals < envelope_down | r1$residuals > envelope_up)
score_weirds <- qnorm(weirds / 111)
points(score_weirds, r1$residuals[weirds], col = "red")

## Performance comparison

In this section eight datasets, previously used in different works, have been used to compare the performance of fitting $h_{\mu, \gamma}$ and $CMP_{\mu, \nu}$ models with the DGLMExtPois package (version 0.2.1, with nloptr version 2.0.0), $CMP_{\theta, \nu}$ models with the COMPoissonReg package (version 0.7.0) and $CMP_{\mu, \nu}$ models with the mpcmp package (version 0.3.6). We think that the selected datasets cover a wide range of examples, with low, medium and large sample sizes, and under- and over-dispersion in their variability structure.

For each dataset we have tried to fit $h_{\mu, \gamma}$, $CMP_{\mu, \nu}$ and $CMP_{\theta, \nu}$ models, and we have computed some goodness of fit criteria (Lord et al., 2010), such as AIC, mean prediction bias (MPB), mean absolute deviance (MAD), mean squared predictive error (MSPE) and the Pearson statistic. Additionally, we have calculated the time needed to fit the models with these R packages. The experimentation has been carried out on a computer equipped with an Intel Core i7 processor (3.20GHz) and 64 GB of RAM.

Table 4 shows the AICs and the computational times obtained by the different models for the different datasets. The first column of the table lists the names of the datasets and the second column lists the number of observations of each dataset. The best AICs results are highlighted in bold. Unfortunately, the COMPoissonReg and mpcmp packages crash with some datasets, which is indicated by a dash in the table. In relation to the crashes, we clarify that all the fits have been carried out with the default optional arguments. To obtain a better approximation of the execution time needed to estimate a model, we have estimated each model 10 times and computed the average time required for the estimate. These average times are presented in Table 4 as follows. A value of 1.0 means the model has the fastest average estimate time and this fastest average time, in seconds, is shown in the last column. A value of, e.g., 41.4 means the average estimate time of the model is 41.4 times larger than the average of the fastest. Table 5 shows the results of the $h_{\mu, \gamma}$ and $CMP_{\mu, \nu}$ models, fitted with the DGLMExtPois package, according to other goodness of fit criteria.

**Table 4:** Comparison of models based on AIC (the smallest value is highlighted in bold) and relative average estimate time. For the estimate times a value of 1.0 means the model is the fastest on average for the dataset. A value of, e.g., 41.4 means the average estimate time of the model is 41.4 times larger than the average of the fastest. The fastest time is expressed in seconds. The symbol – means that the model could not be estimated.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>obs.</th>
<th>$h_{\mu, \gamma}$</th>
<th>$CMP_{\mu, \nu}$</th>
<th>$CMP_{\theta, \nu}$</th>
<th>mpcmp</th>
<th>$h_{\mu, \gamma}$</th>
<th>$CMP_{\mu, \nu}$</th>
<th>$CMP_{\theta, \nu}$</th>
<th>mpcmp</th>
<th>fastest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Takeover bids</td>
<td>126</td>
<td>355.1</td>
<td>354.8</td>
<td>–</td>
<td>357.8</td>
<td>1.0</td>
<td>41.4</td>
<td>–</td>
<td>4.9</td>
<td>0.7</td>
</tr>
<tr>
<td>Days absent</td>
<td>314</td>
<td>1739.2</td>
<td>1736.1</td>
<td>–</td>
<td>–</td>
<td>1.0</td>
<td>5.1</td>
<td>–</td>
<td>–</td>
<td>23.6</td>
</tr>
<tr>
<td>Cotton bolls</td>
<td>125</td>
<td>555.1</td>
<td>446.3</td>
<td>446.4</td>
<td>447.2</td>
<td>1.0</td>
<td>53.4</td>
<td>10.0</td>
<td>11.5</td>
<td>0.7</td>
</tr>
<tr>
<td>Korea</td>
<td>162</td>
<td>218.2</td>
<td>214.5</td>
<td>–</td>
<td>–</td>
<td>1.0</td>
<td>6.0</td>
<td>–</td>
<td>–</td>
<td>2.5</td>
</tr>
<tr>
<td>Toronto</td>
<td>868</td>
<td>5088.4</td>
<td>5067.4</td>
<td>5069.4</td>
<td>17.6</td>
<td>17.6</td>
<td>–</td>
<td>10.0</td>
<td>9.0</td>
<td>–</td>
</tr>
<tr>
<td>Insurance</td>
<td>64</td>
<td>407.8</td>
<td>399.0</td>
<td>469.4</td>
<td>–</td>
<td>1.8</td>
<td>1.7</td>
<td>1.0</td>
<td>7.4</td>
<td>6.7</td>
</tr>
<tr>
<td>Credit card</td>
<td>1319</td>
<td>2392.9</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1.0</td>
<td>–</td>
<td>–</td>
<td>1591.1</td>
<td></td>
</tr>
<tr>
<td>Children</td>
<td>1761</td>
<td>5380.4</td>
<td>5372.4</td>
<td>5393.2</td>
<td>5374.4</td>
<td>583.5</td>
<td>4.6</td>
<td>1.0</td>
<td>6.3</td>
<td></td>
</tr>
</tbody>
</table>

Next, the datasets used in the comparison are described, together with some comments about the fits for the different models:

- **Takeover bids.** The response variable is the number of bids received by 126 US firms that were successful targets of tender offers between 1978–1985. We consider 9 explanatory variables on the defensive actions taken by the management of the target firm, firm-specific characteristics and the intervention taken for both the mean and the dispersion parameter with a log-linear link. See Saez-Castillo and Conde-Sanchez (2013) for more details about these explanatory variables and other references where the dataset was analyzed. For this dataset the COMPoissonReg package crashes with the $CMP_{\theta, \nu}$ model. The lowest AIC is obtained by the $CMP_{\mu, \nu}$ model.
The main difference between the fits of $hP_{µ,γ}$ and $CMP_{µ,ν}$ is that the former identifies 108 out of 126 of the cases as being under-dispersed, whereas the latter identifies only 88. In general, the goodness of fit measures are quite similar, being somewhat smaller for the $hP_{µ,γ}$ model. In addition, the $CMP_{µ,ν}$ model obtained with the DGLMExtPois package is slower, but with a lower AIC, than the one obtained with the mpcmp package.

- **Days absent.** The sample refers to 314 students sampled from two urban high schools. The response variable is the number of days absent from high school, and the explanatory variables for both the mean and the dispersion parameter are gender, math’s score (standardized score out of 100) and academic program (General, Academic and Vocational). The dataset has been analyzed, among others, by Huang (2017). The COMPoissonReg package crashes again with the $CMP_{µ,ν}$ model, as does the mpcmp package with the $CMP_{µ,ν}$ model. $CMP_{µ,ν}$ achieves a slightly better AIC than $hP_{µ,γ}$, but again requires more computational time and the goodness of fit measures are quite similar. In relation to the dispersion structure, both models identify all the cases as being over-dispersed.

- **Cotton bolls.** Here, the response variable is the number of bolls produced by 125 cotton plants. The explanatory variables, again for both the mean and the dispersion parameter, are the growth stage (vegetative, flower-bud, blossom, fig and cotton boll), and the defoliation level (with 5 levels). The model also includes the square of the defoliation level. It reproduces the analysis carried out by Zeviani et al. (2014). This is one of the cases where COMPoissonReg provides a $CMP_{µ,γ}$ fit, although the one provided by the $CMP_{µ,ν}$ model obtains a lower AIC (very similar to both the DGLMExtPois and mpcmp packages). Here $hP_{µ,γ}$ performs worse and obtains a higher AIC. Another interesting point in relation to the $CMP_{µ,γ}$ fit is that, regardless of the strong under-dispersion of the complete dataset, the introduction of covariates in the dispersion parameter allows us to identify 5 cases that are actually described with over-dispersed distributions. Note also that $hP_{µ,γ}$ is the fastest approach.

- **Korea crashes.** The dataset contains crash count data, which is the response variable, collected at 162 railway-highway crossings in Korea. The explanatory variables include the average daily vehicle traffic (in logarithmic scale) and 7 characteristics of the crossing. See, for example, Khazraee et al. (2015) for more details. For this dataset both the COMPoissonReg and mpcmp packages crash. The $CMP_{µ,ν}$ fit obtains the best AIC, although the rest of the goodness of fit measures are quite similar to those provided by the $hP_{µ,γ}$ fit. Once the effect of the covariates is taken into account, both fits detect over 50 cases with under-dispersed underlying distributions.

- **Toronto crashes.** The dataset provides Toronto intersection crash data. The response variable is the number of crashes at 868 intersections, and the explanatory variables for the mean and the dispersion parameter are the average annual daily traffic in the major and minor approaches to the intersection. These data have been widely used as an example of over-dispersed crash data. See, for example, Khazraee et al. (2015). Whereas COMPoissonReg once again leads to a crash in the fit of a $CMP_{µ,ν}$ model, $CMP_{µ,γ}$ and $hP_{µ,γ}$ provide similar fits in terms of goodness of fit measures. The $CMP_{µ,ν}$ model obtained with the DGLMExtPois package has a slightly lower AIC than the model obtained with the mpcmp package. In this case, the mpcmp package is the fastest estimating the model. There are 2 under-dispersed cases for the $hP_{µ,γ}$ fit.

- **Insurance claims.** Data consist of the numbers of policyholders of an insurance company who were exposed to risk, and the number of car insurance claims made by those policyholders in the third quarter of 1973. The response variable is the number of claims, with the number of policyholders being an offset for the mean. Apart from this offset, there are 3 explanatory

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$hP_{µ,γ}$</th>
<th>$CMP_{µ,ν}$</th>
<th>$hP_{µ,γ}$</th>
<th>$CMP_{µ,ν}$</th>
<th>$hP_{µ,γ}$</th>
<th>$CMP_{µ,ν}$</th>
<th>$hP_{µ,γ}$</th>
<th>$CMP_{µ,ν}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Takeover bids</td>
<td>0.0005</td>
<td>-0.0834</td>
<td>0.83</td>
<td>0.90</td>
<td>1.66</td>
<td>1.99</td>
<td>128.4</td>
<td>127.4</td>
</tr>
<tr>
<td>Days absent</td>
<td>0.0287</td>
<td>0.0257</td>
<td>4.51</td>
<td>4.51</td>
<td>40.20</td>
<td>40.20</td>
<td>374.3</td>
<td>370.2</td>
</tr>
<tr>
<td>Cotton bolls</td>
<td>0.0009</td>
<td>0.0183</td>
<td>0.98</td>
<td>1.00</td>
<td>1.60</td>
<td>1.73</td>
<td>29.6</td>
<td>131.4</td>
</tr>
<tr>
<td>Korea</td>
<td>-0.0052</td>
<td>0.0206</td>
<td>0.36</td>
<td>0.35</td>
<td>0.24</td>
<td>0.24</td>
<td>119.9</td>
<td>-</td>
</tr>
<tr>
<td>Toronto</td>
<td>-0.0121</td>
<td>-0.0006</td>
<td>4.14</td>
<td>4.14</td>
<td>32.67</td>
<td>32.66</td>
<td>951.2</td>
<td>935.6</td>
</tr>
<tr>
<td>Insurance</td>
<td>0.0830</td>
<td>-0.0254</td>
<td>3.51</td>
<td>3.45</td>
<td>25.80</td>
<td>25.33</td>
<td>48.0</td>
<td>54.4</td>
</tr>
<tr>
<td>Credit card</td>
<td>0.0156</td>
<td>-0.066</td>
<td>0.95</td>
<td>0.95</td>
<td>1.44</td>
<td>1.44</td>
<td>1653.1</td>
<td>1718.1</td>
</tr>
<tr>
<td>Children</td>
<td>-0.0000</td>
<td>-0.0017</td>
<td>0.95</td>
<td>0.95</td>
<td>1.44</td>
<td>1.44</td>
<td>-11.07</td>
<td>-11.07</td>
</tr>
</tbody>
</table>

Table 5: Comparison of models based on additional goodness of fit criteria. MPB: mean prediction bias; MAD: mean absolute deviance; MSPE: mean squared predictive error and Pearson statistic. The best results are highlighted in bold.
variables for the mean and the dispersion parameter: district of residence of the policyholder, car group and an ordered factor with the age of the insured in 4 groups. The $\text{CMP}_{\mu,\nu}$ model of the \texttt{mpcmp} package obtains the best AIC, but it is the slowest in terms of computational time. On this occasion the \texttt{COMPoissonReg} package does not fail and is the fastest, however its AIC is clearly the highest among the estimated models. Most of the cases show under-dispersion (52 and 44 for $hP_{\mu,\gamma}$ and $\text{CMP}_{\mu,\nu}$ respectively).

- **Credit card reports.** The response variable is the number of major derogatory reports. Three explanatory variables are considered: age in years plus twelfths of a year, yearly income (in USD 10,000) and average monthly credit card expenditure. For this dataset the only model that can be estimated is $hP_{\mu,\gamma}$. This is a case of over-dispersed data, but as a result of introducing covariates in the dispersion parameter, 5 under-dispersed cases appear in the $hP_{\mu,\gamma}$ model. In any case a negative binomial model fits better due to the strong over-dispersion of data.

- **Children.** In this dataset the response variable is the number of children a woman has. The explanatory variables are: age of the woman in years, years of education, nationality of the woman (German or not), belief in God (with 6 categories) and attended university (yes or no). This is an example of equidispersed data (Tutz, 2011), although our models detected under-dispersion (1164 and 1238 cases for $hP_{\mu,\gamma}$ and $\text{CMP}_{\mu,\nu}$ respectively). In this case the \texttt{mpcmp} package provides a very fast $\text{CMP}_{\mu,\nu}$ fit, but with a slightly higher AIC.

In summary, the $hP_{\mu,\gamma}$ model is the only one that can provide fits for all the datasets. Regarding the AIC, the $\text{CMP}_{\mu,\nu}$ model seems to be better than the $hP_{\mu,\gamma}$ model, although the latter usually requires less estimation time. As for the comparison of the $\text{CMP}_{\mu,\nu}$ model in the \texttt{DGLMEstPois} and \texttt{mpcmp} packages, the $\text{CMP}_{\mu,\nu}$ model of the \texttt{mpcmp} package fails in more datasets and provides better results in terms of AIC for only one dataset.

**Simulation**

In order to assess the performance of the estimates and standard errors in finite samples of the $hP_{\mu,\gamma}$ model a simulation has been performed. The simulation process consists of the following steps:

1. 500 values of two covariates $X$ and $Z$ are generated from a uniform distribution on the interval $(0,1)$.
2. The mean and $\gamma$ are obtained as $\mu_i = \exp(\beta_0 + \beta_1 x_i)$ and $\gamma_i = \exp(\delta_0 + \delta_1 z_i)$, $i = 1, \ldots, 500$, for several values of the regression coefficients $\beta_0, \beta_1, \delta_0$ and $\delta_1$.
3. The $\lambda$ parameter is obtained as solution of (2) by means of the base R \texttt{optimize} function.
4. Each value of $Y$ in the sample is randomly generated using the \texttt{rhp} function.
5. Once the dataset $(Y, X, Z)$ is available, the coefficients of the $hP_{\mu,\gamma}$ model are estimated.

This process was repeated 1000 times and the average of the estimates and standard errors, as well as the standard deviation of the estimates, were calculated. The standard deviation of the estimates allows us to check whether the standard errors have been calculated correctly, while the average of the estimates allows us to determine whether there is a bias in the estimation of the coefficients. The results obtained for different values of the coefficients are shown in Table 6.

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>sd</td>
</tr>
<tr>
<td>Comb. I: $\beta = (1, 0.5)$</td>
<td>0.998</td>
<td>0.0504</td>
</tr>
<tr>
<td>Comb. II: $\beta = (1, 0.5)$</td>
<td>0.998</td>
<td>0.0552</td>
</tr>
<tr>
<td>Comb. III: $\beta = (1, 0.5)$</td>
<td>0.998</td>
<td>0.0490</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>$\delta_0$</th>
<th>$\delta_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>sd</td>
</tr>
<tr>
<td>Comb. I: $\delta = (0.5, -1)$</td>
<td>0.461</td>
<td>0.4237</td>
</tr>
<tr>
<td>Comb. II: $\delta = (0.5, 0.25)$</td>
<td>0.456</td>
<td>0.3834</td>
</tr>
<tr>
<td>Comb. III: $\delta = (0, -0.5)$</td>
<td>-0.058</td>
<td>0.4943</td>
</tr>
</tbody>
</table>

Table 6: Simulation process for $hP_{\mu,\gamma}$ models. For each combination of coefficients the mean and standard deviation (sd) of the estimates were calculated, as well as the mean of the standard errors (s.e.).
The first combination of coefficients ($\beta = (1, 0.5)$ and $\delta = (0.5, -1)$) allows for under- and over-dispersion situations. The coefficients in the second combination ($\beta = (1, 0.5)$ and $\delta = (0.5, 0.25)$) determine a model with over-dispersion only. And finally, the third combination of coefficients ($\beta = (1, 0.5)$ and $\delta = (0, -0.5)$) corresponds to an under-dispersed model. It can be seen that the estimates of the $\beta$ coefficients are practically unbiased and very precise, while those of the $\delta$ coefficients are more biased and much less precise. In any case the mean standard errors are very close to the standard deviations of the 1000 estimates, so the calculation of these standard errors seem to be correct.

A similar simulation has been performed for the $CMP_{\mu, \nu}$ model in the DGLMExtPois package, and the results are shown in Table 7. It can be seen that, although the estimates of the $\delta$ coefficients are still less precise than those of the $\beta$ coefficients, these estimates are less biased and more precise than those of the $hP_{\mu, \gamma}$ model. In this sense, the $CMP_{\mu, \nu}$ model leads to better estimates than the hyper-Poisson model.

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\delta_0$</th>
<th>$\delta_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comb. I: $\beta = (1, 0.5)$</td>
<td>0.998</td>
<td>0.0497</td>
<td>0.0498</td>
<td>0.950</td>
</tr>
<tr>
<td>Comb. II: $\beta = (1, 0.5)$</td>
<td>0.999</td>
<td>0.0383</td>
<td>0.0386</td>
<td>0.501</td>
</tr>
<tr>
<td>Comb. III: $\beta = (1, 0.5)$</td>
<td>0.998</td>
<td>0.0557</td>
<td>0.0556</td>
<td>0.502</td>
</tr>
</tbody>
</table>

Table 7: Simulation for $CMP_{\mu, \nu}$ models. For each combination of coefficients the mean and standard deviation (sd) of the estimates were calculated, as well as the mean of the standard errors (s.e.).

The optimization process

As previously mentioned, the estimation of $hP_{\mu, \gamma}$ and $CMP_{\mu, \nu}$ models in the DGLMExtPois package was performed using the SLSQP optimizer included in the nloptr package. Because the SLSQP algorithm uses dense-matrix methods, it requires $O(m^2)$ storage and $O(m^3)$ time in $m$ dimensions (where $m$ is the number of estimated parameters, i.e., the number of observations plus the number of model coefficients), which makes it less practical for optimizing more than a few thousand parameters. Table 8 shows the datasets from the above performance comparison sorted by the dimension of the objective function when fitting an $hP_{\mu, \gamma}$ model. Also, the average estimation time for 10 estimations and the number of iterations taken by the optimizer are listed. Clearly, the dimension of the objective function has the greatest impact on optimization time, while the number of iterations plays a less important role.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dimension</th>
<th>Average Time</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insurance</td>
<td>84</td>
<td>12.0</td>
<td>987</td>
</tr>
<tr>
<td>Takeover bids</td>
<td>146</td>
<td>0.7</td>
<td>78</td>
</tr>
<tr>
<td>Cotton bolls</td>
<td>155</td>
<td>0.7</td>
<td>96</td>
</tr>
<tr>
<td>Korea</td>
<td>180</td>
<td>2.5</td>
<td>283</td>
</tr>
<tr>
<td>Days absent</td>
<td>324</td>
<td>23.6</td>
<td>558</td>
</tr>
<tr>
<td>Toronto</td>
<td>874</td>
<td>158.8</td>
<td>111</td>
</tr>
<tr>
<td>Credit card</td>
<td>1327</td>
<td>1591.1</td>
<td>135</td>
</tr>
<tr>
<td>Children</td>
<td>1781</td>
<td>3221.0</td>
<td>61</td>
</tr>
</tbody>
</table>

Table 8: Optimization process for $hP_{\mu, \gamma}$ models. For each dataset, the dimension of the objective function, the average optimization time (in seconds) and the number of iterations are shown.

The nloptr optimization function from the nloptr package was used to fit the models. This function allows several termination conditions for the optimization process, of which the DGLMExtPois package uses, by default, the following:
• A maximum number of iterations of 1000 and,
• a fractional tolerance (xtol_rel) is used in the parameters $x$, so that the optimizer stops when $|\Delta x_i|/|x_i| < xtol_rel$, with a default value for xtol_rel of 0.01.

Information on the optimization process can be obtained through the component nloptr, of class "nloptr", included in the S3 object returned by the glm.hP function:

```
library(mpcmp)
formula <- daysabs ~ gender + math + prog
fit <- glm.hP(formula, formula, data = attendance)
fit$nloptr$message # termination condition
[1] "NLOPT_XTOL_REACHED: Optimization stopped because xtol_rel or xtol_abs (above) was reached."
fit$nloptr$iterations # number of iterations
[1] 558
```

The user can modify the termination conditions. For example, in the following code the second call toglm.hP removes the termination condition on fractional tolerance used in the parameters and set the maximum number of iterations:

```
fit1 <- glm.hP(formula, formula, data = attendance) # default termination conditions
AIC(fit1)
[1] 1739.192
fit1$nloptr$iterations
[1] 558
fit2 <- glm.hP(formula, formula, data = attendance,
    opts = list(xtol_rel = -1, maxeval = 250))
AIC(fit2)
[1] 1739.369
fit2$nloptr$iterations
[1] 250
```

Another interesting termination condition is an approximate maximum optimization time, expressed in seconds:

```
fit3 <- glm.hP(formula, formula, data = attendance,
    opts = list(xtol_rel = -1, maxeval = -1, maxtime = 5))
AIC(fit3)
[1] 1741.672
fit3$nloptr$iterations
[1] 124
```

The user can also see how the optimization process evolves:

```
fit4 <- glm.hP(formula, formula, data = attendance,
    opts = list(xtol_rel = -1, maxeval = 5, print_level = 1))
iteration: 1
  f(x) = 1550.509229
iteration: 2
  f(x) = -0.000000
iteration: 3
  f(x) = 33443.986687
iteration: 4
  f(x) = 2980.009033
iteration: 5
  f(x) = 1444.775141
```

6 Conclusion

In this paper we have presented DGLMExtPois, the only package on CRAN that allows fitting count data using the hyper-Poisson model. This model is able to fit data with over- or under-dispersion, or both for different levels of covariates. Additionally, DGLMExtPois can also fit COM-Poisson models, considering covariates in the mean and the dispersion parameter. The mpcmp package also presents this feature, so the results obtained by both packages can be compared. We have found that the mpcmp
package fails to fit some datasets and the fits obtained are, in general, somewhat worse than those obtained with the DGLMExtPois package. The COMPoissonReg package allows covariates in the location parameter, but it models a location parameter instead of the mean. In addition, this latter package fails to fit many of the datasets used in the experimentation.

The two models implemented by the DGLMExtPois package achieved similar results in the experimentation, with the COM-Poisson model obtaining, in general, slightly lower AICs and needing more computational time to fit the models. We have also verified, by means of a simulation, that the COM-Poisson model is more accurate in estimating the coefficients of the covariates included in the dispersion parameter. But, overall, it cannot be said that one model is clearly better than the other. Thus, two competitive models are available to deal with count data. As shown in the example, the possibility of including covariates in the dispersion parameter is suitable for situations where there may be both over- and under-dispersion for different levels of the covariates, which makes these models especially attractive compared to those where the dispersion is constant. In fact, in the example with a constant dispersion parameter, equidispersion was obtained, whereas if covariates are considered in the dispersion parameter over- or under-dispersion can be found.

It should also be noted that the algorithm used to estimate the parameters of the hyper-Poisson model is much faster than the previous algorithms (Huang, 2017; Saez-Castillo and Conde-Sanchez, 2013).

7 Acknowledgment

We would like to thank the reviewers; their comments helped improve and clarify this manuscript and the package. In memory of our friend Antonio José, this work would not have been possible without your talent and enthusiasm.

Funding: This work was supported by MCIN/AEI/10.13039/501100011033 [project PID2019-107793GB-I00].

Bibliography


Appendix

Preliminaries

Firstly, given that
\[ 1 \mathcal{F}_1 (1, \gamma; \lambda) = \sum_{y=0}^{\infty} \frac{\lambda^y}{(\gamma)_y}, \]

it is verified that
\[ \frac{\partial}{\partial \lambda} 1 \mathcal{F}_1 (1, \gamma; \lambda) = \sum_{y=0}^{\infty} \frac{\lambda^y y}{(\gamma)_y} = \frac{1}{\lambda} \sum_{y=0}^{\infty} \frac{\lambda^y y}{(\gamma)_y} = 1 \mathcal{F}_1 (1, \gamma; \lambda) \frac{\mu}{\lambda} \]

and therefore
\[ \frac{\partial}{\partial \lambda} \log (1 \mathcal{F}_1 (1, \gamma; \lambda)) = \frac{1}{1 \mathcal{F}_1 (1, \gamma; \lambda)} \frac{\partial}{\partial \lambda} 1 \mathcal{F}_1 (1, \gamma; \lambda) = \frac{\mu}{\lambda}. \]

Furthermore, since
\[ \frac{\partial}{\partial \gamma} (\gamma)_y \frac{\partial}{\partial \gamma} \Gamma (\gamma + y) = \frac{\Gamma (\gamma + y) \psi (\gamma + y) \Gamma (\gamma) - \Gamma (\gamma + y) \Gamma (\gamma) \psi (\gamma)}{\Gamma (\gamma)^2} \]
\[ = \frac{\Gamma (\gamma + y)}{\Gamma (\gamma)} (\psi (\gamma + y) - \psi (\gamma)) = (\gamma)_y (\psi (\gamma + y) - \psi (\gamma)), \]

where we have used
\[ \frac{\partial}{\partial \gamma} \Gamma (\gamma) = \Gamma (\gamma) \psi (\gamma). \]

Thus, you get that
\[ \frac{\partial}{\partial \gamma} 1 \mathcal{F}_1 (1, \gamma; \lambda) = \sum_{y=0}^{\infty} \frac{\lambda^y y}{(\gamma)_y} - \frac{\lambda^y \psi (\gamma)_y}{(\gamma)_y} = \frac{1}{\lambda} \sum_{y=0}^{\infty} \frac{\lambda^y y}{(\gamma)_y} - \frac{\lambda^y \psi (\gamma)_y}{(\gamma)_y} (\psi (\gamma + y) - \psi (\gamma)) \]
\[ = 1 \mathcal{F}_1 (1, \gamma; \lambda) \left\{ \frac{\mu}{\lambda} \frac{\partial}{\partial \gamma} - E [\psi (\gamma + Y)] + \psi (\gamma) \right\}, \]

so
\[ \frac{\partial}{\partial \gamma} \log (1 \mathcal{F}_1 (1, \gamma; \lambda)) = \frac{1}{1 \mathcal{F}_1 (1, \gamma; \lambda)} \frac{\partial}{\partial \gamma} 1 \mathcal{F}_1 (1, \gamma; \lambda) = \frac{\mu}{\lambda} \frac{\partial}{\partial \gamma} - E [\psi (\gamma + Y)] + \psi (\gamma). \]

Proof of result 1

Using the chain rule:
\[ \frac{\partial l_{\beta \gamma}}{\partial \beta} = \frac{\partial l_{\beta \gamma}}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta}. \]
so that the score function for \( \mu_i \) is:

\[
\frac{\partial l_{hp}}{\partial \mu_i} = \frac{\partial l_{hp}}{\partial \lambda_i} \frac{\partial \lambda_i}{\partial \mu_i}
\]

Let us start with the first of the previous derivatives:

\[
\frac{\partial l_{hp}}{\partial \lambda_i} = \sum_{y=0}^{\infty} \left\{ \frac{Y_i}{\lambda_i} - \frac{\partial}{\partial \lambda_i} \log \left( F_1 (1, \gamma_i; \lambda_i) \right) \right\} = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{\lambda_i}, \tag{10}
\]

where the result (7) has been used.

The second of the above derivatives is obtained using implicit differentiation, taking into account that \( \lambda (\mu_i, \gamma_i) \) is solution of equation (2). Thus,

\[
0 = \sum_{y=0}^{\infty} (y - \mu) \frac{\lambda^y}{(\gamma)^y}
\]

wherein subscript \( i \) has been omitted, and by differentiating with respect to \( \mu \)

\[
0 = - \sum_{y=0}^{\infty} \frac{\lambda^y}{(\gamma)^y} + \sum_{y=0}^{\infty} (y - \mu) \frac{\lambda^{y-1}}{(\gamma)^y} \frac{\partial \lambda}{\partial \mu} = - F_1 (1, \gamma; \lambda) + \frac{1}{\lambda} \sum_{y=0}^{\infty} (y - \mu) y \frac{\lambda^y}{(\gamma)^y} \frac{\partial \lambda}{\partial \mu}
\]

= \left[ - F_1 (1, \gamma; \lambda) + \frac{1}{\lambda} \sum_{y=0}^{\infty} (y - \mu)^2 \frac{\lambda^y}{(\gamma)^y} \right] \frac{\partial \lambda}{\partial \mu} = - \frac{\partial}{\partial \mu} \left[ F_1 (1, \gamma; \lambda) V (\mu, \gamma) \frac{\partial \lambda}{\partial \mu} \right],
\]

and clearing

\[
\frac{\partial \lambda}{\partial \mu} = \frac{\lambda}{V (\mu, \gamma)}. \tag{11}
\]

Therefore, plugging (10) and (11) into the score function, we obtain

\[
\frac{\partial l_{hp}}{\partial \mu_i} = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{\lambda_i} \frac{\lambda_i}{V (\mu_i, \gamma_i)} = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{V (\mu_i, \gamma_i)}. \tag{12}
\]

The only thing left to consider to achieve expression (4) is that

\[
\frac{\partial \mu_i}{\partial \beta} = \mu_i \chi_i.
\]

**Proof of result 2**

Using the chain rule we get

\[
\frac{\partial l_{hp}}{\partial \delta} = \frac{\partial l_{hp}}{\partial \gamma_i} \frac{\partial \gamma_i}{\partial \delta}.
\]

We develop the first of these derivatives (score for \( \gamma \))

\[
\frac{\partial l_{hp}}{\partial \gamma_i} = \sum_{i=1}^{n} \left\{ \frac{Y_i}{\lambda_i} \frac{\partial \lambda_i}{\partial \gamma_i} - \psi (\gamma_i + Y_i) + \psi (\gamma_i) - \frac{\partial}{\partial \gamma_i} \log \left( F_1 (1, \gamma_i; \lambda_i) \right) \right\}
\]

= \sum_{i=1}^{n} \left\{ \frac{Y_i}{\lambda_i} \frac{\partial \lambda_i}{\partial \gamma_i} - \psi (\gamma_i + Y_i) + \psi (\gamma_i) - \frac{\partial}{\partial \gamma_i} \left[ \psi (\gamma_i + Y_i) - \psi (\gamma_i) \right] \right\}
\]

= \sum_{i=1}^{n} \left\{ \frac{Y_i - \mu_i}{\lambda_i} \frac{\partial \lambda_i}{\partial \gamma_i} - \psi (\gamma_i + Y_i) + E [\psi (\gamma_i + Y_i)] - \psi (\gamma_i) \right\},
\]

wherein we have used (9).

The derivative of \( \lambda_i \) with respect to \( \gamma_i \) is also obtained by explicit differentiation from equation (2)
with respect to \( \gamma \), getting (without considering subscripts):

\[
0 = \sum_{y=0}^{\infty} (y - \mu) \left\{ \frac{\partial \lambda}{\partial \gamma} \right\}_{y} (\gamma) = \frac{\lambda}{\gamma} \cdot \frac{\partial \gamma}{\partial \gamma}
\]

\[
= \frac{1}{\lambda} \frac{\partial \lambda}{\partial \gamma} \sum_{y=0}^{\infty} (y - \mu) \frac{\lambda}{(\gamma)_{y}} \cdot \frac{\partial \gamma}{(\gamma)_{y}} - \sum_{y=0}^{\infty} (y - \mu) \frac{\lambda}{(\gamma)_{y}} (\gamma + y - \gamma)
\]

\[
= \frac{1}{\lambda} \frac{\partial \lambda}{\partial \gamma} \sum_{y=0}^{\infty} (y - \mu) \frac{\lambda}{(\gamma)_{y}} - \sum_{y=0}^{\infty} (y - \mu) \psi (\gamma + \gamma + \gamma)
\]

\[
= \text{I}_{1} (1, \gamma; \lambda) \left\{ \frac{V (\mu, \lambda)}{\gamma} \right\} = \lambda \cdot \frac{\text{Cov} [Y, \psi (\gamma + \gamma)]}{V (\mu, \gamma)}.
\]

Therefore, replacing (13) in the score of \( \gamma \) we obtain

\[
\frac{\partial \lambda}{\partial \gamma} = \text{I}_{1} (1, \gamma; \lambda) \left\{ \frac{V (\mu, \lambda)}{\gamma} \frac{\partial \gamma}{\gamma} - \psi (\gamma + \gamma) + E [\psi (\gamma + \gamma)] \right\}.
\]

The only thing left to consider to get the expression (5) is that

\[
\frac{\partial \gamma}{\partial \delta} = \gamma; Z_{i}.
\]

**Proof of result 3**

The Fisher information matrix can be calculated by blocks:

\[
E \left[ S_{\beta} S'_{\beta} \right] = E \left[ Y (Y - \mu)^{2} \right] \frac{\mu^{2}}{\sigma^{2}} XX' = \frac{\mu^{2}}{\sigma^{2}} XX'.
\]

\[
E \left[ S_{\beta} S'_{\beta} \right] = \left\{ \frac{E \left[ Y (Y - \mu)^{2} \right]}{\sigma^{4}} \frac{\text{Cov} (Y, \psi (\gamma + \gamma))^{2} + E \left[ \psi (\gamma + \gamma)^{2} \right] + E \left[ \psi (\gamma + \gamma)^{2} \right] - 2 E \left[ \psi (\gamma + \gamma) \right] E \left[ \psi (\gamma + \gamma) \right]}{\gamma^{2}} \right\} ZZ' \gamma^{2}
\]

\[
= \left\{ \frac{\text{Var} (Y + \gamma + \gamma)}{\sigma^{2}} ZZ' \gamma^{2} \right\}
\]

\[
E \left[ S_{\beta} S'_{\beta} \right] = \left\{ \frac{E \left[ Y (Y - \mu)^{2} \right]}{\sigma^{4}} \frac{\text{Cov} (Y, \psi (\gamma + \gamma))^{2} - E \left[ \psi (\gamma + \gamma)^{2} \right] - 2 E \left[ \psi (\gamma + \gamma) \right] E \left[ \psi (\gamma + \gamma) \right]}{\gamma^{2}} \right\} ZZ' \gamma^{2}
\]

\[
= 0.
\]
Gradient of the objective function

Considering the log-likelihood function (3) as a function of \( \beta, \lambda'_i = \log \lambda_i \) and \( \delta \), the gradient of the objective function is:

\[
\frac{\partial l_{HP}}{\partial \beta} = 0_{1 \times (q_1 + 1)};
\]

\[
\frac{\partial l_{HP}}{\partial \lambda_i} = \frac{\partial l_{HP}}{\partial \lambda'_i} \frac{\partial \lambda'_i}{\partial \lambda_i} = \frac{1}{\lambda'_i} \left( \sum_{i=1}^{n} Y_i - n \mu_i \right) \lambda_i = \frac{n}{\lambda_i} \left( \sum_{i=1}^{n} Y_i - n \mu_i \right),
\]

\[
\frac{\partial l_{HP}}{\partial \delta} = \frac{\partial l_{HP}}{\partial \gamma} \frac{\partial \gamma}{\partial \delta} = \left( - \sum_{i=1}^{n} \psi (\gamma_i + Y_i) + E \left[ \psi (\gamma_i + Y_i) \right] \right) \gamma \gamma'_i.
\]

The proof is straightforward from expressions (7) and (8).

Gradient of the restriction (2)

Restriction (2) can be expressed as

\[
ceq_i = e^{x_i \beta} - \sum_{y_i=0}^{\infty} y_i \frac{1}{F_1 (1, \gamma_i; \lambda_i)} \left( \frac{\lambda'_i}{\gamma_i} \right)^y_i, \quad i = 1, \ldots, n,
\]

whose gradient is given by:

\[
\nabla c_{eq_i} = \left( \frac{\partial c_{eq_i}}{\partial \beta}, \frac{\partial c_{eq_i}}{\partial \lambda_i}, \frac{\partial c_{eq_i}}{\partial \delta} \right),
\]

where

\[
\frac{\partial c_{eq_i}}{\partial \beta} = \mu_i x_i,
\]

\[
\frac{\partial c_{eq_i}}{\partial \lambda_i} = -V (\mu_i, \gamma_i) \Rightarrow \frac{\partial c_{eq_i}}{\partial \lambda'_i} = -V (\mu_i, \gamma_i),
\]

\[
\frac{\partial c_{eq_i}}{\partial \delta} = \text{Cov} (Y_i, \psi (\gamma_i + Y_i)) \gamma_i \gamma'_i.
\]

For this, you have to take into account that:

\[
\frac{\partial c_{eq_i}}{\partial \lambda_i} = - \sum_{y_i=0}^{\infty} y_i^2 \frac{1}{F_1 (1, \gamma_i; \lambda_i)} \left( \frac{\lambda'_i}{\gamma_i} \right)^{y'_i} + \sum_{y_i=0}^{\infty} y_i \frac{\mu_i}{\lambda_i} \frac{1}{F_1 (1, \gamma_i; \lambda_i)} \left( \frac{\lambda'_i}{\gamma_i} \right)^{y'_i}
\]

\[
= \frac{1}{\lambda'_i} \sum_{y_i=0}^{\infty} y_i^2 \frac{1}{F_1 (1, \gamma_i; \lambda_i)} \left( \frac{\lambda'_i}{\gamma_i} \right)^{y'_i} + \sum_{y_i=0}^{\infty} y_i \frac{\mu_i}{\lambda_i} \sum_{y_i=0}^{\infty} \frac{1}{F_1 (1, \gamma_i; \lambda_i)} \left( \frac{\lambda'_i}{\gamma_i} \right)^{y'_i}
\]

\[
= \frac{1}{\lambda'_i} E \left[ Y^2_i \right] + \frac{\mu^2_i}{\lambda'_i} = -V (\mu_i, \gamma_i),
\]

where expression (7) has been used. Considering expressions (8) and (9), it is verified that:

\[
\frac{\partial c_{eq_i}}{\partial \gamma_i} = \sum_{y_i=0}^{\infty} y_i (\psi (\gamma_i + Y_i) - \psi (\gamma_i)) \frac{1}{F_1 (1, \gamma_i; \lambda_i)} \left( \frac{\lambda'_i}{\gamma_i} \right)^{y'_i} + \sum_{y_i=0}^{\infty} y_i (\psi (\gamma_i) - E [\psi (\gamma_i + Y_i)]) \frac{1}{F_1 (1, \gamma_i; \lambda_i)} \left( \frac{\lambda'_i}{\gamma_i} \right)^{y'_i}
\]

\[
= E [Y_i \psi (\gamma_i + Y_i)] - E [Y_i] E [\psi (\gamma_i + Y_i)] = \text{Cov} (Y_i, \psi (\gamma_i + Y_i)),
\]

from which it is straightforward to obtain the gradient for \( \delta \).