The implementation of the Conway-Maxwell (COM) Poisson model in **brms**

Wellington José Leite da Silva
School of Applied Mathematics, Fundação Getulio Vargas, Brazil.

March 26, 2025

Contents

1 Proof of stopping criterion for guaranteed error

For $\lambda > 0$ and $\nu > 0$, the Conway-Maxwell-Poisson (COMP) probability mass function (p.m.f.) is given by

$$p_{\lambda,\nu}(n) := \Pr(X = n \mid \lambda, \nu) = \frac{\lambda^n}{Z(\lambda,\nu)(n!)^\nu},$$

where

$$Z(\lambda,\nu) := \sum_{i=0}^{\infty} \frac{\lambda^i}{(i!)^\nu} \tag{1}$$

is the normalising constant. Our goal is to find a function $U_k(\lambda,\nu)$, such that

$$|Z(\lambda,\nu) - Z_k(\lambda,\nu)| \leq U_k(\lambda,\nu) \tag{2}$$

where $Z_k(\lambda,\nu) := \sum_{i=0}^{k} \frac{\lambda^i}{(i!)^\nu}$ (the partial sum of $Z(\lambda,\nu)$). To achieve this, we can use the following result (Braden (1992)):

Proposition 1 (Bounding a series that pass in ratio test). *Let $S_k := \sum_{i=0}^{k} a_i$. Under the assumptions that $(a_k)_{k\geq 0}$ is positive, decreasing and passes the ratio test, then for every $0 \leq k < \infty$ the following holds:*

$$S_k + a_k\left(\frac{L}{1-L}\right) < S < S_k + a_k\left(\frac{1}{1-\frac{a_k}{a_{k-1}}}\right), \tag{3}$$

*if $\frac{a_{k+1}}{a_k}$ **decreases** to L and*

$$S_k + a_k\left(\frac{1}{1-\frac{a_k}{a_{k-1}}}\right) < S < S_k + a_k\left(\frac{L}{1-L}\right), \tag{4}$$

*if $\frac{a_{k+1}}{a_k}$ **increases** to L.*

Proof. See Appendix A. $\square$

To obtain the result of Equation 2 using Proposition 1, we will prove that the series defined in Equation 5 satisfies the ratio test and has a decreasing ratio of terms. First, for the ratio test, we have

$$\lim_{i\to\infty} \frac{\left|\frac{\lambda^{i+1}}{(i+1)!^{\nu}}\right|}{\left|\frac{\lambda^{i}}{(i)!^{\nu}}\right|} = \lim_{i\to\infty} \frac{\lambda^{i+1}}{(i+1)!^{\nu}} \frac{(i)!^{\nu}}{\lambda^{i}} = \lim_{i\to\infty} \frac{\lambda}{(i+1)^{\nu}} = 0.$$

Thus, it passes the ratio test with a limit of 0. Also, note that $\frac{\lambda}{(i+1)^{\nu}}$ is a decreasing function of i (considering $\lambda, \nu > 0$). Then, by Proposition 1, it follows that

$$S < S_k + a_k \left(\frac{1}{1 - \frac{a_k}{a_{k-1}}} \right)$$

where $a_k = \frac{\lambda^k}{k!^{\nu}}$, and since the terms are positive, we have

$$|Z(\lambda,\nu) - Z_k(\lambda,\nu)| \leq a_k \left(1 - \frac{a_k}{a_{k-1}}\right)^{-1}.$$

1.1 Algorithm

We can write the previous result as an algorithm as follows:

Algorithm 1 Adaptive truncation via ratio test

Initialize a_0, a_1 and $k = 0$
while $a_k \geq a_{k-1}$ **or** $a_k \left(1 - \frac{a_k}{a_{k-1}}\right)^{-1} \geq \varepsilon$ **do**
    Set $k = k + 1$
    Evaluate a_k
end while
Evaluate $S_k = \sum_{i=0}^{k} a_i$
Return S_k

Checking $a_k \geq a_{k-1}$ is important because the series may start increasing, but after a point, it always decreases.

2 Examples with new approach

2.1 For simple cases

As mentioned in the PR. Here, we have the *COMP_brms_updated* with the new version for the normalization constant of COM-Poisson; The *COMP_brms* the actual version in the brms repository; The *COMPoissonReg* is the version from the library https://cran.r-project.org/web/packages/COMPoissonReg/index.html; The *COMP_true* evaluates the series with at least 1 million terms (much more than necessary in these cases).

parameters	COMP_brms_updated	COMP_brms	COMPoissonReg	COMP_true
$\mu = 0.50, \nu = 2.00$	1.2660	1.0634	1.2660	1.2660
$\mu = 1.00, \nu = 1.50$	2.4309	2.4309	2.4309	2.4309
$\mu = 1.10, \nu = 1.40$	2.7816	2.9266	2.7816	2.7816
$\mu = 2.00, \nu = 1.30$	8.1933	14.4379	8.2008	8.2008
$\mu = 3.00, \nu = 1.20$	25.0584	59.2945	25.0669	25.0669

Table 1: Normalising constant Z of COM-Poisson with an updated version.

And here, we have the errors in relation to the *COMP_true*

parameters	error_COMP_brms_updated	error_COMP_brms	error_COMPoissonReg
$\mu = 0.50, \nu = 2.00$	0.0000	0.2025	6.8290e-08
$\mu = 1.00, \nu = 1.50$	0.0000	0.0000	1.2823e-07
$\mu = 1.10, \nu = 1.40$	0.0000	0.1450	1.0944e-06
$\mu = 2.00, \nu = 1.30$	0.0074	6.2370	2.9324e-06
$\mu = 3.00, \nu = 1.20$	0.0080	34.2276	9.0006e-06

Table 2: Normalising constant Z of COM-Poisson with errors.

Notice that the last two rows, where the error is not zero, fall under the Gaunt approximation Gaunt et al. (2019), which raises some discussion on whether the speed gain justifies the error (see Section 3.1).

2.2 For complicated cases

We also test for some more complicated cases, here the number of terms necessary to achieve the error is between $\approx$ 150 terms in the first line and $\approx$ 160000 terms in the last line. For the current version of COM-Poisson, the repository version had a bug in the list index that was fixed so that the more demanding tests (with this one) would be fair. The change proposed in Section 1 proves to be robust even in complicated cases, (the error given is the machine-epsilon $\varepsilon = 2^{-52} \approx 2.220\text{e-}16$):

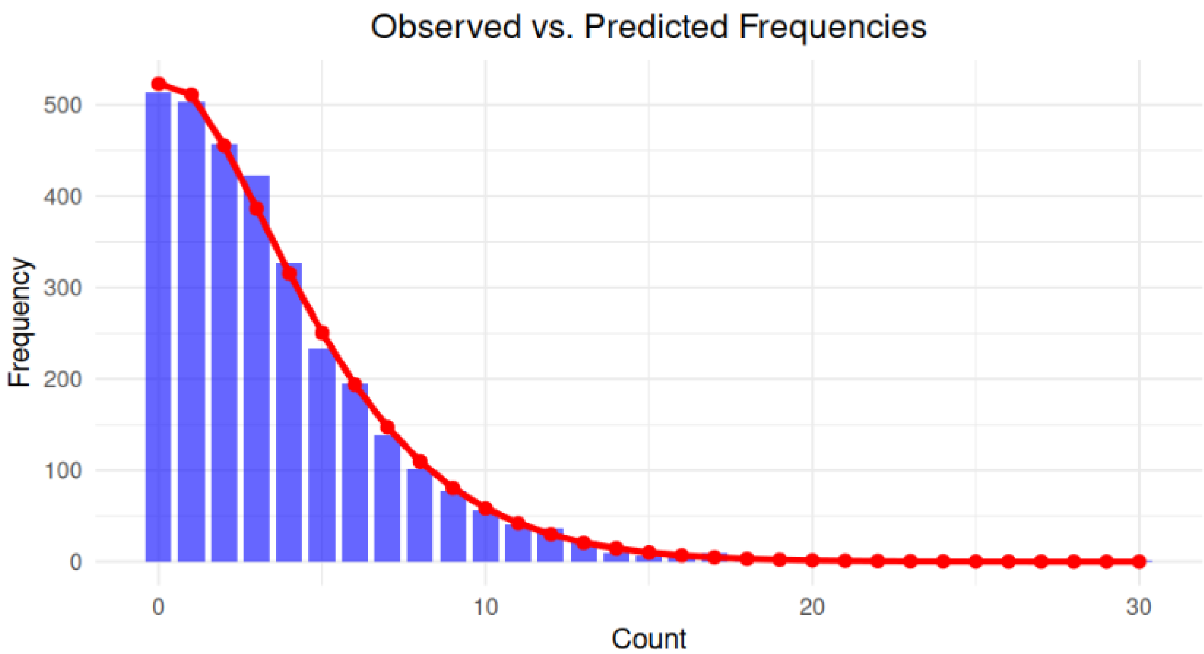
parameters	error_COMP_brms_updated	error_COMP_brms	error_COMPoissonReg
$\mu = 10, \nu = 0.1$	7.9905e-17	1.9569e-14	2.4317e-04
$\mu = 100, \nu = 0.01$	2.0508e-16	7.4670e-13	40.0716
$\mu = 1000, \nu = 0.001$	2.1836e-16	1.1505e-10	412.105
$\mu = 10000, \nu = 0.0001$	2.1972e-16	1.7554e-08	4136.83

Table 3: Normalising constant Z of COM-Poisson with errors.

2.3 MCMC

Additionally, the new version was tested with an example of MCMC (Markov chain Monte Carlo) with noisy approximation (code in: https://github.com/wellington36/MCMC_COMPoisson):

Parameter	Mean	Median	95% BCI	Posterior SD	MCSE	ESS/minute
mu	0.8030483	0.8028546	[0.533, 1.073]	0.1375955	0.0025074	75969.88
nu	0.1269029	0.1268342	[0.105, 0.149]	0.0113593	0.0002071	75871.28
n	81.4820500	81.0000000	[76, 88]	2.9317755	0.0504987	85034.42



More about this case can be seen in section 5.1 of Carvalho et al. (2022).

3 Suggestions

3.1 Remove Gaunt's approximation

As seen from Section 2.1, Gaunt's approximation adds error to the normalization constant. Based on some empirical tests, I believe that the small gain in speed in the evaluation does not compensate for the error. Moreover, doing all of the approximate computations *via* the proposed algorithm simplifies the code and improves maintainability.

3.2 Make the parametrization clearer

I saw two ways of parametrizing COM-Poisson (or COMP):

- For $\lambda > 0$ and $\nu > 0$, the COMP probability mass function (p.m.f.) can be written as

  $$p_{\lambda,\nu}(n) := \Pr(X = n \mid \lambda, \nu) = \frac{\lambda^n}{Z(\lambda,\nu)(n!)^\nu},$$

  where

  $$Z(\lambda,\nu) := \sum_{n=0}^{\infty} \frac{\lambda^n}{(n!)^\nu} \tag{5}$$

  is the normalising constant.

- And

  $$\tilde{p}_{\mu,\nu}(n) = \frac{\mu^{\nu n}}{\tilde{Z}(\mu,\nu)(n!)^\nu},$$

  where

  $$\tilde{Z}(\mu,\nu) := \sum_{n=0}^{\infty} \left(\frac{\mu^n}{n!}\right)^\nu. \tag{6}$$

In the brms we use the first one, but the name of the parameter is `mu`. I believe this creates confusion, perhaps it would be interesting to change the name or the parameterization.

References

Braden, B. (1992). Calculating sums of infinite series. *The American mathematical monthly*, 99(7):649–655.

Carvalho, L. M., Silva, W. J., and Moreira, G. A. (2022). Adaptive truncation of infinite sums: applications to statistics.

Gaunt, R. E., Iyengar, S., Daalhuis, A. B. O., and Simsek, B. (2019). An asymptotic expansion for the normalizing constant of the conway–maxwell–poisson distribution. *Annals of the Institute of Statistical Mathematics*, 71(1):163–180.

A Proofs

Proof of Proposition 1:

Proof. First define the series $r_k = \frac{a_{k+1}}{a_k}$. Now define the remainder $R_k = S - S_k = \sum_{i=k+1}^{\infty} a_i$. Now assume that r_k decreases to L. Then

$$\begin{aligned}
R_k &= a_{k-1}\left(\frac{a_{k+1}}{a_{k-1}} + \frac{a_{k+2}}{a_{k-1}} + \frac{a_{k+3}}{a_{k-1}} + \dots\right) \\
&= a_{k-1}\left(\frac{a_k}{a_{k-1}}\frac{a_{k+1}}{a_k} + \frac{a_k}{a_{k-1}}\frac{a_{k+1}}{a_k}\frac{a_{k+2}}{a_{k+1}} + \frac{a_k}{a_{k-1}}\frac{a_{k+1}}{a_k}\frac{a_{k+2}}{a_{k+1}}\frac{a_{k+3}}{a_{k+2}} + \dots\right) \\
&= a_{k-1}\left(r_{k-1}r_k + r_{k-1}r_k r_{k+1} + r_{k-1}r_k r_{k+1} r_{k+2} + \dots\right) \\
&< a_{k-1}\left(r_{k-1}r_{k-1} + r_{k-1}r_{k-1}r_{k-1} + r_{k-1}r_{k-1}r_{k-1}r_{k-1} + \dots\right) \\
&= a_{k-1}r_{k-1}^2\left(1 + r_{k-1} + r_{k-1}^2 + \dots\right) \\
&= a_k r_{k-1}\sum_{i=0}^{\infty} r_{k-1}^i = a_k\frac{r_{k-1}}{1-r_{k-1}} \\
&= a_k\frac{\frac{a_k}{a_{k-1}}}{1-\frac{a_k}{a_{k-1}}} = a_k\frac{\frac{a_k}{a_{k-1}}}{\frac{a_{k-1}-a_k}{a_{k-1}}} \\
&= a_k\frac{a_k}{a_{k-1}-a_k} = a_k\left(\frac{1}{1-\frac{a_k}{a_{k-1}}}\right).
\end{aligned} \tag{7}$$

On the other hand, since $r_k > L$ for all n,

$$\begin{aligned}
R_k &= a_k\left(\frac{a_{k+1}}{a_k} + \frac{a_{k+2}}{a_k} + \frac{a_{k+3}}{a_k} + \dots\right) \\
&= a_k\left(\frac{a_{k+1}}{a_k} + \frac{a_{k+1}}{a_k}\frac{a_{k+2}}{a_{k+1}} + \frac{a_{k+1}}{a_k}\frac{a_{k+2}}{a_{k+1}}\frac{a_{k+3}}{a_{k+2}} + \dots\right) \\
&= a_k r_k\left(1 + r_{k+1} + r_{k+1}r_{k+2} + k_{n+1}r_{k+2}r_{k+3} + \dots\right) \\
&< a_k L\left(1 + L + L^2 + L^3 + \dots\right) \\
&= a_k L\sum_{i=0}^{\infty} L^i = a_k\frac{L}{1-L}.
\end{aligned} \tag{8}$$

For the case in which r_k increases to L, the proof is analogous, with inequality signs reversed. □