Lower-Truncated Poisson and Negative Binomial Distributions

Charles J. Geyer

June 11, 2021

1 Introduction

This document works through the details of the k-truncated Poisson distribution and the k-truncated negative binomial distribution, which are the distributions of Y conditioned on $Y > k$, where k is a nonnegative integer and Y has a Poisson or negative binomial distribution. It is a design document. There is no reason for ordinary users to read it (except perhaps curiosity). It written for developers.

The negative binomial distribution with *shape parameter* α and *mean parameter* μ has probability mass function (PMF)

$$f_{\alpha,\mu}(y) = \frac{\Gamma(y + \alpha)}{\Gamma(\alpha) y!} p^{\alpha} (1 - p)^{y}, \qquad y = 0, 1, 2, \ldots,$$

where

$$p = \frac{\alpha}{\alpha + \mu}.$$

If one takes the limit as $\alpha \to \infty$ holding μ fixed, one obtains

$$f_{\infty,\mu}(y) = \frac{\mu^{y}}{y!} e^{-\mu} \qquad y = 0, 1, 2, \ldots,$$

which defines the PMF of the Poisson distribution with *mean parameter* μ (we do not prove this, because we do not use this limit in any way and only mention it to explain why we use the ∞ subscript to denote the Poisson PMF).

The PMF of the k-truncated distribution corresponding to the (untruncated) distribution with parameters α and μ has PMF defined by

$$f_{k,\alpha,\mu}(x) = \frac{f_{\alpha,\mu}(x)}{\Pr_{\alpha,\mu}\{Y > k\}}, \qquad x = k + 1, k + 2, \ldots, \tag{1}$$

where $\Pr_{\alpha,\mu}$ indicates probability calculated with respect to the untruncated distribution.

2 Exponential Family Properties

2.1 Untruncated Families

2.1.1 Probability Mass Function

The negative binomial distribution and Poisson are both exponential families. Their densities have the form

$$f_{\alpha,\mu} = \frac{1}{c_\alpha(\theta)} e^{y\theta} m_\alpha(y), \qquad y = 0, 1, 2, \ldots. \tag{2}$$

The function c_α is called the *Laplace transform* of the family. The function m_α is called the *base measure* of the family. The parameter θ is called the *canonical parameter* of the family. We have a different exponential family for each α. If we were to consider this a two-parameter family with parameters α and θ, then it would not have exponential family form.

In order that probabilities sum to one, we must have

$$c_\alpha(\theta) = \sum_{y=0}^{\infty} e^{y\theta} m_\alpha(y), \tag{3}$$

so the choice of base measure determines the family. We consider (3) to define a function on all of $\mathbb{R}$ (the real number system), taking the value $+\infty$ when the sum in (3) does not converge (which makes sense because all terms in the sum are nonnegative). This allows us to define the *canonical parameter space* of the family as the set

$$\Theta_\alpha = \{\, \theta \in \mathbb{R} : c_\alpha(\theta) < \infty \,\}.$$

Then (2) defines a PMF for all $\theta \in \Theta_\alpha$.

Poisson To get the Poisson distribution, we define the base measure by

$$m_\infty(y) = \frac{1}{y!}$$

Then we must have

$$e^{y\theta} = \mu^y$$

from which we see that the transformations between the original parameter μ and the canonical parameter θ are $\mu = \exp(\theta)$ and $\theta = \log(\mu)$.

The Laplace transform is then seen to be

$$c_\infty(\theta) = e^\mu = e^{e^\theta} = \exp(\exp(\theta))$$

and the canonical parameter space Θ_∞ is the whole real line.

Negative Binomial To get the negative binomial distribution, we define the base measure by

$$m_\alpha(y) = \frac{\Gamma(y + \alpha)}{\Gamma(\alpha) y!}$$

Then we must have

$$e^{y\theta} = (1 - p)^y = \left(\frac{\mu}{\mu + \alpha} \right)^y$$

from which we see that the transformations between the *success probability parameter* p and the canonical parameter θ are $\theta = \log(1 - p)$ and $p = 1 - e^\theta$ and that the transformations between the mean parameter μ and the canonical parameter θ are $\theta = \log(\mu) - \log(\mu + \alpha)$ and

$$\mu = \alpha \cdot \frac{e^\theta}{1 - e^\theta} = \alpha \cdot \frac{1 - p}{p} \tag{4}$$

The Laplace transform is then seen to be

$$c_\alpha(\theta) = p^{-\alpha} = \left(1 + \frac{\mu}{\alpha} \right)^\alpha = \left(1 + \frac{1}{e^{-\theta} - 1} \right)^\alpha \tag{5}$$

Since all the y in (3) are positive, c_α is nondecreasing. To have $0 < p < 1$, we must also have $\theta < 0$. We see that (5) makes sense for such θ and that $c_\alpha(\theta) \to +\infty$ as $\theta \uparrow 0$. Hence

$$\Theta_\alpha = \{ \theta \in \mathbb{R} : \theta < 0 \}.$$

2.1.2 Cumulant Function and its Derivatives

The log Laplace transform is called the *cumulant function* because its derivatives are cumulants of the distributions in the family. We write

$$\psi_\alpha(\theta) = \log c_\alpha(\theta)$$

to define the cumulant function.

Then from standard exponential family theory

$$\psi'_\alpha(\theta) = E_{\alpha,\theta}(Y)$$
$$\psi''_\alpha(\theta) = \mathrm{var}_{\alpha,\theta}(Y)$$

One can directly verify this in any particular case by evaluating the derivatives.

Poisson

$$\psi_\infty(\theta) = e^\theta \tag{6a}$$
$$\psi'_\infty(\theta) = e^\theta \tag{6b}$$
$$\psi''_\infty(\theta) = e^\theta \tag{6c}$$

Negative Binomial

$$\psi_\alpha(\theta) = \alpha \log\left(1 + \frac{1}{e^{-\theta} - 1}\right) \tag{7a}$$
$$\psi'_\alpha(\theta) = \alpha \cdot \frac{e^\theta}{1 - e^\theta} \tag{7b}$$
$$\psi''_\alpha(\theta) = \alpha \cdot \frac{e^\theta}{(1 - e^\theta)^2} \tag{7c}$$

Written in terms of the more familiar success probability parameter we have

$$\psi'_\alpha(\theta) = \alpha \cdot \frac{1-p}{p}$$
$$\psi''_\alpha(\theta) = \alpha \cdot \frac{1-p}{p^2}$$

giving the usual formulas for the mean and variance of a negative binomial random variable.

2.2 Truncated Families

The relationship between the PMF of truncated and untruncated families has already been given in (1). Since $\Pr_{\alpha,\mu}\{Y > k\}$ does not involve the data x, we see we again have an exponential family with the same canonical parameter and same canonical parameter space but with Laplace transform

$$c_{k,\alpha}(\theta) = c_\alpha(\theta) \Pr_{\alpha,\mu}\{Y > k\}$$

and hence

$$\psi_{k,\alpha}(\theta) = \psi_\alpha(\theta) + \log \Pr_{\alpha,\mu}\{Y > k\}$$

where μ is given as a function of θ by $\mu = \exp(\theta)$ for the Poisson and (4) for the negative binomial. Hence we can also write this

$$\psi_{k,\alpha}(\theta) = \psi_\alpha(\theta) + \log \Pr_{\alpha,\theta}\{Y > k\} \tag{8}$$

The mean and variance of X are again given by the first and second derivatives of the cumulant function (8)

$$\psi'_{k,\alpha}(\theta) = E_{k,\alpha,\theta}\{X\} = E_{\alpha,\theta}\{Y \mid Y > k\} \tag{9a}$$

$$\psi''_{k,\alpha}(\theta) = \operatorname{var}_{k,\alpha,\theta}\{X\} = \operatorname{var}_{\alpha,\theta}\{Y \mid Y > k\} \tag{9b}$$

These identities must hold by exponential family theory. Of course, they can also be verified directly by evaluating the derivatives seeing that they do indeed give the appropriate expectation.

Note that although we still use μ as a parameter, it is no longer the mean of the k-truncated variable X (it is the mean of the corresponding untruncated variable Y). The mean of X is yet another parameter

$$\tau = \psi'_{k,\alpha}(\theta) \tag{10}$$

which is called the *mean value parameter* of the family. The fact that

$$\frac{d\tau}{d\theta} = \operatorname{var}_{k,\alpha,\theta}(X)$$

is necessarily positive (because it is a variance) means the map $\theta \mapsto \tau$ is one-to-one, an invertible change-of-parameter. We will make no use of this fact. The only point of this paragraph is to stress that the mean of X is not μ; the mean of X is τ.

3 Computing

As always, we wish to compute things, in this case the cumulant function and its first two derivatives, without overflow or cancellation error. Problems arise when μ is nearly zero or when μ is very large.

3.1 Cumulant Function

We consider (8) fairly behaved computationally. The computation of $\log \Pr_{\alpha,\theta}\{Y > k\}$ can be left to the relevant R function (`ppois` or `pnbinom`) using the `lower.tail = FALSE` and `log.p = TRUE` optional arguments to avoid cancellation error and overflow.

Any of the cumulant functions we are discussing are continuous, because differentiable, and strictly increasing, because their derivatives are the mean value parameters, which are strictly positive. It can be checked directly from (6a) and (7a) that

$$\begin{aligned} \psi_\alpha(\theta) &\to 0, &&\text{as } \theta \to -\infty \\ \psi_\alpha(\theta) &\to +\infty, &&\text{as } \theta \to \sup \Theta_\alpha \end{aligned}$$

where, of course, $\sup \Theta_\alpha$ is zero for the negative binomial and $+\infty$ for the Poisson. It can also be checked directly that

$$\begin{aligned} \Pr_{\alpha,\theta}\{Y > k\} &\to 0, &&\text{as } \theta \to -\infty \\ \Pr_{\alpha,\theta}\{Y > k\} &\to 1, &&\text{as } \theta \to \sup \Theta_\alpha \end{aligned}$$

hence

$$\begin{aligned} \log \Pr_{\alpha,\theta}\{Y > k\} &\to -\infty, &&\text{as } \theta \to -\infty \\ \log \Pr_{\alpha,\theta}\{Y > k\} &\to 0, &&\text{as } \theta \to \sup \Theta_\alpha \end{aligned}$$

Thus $\psi_{k,\alpha}$, which is also continuous and strictly increasing (because its derivative is strictly positive), goes from $-\infty$ to $+\infty$ as θ goes from the lower end of Θ_α to the upper end.

Since the addition in the computation of (8) involves terms of opposite sign, $\psi_{k,\alpha}(\theta)$ positive and $\log \Pr_{\alpha,\theta}\{Y > k\}$ negative, cancellation error may occur. Also overflow to `-Inf` or `Inf` (if the machine arithmetic is IEEE, as is true with most machines nowadays) may occur when θ is near an endpoint of Θ_α,

What cannot happen is that we get `-Inf + Inf = NaN` (in IEEE arithmetic) because when the first term is large, the second is near zero, and vice versa. Thus we regard whatever cancellation error occurs as not a problem. There seems to be nothing that can be done about it if we use the floating-point arithmetic native to the machine.

3.2 First Derivative of Cumulant Function

In evaluating derivatives of (8), we have no problem evaluating derivatives of $\psi_\alpha(\theta)$. The only problematic part is derivatives of $\log \Pr_{\alpha,\theta}\{Y > k\}$. Since we have no formulas for that, we proceed differently.

From (9a) we have

$$\begin{aligned}\psi'_{k,\alpha}(\theta) &= E_{\alpha,\theta}\{Y \mid Y > k\} \\ &= \frac{E_{\alpha,\theta}\{Y I(Y > k)\}}{\Pr_{\alpha,\theta}\{Y > k\}}\end{aligned}$$

where Y denotes a random variable having the corresponding untruncated distribution, and $I(Y > k)$ is one if $Y > k$ and zero otherwise. There being no functions that evaluate expectations with respect to Poisson and negative binomial distributions, we need to rewrite this in terms of probabilities using special properties of each distribution.

3.2.1 Poisson

$$\begin{aligned}E_{\infty,\theta}\{Y I(Y > k)\} &= \sum_{y=k+1}^{\infty} \frac{\mu^y}{(y-1)!} e^{-\mu} \\ &= \mu \Pr_{\infty,\theta}\{Y \geq k\}\end{aligned} \tag{11}$$

Hence

$$\begin{aligned}E_{\infty,\theta}\{Y \mid Y > k\} &= \frac{\mu \Pr_{\infty,\theta}\{Y \geq k\}}{\Pr_{\infty,\theta}\{Y > k\}} \\ &= \mu + \frac{\mu \Pr_{\infty,\theta}\{Y = k\}}{\Pr_{\infty,\theta}\{Y > k\}} \\ &= \mu + \frac{\mu^{k+1} e^{-\mu} / k!}{\mu^{k+1} e^{-\mu}/(k+1)! + \Pr_{\infty,\theta}\{Y > k+1\}} \\ &= \mu + \frac{k+1}{1 + \frac{\Pr_{\infty,\theta}\{Y > k+1\}}{\Pr_{\infty,\theta}\{Y = k+1\}}}\end{aligned}$$

To simplify notation we give the fraction in the denominator a name

$$\beta = \frac{\Pr_{\alpha,\theta}\{Y > k+1\}}{\Pr_{\alpha,\theta}\{Y = k+1\}} \tag{12}$$

(We use subscript α because we will use the same definition of β for both cases. The Poisson case has $\alpha = \infty$.) Then

$$\psi'_{k,\infty}(\theta) = \mu + \frac{k+1}{1+\beta} \tag{13}$$

We are pleased with this formula, which took a bit of formula bashing to find, since it behaves very well computationally.

Since both terms in (13) are positive, we never have cancellation error. When μ is near zero, we have β near zero, and (13) calculates a result near $k + 1$ accurately. When μ is large, we have β also large, and (13) calculates a result near μ accurately. The result may overflow (to `Inf` in IEEE arithmetic), but only when μ itself is near overflow.

3.2.2 Negative Binomial

$$
\begin{aligned}
E_{\alpha,\theta}\{Y I(Y > k)\} &= \sum_{y=k+1}^{\infty} y \frac{\Gamma(y+\alpha)}{\Gamma(\alpha) y!} p^{\alpha} (1-p)^{y} \\
&= (1-p) \sum_{y=k+1}^{\infty} (y - 1 + \alpha) \frac{\Gamma(y-1+\alpha)}{\Gamma(\alpha)(y-1)!} p^{\alpha} (1-p)^{y-1} \\
&= (1-p) E_{\alpha,\theta}\{(Y + \alpha) I(Y \geq k)\} \\
&= (1-p) E_{\alpha,\theta}\{Y I(Y \geq k)\} + \alpha(1-p) \Pr_{\alpha,\theta}\{Y \geq k\} \\
&= (1-p) E_{\alpha,\theta}\{Y I(Y > k)\} + k(1-p) \Pr_{\alpha,\theta}\{Y = k\} \\
&\qquad + \alpha(1-p) \Pr_{\alpha,\theta}\{Y \geq k\}
\end{aligned}
$$

Moving the term containing the expectation (rather than probability) from the right hand side to the left, we obtain

$$
\begin{aligned}
p E_{\alpha,\theta}\{Y I(Y > k)\} &= k(1-p) \Pr_{\alpha,\theta}\{Y = k\} + \alpha(1-p) \Pr_{\alpha,\theta}\{Y \geq k\} \\
&= (k + \alpha)(1-p) \Pr_{\alpha,\theta}\{Y = k\} + \alpha(1-p) \Pr_{\alpha,\theta}\{Y > k\}
\end{aligned}
\tag{14}
$$

and have expressed this expectation in terms of probability functions (which are implemented in R).

Hence

$$
\begin{aligned}
E_{\alpha,\theta}\{Y \mid Y > k\} &= \frac{\alpha(1-p)}{p} + \frac{(k+\alpha)(1-p) \Pr_{\alpha,\theta}\{Y = k\}}{p \Pr_{\alpha,\theta}\{Y > k\}} \\
&= \mu + \frac{(k+\alpha)(1-p) \Pr_{\alpha,\theta}\{Y = k\}}{p \Pr_{\alpha,\theta}\{Y > k\}}
\end{aligned}
\tag{15}
$$

We work on

$$
\begin{aligned}
(1-p)(k+\alpha)\Pr_{\alpha,\theta}\{Y=k\} &= (1-p)(k+\alpha)\frac{\Gamma(k+\alpha)}{\Gamma(\alpha)k!}p^\alpha(1-p)^k \\
&= \frac{\Gamma(k+1+\alpha)}{\Gamma(\alpha)k!}p^\alpha(1-p)^{k+1} \\
&= (k+1)\frac{\Gamma(k+1+\alpha)}{\Gamma(\alpha)(k+1)!}p^\alpha(1-p)^{k+1} \\
&= (k+1)\Pr_{\alpha,\theta}\{Y=k+1\}
\end{aligned}
$$

Hence

$$
\begin{aligned}
E_{\alpha,\theta}\{Y \mid Y > k\} &= \mu + \frac{(k+1)\Pr_{\alpha,\theta}\{Y=k+1\}}{p\Pr_{\alpha,\theta}\{Y>k\}} \\
&= \mu + \frac{(k+1)}{p + p\frac{\Pr_{\alpha,\theta}\{Y>k+1\}}{\Pr_{\alpha,\theta}\{Y=k+1\}}}
\end{aligned}
$$

Defining β by (12) as in the Poisson case, we get

$$
\psi'_{k,\alpha}(\theta) = \mu + \frac{k+1}{p(1+\beta)} \tag{16}
$$

as our simple computational formula for the mean value parameter in the negative binomial case. Formula (16) is not as well behaved as its Poisson analogue (13), but only in that we can have p underflow to zero and $\Pr_{\alpha,\theta}\{Y = k+1\}$ also underflow to zero, giving a possible `NaN` result for $p(1+\beta)$. However, when p underflows to zero, μ evaluates to `Inf`, and hence (16), being the sum of positive terms should evaluate to `Inf` as well. Thus, if we simply return `Inf` when computing (16) and `p == 0.0`, we avoid `NaN` and produce a correct result. We may produce `Inf`, but consider that not a problem.

In testing (Section 3.4 below) we discovered another issue. The R functions `pnbinom` and `dnbinom` just punt on very small μ and so β can be calculated as `0.0 / 0.0 = NaN` when μ is very near zero (and θ very large negative). What should we get in this case? From (12)

$$\begin{aligned}
\beta &= \frac{\Pr_{\alpha,\theta}\{Y > k+1\}}{\Pr_{\alpha,\theta}\{Y = k+1\}} \\
&= \frac{\sum_{y=k+2}^{\infty} \frac{\Gamma(y+\alpha)}{\Gamma(\alpha) y!} p^\alpha (1-p)^y}{\frac{\Gamma(k+1+\alpha)}{\Gamma(\alpha)(k+1)!} p^\alpha (1-p)^{k+1}} \\
&= \frac{(k+1)!}{\Gamma(k+1+\alpha)} \sum_{y=k+2}^{\infty} \frac{\Gamma(y+\alpha)}{y!} (1-p)^{y-(k+1)}
\end{aligned}$$

Now $1 - p \to 0$ as $\mu \to 0$ and the sum in the last expression converges to zero (by dominated convergence). Thus we should replace `NaN` calculated for β when μ is near zero by zero.

3.3 Second Derivative of Cumulant Function

We obtain a formula for the second derivative of $\psi_{k,\alpha}$ by differentiating our computationally stable formulas (13) and (16) and using

$$\frac{\partial \mu}{\partial \theta} = \psi''_\alpha(\theta)$$

for which we already have the formulas (6c) and (7c) so we only need formulas for the derivatives of the second terms on the right hand sides of (13) and (16).

3.3.1 Poisson

$$\frac{\partial}{\partial \theta} \left(\frac{k+1}{1+\beta} \right) = -\frac{k+1}{(1+\beta)^2} \cdot \frac{\partial \beta}{\partial \theta}$$

and

$$
\begin{aligned}
\frac{\partial \beta}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(\frac{\Pr_{\infty,\theta}\{Y > k+1\}}{\Pr_{\infty,\theta}\{Y = k+1\}}\right) \\
&= \frac{\partial}{\partial \theta} \sum_{y=k+2}^{\infty} \frac{e^{[y-(k+1)]\theta}(k+1)!}{y!} \\
&= \sum_{y=k+2}^{\infty} [y-(k+1)] \frac{e^{[y-(k+1)]\theta}(k+1)!}{y!} \\
&= \frac{E_{\infty,\theta}\{[Y-(k+1)]I(Y > k+1)\}}{\Pr_{\infty,\theta}\{Y = k+1\}} \\
&= \frac{E_{\infty,\theta}\{Y I(Y > k+1)\} - (k+1)\Pr_{\infty,\theta}\{Y > k+1\}}{\Pr_{\infty,\theta}\{Y = k+1\}} \\
&= \frac{\mu \Pr_{\infty,\theta}\{Y \geq k+1\} - (k+1)\Pr_{\infty,\theta}\{Y > k+1\}}{\Pr_{\infty,\theta}\{Y = k+1\}}
\end{aligned}
$$

the preceding step resulting from plugging in (11)

$$
\begin{aligned}
&= \frac{\mu \Pr_{\infty,\theta}\{Y = k+1\} + [\mu - (k+1)]\Pr_{\infty,\theta}\{Y > k+1\}}{\Pr_{\infty,\theta}\{Y = k+1\}} \\
&= \mu\left(1 + \beta - \frac{k+1}{\mu}\beta\right)
\end{aligned}
$$

Putting this all together we get

$$
\psi''_{k,\infty}(\theta) = \mu\left[1 - \frac{k+1}{1+\beta}\left(1 - \frac{k+1}{\mu} \cdot \frac{\beta}{1+\beta}\right)\right] \tag{17}
$$

The only particular care required in evaluating (17) is to evaluate $\beta/(1+\beta)$ as written when β is small but as $1/(1/\beta+1)$ when β is large.

3.3.2 Negative Binomial

$$
\frac{\partial}{\partial \theta}\left(\frac{k+1}{p(1+\beta)}\right) = -\frac{k+1}{(1+\beta)^2}\left(\frac{\partial p}{\partial \theta}(1+\beta) + p\frac{\partial \beta}{\partial \theta}\right)
$$

and

$$
\begin{aligned}
\frac{\partial p}{\partial \theta} &= \frac{\partial}{\partial \theta}(1 - e^{\theta}) \\
&= -e^{\theta} \\
&= -(1-p)
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\partial \beta}{\partial \theta} &= \frac{\partial}{\partial \theta} \sum_{y=k+2}^{\infty} \frac{\frac{\Gamma(y+\alpha)}{\Gamma(\alpha) y!} p^{\alpha}(1-p)^{y}}{\frac{\Gamma(k+1+\alpha)}{\Gamma(\alpha)(k+1)!} p^{\alpha}(1-p)^{k+1}} \\
&= \frac{\partial}{\partial \theta} \sum_{y=k+2}^{\infty} \frac{\Gamma(y+\alpha)}{\Gamma(k+1+\alpha)} \cdot \frac{(k+1)!}{y!} \cdot e^{[y-(k+1)]\theta} \\
&= \frac{\partial}{\partial \theta} \sum_{y=k+2}^{\infty} \frac{\Gamma(y+\alpha)}{\Gamma(k+1+\alpha)} \cdot \frac{(k+1)!}{y!} \cdot (1-p)^{y-(k+1)} \cdot [y-(k+1)] \\
&= \frac{E_{\alpha,\theta}\{[Y-(k+1)] I(Y>k+1)\}}{\Pr_{\alpha,\theta}\{Y=k+1\}}
\end{aligned}
$$

So

$$
\begin{aligned}
p\frac{\partial \beta}{\partial \theta} &= \frac{pE_{\alpha,\theta}\{[Y-(k+1)] I(Y>k+1)\}}{\Pr_{\alpha,\theta}\{Y=k+1\}} \\
&= \frac{pE_{\alpha,\theta}\{Y I(Y>k+1)\} - p(k+1)\Pr_{\alpha,\theta}\{Y>k+1\}}{\Pr_{\alpha,\theta}\{Y=k+1\}}
\end{aligned}
$$

which using (14) becomes

$$
\begin{aligned}
&= \frac{1}{\Pr_{\alpha,\theta}\{Y=k+1\}} \big[(k+1+\alpha)(1-p)\Pr_{\alpha,\theta}\{Y=k+1\} \\
&\qquad + \alpha(1-p)\Pr_{\alpha,\theta}\{Y>k+1\} - p(k+1)\Pr_{\alpha,\theta}\{Y>k+1\}\big] \\
&= \frac{1}{\Pr_{\alpha,\theta}\{Y=k+1\}} \big[(k+1+\alpha)(1-p)\Pr_{\alpha,\theta}\{Y=k+1\} \\
&\qquad + [\alpha - p(k+1+\alpha)]\Pr_{\alpha,\theta}\{Y>k+1\}\big] \\
&= (k+1+\alpha)(1-p) + [\alpha - p(k+1+\alpha)]\beta
\end{aligned}
$$

Putting everything together we get

$$
\begin{aligned}
\psi''_{k,\alpha}(\theta) &= \psi''_{\alpha}(\theta) - \frac{k+1}{[p(1+\beta)]^2}\left[\frac{\partial p}{\partial \theta}(1+\beta) + p\frac{\partial \beta}{\partial \theta}\right] \\
&= \psi''_{\alpha}(\theta) - \frac{k+1}{[p(1+\beta)]^2}\big[-(1-p)(1+\beta) \\
&\qquad + (k+1+\alpha)(1-p) + [\alpha - p(k+1+\alpha)]\beta\big] \qquad (18) \\
&= \psi''_{\alpha}(\theta) - \frac{k+1}{p^2(1+\beta)}\Bigg[-(1-p) \\
&\qquad + \frac{(k+1+\alpha)(1-p)}{1+\beta} + [\alpha - p(k+1+\alpha)]\frac{\beta}{1+\beta}\Bigg]
\end{aligned}
$$

As with (17) the only particular care needed with (18) is in carefully calculating $\beta/(1+\beta)$.

3.4 Checks

We do a few examples to check the formulas.

3.4.1 Poisson

```
> k <- 2
> theta <- seq(-100, 100, 10)
> psi <- function(theta) {
+     mu <- exp(theta)
+     mu + ppois(k, lambda = mu, lower.tail = FALSE, log.p = TRUE)
+ }
> tau <- function(theta) {
+     mu <- exp(theta)
+     beeta <- ppois(k + 1, lambda = mu, lower.tail = FALSE) /
+         dpois(k + 1, lambda = mu)
+     mu + (k + 1) / (beeta + 1)
+ }
> qux <- function(theta) {
+     mu <- exp(theta)
+     beeta <- ppois(k + 1, lambda = mu, lower.tail = FALSE) /
+         dpois(k + 1, lambda = mu)
+     pbeeta <- ifelse(beeta < 1, beeta / (1 + beeta), 1 / (1 / beeta + 1))
+     mu * (1 - (k + 1) / (beeta + 1) * (1 - (k + 1) / mu * pbeeta))
+ }
```

First we check the derivative property.

```
> epsilon <- 1e-6
> pfoo <- tau(theta)
> pbar <- (psi(theta + epsilon) - psi(theta)) / epsilon
> all.equal(pfoo, pbar, tolerance = 10 * epsilon)

[1] TRUE

> pfoo <- qux(theta)
> pbar <- (tau(theta + epsilon) - tau(theta)) / epsilon
> all.equal(pfoo, pbar, tolerance = 10 * epsilon)
```

```
[1] TRUE
```

Then we check the mean property.

```
> theta <- seq(log(0.01), log(100), length = 51)
> pfoo <- tau(theta)
> pqux <- qux(theta)
> pbar <- double(length(theta))
> pbaz <- double(length(theta))
> for (i in seq(along = theta)) {
+     mu <- exp(theta[i])
+     xxx <- seq(0, 10000)
+     ppp <- dpois(xxx, lambda = mu)
+     ppp[xxx <= k] <- 0
+     ppp <- ppp / sum(ppp)
+     pbar[i] <- sum(xxx * ppp)
+     pbaz[i] <- sum((xxx - pbar[i])^2 * ppp)
+ }
> all.equal(pfoo, pbar)

[1] TRUE

> all.equal(pqux, pbaz)

[1] TRUE
```

3.4.2 Negative Binomial

```
> k <- 2
> alpha <- 2.22
> mu <- 10^seq(-2, 2, 0.1)
> theta <- log(mu) - log(mu + alpha)
> psi <- function(theta) {
+     stopifnot(all(theta < 0))
+     mu <- (- alpha * exp(theta) / expm1(theta))
+     alpha * log1p(1 / expm1(- theta)) +
+         pnbinom(k, size = alpha, mu = mu, lower.tail = FALSE, log.p = TRUE)
+ }
> tau <- function(theta) {
+     stopifnot(all(theta < 0))
+     mu <- (- alpha * exp(theta) / expm1(theta))
```

```
+     p <- alpha / (mu + alpha)
+     beetaup <- pnbinom(k + 1, size = alpha, mu = mu, lower.tail = FALSE)
+     beetadn <- dnbinom(k + 1, size = alpha, mu = mu)
+     beeta <- beetaup / beetadn
+     beeta[beetaup == 0] <- 0
+     result <- mu + (k + 1) / (beeta + 1) / p
+     result[p == 0] <- Inf
+     return(result)
+ }
> qux <- function(theta) {
+     stopifnot(all(theta < 0))
+     mu <- (- alpha * exp(theta) / expm1(theta))
+     p <- alpha / (mu + alpha)
+     omp <- mu / (mu + alpha)
+     beetaup <- pnbinom(k + 1, size = alpha, mu = mu, lower.tail = FALSE)
+     beetadn <- dnbinom(k + 1, size = alpha, mu = mu)
+     beeta <- beetaup / beetadn
+     beeta[beetaup == 0] <- 0
+     pbeeta <- ifelse(beeta < 1, beeta / (1 + beeta), 1 / (1 / beeta + 1))
+     alpha * omp / p^2 - (k + 1) / p^2 / (1 + beeta) * ( - omp +
+         (k + 1 + alpha) * omp / (1 + beeta) +
+         ( alpha - p * (k + 1 + alpha) ) * pbeeta )
+ }
```

First we check the derivative property.

```
> epsilon <- 1e-6
> pfoo <- tau(theta)
> pbar <- (psi(theta + epsilon) - psi(theta)) / epsilon
> all.equal(pfoo, pbar, tolerance = 20 * epsilon)

[1] TRUE

> pfoo <- qux(theta)
> pbar <- (tau(theta + epsilon) - tau(theta)) / epsilon
> all.equal(pfoo, pbar, tolerance = 40 * epsilon)

[1] TRUE
```

Then we check the mean property.

```
> pfoo <- tau(theta)
> pqux <- qux(theta)
> pbar <- double(length(theta))
> pbaz <- double(length(theta))
> for (i in seq(along = theta)) {
+     mu <- (- alpha * exp(theta[i]) / expm1(theta[i]))
+     xxx <- seq(0, 10000)
+     ppp <- dnbinom(xxx, size = alpha, mu = mu)
+     ppp[xxx <= k] <- 0
+     ppp <- ppp / sum(ppp)
+     pbar[i] <- sum(xxx * ppp)
+     pbaz[i] <- sum((xxx - pbar[i])^2 * ppp)
+ }
> all.equal(pfoo, pbar)

[1] TRUE

> all.equal(pqux, pbaz)

[1] TRUE
```

4 Random Variate Generation

To simulate a k-truncated random variate, the simplest method rejection sampling with "proposal" the corresponding untruncated random variate (assuming code to simulate that already exists), that is we simulate Y from the untruncated distribution of the same family having the same parameter values and accept only Y satisfying $Y > k$.

Although this works well when $\mu = E_{\alpha,\theta}(Y)$ is large, it works poorly when $\mu/(k+1)$ is small.

A more general proposal simulates Y from some nonnegative integer valued distribution for which simulation code exists. We then define $X = Y + m$ for some integer m and accept X satisfying $X > k$ with some probability $a(x)$ determined by the rejection sampling algorithm. Necessarily $0 \leq m \leq k+1$. Otherwise rejection sampling is not possible.

4.1 Negative Binomial

We start with the negative binomial because it is harder, having fewer algorithms that are analyzable.

4.1.1 Algorithm

Suppose Y is negative binomial with parameters α^* and p^* and X is k-truncated negative binomial with parameters α and p. The "proposal" is $Y + m$ and the "target" distribution is that of X. (After much analysis, it turns out that there is no advantage obtained by allowing $p^* \neq p$. Hence we consider only the case $p^* = p$.)

Then the ratio of target PMF to proposal PMF is proportional to (dropping terms that do not contain x)

$$r(x) = \frac{\Gamma(x + \alpha)}{x!} \cdot \frac{(x - m)!}{\Gamma(x - m + \alpha^*)} \cdot I(x > k)$$

Since the gamma functions are fairly obnoxious when α and α^* are arbitrary, we need to enforce some relationship. We try the simplest, $\alpha^* = \alpha + m$, which makes them cancel. Then

$$r(x) = \frac{(x - m)!}{x!} \cdot I(x > k)$$

is a decreasing function of x. Then the acceptance probability can be taken to be

$$a(x) = \frac{r(x)}{r(k + 1)} = \frac{(x - m)!}{x!} \cdot \frac{(k + 1)!}{(k + 1 - m)!} \cdot I(x > k)$$

4.1.2 Performance

Now the question arises whether we can calculate the performance of the the algorithm, which is characterized by its acceptance rate (expected acceptance probability) $E^*\{a(Y + m)\}$ where E^* denotes expectation with respect to the distribution of Y where $Y + m$ is the proposal, that is,

$$\rho(m) = \sum_{y=0}^{\infty} a(y + m) \frac{\Gamma(y + m + \alpha)}{\Gamma(m + \alpha) y!} p^{m+\alpha} (1 - p)^y$$

(we used $\alpha^* = m + \alpha$)

$$
\begin{aligned}
&= \sum_{y=0}^{\infty} \frac{(y+m-m)!}{(y+m)!} \cdot \frac{(k+1)!}{(k+1-m)!} \cdot I(y+m>k) \\
&\quad \times \frac{\Gamma(y+m+\alpha)}{\Gamma(m+\alpha)y!} p^{m+\alpha}(1-p)^y \\
&= \frac{(k+1)!}{(k+1-m)!} \sum_{y=k+1-m}^{\infty} \frac{\Gamma(y+m+\alpha)}{\Gamma(m+\alpha)(y+m)!} p^{m+\alpha}(1-p)^y \\
&= \frac{(k+1)!}{(k+1-m)!} \sum_{x=k+1}^{\infty} \frac{\Gamma(x+\alpha)}{\Gamma(m+\alpha)x!} p^{m+\alpha}(1-p)^{x-m} \\
&= \frac{(k+1)!}{(k+1-m)!} \cdot \frac{\Gamma(\alpha)}{\Gamma(m+\alpha)} \cdot \left(\frac{p}{1-p}\right)^m \sum_{x=k+1}^{\infty} \frac{\Gamma(x+\alpha)}{\Gamma(\alpha)x!} p^{\alpha}(1-p)^x \\
&= \frac{(k+1)!}{(k+1-m)!} \cdot \frac{\Gamma(\alpha)}{\Gamma(m+\alpha)} \cdot \left(\frac{p}{1-p}\right)^m \cdot \Pr_{\alpha,p}\{Y>k\}
\end{aligned}
$$

Now

$$
\frac{\rho(m+1)}{\rho(m)} = \frac{k+1-m}{m+\alpha} \cdot \frac{p}{1-p}
$$

This is greater than one (so it pays to increase m) if and only if

$$
m + \alpha < (k+1+\alpha)p
$$

Hence we set

$$
m = \lceil (k+1+\alpha)p - \alpha \rceil
$$

or zero, whichever is greater.

4.1.3 Checks

There are a lot of thing to check about our analysis. First we need to check that we actually have a valid rejection sampling algorithm.

```
> alpha <- 2.22
> p <- 0.5
> k <- 20
> m <- max(ceiling((k + 1 + alpha) * p - alpha), 0)
> m
```

```
[1] 10

> nsim <- 1e6
> y <- rnbinom(nsim, size = alpha + m, prob = p)
> xprop <- y + m
> aprop <- exp(lfactorial(y) - lfactorial(xprop) + lfactorial(k + 1) -
+     lfactorial(k + 1 - m)) * as.numeric(xprop > k)
> max(aprop)

[1] 1

> x <- xprop[runif(nsim) < aprop]
> n <- length(x)
> fred <- tabulate(x)
> xfred <- seq(along = fred)
> pfred <- dnbinom(xfred, size = alpha, prob = p)
> pfred[xfred <= k] <- 0
> pfred <- pfred / sum(pfred)
> mfred <- max(xfred[n * pfred > 5])
> o <- fred
> o[mfred] <- sum(o[seq(mfred, length(fred))])
> o <- o[seq(k + 1, mfred)]
> e <- n * pfred
> e[mfred] <- sum(e[seq(mfred, length(fred))])
> e <- e[seq(k + 1, mfred)]
> chisqstat <- sum((o - e)^2 / e)
> pchisq(chisqstat, lower.tail = FALSE, df = length(o))

[1] 0.5276414
```

Seems to be o. k. (This number changes every time `Sweave` is run due to randomness in the simulation.)

Next we check that our performance formula is correct.

```
> length(x) / nsim

[1] 0.176283

> rho <- function(m, p) {
+     exp(lfactorial(k + 1) - lfactorial(k + 1 - m) +
+     lgamma(alpha) - lgamma(m + alpha) + m * (log(p) - log1p(- p)) +
```

```
+     pnbinom(k, size = alpha, prob = p, lower.tail = FALSE, log.p = TRUE))
+ }
> rho(m, p)
```

```
[1] 0.1765559
```

Finally, we check the performance of the algorithm over the range of mean values for which it may have trouble, from zero to a little more than k.

```
> mu <- seq(0.01, k + 5, 0.01)
> p <- alpha / (alpha + mu)
> m <- pmax(ceiling((k + 1 + alpha) * p - alpha), 0)
> r <- rho(m, p)
```

Figure 1 (page 21) shows the performance as a function of μ.

The performance is not great, but it will have to do until we find a better algorithm.

4.2 Poisson

We now work out the analogous algorithm for the Poisson distribution.

4.2.1 Algorithm

It is clear that taking limits as $\alpha \to \infty$ that the analogous algorithm for Poisson variates is as follows. The target distribution is k-truncated Poisson with untruncated mean μ. The proposal is $Y + m$, where Y is untruncated Poisson with mean μ.

Then the ratio of target PMF to proposal PMF is proportional to (dropping terms that do not contain x)

$$r(x) = \frac{(x - m)!}{x!} \cdot I(x > k)$$

This is a decreasing function of x. So the acceptance probability can be taken to be

$$a(x) = \frac{r(x)}{r(k + 1)} = \frac{(k + 1)!}{(k + 1 - m)!} \cdot \frac{(x - m)!}{x!} \cdot I(x > k)$$

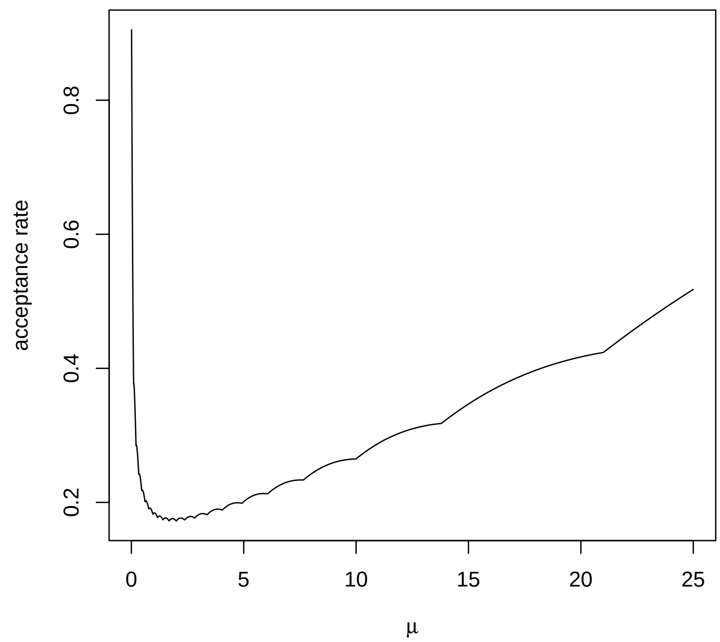

Figure 1: Performance of our algorithm for simulating k-truncated negative binomial with $k = 20$, $\alpha = 2.22$ and μ plotted.

4.2.2 Performance

To understand the performance of this algorithm, hence to understand how to chose m, we need to calculate the acceptance rate

$$
\begin{aligned}
\rho(m) &= E^*\{a(Y+m)\} \\
&= \sum_{y=0}^{\infty} a(y+m) \frac{\mu^y}{y!} e^{-\mu} \\
&= \sum_{y=0}^{\infty} \frac{(k+1)!}{(k+1-m)!} \cdot \frac{(y+m-m)!}{(y+m)!} \cdot I(y+m>k) \cdot \frac{\mu^y}{y!} e^{-\mu} \\
&= \frac{(k+1)!}{(k+1-m)!} \sum_{y=k+1-m}^{\infty} \frac{\mu^y}{(y+m)!} e^{-\mu} \\
&= \frac{(k+1)!}{(k+1-m)!} \cdot \mu^{-m} \sum_{x=k+1}^{\infty} \frac{\mu^x}{x!} e^{-\mu} \\
&= \frac{(k+1)!}{(k+1-m)!} \cdot \mu^{-m} \cdot \Pr_{\infty,\mu}\{Y>k\}
\end{aligned}
$$

Everything is fixed in our formula for acceptance rate except m, which we many choose to be any integer $0 \le m \le k+1$. Consider

$$
\frac{\rho(m+1)}{\rho(m)} = \frac{(k+1-m)}{\mu}.
$$

This is greater than one (so it pays to increase m) when

$$
k+1-m<\mu.
$$

Hence we set

$$
m = \lceil k+1-\mu \rceil
$$

or zero, whichever is greater.

4.2.3 Checks

There are a lot of thing to check about our analysis. First we need to check that we actually have a valid rejection sampling algorithm.

```
> mu <- 2.22
> k <- 20
> m <- max(ceiling(k + 1 - mu), 0)
> m
```

```
[1] 19
```

```
> nsim <- 1e6
> y <- rpois(nsim, lambda = mu)
> xprop <- y + m
> aprop <- exp(lfactorial(y) - lfactorial(xprop) + lfactorial(k + 1) -
+     lfactorial(k + 1 - m)) * as.numeric(xprop > k)
> max(aprop)
```

```
[1] 1
```

```
> x <- xprop[runif(nsim) < aprop]
> n <- length(x)
> fred <- tabulate(x)
> xfred <- seq(along = fred)
> pfred <- dpois(xfred, lambda = mu)
> pfred[xfred <= k] <- 0
> pfred <- pfred / sum(pfred)
> mfred <- max(xfred[n * pfred > 5])
> o <- fred
> o[mfred] <- sum(o[seq(mfred, length(fred))])
> o <- o[seq(k + 1, mfred)]
> e <- n * pfred
> e[mfred] <- sum(e[seq(mfred, length(fred))])
> e <- e[seq(k + 1, mfred)]
> chisqstat <- sum((o - e)^2 / e)
> pchisq(chisqstat, lower.tail = FALSE, df = length(o))
```

```
[1] 0.6070105
```

Seems to be o. k. (This number changes every time `Sweave` is run due to randomness in the simulation.)

Next we check that our performance formula is correct.

```
> length(x) / nsim
```

```
[1] 0.297668
```

```
> rho <- function(m, mu) {
+     exp(lfactorial(k + 1) - lfactorial(k + 1 - m) - m * log(mu) +
+     ppois(k, lambda = mu, lower.tail = FALSE, log.p = TRUE))
+ }
> rho(m, mu)
```

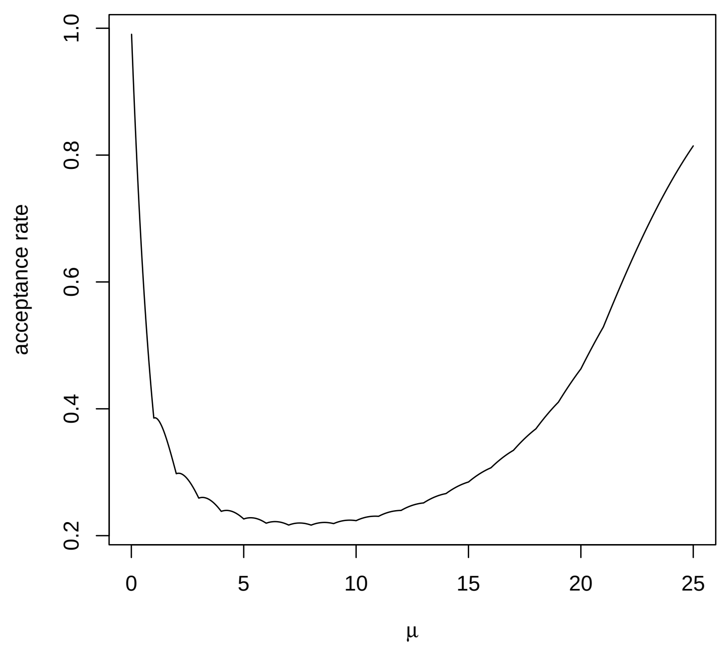

Figure 2: Performance of our algorithm for simulating k-truncated poisson with $k = 20$ and μ plotted.

```
[1] 0.2975126
```

Finally, we check the performance of the algorithm over the range of mean values for which it may have trouble, from zero to a little more than k.

```
> mu <- seq(0.01, k + 5, 0.01)
> m <- pmax(ceiling(k + 1 - mu), 0)
> r <- rho(m, mu)
```

Figure 2 (page 24) shows the performance as a function of μ.

```
> kseq <- c(0, 1, 2, 20, 100)
> mseq <- double(length(kseq))
> for (i in seq(along = kseq)) {
```

```
+       k <- kseq[i]
+       mu <- seq(0.01, k + 5, 0.01)
+       m <- pmax(ceiling(k + 1 - mu), 0)
+       r <- rho(m, mu)
+       mseq[i] <- min(r)
+ }
```

The performance of this algorithm seems to be fine for small k. However the worst case acceptance rate, which occurs for μ between $k/4$ and $k/2$, does seem to go to zero as k goes to infinity. For a zero-truncated Poisson distribution the worst case acceptance rate is 63.2%. For a two-truncated Poisson distribution the worst case acceptance rate is 48.2%. For a twenty-truncated Poisson distribution the worst case acceptance rate is 21.7%. For a one-hundred-truncated Poisson distribution the worst case acceptance rate is 10.2%.