Zero-inflated models: the devil in the tails

Timothy Daley

7/9/2019

Abstract

Zero-inflated models can sometimes suck really bad

I will begin as many statistics papers do with a quote from George Box, "Since all models are wrong the scientist must be alert to what is importantly wrong" (box1976science). There is currently a discussion on the "correctness" on the assumption of zero-inflation in single cell data (hicks2017missing, risso2018general, svensson2019droplet, and townes2019feature). I believe that this discussion misses the point. We should instead be talking about the utility of the models. Any specific model is going to be wrong in some way or another. The question then becomes whether the model is sufficiently useful in spite of its incorrectness.

In this paper we illustrate a specific and peculiar issue in zero-inflated models. This is their dependence on specifying the correct model. If the incorrect model is chosen, then the level of zero-infation can be wildly mis-estimated. This then results in incorrect inferences. This is because estimating the zero-inflation is equivalent to estimating the number of missing or unsampled genes (dietz2000estimation, deng2019molecular). There is a long literature in estimating the missing number of species in a sampling experiment, with most of the applications in estimating the number of missing species in ecology (fisher1943relation, bunge1993estimating, bunge2014estimating). We can take a few lessons from this literature. First, estimating the number of missing species is a very hard problem. The missing species are necessarily the rarest and least abundant species. It's difficult to rule out the possibility that there are a large number of missing species at very low abudance. This so-called length bias makes inference difficult and can lead to arbitrarily large risk in estimating the missing species (if the number species can be infinite, in our case the number of genes is bounded but the risk can still be extremely large) (johndrow2016estimating).

Consider the following situation. We observe the following counts from a single cell, with j equal to count and n_j equal to the number of genes with that count:

j	1	2	3	4	5	6	8	10	15	26
n_j	323	30	10	1	3	1	1	1	1	1

As an illustration, suppose we fit a negative binomial model (using the preseqR preseqR.ztnb.em function, dengpreseqr) and a lognormal-Poisson model (using the poilogMLE function of the poilog package, grotan2008poilog). The fits are shown below.

Since both models are two parameter models and the log-likelihood of the lognormal-Poisson is higher, this indicates that the lognormal-Poisson should be preferred. Now what happens when we extend the fit to include the zeros?

We see that the estimated number of zeros differs by an order of magnitude. Which is correct? Well, actually neither (naturally). The counts were simulated from twenty thousand genes with relative proportions following a Zipf-Mandelbrot model (mouillot1999comparison), e.g. $p_i \propto (i^{0.9} + 0.25)^{-1}$. Both models vastly underestimate the missing number of genes, and despite the fact that the log-normal Poisson model has a higher likelihood, the Gamma Poisson model comes closer to estimating the level of zero-inflation.

Though this problem is general in the analysis of sparse single cell data, we illustrate the issue in the context of differential expression. Differential expression analysis is typically framed as testing the difference of means between populations. In a zero-inflated model the mean parameter is typically conditional on positive expression (or non-zeroness), as the zero-inflation is considered a technical effect (kharchenko2014bayesian). The mean expression is then the total counts divided by the estimated number of cells that express that gene,

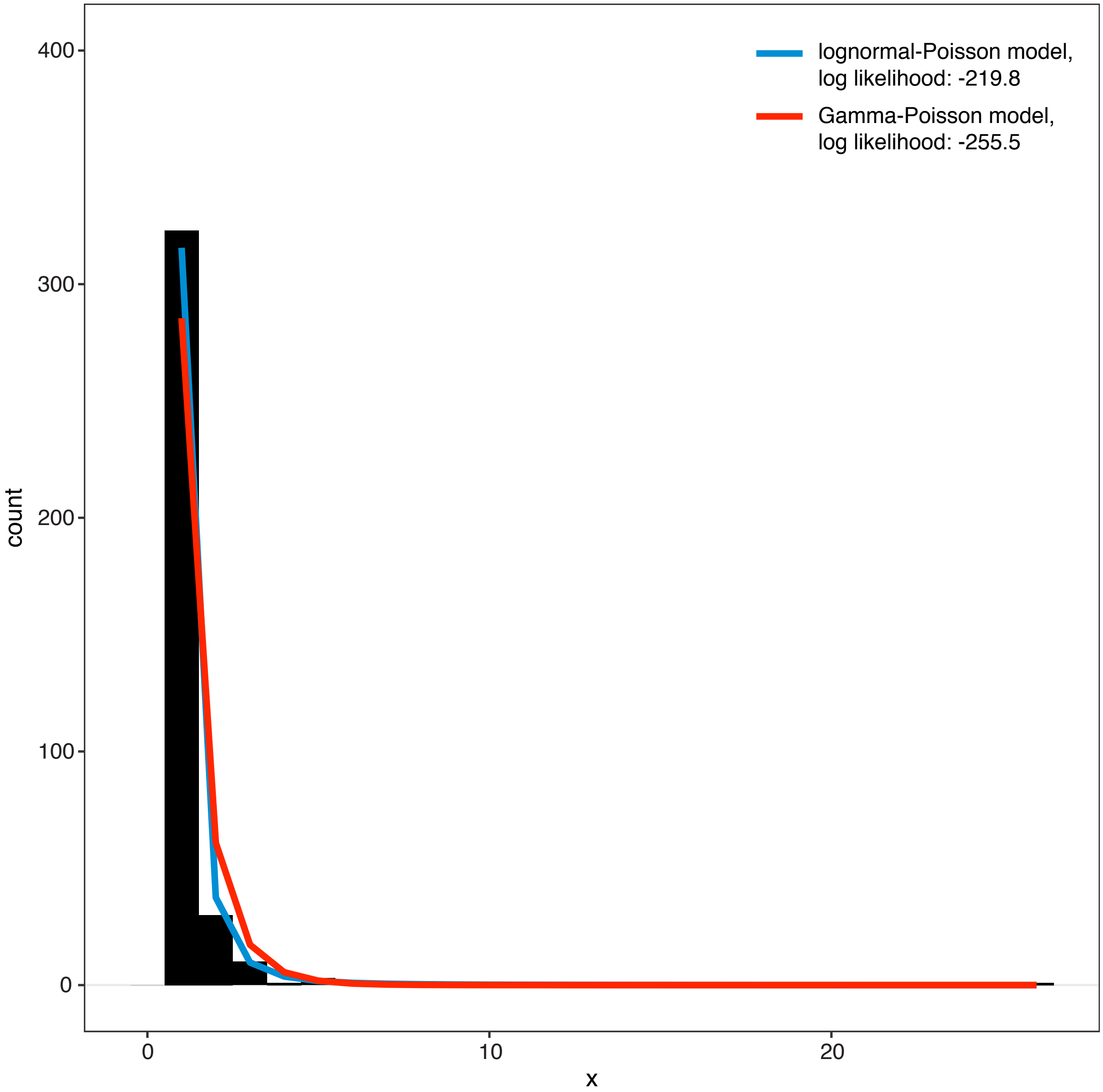

Figure 1: Zero-truncated histogram and estimated fits

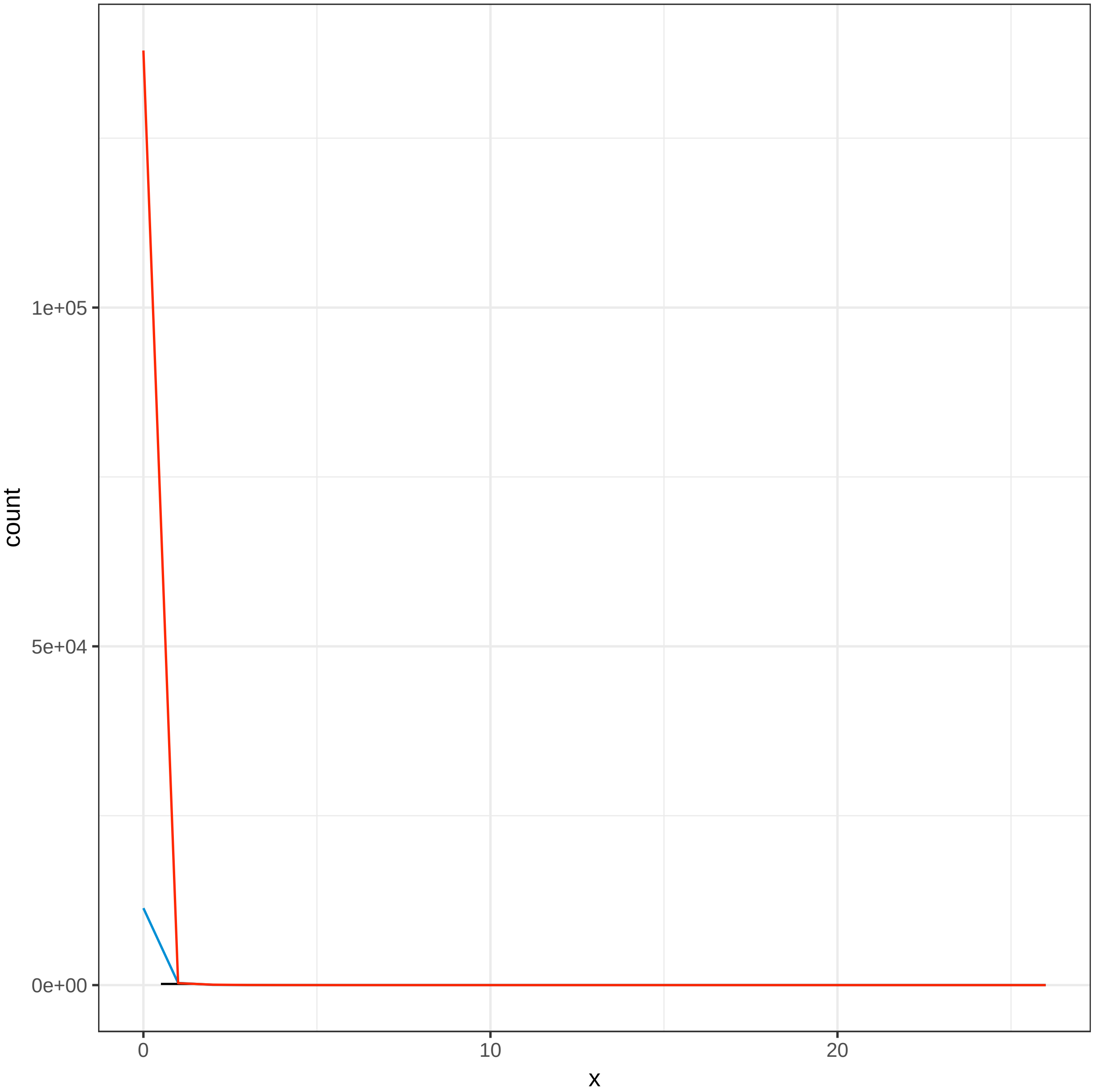

Figure 2: Full fits

so that the mean expression is inversely proportional to the estimated dropout. We can see immediately that errors in estimating the dropout will have a profound effect on the estimated expression, and consequently the estimated dropout. Using the previous example as a motivation, we will look at the performance of differential expression in two cases: when the model is correctly specified and when the model is incorrectly specified. To ensure that the only impact is the model specification, we will keep all other quantities the same.

We simulate from the following model, which is combination of the simulation models presented by lun2016pooling and hicks2015widespread. The parameters were chosen so that summary statistics (e.g. cell-level and gene-level variance) are similar to observed summary statistics from droplet-based single cell RNA-seq data.

$$\begin{aligned}
x_{gi} &= \text{count of gene } g \text{ in cell } i; \\
g &= 1, \dots, 2 \cdot 10^4; \\
i &= 1, \dots, 2 \cdot 10^3, \text{ half in group 1 and half in group 2;} \\
\phi_{ij} &= 5 \text{ for 10\% of the genes in group 1} \\
&= 0.2 \text{ for 10\% of the genes in group 1} \\
&= 1 \text{ otherwise;} \\
\lambda_{0j} &\sim \text{Gamma(shape } = 1, \text{ rate } = 1); \\
\theta_i &\sim \log \text{Normal}(-1/2, 1). \\
d &= 0.1; \\
\pi_{ij} &= 1/(1 + \exp(\log(1/\pi_{0j} - 1) + 0.5 * (\phi_{i,j} - \bar{\phi}_{g\cdot}))) \\
\pi_{0j} &\sim \text{Beta}(1, 3) \\
\lambda_{gi} &\sim d * \phi_{gi} * \lambda_{0g} * \theta_i * \psi_{gi}; \\
x_{gi} &\sim \pi_{gi} 1(0) + (1 - \pi_{gi}) \text{Poisson}(\lambda_{gi});
\end{aligned}$$

To ensure that all other parameters are identical between simulations (and to ensure that the only difference is due to the different models) we set the random seed and simulate these first. The difference between the models will be only in the random variables ψ_{gi}, a positive random variable with expecation equal to 1. We consider two cases, where ψ_{gi} are independent Gamma random variables and where ψ_{gi} are independent log-Normal random variables. We vary the variance of ψ_{gi} from 1 to 64 to look at the performance as the technical variance increases. We look at how a negative binomial differential expression algorithm (risso2018general, van2018observation) performs when the model in both cases: when the model is correctly specified and when the model is incorrectly specified.

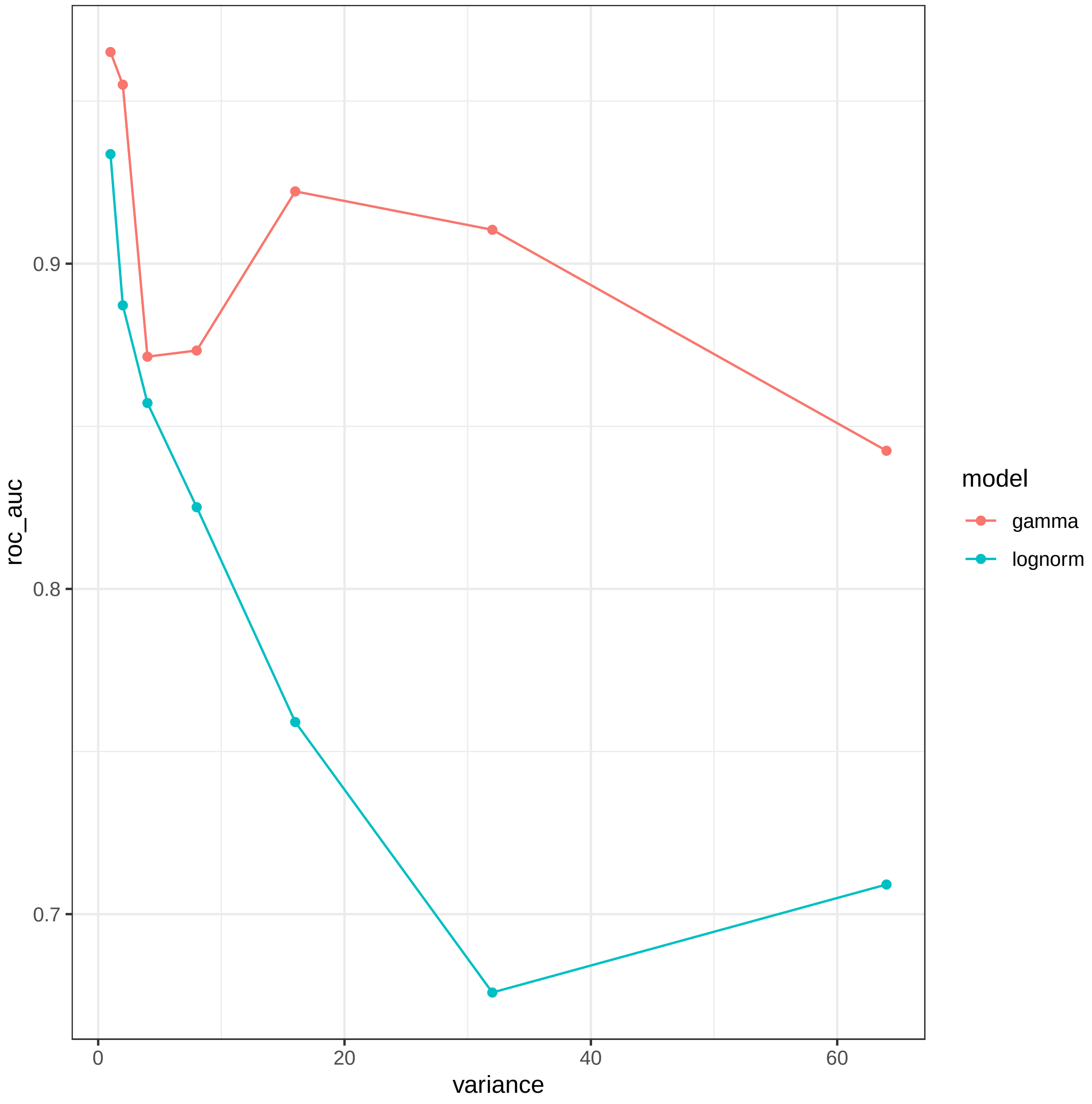

Figure 3: Area under the receiver-operator curve (roc_auc) when the assumed model is correct (gamma) and incorrect (lognorm).

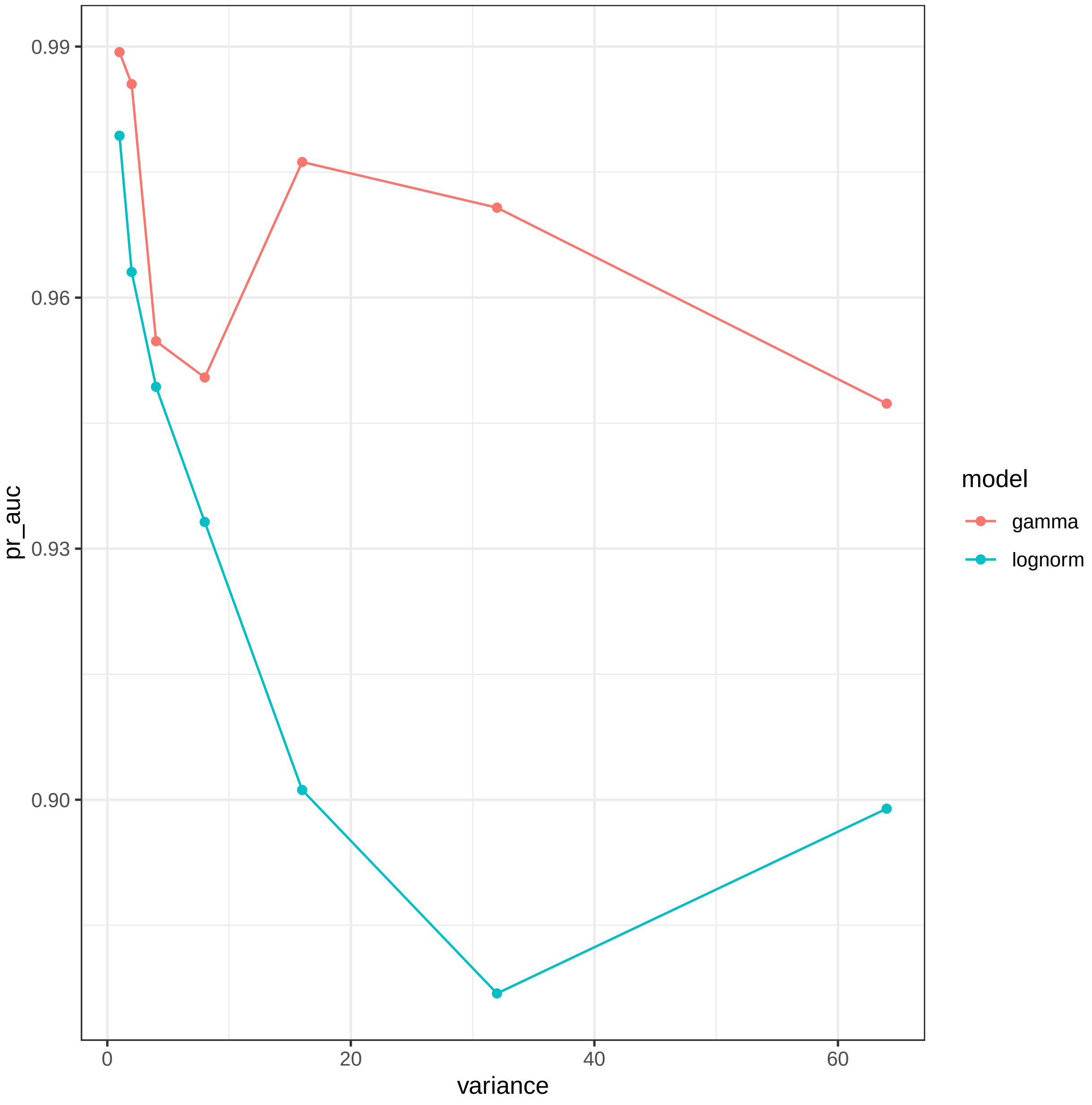

Figure 4: Area under the precision-recall curve (pr_auc) when the assumed model is correct (gamma) and incorrect (lognorm).