Computationally Efficient Models for Count Data with Varying Levels of Dispersion*

E. Huch†[1], C. Berrett[2], M. Ferlic[1], and K. Sellers[3]

[1]Department of Statistics, University of Michigan, Ann Arbor, MI, USA

[2]Department of Statistics, Brigham Young University, Provo, UT, USA

[3]Department of Statistics, North Carolina State University, Raleigh, NC, USA

June 8, 2024

This is a working paper. The most recent version can be downloaded from the first author's website: eastonhuch.com.

Abstract

The Poisson distribution is ubiquitous for modeling count data but is limited in application by its assumption of equidispersion. Alternative models exist that do not suffer from this challenge but most are (1) limited to either under- or over-dispersed data, (2) only provide estimates of moments, or (3) are computationally prohibitive for large data sets. We review existing methods for count data with varying levels of dispersion, including some new results for generalized Poisson regression. We introduce a latent-variable model based on a discretized log-normal distribution, develop a scalable EM algorithm to estimate it, and provide straightforward likelihood-based theory for performing statistical inference. Finally we illustrate these methods in simulation and on a case study involving algae blooms; we find that our latent-variable model performs nearly as well as the leading method (COM-Poisson regression) but at a fraction of the computational cost.

Keywords: underdispersion, overdispersion, GLM, count data, latent-variable models

1 Introduction

The Poisson distribution is ubiquitous in the statistical modeling of count data (see, for example, Frome et al., 1973; McCullagh and Nelder, 1989; Agresti, 2013). However, its

*Work in progress.

†Corresponding author. Email: ekhuch@umich.edu.

assumption of equidispersion—that the variance is equal to the mean—is often unrealistic in real-world applications. Historically, overdispersion has received a great deal of attention due to (1) its frequent occurrence in real data sets and (2) compelling theoretical reasons why it exists (see, for example, the discussion in Hilbe, 2011). More recently, however, researchers have begun to devote more effort to understanding and accounting for underdispersion in count data (e.g., Sellers and Morris, 2017).

Researchers have proposed various extensions of the standard Poisson model to account for deviations from equidispersion. Some of these solutions—most notably the negative binomial model (Lawless, 1987)—are appropriate for overdispersed but not underdispersed data. Others, such as the condensed Poisson (Sellers and Morris, 2017) or binomial (Kokonendji, 2014) models, suffer from the opposite problem and are appropriate only for equidispersed or underdispersed data.

Models appropriate for both underdispersion *and* overdispersion are relatively less common. These models are particularly useful for situations in which one model will be applied to multiple data sets, each of which could be either underdispersed or overdispersed conditional on covariates. While the benefit of these models is clear, their flexibility often comes at the expense of complexity. For example, the Conway-Maxwell-Poisson (COM-Poisson) distribution is suitable for both cases but fitting it requires (1) approximations for moments and (2) computationally demanding subroutines for estimating the normalizing constant (Shmueli et al., 2005).

Among these more flexible models, the generalized Poisson distribution (GPD) is notable for its relative simplicity, both in terms of its probability mass function (pmf) and its moments, both of which have simple, closed-form expressions. Consul and Famoye (1992) and Famoye (1993) developed GPD regression models now referred to as the GP-1 and GP-2, respectively. Further, Zamani and Ismail (2012) generalized these models into a larger class called the GP-P that parametrically nests these models. In reviewing existing models, we devote particular attention to the GP-P model, including deriving its expected information matrix (EIM) and a lower bound on it dispersion parameter, φ. While the appeal of the GP-P is clear, it suffers from three notable disadvantages for underdispersed data:

1. The support is limited to counts below an upper bound.
2. The probability mass function is only approximate and does not sum to exactly to one.
3. The parameter space involves non-smooth restrictions designed to maintain the summed probability mass within 0.5% of one.

These disadvantages limit the general applicability of the GP-P model because they restrict its applicability to data sets that are overdispersed, equidispersed, or—perhaps—slighly underdispersed; though, the latter case can quickly become problematic, especially with small counts. Thus, we summarize and consolidate results for this model largely for historical purposes and as a benchmark for our proposed latent-variable method, which is based on a latent Gaussian random variable, Z_i, with mean and standard deviation modeled as a function of covariates, x_i, as follows:

$$\mu_i = f(\mathbf{x_i})^\intercal \boldsymbol{\beta} \quad \log(\sigma_i) = g(\mathbf{x_i})^\intercal \boldsymbol{\alpha}, \tag{1}$$

for some analyst-specified covariate transformations, f and g. We could, for instance, employ a linear model for the mean function and an intercept-only model for the log standard deviation by setting these functions as follows: $f(\mathbf{x})) = (1, x^\intercal)^\intercal$ and $g(\mathbf{x})) = 1$. We then implicitly define the outcome variable as $Y_i = \lfloor \exp(Z_i) \rfloor$, effectively setting

$$\Pr\left(Y_i = y; x, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = \Phi\left[\frac{\log(y+1) - f(\mathbf{x})^\intercal \boldsymbol{\beta}}{\exp\left\{g(\mathbf{x})^\intercal \boldsymbol{\alpha}\right)\}}\right] - \Phi\left[\frac{\log(y) - f(\mathbf{x})^\intercal \boldsymbol{\beta}}{\exp\left\{g(\mathbf{x})^\intercal \boldsymbol{\alpha}\right)\}}\right], \tag{2}$$

where Φ is the standard normal CDF. Whereas generalized Poisson models encounter challenges with underdispersed data, this latent-variable model does not because Z_i—and consequently Y_i—has a valid probability distribution for any values of $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$. The primary drawbacks of our latent-variable model are (1) it does not nest the Poisson distribution and (2) the parameters are difficult to interpret directly. While these drawbacks are worth considering, we argue that its flexibility and computational convenience can, at least in some settings, compensate for these challenges. In particular, when interest lies in quantifying prediction uncertainty in the presence of varying levels of dispersion, our latent-variable model is an appealing alternative to existing methods.

The rest of the paper proceeds as follows. Section 2 reviews existing methods for modeling count data with varying levels of dispersion. Section 3 further develops the discrete log-normal model discussed above, including algorithms for computing the maximum likelihood estimate and its large-sample properties. Sections 4 and 5 compare the methods in simulation and a case study, respectively. Finally, Section 6 concludes with a brief summary and discussion.

2 Existing Methods

In this section, we review three existing methods for modeling count data with varying levels of dispersion, the first two of which are likelihood based. Throughout, we restrict our attention to models that are capable of describing both underdispersed and overdispersed data—omitting, for example, negative binomial models that are appropriate only for overdispersed data. In Sections 4 and 5, we compare several variants of these methods to our proposed discrete log-normal model. To keep the notation light, we use $\boldsymbol{\beta}$ to represent regression parameters for the mean and $\boldsymbol{\alpha}$, for the variance. Note, however, that these parameters are not directly comparable across models. We handle this complexity in our simulation study by comparing estimates of predictive quantities, such as conditional means (fitted values).

2.1 Generalized Poisson Models

The family of generalized Poisson distributions (GPD) includes distributions that are both underdispersed and overdispersed relative to the standard Poisson distribution. Further, its moments are available in closed form—a desirable property from a computational vantage point. For these two reasons, the GPD—at least at first glance—appears to be a promising candidate method in our search. As we describe its properties, however, we will see that it suffers from some theoretical challenges that limit its applicability to underdispersed data. Thus, we include it as a comparator, but we suggest caution in applying it when underdispersion may be present.

The generalized Poisson distribution (GPD) introduces an additional parameter, $\delta \in (-1, 1)$, to the standard Poisson distribution. Following Consul (1989), the pmf can be expressed as

$$p(y \mid \theta, \delta) = \frac{\theta\,(\theta + \delta y)^{y-1} \exp(-\theta - \delta y)}{y!} \qquad \text{for } y = 0, 1, 2, ... \tag{3}$$

with mean $\mathrm{E}(Y) = \theta/(1-\delta)$ and variance $\mathrm{Var}(Y) = \theta/(1-\delta)^3$. When $\delta = 0$, the GPD reduces to the standard Poisson distribution. When $0 < \delta < 1$[1] (the *overdispersed* case), Y can equal any non-negative integer and θ, any positive real value. When $-1 < \delta < 0$ (the *underdispersed* case), two restrictions are necessary. First, the support is truncated such that $\theta + \delta y > 0$. This restriction ensures that the pmf is non-negative. Second, the parameters are constrained such that $\max(-1, -\theta/4) < \delta$. Although the infinite summation, $\sum_{y=0}^{\infty} p(y \mid \theta, \delta)$, does equal unity, the summation over the truncated support, $\sum_{y=0}^{m} p(y \mid \theta, \delta)$,

[1]Some authors allow $\delta \in [-1, 1]$, but we exclude equailty to avoid some technicalities (e.g., infinite moments).

does not. However, the above restriction on the parameter space ensures that the truncation error is less than 0.5%, which several authors have argued is sufficient for many practical applications (Consul and Shoukri, 1985; Consul and Famoye, 2006).

Consul and Famoye (1992) and Famoye (1993) showed how to introduce covariates to model the GPD's mean function, similar to the Poisson generalized linear model (GLM). The two models they introduced have found plentiful applications in the statistics literature (see, for example, Wang and Famoye, 1997; Famoye et al., 2004) and have come to be known as the GP-1 and GP-2 regression models (Yang et al., 2009). More recently, Zamani and Ismail (2012) introduced the GP-P model, a model that parametrically nests the GP-1 and GP-2 models. The development of the GP-P mirrors that of the NB-P, the family of models that nests the two primary variants of negative binomial regression (Hilbe, 2011)[2]. The GP-P facilitates comparisons of the GP-1 and GP-2 models and, perhaps more importantly, allows greater flexibility in selecting a variance function.

The GP-P model transforms the parameters of the standard GPD such that $\theta = \mu/(1 + \phi\mu^{P-1})$ and $\delta = \phi\mu^{P-1}/(1 + \phi\mu^{P-1})$. The pmf for the GP-P is then given as follows:

$$p(y \mid \mu, \phi, P) = \frac{\mu(\mu + \phi\mu^{P-1}y)^{y-1}\exp\big(-\frac{\mu+\phi\mu^{P-1}y}{1+\phi\mu^{P-1}}\big)}{(1 + \phi\mu^{P-1})^{y}\, y!}, \quad \text{for } \mu > 0, P \in (-\infty, \infty), \phi \in (\phi_{\min}(\mu, P), \infty), \tag{4}$$

where $\mu = \mathrm{E}(Y)$, ϕ is the dispersion parameter with minimum value $\phi_{\min}(\mu, P)$, and P determines the variance function as follows: $\mathrm{Var}(Y) = (1 + \phi\mu^{P-1})^2\mu$. When $\phi = 0$, the GP-P reduces to the standard Poisson distribution. $\phi > 0$ also produces a valid pmf, but it is overdispersed relative to the Poisson. $\phi < 0$, on the other hand, corresponds to an underdispersed distribution that is only *approximately* valid.

To ensure the pmf sums to a value close to one, we must restrict ϕ such that the corresponding parameters in the original parameterization (θ, δ) respect the restrictions described above. Because these restrictions are somewhat complicated, we denote the minimum value of ϕ generically as $\phi_{\min}(\mu, P)$. We show in Appendix B that $\phi_{\min}(\mu, P) \geq -2^{-P}$ when $P \in [1, 2]$; for other values of P, ϕ has no lower bound.

These complications raise serious concerns regarding the applicability of the GP-P model—and, more generally, the GPD model—to underdispersed data. In particular, allowing ϕ to depend on covariates is not advisable unless (1) underdispersion is unlikely to be present and (2) we employ a link function that ensures $\phi > 0$.

[2]Note, however, that the variance functions for the NB-P and GP-P generally do not agree.

We include the GP-P in our simulation study (Section 4) for the sake of comparison. In contrast with the other methods, however, we assume that ϕ is fixed across observations to avoid computational and theoretical issues related to the challenges described above. We estimate the GP-P model via maximum likelihood estimation and employ asymptotic likelihood-based standard errors for inference; the details can be found in Appendix A. In particular, we were unable to locate the expected information matrix (EIM) in the literature, so we derive it in Appendix A.1.

2.2 COM-Poisson Models

The COM-Poisson distribution is another appealing alternative for modeling count data with varying degrees of dispersion. It was revived in the statistics literature by Shmueli et al. (2005). Subsequently, Lord et al. (2010) and Sellers and Shmueli (2010) introduced COM-Poisson generalized linear models, which have since been generalized in various directions; see Sellers (2023) Chapter 5 for an overview. Our development most closely follows Chatla and Shmueli (2018), which presents an efficient iteratively reweighted least squares algorithm for fitting COM-Poisson models in which the dispersion is modeled as a function of covariates.

Similar to the generalized Poisson distribution, the COM-Poisson distribution introduces an additional parameter into the Poisson distribution that controls the degree of dispersion. The probability mass function for the COM-Poisson distribution is defined as follows:

$$\Pr(Y = y \mid \lambda, \nu) = \frac{\lambda^y}{(y!)^\nu \zeta(\lambda, \nu)}, \quad \text{where} \quad \zeta(\lambda, \nu) = \sum_{k=0}^{\infty} \frac{\lambda^k}{(k!)^\nu} \tag{5}$$

with parameters $\lambda > 0$ and $\nu \geq 0$. When $\nu = 0$, we constrain $\lambda \in (0, 1)$ such that the sum converges and the distribution is non-degenerate; note that this case reduces to the geometric distribution. When $\nu > 0$, we require only that $\lambda > 0$. Note that $\nu = 1$ corresponds to the standard Poisson distribution. $\nu < 1$ corresponds to over dispersion and $\nu > 1$, to under dispersion. As $\nu \to \infty$, we obtain the Bernoulli distribution as a limiting case with parameter $\frac{\lambda}{1+\lambda}$. $\zeta(\lambda, \nu)$ plays the role of normalizing constant. Unfortunately, it does not have a closed-form representation, so it must be approximated in practice, either analytically or via numerical approximation.

One particularly attractive aspect of the COM-Poisson distribution is that it is a member of the exponential family, having natural parameters $\log(\lambda)$ and $-\nu$. Thus, it inherits some desirable properties from that class, such as useful expressions for its moments and direct application of many aspects of the theory of GLMs. We employ the following common GLM

formulation:

$$\log(\lambda_i) = f(\mathbf{x_i})^\intercal \boldsymbol{\beta} \tag{6}$$

$$\log(\nu_i) = g(\mathbf{x_i})^\intercal \boldsymbol{\alpha} \tag{7}$$

The log links ensure that this formulation always produces valid values of λ_i and ν_i. We denote the regression parameter for λ_i as $\boldsymbol{\beta}$ because $\lambda_i \approx \mathrm{E}(Y_i \,|\, \mathbf{x_i})$ when $\nu_i \approx 1$. Note, however, that λ_i is quite far from $\mathrm{E}(Y_i \,|\, \mathbf{x_i})$ when ν_i is far from one. Thus, although we denote the regression parameter for λ_i as $\boldsymbol{\beta}$, it may not correspond closely with the mean in practice and, consequently, is not be directly comparable to regression parameters produced by other methods.

In our simulation study, we employ the model-fitting procedure provided in Chatla and Shmueli (2018). The procedure is a two-step IRLS algorithm that leverages the expected information matrix for efficiency and robustness. It makes use of two approximations for increased efficiency: (1) Stirling's approximation for large values of $\log(y_i!)$ and (2) asymptotic expressions for the mean and variance due to Gaunt et al. (2019). Inference proceeds via standard large-sample asymptotics, using the inverse expected information matrix at the estimated parameter values to estimate their covariance.

2.3 Methods Based on Moment Conditions

The next family of methods we discuss is those based on moment conditions, which encompasses both 'quasi-likelihood' and 'pseudo-likelihood' methods. Both quasi-likelihood and pseudo-likelihood methods generalize maximum likelihood estimation. The primary difference is that quasi-likelihood is based on deviance residuals while pseudo-likelihood methods are based on Pearson residuals (Nelder and Lee, 1991).

Rather than assuming a full probability distribution, these methods assume a parametric form only for certain moments of the data-generating distribution. For instance, the quasipoisson model assumes that $\log(\mu) = f(\mathbf{x})^\intercal \boldsymbol{\beta}$ and $\mathrm{var}(Y) = \phi\,\mu$, where $\mu = \mathrm{E}(Y)$, $\boldsymbol{\beta}$ is a vector of regression coefficients, and ϕ is an 'overdispersion parameter.' Nelder and Lee (1991) and Lee and Nelder (2000) discuss extensions that also model ϕ using covariates, a suggestion first proposed in the GLM literature by Pregibon (1984). In Sections 4 and 5, we implement a version of this method with $\log(\phi) = g(\mathbf{x})^\intercal \boldsymbol{\alpha}$, where g maps the covariate x to the linear predictors for the variance function and $\boldsymbol{\alpha}$ is an unknown vector of coefficients.

The key benefit of these moment-based models is that the analyst does not need to make any distributional assumptions beyond the first and second moments. Estimation is performed by solving a set of estimating equations. Using the pseudo-likelihood formulation of our quasipoisson model with $\log(\phi) = g(\mathbf{x})^\intercal \boldsymbol{\alpha}$, there are two such equations:

$$\mathbf{0} = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{\phi_i} f(\mathbf{x_i}) = \sum_{i=1}^{n} \frac{Y_i - \exp\{f(\mathbf{x_i})^\intercal \boldsymbol{\beta}\}}{\exp\{g(\mathbf{x_i})^\intercal \boldsymbol{\alpha}\}} f(\mathbf{x_i}) \tag{8}$$

$$\mathbf{0} = \sum_{i=1}^{n} \left(\frac{R_i^2}{\phi_i} - 1 \right) g(\mathbf{x_i}) = \sum_{i=1}^{n} \left(\frac{[Y_i - \exp\{f(\mathbf{x_i})^\intercal \boldsymbol{\beta}\}]^2}{\exp\{f(\mathbf{x_i})^\intercal \boldsymbol{\beta} + g(\mathbf{x_i})^\intercal \boldsymbol{\alpha}\}} - 1 \right) g(\mathbf{x_i}), \tag{9}$$

where $R_i = (Y_i - \mu_i)/\sqrt{\mu_i}$ denotes the Pearson residuals. As noted by Nelder and Lee (1991), the challenge with these estimating equations is that they are not independent; equation (8) depends on (9) and vice versa. Nelder and Lee (1991) suggests solving the equations in alternating form but concludes the paper noting that "there remain substantial statistical problems in making inferences from these models." The key issue is that the asymptotic covariance matrices assume (in our notation) that either $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ is known; i.e., they do not account for the fact that $\boldsymbol{\alpha}$ is estimated when making inferences for $\boldsymbol{\beta}$, or vice versa.

We propose a simple solution to this problem in the form of stacked estimating equations (Carroll et al., 2006, Appendix A.6.6). We start by fixing $\boldsymbol{\alpha} = \mathbf{0}$ and solving Equation (8) for $\boldsymbol{\beta}$. We then fix $\boldsymbol{\beta}$ at this solution and solve Equation (9) for $\boldsymbol{\alpha}$. We continue iterating between solving Equations (8) and (9), each time fixing one of the parameters at its most recent estimated value, until solving each equation $k \in \mathbb{N}$ times (we set $k = 3$ in Sections 4 and 5). We then calculate asymptotic standard errors for the final estimates, $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$, using a sandwich estimator from the full set of $2k$ equations.

In Section 4, we show in simulation that this procedure is computationally efficient and produces statistically calibrated estimates for the first two moments with modest sample sizes. The primary drawback of this method compared to the others that we investigate is that inference is limited to the first two moments of the data distribution.

3 Discrete Log-Normal Model

In the last section, we introduced three existing methods for modeling count data with varying levels of dispersion, each of which possessed certain desirable qualities. The GP-P model is computationally efficient (because its moments are available in closed form), but it suffers from some theoretical issues that restrict its use to only mild levels of underdispersion.

In contrast, the COM-Poisson model is always a valid probability distribution and, moreover, is a member of the exponential family. However, its moments and normalizing constant are not available in closed form, so they must be approximated, which can be computationally demanding.

The moment-based models are both computationally efficient *and* always result in valid statistical models, so they are an appealing alternative when interest lies in the first two moments of the data. There is a cost to making limited assumptions, however. For instance, various GLM model diagnostics and modeling tools are not directly applicable. Additionally, inference for predictive quantities—in particular, prediction intervals for a new response—are complicated by the fact that these methods do not possess a predictive distribution. Some methods for the latter challenge have been introduced in the literature; e.g., conformal methods (Lei et al., 2018; Foygel Barber et al., 2021) and prediction intervals specifically designed for count data (Kim et al., 2022), some of which do not require a predictive distribution. While these methods are promising, practitioners should also be aware of their shortcomings, which often include substantial increases in required computation time, overly wide intervals, or coverage guarantees that apply only marginally or require additional assumptions.

In this section, we develop a discrete log-normal regression model that offers an attractive compromise among the qualities discussed above. In particular, the discrete log-normal model possesses the following desirable properties:

- It allows a wide range of dispersion from almost constant variance to extreme overdispersion
- Model-fitting and inference is computationally efficient
- The parameters do not need to obey any constraints
- The model possesses a valid predictive distribution

In the remainder of this section, we describe the procedures for applying this model in practice. Section 3.1 describes maximum likelihood model-fitting procedures, including an efficient second-order EM algorithm. Section 3.2 discusses large-sample inference procedures, including several approaches for forming prediction intervals.

3.1 Model-fitting Procedures

This section describes two procedures for fitting the discrete log-normal model. The first is a standard Newton–Raphson algorithm. The second is a an efficient second-order EM

algorithm.

3.1.1 Newton–Raphson Algorithm

The first approach we introduce is to maximize the log likelihood via the Newton–Raphson algorithm. Let $\boldsymbol{\theta} = (\boldsymbol{\alpha}^\intercal, \boldsymbol{\beta}^\intercal)^\intercal$ and let $\ell(\boldsymbol{\theta})$ denote the log-likelihood—the log of Equation (2) as a function of $\boldsymbol{\theta}$. The Newton–Raphson updates can then be expressed as follows:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \mathbf{H}\left(\boldsymbol{\theta}^{(t)}\right)^{-1} \mathbf{s}\left(\boldsymbol{\theta}^{(t)}\right), \tag{10}$$

where $\mathbf{s}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta})$ is the score function and $\mathbf{H}(\boldsymbol{\theta}) = \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal} \ell(\boldsymbol{\theta})$ is the observed information matrix. Appendix C.1 provides the gradient and Hessian required for these Newton–Raphson updates. They rely on moments of a truncated normal distribution which, fortunately, are available in closed form. However, the moments are ratios with (potentially very small) probabilities in the denominator, so accurately computing them requires some care. Appendix C.1 shows how to do this using the identity $\log(b-a) = \log(a) + \log(b/a - 1)$ for $0 < a < b$.

We note that first-order or quasi-Newton methods (e.g., BFGS) could also be applied. However, in simulations not reported in this paper, we found that the standard Newton–Raphson algorithm performed better, so it is the only one detailed here. With very large sample sizes, we expect that first-order methods such as stochastic gradient ascent might perform relatively better.

3.1.2 Expectation Maximization Algorithm

We now provide a second-order EM algorithm for fitting the discrete log-normal model. Because Y_i is a deterministic function of the latent Z_i, the 'full-data' log-likelihood is simply $\sum_{i=1}^n \log p(z_i; \boldsymbol{\alpha}, \boldsymbol{\beta})$, where $p(z_i \,|\, \mathbf{x_i}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is the normal pdf with mean $\mu_i = \exp\{f(\mathbf{x_i}^\intercal \boldsymbol{\beta})\}$ and standard deviation $\sigma_i = \exp\{g(\mathbf{x_i}^\intercal \boldsymbol{\alpha})\}$.

The E-step of the algorithm involves calculating the expected value of the full-data log likelihood under the conditional distribution of Z_i given $Y_i = y_i$. This conditional distribution is a truncated normal distribution whose moments—conveniently—are available in closed form (Kotz et al., 1994); Equation (50) in Appendix C.2 provides the formulas. Denoting the first and second moments of this distribution as e_{1i} and e_{2i}, the full expectation becomes

$$q(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^n \left\{ -\log\left(\sqrt{2\pi}\right) - g(\mathbf{x_i})^\intercal \boldsymbol{\alpha} - \frac{e_{2i} - 2e_{1i}\, f(\mathbf{x_i})^\intercal \boldsymbol{\beta} + [f(\mathbf{x_i})^\intercal \boldsymbol{\beta}]^2}{2\exp\{2\, g(\mathbf{x_i})^\intercal \boldsymbol{\alpha}\}} \right\}. \tag{11}$$

The M-step of the algorithm then involves finding values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that maximize $q(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Because $q(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is quadratic in $\boldsymbol{\beta}$, straightforward matrix calculus reveals that the optimal value of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta}^* = \left(\mathbf{F}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{F}\right)^{-1} \mathbf{F}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{e_1}, \tag{12}$$

where $\mathbf{F}$ is an n-row matrix with row i equal to $f(\mathbf{x_i})^\intercal \boldsymbol{\beta}$, $\boldsymbol{\Sigma}$ is an $n \times n$ diagonal matrix having the i-th diagonal element equal to $\sigma_i^2 = \exp\{2\, g(\mathbf{x_i})^\intercal \boldsymbol{\alpha}\}$, and $\mathbf{e_1}$ is an n-dimensional column vector having element i equal to e_{1i}.

In contrast, $\boldsymbol{\alpha}$ does not have a closed-form update, so we use Newton–Raphson updates to approximately maximize $q(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\alpha}$. We provide the gradient and Hessian in Appendix C.2. Because these values depend on $\boldsymbol{\beta}$, we first update $\boldsymbol{\beta}$ using Equation (12) and then perform Newton–Raphson updates for $\boldsymbol{\alpha}$ using the updated value of $\boldsymbol{\beta}$.

EM iterations proceed until changes in the marginal log-likelihood (or parameters) falls below a pre-specified threshold. The full EM algorithm is summarized in Algorithm 1.

Algorithm 1 Expectation-Maxmization Algorithm for the Discrete Log-Normal Model

$\ell_{\text{prev}} \leftarrow \ell(\boldsymbol{\alpha}, \boldsymbol{\beta})$
$\ell_{\text{curr}} \leftarrow \infty$
while $\ell_{\text{curr}} - \ell_{\text{prev}} > \epsilon$ **do**
  $\ell_{\text{prev}} \leftarrow \ell_{\text{curr}}$
  Calculate $\mathbf{e_1}$ and $\mathbf{e_2}$ using Equation (50) in Appendix C.2
  $\Sigma_{ii} \leftarrow \exp\{2\, g(\mathbf{x_i})^\intercal \boldsymbol{\alpha}\}$
  $\boldsymbol{\beta} \leftarrow \left(\mathbf{F}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{F}\right)^{-1} \mathbf{F}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{e_1}$
  for k in $1, 2, \ldots, K$ **do**
    Calculate $\mathbf{e_1}$ and $\mathbf{e_2}$ using Equation (50) in Appendix C.2
    Calculate $\tilde{\mathbf{s}}_{\boldsymbol{\beta}}(\boldsymbol{\alpha}), \tilde{\mathbf{H}}_{\boldsymbol{\beta}}(\boldsymbol{\alpha})$ using Equation (51) in Appendix C.2
    $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \tilde{\mathbf{H}}_{\boldsymbol{\beta}}(\boldsymbol{\alpha})^{-1} \tilde{\mathbf{s}}_{\boldsymbol{\beta}}(\boldsymbol{\alpha})$
  end for
  $\ell_{\text{curr}} \leftarrow \ell(\boldsymbol{\alpha}, \boldsymbol{\beta})$
end while

3.2 Inference Procedures

Both the Newton–Raphson and EM algorithms provided in the previous section provide strategies for computing the maximum likelihood estimate. Because the discrete log-normal model satisfies the standard regularity conditions for large-sample theory of maximum likelihood estimators, the inference procedures are straightforward. In particular, under correct model specification, standard large-sample theory guarantees that $\hat{\boldsymbol{\theta}}$ is consistent and asymptotically normal:

$$\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \xrightarrow{d} \text{Normal}\left(\mathbf{0}, \mathcal{H}(\boldsymbol{\theta})^{-1}\right), \tag{13}$$

where $\mathcal{H}(\boldsymbol{\theta})$ is the expected (Fisher) information matrix. In practice, we do no know the expected information matrix because (1) it is not available analytically and (2) we do not know the true value of $\boldsymbol{\theta}$. Fortunately, it suffices to use observed information matrix for statistical inference because it is a consistent estimate of the expected information matrix.

Under incorrect model specification, the theory of m-estimation can be leveraged to show that $\hat{\boldsymbol{\theta}}$ converges in probability to a parameter vector that solves the score equations. Further, $\hat{\boldsymbol{\theta}}$ is asymptotically normal with covariance matrix that can be consistently estimated via a sandwich estimator. In our simulation study, we find that the observed information matrix provides adequate statistical inference under our assumed data-generating model. However, a robust sandwich estimator may be more appropriate when model mispecification is suspected.

We now turn our attention to generating prediction intervals. Let Z^* denote a new value of the latent response variable with associated covariate vector $\mathbf{x}$. Then by Slutsky's Theorem, we have

$$\frac{Z^* - f(\mathbf{x_i}^\intercal \hat{\boldsymbol{\beta}})}{\sqrt{\exp\left\{2 \cdot g(\mathbf{x_i})^\intercal \hat{\boldsymbol{\alpha}}\right\} + f(\mathbf{x})^\intercal [\mathcal{H}(\boldsymbol{\theta})^{-1}]_{11} f(\mathbf{x_i})/n}} \xrightarrow{d} \text{Normal}(0, 1), \tag{14}$$

where $[\mathcal{H}(\boldsymbol{\theta})^{-1}]_{11}$ is the top-left block of $\mathcal{H}(\boldsymbol{\theta})^{-1}$ corresponding to $\boldsymbol{\beta}$. Thus, we can form a large-sample $(1-\epsilon)\cdot 100\%$ prediction interval for Z^* as follows:

$$\text{PI} = f(\mathbf{x_i}^\intercal \hat{\boldsymbol{\beta}}) \pm \Phi^{-1}(\epsilon/2) \sqrt{\exp\left\{2 \cdot g(\mathbf{x_i})^\intercal \hat{\boldsymbol{\alpha}}\right\} + f(\mathbf{x})^\intercal [\mathcal{H}(\boldsymbol{\theta})^{-1}]_{11} f(\mathbf{x_i})/n}. \tag{15}$$

While this approach is asymptotically valid, it is somewhat unsatisfying because it propagates only *some* of the uncertainty arising from estimating $\boldsymbol{\alpha}$. Specifically, it propagates the uncertainty insofar as it affects the asymptotic variance of $\hat{\boldsymbol{\beta}}$, but it does not address the uncertainty arising from the direct inclusion of $\hat{\boldsymbol{\alpha}}$ in the standard error computation.

To address this challenge, we introduce an approximate Bayesian approach that could be justified via the Bernstein—von Mises theorem or, alternatively, based on its asymptotic equivalence with the approach detailed above. We first generate a large number of samples of $\boldsymbol{\theta}$ from its approximate posterior distribution—Normal$(\hat{\boldsymbol{\theta}}, \mathbf{H}(\hat{\boldsymbol{\theta}}))$. Then for each sample we generate a value of Z^* conditional on the sampled value of $\boldsymbol{\theta}$. Finally, we select quantiles of the sampled Z^* values such that $(1-\epsilon)\cdot 100\%$ of the sampled values fall between the quantiles.

Note that both of these approaches generate prediction intervals for Z^*, not a new value of the count—call it Y^*. Since there is a deterministic mapping from Z^* to Y^*, this is not particularly problematic. For the Bayesian approach, we must accept that our nominal coverage rate will not be exactly $(1-\epsilon)\cdot 100\%$. For the frequentist intervals, we can either (1) create conservative intervals with at least $(1-\epsilon)\cdot 100\%$ asymptotic coverage or (2) randomize the procedure such that the coverage is exactly $(1-\epsilon)\cdot 100\%$ in repeated samples. We illustrate the latter approach for the upper limit only (the lower limit is similar). Suppose that the upper limit for Z^* is 3.6, corresponding to a count of 36. We would then compute the additional coverage probabilities of extending our interval from log(36) to 3.6 and from log(36) to log(37); call these values a and b, respectively. The upper limit of our interval would then be 35 (inclusive) with probability $b/(a+b)$ and 36 with probability $a/(a+b)$.

4 Simulation Studies

In this section, we use simulation studies to more fully explore (1) parameter estimation, particularly for $E(Y_i) \equiv \mu_i$ and $SD(Y_i) \equiv \sigma_i$, in the face of model mispecification, (2) computation time for large sample sizes, and (3) prediction intervals for unobserved data.

4.1 Method Comparison

The DLN model allows for varying (both over- and under-) dispersion for a single count data set and does so for large sample sizes in a computationally fast manner. The COM-Poisson model is the competing model that can account for varying dispersion, but it is known for being computationally burdensome. Thus, we chose to simulate data under the COM-Poisson model, (6), and estimate $\{(\mu_i, \sigma_i); i = 1, \ldots, n\}$ for the various competing models.

We expect that the COM-Poisson model will be the most accurate model for estimation as it is the data-generating model, but that it will also be the slowest-fitting model. We expect the DLN model to be the strongest competitor – as it accounts for varying dispersion – but whose likelihood will be different than the data-generating model; thus, we do not expect as accurate estimates as the COM-Poisson model. We fit the DLN model using both the Newton-Raphson and EM algorithms. We also fit two other models for count data, the GP-1 model and "quasi"-Poisson model (EPL). Both of these models account for *constant* over- and under-dispersion for a single data set. These two methods should be fast, but will not account for the *varying* dispersion.

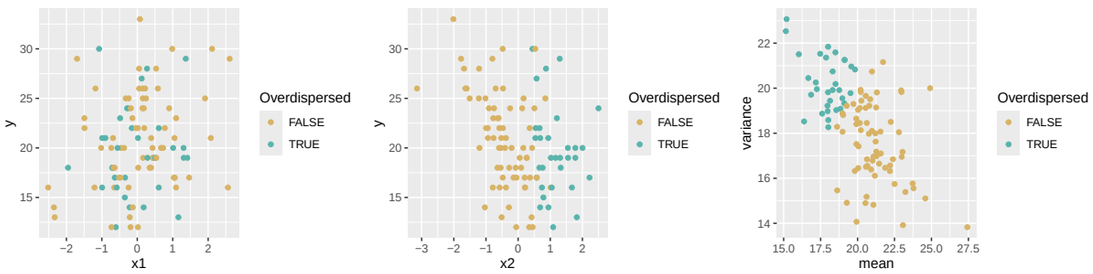


Figure 1: Simulated data set for $n = 100$. Left: values of y_i versus x_{i1}; Center: values of y_i versus x_{i2}; data-generating variance, σ_i^2, versus mean, μ_i. The color of the points indicates whether the dispersion is greater than 1 (green) or not (tan).

To simulate our data, we draw values for two covariates, x_{i1} and x_{i2}, independently from a $N(0, 1)$ distribution for $i = 1, \ldots, n$. We leave these values fixed for all simulations of the same sample size, n. We include both main effects and an interaction in the linear terms of (6). Specifically,

$$f(\mathbf{x}_i)^T\boldsymbol{\beta} = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + x_{i1}x_{i2}\beta_3,$$
$$g(\mathbf{x}_i)^T\boldsymbol{\alpha} = \alpha_0 + x_{i1}\alpha_1 + x_{i2}\alpha_2 + x_{i1}x_{i2}\alpha_3,$$

and fix $\boldsymbol{\beta} = (3, 0.05, -0.1, 0.02)^T$ and $\boldsymbol{\alpha} = (0.1, 0, -0.2, 0.05)^T$. These parameter values create data with an average mean of approximately 20 and variance of approximately 18 (slightly under-dispersed on average, but close to equi-dispersed). Note, however, that the variance-to-mean dispersion ranges from approximately 0.4 to 1.8. We draw $\{y_i; i = 1, \ldots, n\}$ from Equation (5). For each sample size we generate a new response variable 1000 times and estimate $\hat{\mu}_i$ and $\hat{\sigma}_i$ for each model. We do this for $n \in \{50, 100, 250, 500\}$.

Figure 1 shows an example of the simulated data for $n = 100$. The color of the points shows whether the true variance is greater than the mean (green points) or the variance is less than the mean (tan points). The left and center plots show the relationship of y_i to the covariates, x_{i1} and x_{i2} respectively. What's noteworthy is that the data-generating relationships are visible in these plots. For example, $\beta_1 = 0.05$, indicating that there should be a small positive relationship between y_i and x_{i1}, as seen in the plot on the left. Additionally, $\alpha_1 = 0$ and thus, as expected, it is difficult to see a pattern in the dispersion relative to x_{i1}; however, $\alpha_2 = -0.2$ and the center plot shows a clear relationship between the dispersion and x_{i2}. The right plot shows the variance versus the mean, illustrating the varying dispersion for different observations.

Figures 2 and 3 show how the 95% confidence interval coverage changes for each model as the sample size increases for μ_i and σ_i, respectively. Each boxplot shows the distribution for the coverage of the n observations. For the mean, all methods perform comparably (see Figure 2). For the standard deviation, there is a different story. When n is relatively small ($n = 50$

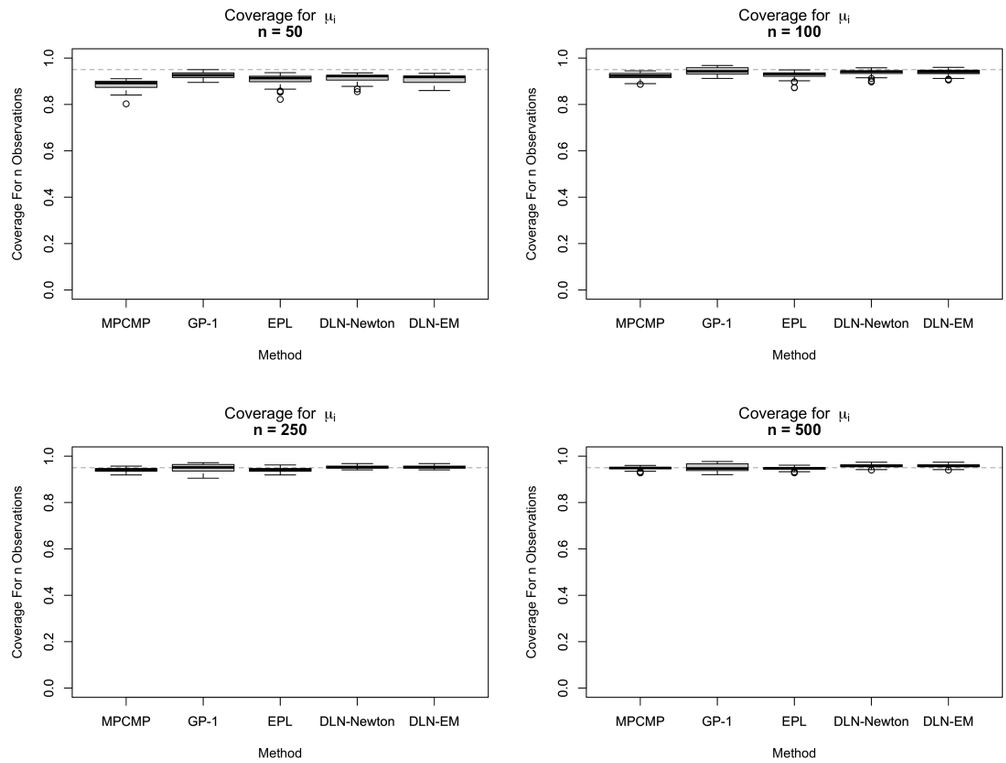


Figure 2: Coverage of 95% confidence intervals for μ_i for the n observations for each of the five model-fitting methods when $n = 50$ (top left), $n = 100$ (top right), $n = 250$ (bottom left), and $n = 500$ (bottom right).

or 100), the DLN model (particularly fit via Newton-Raphson method) has coverage closest to 95% for all observations, even outperforming the "true" data-generating model, MPCMP. However, for large values of n ($n = 500$), the DLN model does not perform as well as the true model, MPCMP, nor the EPL model, which assumes constant variance across observations. This makes sense since as n gets larger, the approximation of the DLN likelihood will be less similar to the true data-generating generalized Poisson likelihood, which the MPCMP and EPL are better able to capture. What is surprising is how the coverage for the GP-1 model gets much worse as n increases, even though it is a generalized Poisson model. This drives home the point that an incorrect mean-variance specification can result in poor inferences no matter the sample size.

Computationally we expected the GP-1 and EPL models to outperform the other methods and they do, with each fitting the models when $n = 500$ in an average of 0.0063 and 0.0026 minutes (0.38 and 0.15 seconds), respectively. Also as expected, the MPCMP model is the slowest, taking 2.1132 minutes on average to fit the same data. The DLN-Newton and DLN-EM methods are a happy medium and comparable to each other in computation time, taking 0.04 minutes ($\approx$2.5 seconds) to fit these methods. We note that this takes into account fitting the DLN-Newton method twice: once to identify better starting values for $\boldsymbol{\alpha}$ and once to get the final parameter estimates. The data were generated and models fit consecutively on an AMD EPYC 7502 processor running at 2.50GHz.

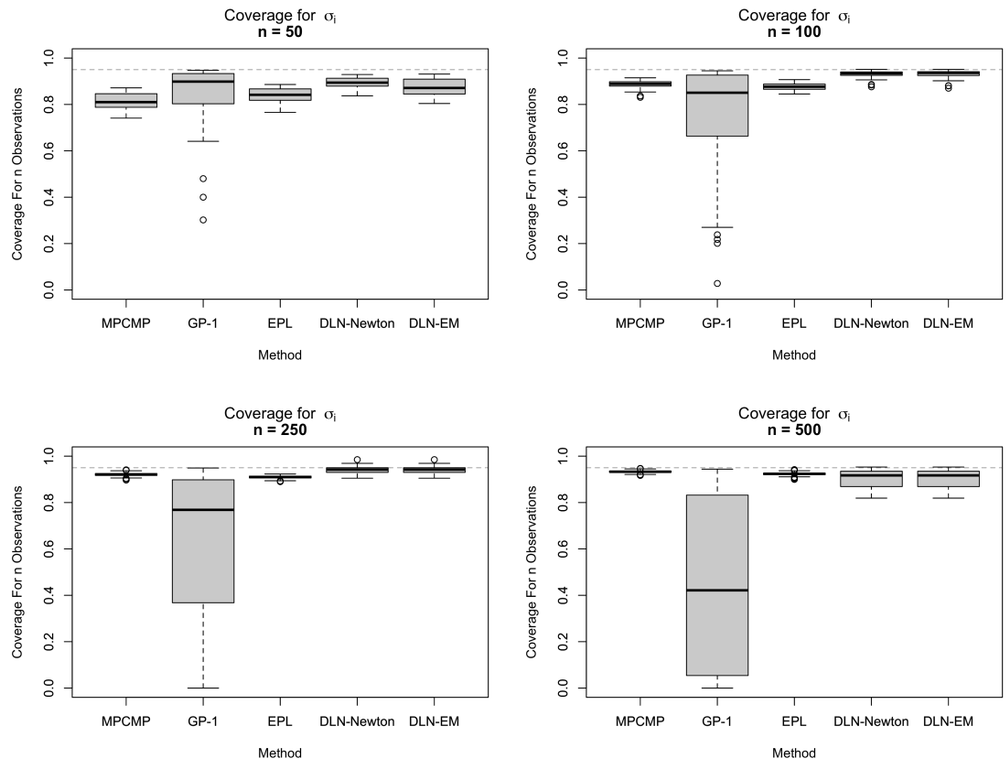


Figure 3: Coverage of 95% confidence intervals for σ_i for the n observations for each of the five model-fitting methods when $n = 50$ (top left), $n = 100$ (top right), $n = 250$ (bottom left), and $n = 500$ (bottom right).

4.2 Prediction Intervals for the DLN Model

In this section, we test the two prediction interval methods discussed in Section 3.2: (1) the plug-in method and (2) the asymptotic Bayes approach in which we draw samples of the estimated parameters. For comparison, we also included a correctly specified fully Bayesian approach, which we estimated using the sampling importance resampling algorithm (Rubin, 1987, 1988; Smith and Gelfand, 1992). Within the simulation, we sampled α and β as follows:

$$\alpha \sim \mathrm{N}\left(\begin{bmatrix}4\\0\end{bmatrix}, \begin{bmatrix}0.2^2 & 0\\0 & 0.05^2\end{bmatrix}\right), \quad \beta \sim \mathrm{N}\left(\begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}0.2^2 & 0\\0 & 0.05^2\end{bmatrix}\right).$$

We then sampled the outcomes from the DLN model conditional on these values of α and β. The fully Bayesian approach uses the above (independent) distributions for its prior. We tested four sample sizes: 10, 30, 100, and 400. With each sample size, we ran 400 repetitions and computed the marginal coverage, coverage rMSE (as detailed in Section 4.1), and median interval length. Table 1 displays the results.

All three methods produce approximately calibrated prediction intervals in large ($n \geq 100$) sample sizes. Because the observed values for the coverage rMSE are near their theoretical value ($\sqrt{0.95 * 0.05/400} \approx 0.11$), the methods are both marginally *and* conditionall calibrated. In small sample sizes, we observe that only the fully Bayesian approach achieves

Summary	Method	n=10	n=30	n=100	n=400
Marginal Coverage	Plug-in	0.908 (0.006)	0.920 (0.004)	0.939 (0.002)	0.942 (0.002)
	Asymp. Bayes	0.932 (0.005)	0.926 (0.004)	0.941 (0.002)	0.941 (0.002)
	Full Bayes	0.944 (0.004)	0.947 (0.002)	0.950 (0.002)	0.946 (0.001)
Coverage rMSE	Plug-in	0.044 (0.006)	0.033 (0.004)	0.017 (0.002)	0.014 (0.001)
	Asymp. Bayes	0.022 (0.004)	0.027 (0.003)	0.015 (0.002)	0.015 (0.001)
	Full Bayes	0.011 (0.003)	0.010 (0.002)	0.011 (0.001)	0.012 (0.001)
Median Length	Plug-in	96.5 (1.0)	88.8 (1.3)	87.0 (0.6)	86.0 (0.4)
	Asymp. Bayes	111.8 (2.8)	91.0 (1.3)	88.0 (0.9)	86.0 (0.4)
	Full Bayes	91.5 (1.2)	89.0 (1.4)	84.8 (1.7)	85.8 (1.4)

Table 1: Coverage and median interval lengths for three prediction interval methods. The Plug-in and Asymp. Bayes methods correspond with those explained in Section 3.2. The prior for the fully Bayesian method matches the data generating distribution.

near-nominal coverage. In practice, however, the empirical coverage of this method will depend on prior specification. The asymptotic Bayes approach offers a compromise between the fully Bayesian approach and the plug-in method in that it produces coverage rates within two percentage points of the nominal level without requiring 'correct' prior specification. Because it does not leverage prior information, the increased coverage necessarily comes at the expense of slightly wider interval widths.

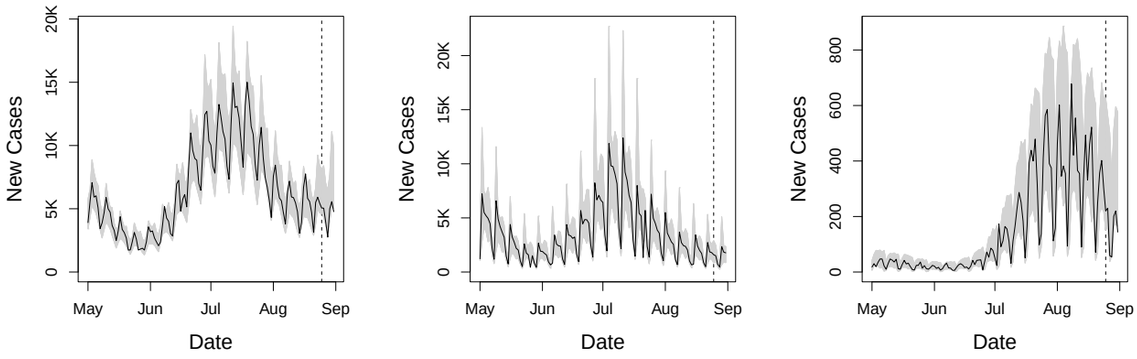
5 Case Study

Our simulation results indicate that the COM-Poisson, EPL, and DLN models can achieve comparable statistical performance; however, the COM-Poisson model requires much longer computational time. To further assess the computational scalability of the EPL and DLN methods, we applied them to a large-scale forecasting problem: week-ahead COVID-19 case count prediction for the European area. We did not consider the GP-P model because our simulation study indicated that the GP-P model performs poorly in the presence of varying dispersion. Relatedly, we were unable to fit the COM-Poisson model due to the size of the data set.

We used openly available case count data from Google for the comparison (Wahltinez et al., 2020). We filtered down to the time period July 1, 2020 – August 31, 2022 and removed countries with data abnormalities; e.g., gaps in the data. The resulting data set includes 27 countries.

The functions f and g include separate intercepts, day-of-week effects, and cubic natural splines over time. We selected the degrees of freedom for the natural splines via grid-search cross validation, which resulted in 50 degrees of freedom for f and 12 for g. This specification allows both the mean and dispersion to flexibly fit nonlinear patterns in the data; however, it allows the mean function to change over shorter time scales. The resulting design matrices

have 1,539 and 513 columns, respectively.



(a) Austria (b) Belgium (c) Bosnia and Herzegovina

Figure 4: New cases of COVID-19 (black line) in the first 3 of 27 European countries and 95% pointwise prediction bands (gray shaded area) from the discrete log-normal (DLN) model. The model is fit to case-count data from July 1, 2020 to August 24, 2022. The last week of plotted counts (separated by the dashed gray line) offer an out-of-sample performance comparison.

We fit the models using case-count data through August 24, 2022 and used the final week (August 25 – August 31 2022) as a test set to evaluate model performance. Under this configuration, the training and test sets include 21,194 and 189 case counts, respectively. Figure 4 plots the observed case counts against 95% prediction bands from the DLN method for the first three countries (alphabetically), with the final week representing an out-of-sample comparison. The figure indicates that the DLN method is able to accurately model the changes in case counts over time, including varying dispersion. Plots for the remaining 24 countries are available in Appendix D. The coverage rate of the DLN prediction intervals was 88.9% with a Monte Carlo standard error of 2.7%, indicating some undercoverage, likely due to linear extrapolation error in the natural splines.

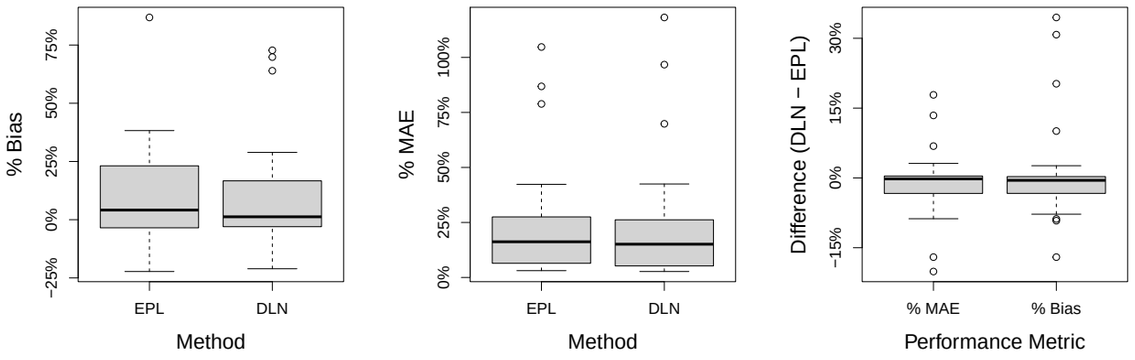


Figure 5: Performance comparison of the extended pseudo-likelihood (EPL) and discrete log-normal (DLN) models in predicting COVID case counts for 27 European countries. Each data point represents a single country. The final panel shows paired differences. Averages and standard errors are displayed in Table 2.

Figure 5 plots (point) predictive performance for the EPL and DLN models. Each data point represents one of the 27 countries. The figure shows bias and mean absolute error (MAE) as a percentage of historical (over the time period of the training data) average case counts at the country level; this normalization allows an appropriate comparison across countries of

different sizes and infection intensities. The figure indicates that the EPL and DLN models achieved similar predictive performance with neither clearly dominating the other.

Table 2 displays the average values of the performance metrics and their associated standard errors, including pairwise comparisons. The results indicate that the DLN method had slightly higher bias and slightly lower MAE on average compared to the EPL method. The table also displays the compute time for the two methods. The DLN method required slightly less computation time at 5.7 minutes compared to 8.7 minutes for the EPL method on a personal computer with 16 GB of memory and 8 CPUs. We fit the DLN method using the EM algorithm because we found the EM algorithm to be more stable and insensitive to starting values than the Newton–Raphson algorithm.

	EPL	DLN	Difference
Bias %	9.6% (0.8%)	10.6% (0.9%)	1.0% (0.4%)
MAE %	24.7% (1.0%)	23.7% (1.1%)	-1.0% (0.3%)
Elapsed Time (Minutes)	8.72	5.72	N/A

Table 2: Performance comparison of the extended pseudo-likelihood (EPL) and discrete log-normal (DLN) models in predicting COVID case counts for 27 European countries. The DLN method exhibits slightly more bias, lower MAE, and a shorter model-fitting time. Figure 5 plots the country-level data.

These results agree with those of the simulation study, showing that the EPL and DLN methods achieve similar predictive performance on count data with varying dispersion, and they are both scalable to large data sets.

6 Discussion

This paper presents several statistical methods for modeling count data with varying levels of dispersion. We focus specifically on methods that are scalable to large-scale data sets, such as the COVID-19 data set analyzed in our case study. Although COM-Poisson models have strong theoretical backing, these models do not scale well due to the presence of a computationally demanding normalizing constant, which can limit their applicability in practice. The GP-P model, on the other hand, is computationally scalable, but it poses some theoretical and computational issues in the presence of underdispersion because its PMF does not sum exactly to one in that case. For data sets known to be overdispersed (at all $\mathbf{x}$), however, the GP-P is a viable choice and a full-fledged competitor to negative binomial models.

The results from our simulation study and case study indicate that moment-based methods (e.g., the EPL method) are a suitable alternative when objective of the analysis is to estimate only the first two moments of the data distribution. If only the first moment is of interest,

it would suffice to estimate the regression parameters via Poisson regression provided robust standard errors are employed for inference. The benefits of the EPL method (and its quasi-likelihood cousin) are that (a) it can achieve better efficiency by modeling the variance and (b) it provides a richer set of inferences in that it estimates both the mean and variance of the counts at all $\mathbf{x}$.

In some cases, however, estimates of only the first two moments may not be sufficient. Our case study offers one such example. For COVID-19 forecasting, it would be preferable to produce a predictive distribution to fully convey the forecast uncertainty and appropriately calibrate downstream decision-making, such as inventory planning and staffing. Another example is anomaly detection with application monitoring data (Veasey and Dodson, 2014); i.e., monitoring counts of website events over time to detect outages or sudden changes in user behavior.

When a full predictive distribution is required, the discrete log-normal (DLN) introduced in Section 3 is a viable alternative. Our results indicate that the DLN model is computationally scalable, achieves comparable statistical performance compared to moment-based methods, and enables researchers to create calibrated prediction intervals. Compared to transformation-based methods, the DLN method offers the additional benefit of respecting the natural domain of count data. This benefit is especially important when the counts are small or interest lies in low-probability quantiles close to zero.

We see several possible extensions of the DLN model that would further increase its utility. One such extension would be to allow correlation across proximal time points, which would be accommodated via the latent Gaussian formulation. Another useful extension would be to generalize the form of the transformation from the latent Gaussian variates to the counts. This extension would enable application of the method to non-Poissonian count data exhibiting, for example, zero inflation or high tail probability.

References

Agresti, A. (2013), *Categorical Data Analysis*, Jon Wiley & Sons, Hoboken, New Jersey, Third edn.

Carroll, R. J., Ruppert, D., Stefanski, L. A., and Crainiceanu, C. M. (2006), *Measurement error in nonlinear models: a modern perspective*, Chapman and Hall/CRC.

Chatla, S. B. and Shmueli, G. (2018), "Efficient estimation of COM–Poisson regression and a generalized additive model," *Computational Statistics & Data Analysis*, 121, 71–88.

Consul, P. and Famoye, F. (1992), "Generalized Poisson regression model," *Communications in Statistics – Theory and Methods*, 21, 89–109.

Consul, P. and Shoukri, M. (1985), "The generalized Poisson distribution when the sample mean is larger than the sample variance," *Communications in Statistics – Simulation and Computation*, 14, 667–681.

Consul, P. C. (1989), *Generalized Poisson Distributions: Properties and Applications*, M. Dekker, New York.

Consul, P. C. and Famoye, F. (2006), *Lagrangian Probability Distributions*, Springer, New York, New York.

Famoye, F. (1993), "Restricted generalized Poisson regression model," *Communications in Statistics – Theory and Methods*, 22, 1335–1354.

Famoye, F., Wulu, J. T., and Singh, K. P. (2004), "On the generalized Poisson regression model with an application to accident data," *Journal of Data Science*, 2, 287–295.

Foygel Barber, R., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2021), "The limits of distribution-free conditional predictive inference," *Information and Inference: A Journal of the IMA*, 10, 455–482.

Frome, E. L., Kutner, M. H., and Beauchamp, J. J. (1973), "Regression analysis of Poisson-distributed data," *Journal of the American Statistical Association*, 68, 935–940.

Gaunt, R. E., Iyengar, S., Olde Daalhuis, A. B., 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, 163–180.

Hilbe, J. M. (2011), *Negative Binomial Regression*, Cambridge University Press, Cambridge, Second edn.

Kim, T., Lieberman, B., Luta, G., and Peña, E. A. (2022), "Prediction intervals for Poisson-based regression models," *Wiley Interdisciplinary Reviews: Computational Statistics*, 14, e1568.

Kokonendji, C. C. (2014), "Over-and underdispersion models," *Methods and applications of statistics in clinical Trials: Planning, analysis, and inferential methods*, 2, 506–526.

Kotz, S., Johnson, N. L., and Balakrishnan, N. (1994), "Continuous Univariate Distributions, Vol. 1 of Wiley Series in Probability and Statistics," .

Lawless, J. F. (1987), "Negative binomial and mixed Poisson regression," *The Canadian Journal of Statistics*, 15, 209–225.

Lee, Y. and Nelder, J. (2000), "The relationship between double-exponential families and extended quasi-likelihood families, with application to modelling Geissler's human sex ratio data," *Journal of the Royal Statistical Society, Series C*, 49, 413–419.

Lehmann, E. L. and Casella, G. (2006), *Theory of Point Estimation*, Springer Science & Business Media, New York, New York.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018), "Distribution-free predictive inference for regression," *Journal of the American Statistical Association*, 113, 1094–1111.

Lord, D., Geedipally, S. R., and Guikema, S. D. (2010), "Extension of the application of Conway-Maxwell-Poisson models: Analyzing traffic crash data exhibiting underdispersion," *Risk Analysis: An International Journal*, 30, 1268–1276.

McCullagh, P. and Nelder, J. (1989), *Generalized Linear Models, Second Edition*, Chapman & Hall, Boca Raton, Florida.

Nelder, J. A. and Lee, Y. (1991), "Generalized linear models for the analysis of Taguchi-type experiments," *Applied stochastic models and data analysis*, 7, 107–120.

Pregibon, D. (1984), "Review of 'Generalized Linear Models' by P. McCullagh and J. Nelder," *The Annals of Statistics*, 12, 1589–1596.

Rubin, D. B. (1987), "Comment on 'The calculation of posterior distributions by data augumentation' by MA Tanner and WH Wong," *Journal of the American Statistical Association*, 82, 543–546.

Rubin, D. B. (1988), "Using the SIR algorithm to simulate posterior distributions," in *Bayesian statistics 3. Proceedings of the third Valencia international meeting, 1-5 June 1987*, pp. 395–402, Clarendon Press.

Sellers, K. F. (2023), *The Conway–Maxwell–Poisson distribution*, vol. 8, Cambridge University Press.

Sellers, K. F. and Morris, D. S. (2017), "Underdispersion models: Models that are 'under the radar'," *Communications in Statistics – Theory and Methods*, 46, 12075–12086.

Sellers, K. F. and Shmueli, G. (2010), "A flexible regression model for count data," *The Annals of Applied Statistics*, pp. 943–961.

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S., and Boatwright, P. (2005), "A useful distribution for fitting discrete data: revival of the Conway–Maxwell–Poisson distribution," *Journal of the Royal Statistical Society, Series C*, 54, 127–142.

Smith, A. F. and Gelfand, A. E. (1992), "Bayesian statistics without tears: a sampling–resampling perspective," *The American Statistician*, 46, 84–88.

Veasey, T. J. and Dodson, S. J. (2014), "Anomaly detection in application performance monitoring data," *International Journal of Machine Learning and Computing*, 4, 120.

Wahltinez, O. et al. (2020), "COVID-19 Open-Data: curating a fine-grained, global-scale data repository for SARS-CoV-2," Work in progress.

Wang, W. and Famoye, F. (1997), "Modeling household fertility decisions with generalized Poisson regression," *Journal of Population Economics*, 10, 273–283.

Yang, Z., Hardin, J. W., and Addy, C. L. (2009), "A score test for overdispersion in Poisson regression based on the generalized Poisson-2 model," *Journal of Statistical Planning and Inference*, 139, 1514–1521.

Zamani, H. and Ismail, N. (2012), "Functional form for the generalized Poisson regression model," *Communications in Statistics – Theory and Methods*, 41, 3666–3675.

A Details for Fitting the GP-P Model

In this appendix, we provide the details for the GP-P model, including the procedures for estimation and inference. We first derive the expected information matrix (EIM). Then we show how to leverage it to maximize the GP-P likelihood via iteratively reweighted least squares. Finally, we describe the likelihood-based inference procedures we employed in Section 4.

A.1 Fisher information for the GP-P

In this section, we derive the Fisher information for the GP-P using the reparameterization formula. Section A.1.1 gives the derivation, and Section A.1.2 displays the results of some numerical checks.

A.1.1 Derivation

We will start from the GP-2 because its Fisher information matrix is diagonal (Famoye (1993), equations (3.9)–(3.11)), which simplifies computations.

The reparameterization formula can be described as follows. Suppose we know the Fisher information of a probability distribution with parameter vector $\boldsymbol{\xi}$. Further, suppose we would like to know the Fisher information matrix for the distribution parameterized with a different vector $\boldsymbol{\tau}$, where $\boldsymbol{\xi}$ is a continuously differentiable function of $\boldsymbol{\tau}$. Then, under some technical assumptions, the Fisher information with respect to $\boldsymbol{\tau}$ can be calculated as follows (See Lehmann and Casella (2006) equation (2.6.16)):

$$\boldsymbol{\mathcal{J}}_{\boldsymbol{\tau}}(\boldsymbol{\tau}) = \mathbf{J}' \, \boldsymbol{\mathcal{J}}_{\boldsymbol{\xi}}\Big(\boldsymbol{\xi}(\boldsymbol{\tau})\Big) \, \mathbf{J}, \tag{16}$$

where $\mathbf{J}$ is the Jacobian of the transformation whose ij-th element is equal to $\frac{d\xi_i}{d\tau_j}$. The technical assumptions for this result are not met in the underdispersed case because the support of the distribution depends on the parameter space. We address this separately in the next section, showing that the formula produces a useful approximation provided (1) the mean is sufficiently large and/or (2) the degree of underdispersion is minimal.

We relate the GP-P and GP-2 as follows. Let $\boldsymbol{\xi} = (\mu, \alpha)'$, be the parameters of the GP-2 distribution, where μ is the mean and α is the dispersion parameter, and $\boldsymbol{\tau} = (\mu, \phi)'$ the mean and dispersion parameters of the GP-P distribution. Note that the mean from the two distributions are the same, but the dispersion parameters are different to accommodate the

different values of P in the GP-P. The dispersion parameters can be related by setting the variance functions to be equal:

$$\begin{aligned}(1+\phi\,\mu^{P-1})^2\,\mu =&(1+\alpha\,\mu)^2\,\mu\\ \phi\,\mu^{P-1} =&\alpha\,\mu\\ \phi\,\mu^{P-2} =&\alpha.\end{aligned} \tag{17}$$

The Jacobian is then

$$\mathbf{J} = \begin{bmatrix} 1 & 0 \\ (P-2)\,\phi\,\mu^{P-3} & \mu^{P-2} \end{bmatrix} \tag{18}$$

So the Fisher information can be calculated as follows

$$\begin{aligned}
&\boldsymbol{\mathcal{J}}_{\boldsymbol{\tau}}(\boldsymbol{\tau})\\
&= \boldsymbol{\mathcal{J}}_{\boldsymbol{\tau}}(\boldsymbol{\tau}) = \mathbf{J}'\;\boldsymbol{\mathcal{J}}_{\boldsymbol{\xi}}\Big(\boldsymbol{\xi}(\boldsymbol{\tau})\Big)\;\mathbf{J}\\
&= \frac{1}{(1+\alpha\,\mu)^2}\begin{bmatrix}1 & (P-2)\,\phi\,\mu^{P-3}\\ 0 & \mu^{P-2}\end{bmatrix}\begin{bmatrix}1/\mu & 0\\ 0 & \frac{2\mu^2}{1+2\alpha}\end{bmatrix}\begin{bmatrix}1 & 0\\ (P-2)\,\phi\,\mu^{P-3} & \mu^{P-2}\end{bmatrix}\\
&= \frac{1}{(1+\phi\mu^{P-1})^2}\begin{bmatrix}1 & (P-2)\,\phi\,\mu^{P-3}\\ 0 & \mu^{P-2}\end{bmatrix}\begin{bmatrix}1/\mu & 0\\ 0 & \frac{2\mu^2}{1+2\phi\mu^{P-2}}\end{bmatrix}\begin{bmatrix}1 & 0\\ (P-2)\,\phi\,\mu^{P-3} & \mu^{P-2}\end{bmatrix}\\
&= \frac{1}{(1+\phi\mu^{P-1})^2}\begin{bmatrix}1/\mu & \frac{2(P-2)\phi\mu^{P-1}}{1+2\phi\mu^{P-2}}\\ 0 & \frac{2\mu^P}{1+2\phi\mu^{P-2}}\end{bmatrix}\begin{bmatrix}1 & 0\\ (P-2)\,\phi\,\mu^{P-3} & \mu^{P-2}\end{bmatrix}\\
&= \frac{1}{(1+\phi\mu^{P-1})^2}\begin{bmatrix}1/\mu+\frac{2(P-2)^2\phi^2\mu^{2P-4}}{1+2\phi\mu^{P-2}} & \frac{2(P-2)\phi\mu^{2P-3}}{1+2\phi\mu^{P-2}}\\ \frac{2(P-2)\phi\mu^{2P-3}}{1+2\phi\mu^{P-2}} & \frac{2\mu^{2(P-1)}}{1+2\phi\mu^{P-2}}\end{bmatrix}\\
&= \frac{1}{(1+\phi\mu^{P-1})^2}\begin{bmatrix}1/\mu+\frac{2\left[(P-2)\phi\mu^{P-2}\right]^2}{1+2\phi\mu^{P-2}} & \frac{2(P-2)\phi\mu^{2P-3}}{1+2\phi\mu^{P-2}}\\ \frac{2(P-2)\phi\mu^{2P-3}}{1+2\phi\mu^{P-2}} & \frac{2\mu^{2(P-1)}}{1+2\phi\mu^{P-2}}\end{bmatrix}.
\end{aligned} \tag{19}$$

A.1.2 Numerical Checks on the EIM

As discussed above, the GPD does not satisfy the standard regularity conditions for maximum likelihood estimation when $\phi < 0$. The purpose of this section is to compare our analytical expression for the EIM to its true value (determined via numerical integration) to determine whether and when our derived expression is a useful approximation. To calculate the EIM numerically, we constructed a fine grid of parameter values for μ and ϕ and a long sequence of values (ranging from 0 to 1,000) for the response variable Y. Then for each possible outcome, y, we calculated the derivatives given in equations (21) and (22) and the probability $Pr(Y = y|, \mu, \phi, P)$. We then used these values to numerically approximate the

covariance of the score function: the EIM.

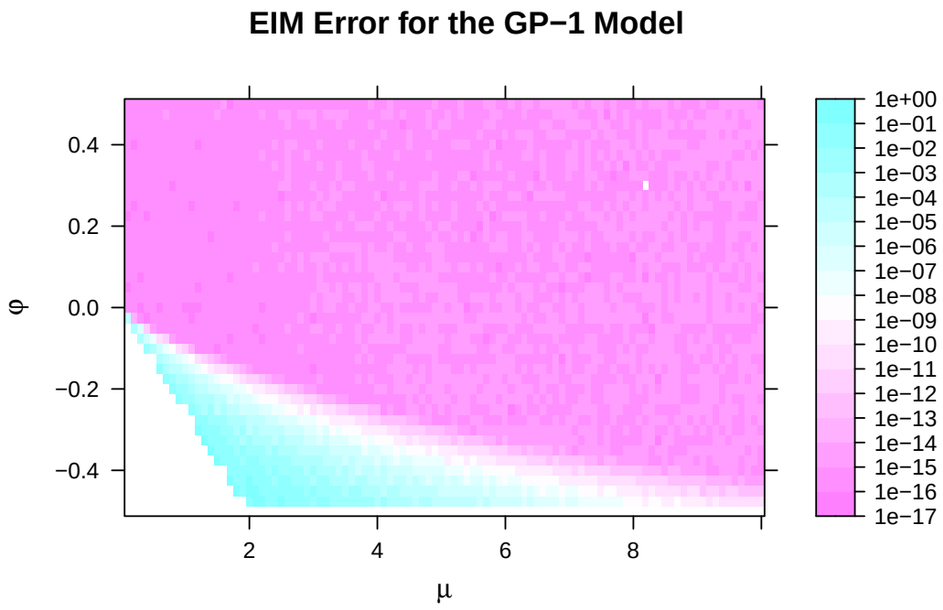


Figure 6: *L^1 error in the expected information matrix (EIM) as a proportion of the L^1 norm of the true EIM, which was calculated by numerical integration. The plot shows results for the GP-1; similar results were obtained for other values of P. The derivation is exact when $\phi \geq 0$. However, it's only approximate when $\phi < 0$ (which corresponds to underdispersion) and the error becomes quite large close to the boundary of the parameter space.*

Figure 6 shows the results of our numerical calculations for the GP-1. The color displayed in the figure represents the L^1 error in the expected information matrix (EIM) as a proportion of the L^1 norm of the true EIM; i.e., denoting the analytical and numerical values of the EIM as $\mathcal{J}_a$ and $\mathcal{J}_n$, respectively, the plot shows $\frac{||\mathcal{J}_a - \mathcal{J}_n||_1}{||\mathcal{J}_n||_1}$. We note that this approach does not account for error due to the GP-P's pmf not summing to unity. Rather, it measures potential discrepancies between our analytical expression for the EIM and a numerical approximation of it, treating the GP-P as a valid probability distribution.

Figure 6 shows essentially no error when $\phi \geq 0$ or the parameters are far from their boundary. As the parameters approach the boundary, however, the error is non-negligible. Although Figure 6 shows the error for the GP-1 only, similar results hold for other values of P.

These results suggest that researchers should exercise caution in applying the GP-P to underdispersed data. Doing so may be reasonable when the degree of underdispersion is low or the mean is relatively high, but researchers should be wary of situations where the parameters are close to their boundary.

A.2 Estimation via Iteratively Reweighted Least Squares

The derivations for the Fisher scoring algorithm for a GLM follow the derivation for general GLM's and we refer the reader to any GLM book, such as McCullagh and Nelder (1989).

From Equation (4), the log likelihood of a single observation, y, can be found to be

$$\ell(\mu, \phi, P) = \log(\mu) + (y-1)\log(\mu + \phi\mu^{P-1}y) - \frac{\mu + \phi\mu^{P-1}y}{1 + \phi\mu^{P-1}} - y\log(1 + \phi\mu^{P-1}) - \log(y!). \quad (20)$$

The Fisher scoring model-fitting algorithm requires the derivative of the log likelihood with respect to both μ and ϕ; thus, we provide those expressions here. The derivative with respect to μ is given as follows:

$$\frac{d\ell}{d\mu} = \mu^{-1} + \frac{(y-1)(1+\zeta(P-1)y)}{\mu(1+\zeta\mu)} - \frac{1+2(P-1)\zeta y}{1+\zeta y} + \frac{(P-1)\zeta\mu(1+\zeta\mu)}{(1+\zeta\mu)^2}, \quad (21)$$

where, to clean the notation, we use ζ here to represent $\phi\mu^{P-2}$. The derivative with respect to ϕ is

$$\frac{d\ell}{d\phi} = \frac{\mu^{P-2}\, y\,(y-1)}{1+\zeta\, y} - \frac{\mu^{P-1}(y-\mu)}{(1+\zeta\mu)^2} - \frac{\mu^{P-1}y}{1+\zeta\mu}. \quad (22)$$

We now build on these derivations for one observation to multiple observation in a regression framework. The GP-P model can be represented using the generalized linear model (GLM) framework of stochastic, link, and linear terms. Specifically, for a response variable, Y_i, model

$$\begin{aligned} Y_i &\overset{ind}{\sim} \text{GP-P}(\mu_i, \phi, P) \\ \log(\mu_i) &= \eta_i \\ \eta_i &= \mathbf{x}_i'\boldsymbol{\beta} + o_i \end{aligned} \quad (23)$$

where $\mathbf{x}_i$ is a k-dimensional vector of covariates for observation i, $\boldsymbol{\beta}$ is the corresponding vector of coefficients, and o_i is a fixed and known "offset." P is treated as a known value and could be either fully specified *a priori* or optimized by fitting the model with multiple values and comparing the resulting likelihoods. Although the only theoretical requirement is that $P \in \mathbb{R}$, it will typically be set within several integers of the standard values of 1 and 2. ϕ is an unknown constant that plays the role of dispersion parameter. As described in the main text, its minimum value $\phi_{\min}(\mu, P)$ is a complicated function of the other parameters.

We now describe the Fisher scoring updates We first derive them for a fixed value of ϕ and then discuss how to jointly estimate both $\boldsymbol{\beta}$ and ϕ.

A.2.1 Fixed ϕ

In both this section and the next, the Fisher scoring algorithm is equivalent to maximum likelihood estimation. Its development closely follows that of a standard generalized linear

model (GLM; see, for example, McCullagh and Nelder, 1989).

The Fisher information is the covariance of the score equations for the unknown parameters, in this case $\boldsymbol{\beta}$, where the score equation for the j-th coefficient is

$$\mathcal{U}_{\beta_j} = \sum_{i=1}^{n} \frac{d\ell_i}{d\beta_j} = \sum_{i=1}^{n} \frac{d\ell_i}{d\mu_i}\frac{d\mu_i}{d\eta_i}\frac{d\eta_i}{d\beta_j} = \sum_{i=1}^{n} \frac{d\ell_i}{d\mu_i}\mu_i x_{ij} \tag{24}$$

Applying equation (21) and the EIM derived in Appendix A.1, the Fisher information for predictors j and q is given by

$$\begin{aligned}
\mathcal{J}_{\beta_j\beta_q} &= \mathrm{Cov}(\mathcal{U}_{\beta_j},\, \mathcal{U}_{\beta_q}) \\
&= \mathrm{E}(\mathcal{U}_{\beta_j}\, \mathcal{U}_{\beta_q}) \\
&= \mathrm{E}\left\{ \Big[\sum_{i=1}^{n} \frac{d\ell_i}{d\mu_i}\, \mu_i\, x_{ij}\Big] \Big[\sum_{r=1}^{n} \frac{d\ell_r}{d\mu_r}\, \mu_r\, x_{rq}\Big] \right\} \\
&= \mathrm{E}\left\{ \sum_{i=1}^{n} \Big(\frac{d\ell_i}{d\mu_i}\Big)^2 \mu_i^2\, x_{ij}\, x_{iq} \right\} \\
&= \sum_{i=1}^{n} \mu_i^2\, x_{ij}\, x_{iq}\, \mathrm{E}\Big(\frac{d\ell_i}{d\mu_i}\Big)^2 \\
&= \sum_{i=1}^{n} \mu_i^2\, x_{ij}\, x_{iq} \left[\frac{1}{(1+\phi\mu_i^{P-1})^2} \left\{ \mu_i^{-1} + \frac{2\big[(P-2)\phi\mu_i^{P-2}\big]^2}{1+2\phi\mu_i^{P-2}} \right\} \right] \\
&= \sum_{i=1}^{n} x_{ij}\, x_{iq} \left[\frac{1}{(1+\phi\mu_i^{P-1})^2} \left\{ \mu_i + \frac{2\big[(P-2)\phi\mu_i^{P-1}\big]^2}{1+2\phi\mu_i^{P-2}} \right\} \right].
\end{aligned} \tag{25}$$

$E\left(\frac{d\ell_i}{d\mu_i}\right)^2$ is one of the terms in the EIM from Appendix A.1. We can write the Fisher information in the familiar form, $\boldsymbol{\mathcal{J}}_{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{W}_{\boldsymbol{\beta}}\mathbf{X}$, where $\mathbf{X}$ is the $n \times k$ design matrix and $\mathbf{W}_{\boldsymbol{\beta}}$ is a diagonal weight matrix with the i-th diagonal element equal to

$$w_i = \frac{1}{(1+\phi\mu_i^{P-1})^2} \left\{ \mu_i + \frac{2\big[(P-2)\phi\mu_i^{P-1}\big]^2}{1+2\phi\mu_i^{P-2}} \right\}. \tag{26}$$

Armed with this result, we can fit the GP-P model via Fisher scoring updates as follows:

$$\mathbf{b}^{(t)} = \mathbf{b}^{(t-1)} + \boldsymbol{\mathcal{J}}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mathcal{U}}_{\boldsymbol{\beta}}, \tag{27}$$

where $\mathbf{b}^{(t)}$ represents the estimate of $\boldsymbol{\beta}$ at the t-th iteration of the model-fitting algorithm. Note that these Fisher scoring updates are equivalent to updating $\mathbf{b}^{(t)}$ according to the iteratively reweighted least squares (IWLS; see, for example, McCullagh and Nelder, 1989) algorithm:

$$\mathbf{b}^{(t)} = (\mathbf{X}'\mathbf{W}_{\boldsymbol{\beta}}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}_{\boldsymbol{\beta}}\,\mathbf{z}, \tag{28}$$

where $\mathbf{z}$ is an $n \times 1$ vector of "working responses," defined as follows:

$$z_i = \eta_i + \frac{d\ell_i}{d\mu_i}\frac{\mu_i}{w_i}, \tag{29}$$

where η_i and μ_i are defined in equation (23), $\frac{d\ell_i}{d\mu_i}$ is given in equation (21), and w_i is defined in equation (26).

If ϕ were a known value, we could simply perform updates using (28) until reaching satisfactory convergence. However, because ϕ is typically unknown, we now build on these results in Section A.2.2, showing how to adapt them to jointly estimate ϕ.

A.2.2 Joint Estimation of ϕ

In this section, we adapt the algorithm given above to estimate ϕ via maximum likelihood. To jointly estimate both $\boldsymbol{\beta}$ and ϕ, we derive the joint score and EIM. We begin by placing both $\boldsymbol{\beta}$ and ϕ in one vector, $\boldsymbol{\theta}$, such that, $\boldsymbol{\theta} = (\beta_1, \beta_1, \ldots, \beta_k, \phi)'$. The first k elements of the unified score vector, $\boldsymbol{\mathcal{U}}_{\boldsymbol{\theta}}$, are given by $\boldsymbol{\mathcal{U}}_{\boldsymbol{\beta}}$, where the jth element is given in equation (24). The last element is given by $\mathcal{U}_\phi = \sum_{i=1}^n \frac{d\ell_i}{d\phi}$, the derivative given in equation (22). Similarly, the top left $k \times k$ block of the unified EIM, $\boldsymbol{\mathcal{J}}_{\boldsymbol{\theta}}$, is equal to $\boldsymbol{\mathcal{J}}_{\boldsymbol{\beta}}$, given in the previous section. The bottom-right element comes directly from the EIM derivation in Appendix A.1:

$$\mathcal{J}_{\phi\phi} = \sum_{i=1}^{n} \frac{2\mu_i^{2(P-1)}}{(1+\phi\mu_i^{P-1})^2\,(1+2\phi\mu_i^{P-2})}. \tag{30}$$

Finally, the covariance terms can be derived as follows:

$$
\begin{aligned}
\mathcal{J}_{\beta_j \phi} &= \mathrm{Cov}(\mathcal{U}_{\beta_j},\, \mathcal{U}_{\phi}) \\
&= \mathrm{E}(\mathcal{U}_{\beta_j}\, \mathcal{U}_{\phi}) \\
&= \mathrm{E}\left\{ \left[\sum_{i=1}^{n} \frac{d\ell_i}{d\mu_i}\, \mu_i\, x_{ij} \right] \left[\sum_{r=1}^{n} \frac{d\ell_r}{d\phi} \right] \right\} \\
&= \sum_{i=1}^{n} \mu_i\, x_{ij} \mathrm{E}\left\{ \frac{d\ell_i}{d\mu_i}\, \frac{d\ell_i}{d\phi} \right\} \\
&= \sum_{i=1}^{n} \mu_i\, x_{ij} \frac{(P-2)\phi\mu_i^{2P-3}}{(1+\phi\mu_i^{P-1})^2\,(1+2\phi\mu_i^{P-2})} \\
&= \sum_{i=1}^{n} x_{ij} \frac{(P-2)\phi\mu_i^{2(P-1)}}{(1+\phi\mu_i^{P-1})^2\,(1+2\phi\mu_i^{P-2})}.
\end{aligned}
\tag{31}
$$

Computationally, the values of the covariance are most easily computed by placing the values of the fraction in the final line of Equation (31) in a vector, $\mathbf{a}$, and calculating the matrix multiplication $\mathbf{X}'\mathbf{a}$. The algorithm then proceeds with Fisher scoring updates as in equation (27), but for all $k+1$ parameters jointly.

A.3 Inference Procedures

Standard maximum likelihood theory tells us that

$$
\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \xrightarrow{d} \mathrm{Normal}(\mathbf{0}, \boldsymbol{\mathcal{J}}_{\boldsymbol{\theta}}), \tag{32}
$$

where $\boldsymbol{\mathcal{J}}_{\boldsymbol{\theta}}$ is the Fisher information computed at the true value of the parameter, $\boldsymbol{\theta}$. In practice, we only have a plug-in estimate, $\boldsymbol{\mathcal{J}}_{\hat{\boldsymbol{\theta}}}$, of $\boldsymbol{\mathcal{J}}_{\boldsymbol{\theta}}$. However, because $\boldsymbol{\mathcal{J}}_{\hat{\boldsymbol{\theta}}} \xrightarrow{p} \boldsymbol{\mathcal{J}}_{\boldsymbol{\theta}}$, using $\boldsymbol{\mathcal{J}}_{\hat{\boldsymbol{\theta}}}$ to approximate the standard error is sufficient to form asymptotically confidence intervals.

In Section 4, we form Wald-type confidence intervals for the unknown parameters in this fashion. To form prediction intervals, we generate a Monte Carlo sample of predicted values in a hierarchical fashion. We sample $\boldsymbol{\theta}$ from its asymptotic distribution and, conditional on each sampled value, we then sample a value for the response variable, Y_i. The limits of the prediction intervals are then formed using quantiles of these Monte Carlo samples. This procedure could be justified in the Bayesian paradigm by an appeal to the Bernstein von-Mises theorem because in large samples the posterior distribution is approximately equal to a normal distribution centered at the maximum likelihood estimate with variance equal to the asymptotic sampling variance.

B Bounds for φ

Proposition 1. *Let $y \sim$ GP-P(μ, φ, P) and constrain the parameter space as in Section 2.1. Then the following statements hold:*

1. *Regardless of the value of P, φ has no upper bound.*
2. *For $P \in (-\infty, 1) \cup (2, \infty)$, φ has no lower bound.*
3. *For $P \in [1, 2]$, $\varphi > \varphi_{min} = -2^{-P}$.*

Proof. Take the expression for θ in Section 2.1, substitute $\frac{\theta}{1-\delta}$ for μ, and solve for φ to arrive at the following:

$$\varphi = \frac{\delta(1-\delta)^{P-2}}{\theta^{P-1}} \tag{33}$$

To see that statement (1) is true, let $P \in \mathbb{R}$ and $M \in \mathbb{R}^+$ be arbitrary. We proceed in three cases.

- *Case 1:* $P = 1$. In this case, $\varphi = \frac{\delta}{1-\delta}$. Set $\delta > \frac{M}{1+M}$, which implies $\varphi = \frac{\delta}{1-\delta} > M$.
- *Case 2:* $P > 1$. Fix $\delta = 1/2$, $\theta \in (0, \frac{1}{2}\, M^{-1/(P-1)})$, and substitute:

$$\varphi = \frac{\delta(1-\delta)^{P-2}}{\theta^{P-1}} = \left(\frac{1}{2\,\theta}\right)^{P-1} > \left(\frac{1}{2\left(\frac{1}{2}\, M^{-1/(P-1)}\right)}\right)^{P-1} = \left(M^{1/(P-1)}\right)^{P-1} = M \tag{34}$$

  Case 3: $P < 1$. Fix $\delta = 1/2$ and $\theta > \frac{1}{2}\, M^{-1/(P-1)}$. Now Equation (34) holds for this case as well.

We now show that statement (2) is true. Let $P \in (-\infty, 1) \cup (2, \infty)$ and $M \in \mathbb{R}^+$ be arbitrary. We again proceed in cases.

- *Case 1:* $P < 1$. Fix $\delta = -1/2$, which implies that $\varphi = -3^{P-2}(2\theta)^{1-P}$. Consequently, $\varphi < -M$ is equivalent to the statement $(2\theta)^{1-P} > 3^{2-P}M = \widetilde{M}$. Now set $\theta > 1/2\widetilde{M}^{1/(1-P)}$, which is always possible within the GPD's parameter space because δ can take on any value in $(-1, 1)$ for $\theta > 4$. Now we substitute:

$$(2\theta)^{1-P} > \left[2\left(\frac{1}{2}\,\widetilde{M}^{1/(1-P)}\right)\right]^{1-P} = \widetilde{M} \tag{35}$$

- *Case 2:* $P > 2$. Consider the related sequences $\theta_n = 1/n$ and $\delta_n = -\theta_n/5 = -1/(5n)$. Because $\delta_n > -\theta_n/4$, we know that θ_n and δ_n are valid parameter values for the GPD. With some algebra, it can be shown that the related sequence for φ is $\varphi_n = -\frac{1}{5}(n + 1/5)^{P-2}$. Now set $n > (5M^{1/(P-2)} - 1/5)$ and substitute:

$$\varphi = -\frac{1}{5}(n + 1/5)^{P-2} < -\frac{1}{5}(5M^{1/(P-2)} - 1/5 + 1/5)^{P-2} = -\frac{1}{5}(5M) = -M \tag{36}$$

We now show that statement (3) is true. Let $P \in [1, 2]$. Because $\delta \geq 0$ corresponds to non-negative values of φ, we can consider the case where $\delta \in [-1, 0)$ without loss of generality. We now calculate the following partial derivative:

$$\frac{\partial \varphi}{\partial \theta} = \frac{(1 - P)\,\delta(1 - \delta)^{P-2}}{\theta^P} \geq 0 \tag{37}$$

Thus, for a fixed value of δ, φ is minimized when θ is equal to its minimum value: -4δ. We now use this result to simplify the expression for φ:

$$\varphi = \frac{\delta(1 - \delta)^{P-2}}{(-4\delta)^{P-1}} = \frac{\delta(1 - \delta)^{P-2}}{4^{P-1}(-\delta)^{P-1}} = -4^{1-P}\left(\frac{1 - \delta}{-\delta}\right)^{P-2} = -4^{1-P}(1 - 1/\delta)^{P-2} \tag{38}$$

The derivative with respect to δ is then

$$\frac{\partial \varphi}{\partial \delta} = -4^{1-P}(P - 2)(1 - 1/\delta)^{P-3}\delta^{-2} \geq 0. \tag{39}$$

Thus, φ is minimized as δ approaches it's lower bound of -1. Substituting into Equation (38) completes the proof:

$$\varphi_{min} = -4^{1-P}\,2^{P-2} = -2^{2-2P}\,2^{P-2} = -2^{-P} \tag{40}$$

Note that this bound is attainable if we allow $\delta = -1$.

□

C Log-normal Model-fitting Procedures

This appendix describes the model-fitting procedures for the log normal regression model introduced in the main text. Section C.1 derives the gradient and Hessian for our Newton–Raphson algorithm, and Section C.2 provides the details for the EM algorithm.

C.1 Gradient and Hessian Calculations

We drop the i subscript in this section for simplicity. We also define the following quantities for ease of notation:

- $\mu = \exp\left\{f(\mathbf{x})^\intercal \boldsymbol{\beta}\right\}$
- $\sigma = \exp\left\{g(\mathbf{x})^\intercal \boldsymbol{\alpha}\right\}$
- $\underline{z} = (\log(y) - \mu)/\sigma$ for $y \geq 1$; if $y = 0$, set $\underline{z} = -\infty$
- $\bar{z} = (\log(y+1) - \mu)/\sigma$
- $\Phi(\cdot)$: Standard normal cdf with $\Phi(-\infty) = 0$
- $\phi(\cdot)$: Standard normal pdf with $\phi(-\infty) = 0$
- $\kappa_d = \dfrac{\bar{z}^d\,\phi(\bar{z}) - \underline{z}^d\,\phi(\underline{z})}{\Phi(\bar{z}) - \Phi(\underline{z})}$, for $d \in \{0, 1, 2, 3\}$

Notice that $\mu, \sigma, \underline{z}, \bar{z},$ and κ_d implicitly depend on $\mathbf{x}, \boldsymbol{\alpha},$ and $\boldsymbol{\beta}$; we suppress this dependence for ease of notation. In a slight abuse of notation, in this appendix we will write ℓ as a function of its individual arguments—$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$—rather than as a function of $\boldsymbol{\theta} = (\boldsymbol{\alpha}^\intercal, \boldsymbol{\beta}^\intercal)^\intercal$. From Equation (2), we can then work out that

$$\frac{\partial}{\partial \boldsymbol{\beta}} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\frac{\kappa_0}{\sigma} f(\mathbf{x}). \tag{41}$$

Similarly, the gradient with respect to α is

$$\frac{\partial}{\partial \boldsymbol{\alpha}} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\kappa_1\, g(\mathbf{x}). \tag{42}$$

Thus, the full gradient is given by

$$\mathbf{s}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \begin{bmatrix} \kappa_{1i}\, f(\mathbf{x_i}) \\ (\kappa_{0i}/\sigma_i)\, g(\mathbf{x_i}) \end{bmatrix}. \tag{43}$$

Similar calculations yield the components of the Hessian, $\mathbf{H}(\boldsymbol{\theta})$:

$$\frac{\partial}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\intercal} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \frac{\kappa_{0i}^2 + \kappa_{1i}}{\sigma_i^2} f(\mathbf{x_i}) f(\mathbf{x_i})^\intercal \tag{44}$$

$$\frac{\partial}{\partial \boldsymbol{\beta} \partial \boldsymbol{\alpha}^\intercal} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \frac{\kappa_{2i} + \kappa_{0i} (\kappa_{1i} - 1)}{\sigma_i} f(\mathbf{x_i}) g(\mathbf{x_i})^\intercal \tag{45}$$

$$\frac{\partial}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^\intercal} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\sum_{i=1}^{n} [\kappa_{1i} (\kappa_{1i} - 1) + \kappa_{3i}] g(\mathbf{x_i}) g(\mathbf{x_i})^\intercal. \tag{46}$$

Note that $\frac{\partial}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\beta}^\intercal} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left[\frac{\partial}{\partial \boldsymbol{\beta} \partial \boldsymbol{\alpha}^\intercal} \ell(\boldsymbol{\alpha}, \boldsymbol{\beta})\right]^\intercal$. One potential challenge in performing these calculations is that the numerator and denominator in κ_d are often quite small, so a naive implementation can easily result in divide-by-zero errors as the denominator could, without too much difficulty, be computationally indistinguishable from zero. One way of circumventing this issue is to perform the calculations on the log scale. Doing so requires carefully using the following identity for both numerator and denominator: Given $0 < a < b$,

$$\log(b - a) = \log[a(b/a - 1)] = \log(a) + \log(b/a - 1). \tag{47}$$

The ratio b/a can also be written as $\exp\{\log(b) - \log(a)\}$. Together, these identities allow us to compute κ_d using the log-PDF and log-CDF of the normal distribution, both of which are readily available in common statistical computing environments.

C.2 Expectation-Maximization Details

In this section, we provide details for the EM algorithm. The E-step involves calculating the following expectation:

$$q(\boldsymbol{\alpha}, \boldsymbol{\beta}) \coloneqq \sum_{i=1}^{n} \mathbb{E}_{Z_i \sim p(z_i \,|\, y_i, \boldsymbol{\alpha}, \boldsymbol{\beta})} \log p(Z_i | \boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{48}$$

Where $p(Z_i|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the 'full-data' likelihood and $p(z_i \,|\, y_i, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is the conditional distribution of z_i given y_i, treating $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as fixed. We can work out this conditional distribution as follows:

$$p(z_i \,|\, y_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto \; p(z_i \,|\, \boldsymbol{\alpha}, \boldsymbol{\beta})\, p(y_i \,|\, z_i) = \text{Normal}(\mu_i, \sigma_i^2)\, \mathrm{I}\{y_i = \lfloor \exp(z_i) \rfloor\} \tag{49}$$

Thus, given $Y_i = y_i$, Z_i follows a truncated normal distribution with parameters μ_i, σ_i^2, $\log(y_i)$, and $\log(y_i + 1)$, where the latter two parameters indicate the interval of truncation. Calculating the above expectation requires the first and second moments of a truncated normal random variable. Following Section 10.1 of Kotz et al. (1994), the mean and variance can be computed as follows for a given observed value of y:

$$\begin{aligned} \mathrm{E}(Z \mid y) &= \mu - \kappa_0 \sigma \\ \mathrm{Var}(Z \mid y) &= \sigma^2(1 - \kappa_1 - \kappa_0^2) \end{aligned} \tag{50}$$

We can then easily solve for the second moment of Z as $\mathrm{Var}(Z \mid y) + [\mathrm{E}(Z \mid y)]^2$. For ease of notation, we denote the first and second moments as e_{1i} and e_{2i}, respectively, for observation i. Plugging these values into the normal pdf in Equation (48) then yields the expression given for q in Equation (11).

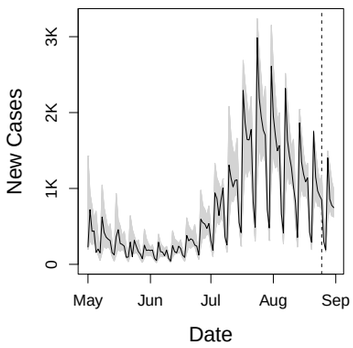
We now provide the gradient and Hessian of $q(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to α:

$$\begin{aligned} \tilde{\mathbf{s}}_{\boldsymbol{\beta}}(\boldsymbol{\alpha}) &= \frac{\partial}{\partial \boldsymbol{\alpha}} q(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{G}^\intercal(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-1}\mathbf{c} - \mathbf{1}) \\ \tilde{\mathbf{H}}_{\boldsymbol{\beta}} &= \frac{\partial}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^\intercal} q(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -2\mathbf{G}^\intercal \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-1} \mathbf{C}\mathbf{G} \end{aligned} \tag{51}$$

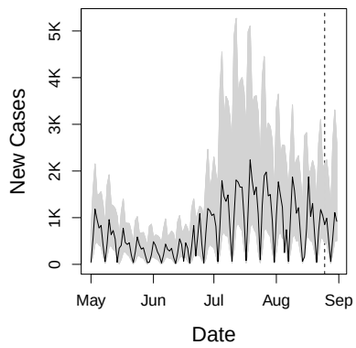
Where $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ is a diagonal matrix with i-th element equal to $\sigma_i^2 = \exp(2\, g(\mathbf{x_i})^\intercal \boldsymbol{\alpha})$, $\mathbf{C}$ is a diagonal matrix with i-th element equal to $e_{2i} - 2\, e_{1i}\, \mu_i + \mu_i^2$, $\mathbf{c} = \mathrm{diag}(\mathbf{C})$, $\mathbf{1}$ is a vector of ones, and $\mathbf{G}$ is an n-row matrix with row i equal to $g(\mathbf{x_i})$. We use these derivatives to perform Newton–Raphson updates to approximately maximize q. In principle, one could perform many such updates within each iteration of the EM algorithm. In our simulations, however, we found that it was sufficient to perform a single update within each iteration.

D Additional Case Study Figures

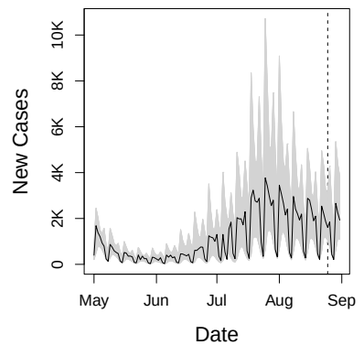
Figure 7 shows case counts and 95% prediction bands for the remaining 24 countries not shown in the main paper.



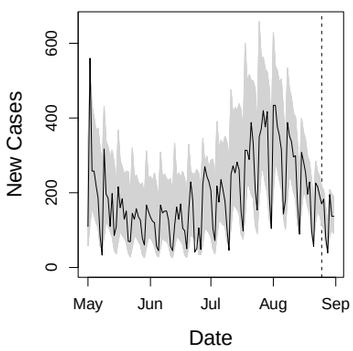
(a) Bulgaria



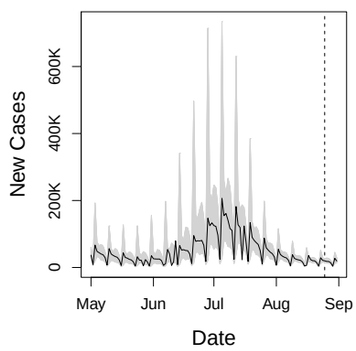
(b) Croatia



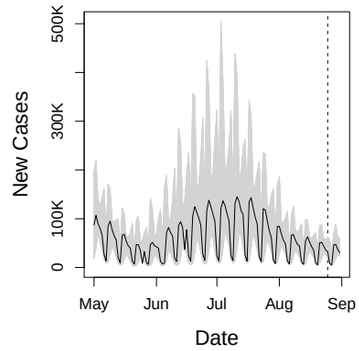
(c) Czech Republic



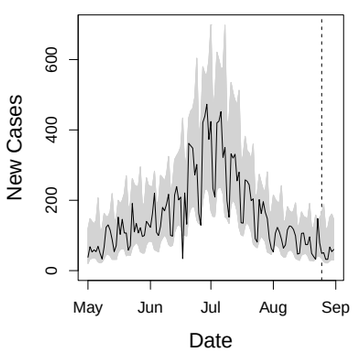
(d) Estonia



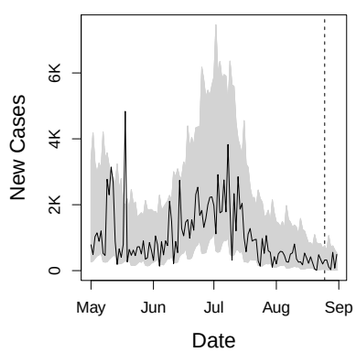
(e) France



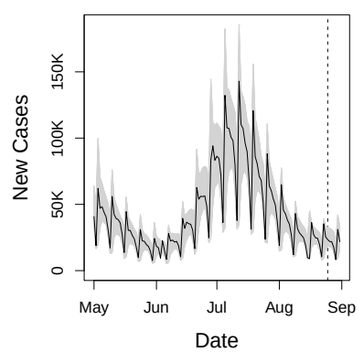
(f) Germany



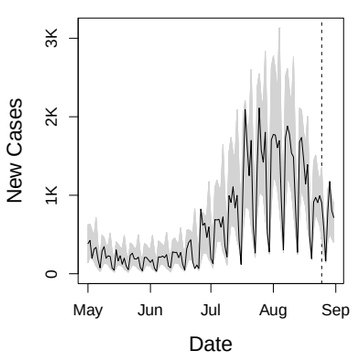
(g) Iceland



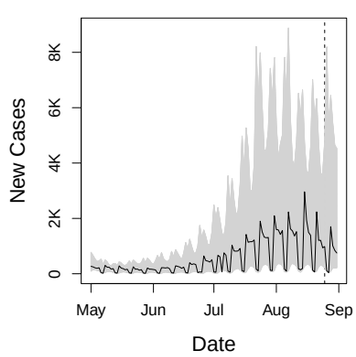
(h) Ireland



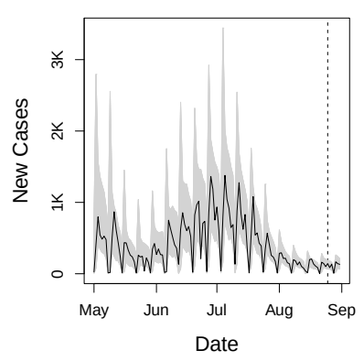
(i) Italy



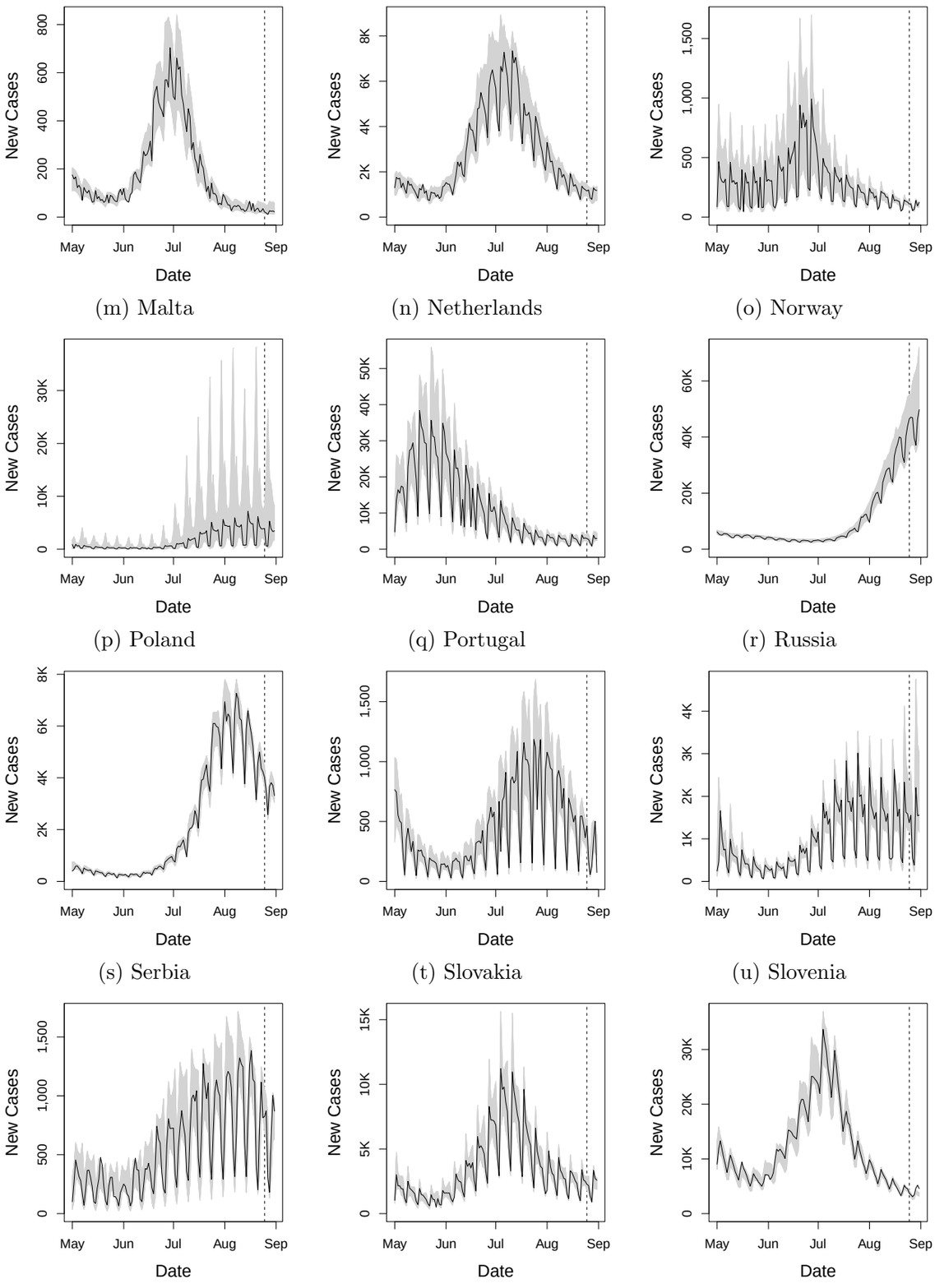
(j) Latvia



(k) Lithuania



(l) Luxembourg



(m) Malta (n) Netherlands (o) Norway

(p) Poland (q) Portugal (r) Russia

(s) Serbia (t) Slovakia (u) Slovenia

(v) Sweden (w) Switzerland (x) United Kingdom

Figure 7: New cases of COVID-19 (black line) in the remaining 24 of 27 European countries and 95% pointwise prediction bands (gray shaded area) from the discrete log-normal (DLN) model. The model is fit to case-count data from July 1, 2020 to August 24, 2022. The last week of plotted counts (separated by the dashed gray line) offer an out-of-sample performance comparison.