Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in
Statistica per le Tecnologie e le Scienze

Relazione Finale

Tweedie exponential dispersion models: theory, properties and inference

Relatore Prof. Alessandra Salvan
Dipartimento di Scienze Statistiche

Laureanda Greta Schiappacasse
Matricola 2037095

Anno Accademico 2023/2024

Contents

Introduction **1**

1 Exponential dispersion models **3**
1.1 Introduction . . . 3
1.2 Definitions . . . 4
1.3 Properties . . . 4
1.3.1 Moment and cumulant generating functions . . . 4
1.3.2 Expected value and variance . . . 5
1.3.3 Unique correspondence between θ and μ . . . 5
1.3.4 The variance function . . . 6
1.3.5 The unit deviance and the dispersion model form . . . 6
1.3.6 The saddlepoint approximation . . . 7

2 Generalised linear models **9**
2.1 Introduction . . . 9
2.2 Definitions . . . 9
2.3 Inference . . . 11
2.3.1 Loglikelihood, score function and information for β . . . 11
2.3.2 Estimates of β and ϕ . . . 12
2.4 Randomised quantile residuals . . . 13

3 Tweedie exponential dispersion models **17**
3.1 Introduction . . . 17
3.2 Definitions . . . 17
3.3 Properties . . . 18
3.3.1 Different types of Tweedie models and notable cases . . . 18
3.3.2 The parameterisation $\mu_\xi(\theta)$ and $\theta_\xi(\mu)$ . . . 19
3.3.3 The functions $b_\xi(\theta)$ and $K_\xi(t;\theta,\phi)$ . . . 19
3.3.4 The unit deviance . . . 20
3.3.5 Scale transformations . . . 20
3.3.6 The Tweedie convergence theorem . . . 21
3.3.7 Link with Box-Cox transformations . . . 21
3.4 The density function . . . 24
3.4.1 The saddlepoint approximation . . . 24

3.4.2 Infinite series expansion . . . 25
3.4.3 Inversion of the cumulant generating function . . . 26
3.5 Tweedie EDMs with $\xi < 0$ . . . 27
3.6 Tweedie EDMs with $\xi = 1$ . . . 27
3.7 Tweedie EDMs with $\xi \in (1, 2)$ . . . 29
3.7.1 The link with the compound Poisson-gamma distribution . . . 29
3.7.2 Density function . . . 30
3.7.3 Properties . . . 31
3.7.4 Type of data modelled . . . 33
3.7.5 Taylor's power law . . . 34
3.8 Tweedie EDMs with $\xi > 2$ . . . 36
3.8.1 Type of data modelled . . . 36

4 Inference **39**
4.1 Assumptions . . . 39
4.2 Maximum likelihood estimation . . . 40
4.2.1 Likelihood, score function and information for (β, ξ, ϕ) . . . 40
4.3 Estimation of the parameters β and ϕ . . . 42
4.4 Estimation of the power parameter ξ . . . 43
4.4.1 The estimation of ξ: a method based on grouping . . . 43
4.4.2 Inference about ξ . . . 44
4.4.3 Profile log-likelihood for ξ . . . 44
4.4.4 Computing the profile log-likelihood for ξ and related quantities . . . 46

5 Simulation studies **57**

Conclusion **62**

Appendix **65**

Bibliography **77**

Introduction

Tweedie models are a class of exponential dispersion models (EDMs) characterised by scale invariance and a variance-to-mean power relationship. First introduced by Tweedie (1984), their properties were studied extensively by Jorgensen (1997), who also coined the name 'Tweedie models'. Depending on the value of the power parameter ξ, the Tweedie family includes positive continuous distributions, real-valued continuous distributions, discrete distributions and non-negative mixed distributions, which have a positive probability of taking a value of zero. Several well-known probability distributions, such as the normal, Poisson, gamma and inverse Gaussian, are included in this family. Furthermore, Tweedie models act as a point of convergence for all EDMs that present variance functions that asymptotically behave as powers of the mean, for large or small values of the expected value.

Tweedie EDMs have been extensively used to model total amounts of physical phenomena (such as total rainfall and total insurance payouts), time duration and variables with highly skewed distributions, consistently outperforming alternative models. However, except for the aforementioned notable cases, their density function does not possess an explicit analytic expression, complicating the use of this family of distributions in statistical modelling.

The discussion proceeds as follows: Chapter 1 introduces exponential dispersion models and their main properties. In Chapter 2, generalised linear models are presented and their inferential framework is established. The first part of Chapter 3 focuses on Tweedie models as probability distributions, exploring their general properties. The second part of Chapter 3 examines different types of Tweedie models based on the value of the power parameter ξ, while their real-world applications are discussed. In Chapter 4, likelihood inference is illustrated with a particular focus on estimating the power parameter ξ via profile likelihood. This latter chapter includes the implementation of a function in R that computes an estimate and a confidence interval for the Tweedie parameter ξ,

analogous to the `tweedie.profile` function from the package `tweedie` (Dunn, 2017). In Chapter 5, simulation studies are conducted to examine the properties of likelihood inference about ξ, as computed by the aforementioned R function, for various values of the power parameter and two different sample sizes. These simulations indicate that the maximum likelihood estimator (MLE) for ξ tends to be negatively biased for smaller values of ξ and positively biased for larger values of the power parameter. Additionally, the actual coverage of the confidence interval is generally lower than the nominal level. The MLE performs better when the sample size is large and covariates are included. However, the case with covariates produces confidence intervals with actual coverage the furthest from the nominal one.

Chapter 1

Exponential dispersion models

1.1 Introduction

Among models employed to describe how the distribution of a response variable depends on the covariates, normal linear regression models are the most widely used, mainly because of their simplicity. However, there are several common scenarios in which they fall short: these models assume that the response is a realisation of independent normally distributed random variables with a common, but usually unknown, variance.

Not all data are continuous, can take on all real values and have a constant variance: consequently, not all data are suitable to be modelled as realisations of the normal distribution. For instance, consider an experiment that consists of n attempts where each attempt has only two possible outcomes: success or failure. Then, the variable that describes the outcome of each attempt can only take on two possible values: 0, if the experiment failed, or 1, if the experiment succeeded. In this scenario, we talk about binary data. The variable that counts the number of successes (or, equivalently, of failures) can take value on all natural numbers from 0 to n, or $\mathbb{N}$, if $n \to \infty$. In this case, we talk about count data. The variable that describes the proportion of successes in each set of attempts can take values on the interval [0,1]. Alternatively, non-discrete variables that measure physical quantities or time duration can only take on positive (or non-negative) values.

It is possible to define a more general family of distributions, known as the exponential dispersion family, that possesses many advantageous properties, encompasses both continuous and discrete distributions and can model all of the scenarios previously described. The normal distribution and the other most common probability distributions studied in introductory probability courses are special cases of this family.

Unless specified otherwise, the discussion presented in the following section is based on Dunn & Smyth (2018, Chapter 5), and Salvan et al. (2020, Chapter 2).

1.2 Definitions

Let

- θ be a parameter having parameter space an open interval $\Theta \subseteq \mathbb{R}$,
- ϕ be a parameter having parameter space $(0, +\infty)$,
- Y be a random variable, continuous or discrete, having mean μ and support $S \subseteq \mathbb{R}$ that does not depend on the parameters ϕ and θ,
- $p_Y(y; \theta, \phi)$ be the probability density function (if Y is continuous) or probability mass function (if Y is discrete) of Y,
- $b : \Theta \to \mathbb{R}$, $a : \mathbb{R}^+ \times S \to \mathbb{R}^+$ be two known functions,
- m be an a-priori known weight, which usually equals 1.

Then the random variable Y is an exponential dispersion model (EDM) having mean μ and dispersion parameter ϕ (in notation, $Y \sim ED(\mu, \phi)$) if $p_Y(y; \theta, \phi)$ can be written as

$$p_Y(y; \theta, \phi) = \exp\left\{\frac{\theta y - b(\theta)}{\phi/m} + a(y, \phi)\right\} = \widetilde{a}(y, \phi) \exp\left\{\frac{\theta y - b(\theta)}{\phi/m}\right\}, \tag{1.1}$$

where $\widetilde{a}(y, \phi) = \exp\{a(y, \phi)\}$ is sometimes referred to as normalising function (Dunn & Smyth, 2018, page 212).

1.3 Properties

1.3.1 Moment and cumulant generating functions

Let Y be a random variable having support $S \subseteq \mathbb{R}$ and probability density function, if Y is continuous, or probability mass function, if Y is discrete, $p_Y(y)$. Then, the moment generating function (MGF) of Y is defined as (Pace & Salvan, 2001, pages 17-18)

$$M_Y(t) = E[e^{tY}] = \begin{cases} \sum\limits_{y \in S} e^{ty} p_Y(y) & \text{if Y is discrete} \\ \int\limits_S e^{ty} p_Y(y)\, dy & \text{if Y is continuous} \end{cases} \qquad \forall\ t \mid E[e^{tY}] \text{ is finite.}$$

Instead, the cumulant generating function (CGF) of Y is defined as

$$K_Y(t) = \log(M_Y(t)) \qquad \forall\ t \mid E[e^{tY}] \text{ is finite.}$$

The MGF and CGF of EDMs have a simple form even when the denisty function does not have an explicit analytic expression.

Let $Y \sim ED(\mu, \phi)$ having canonical parameter θ, then its moment and cumulant generating functions are

$$M_Y(t; \theta, \phi) = \exp\left\{ \frac{b(\theta + t\phi/m) - b(\theta)}{\phi/m} \right\} \tag{1.2}$$

$$K_Y(t; \theta, \phi) = \frac{b(\theta + t\phi/m) - b(\theta)}{\phi/m}. \tag{1.3}$$

For details on the proof, see Dunn & Smyth (2018, page 214) or Salvan et al. (2020, page 69).

1.3.2 Expected value and variance

Let Y be a random variable with CGF $K_Y(t)$ then, the r-th derivative of the CGF, with respect to t, evaluated at $t = 0$, is equal to κ_r, the r-th cumulant of the distribution (Dunn & Smyth, 2018, page 214).

$$\kappa_r = \left. \frac{d^r K_Y(t)}{dt^r} \right|_{t=0}.$$

Applying this property, we can derive the mean, i.e. the first cumulant, and the variance, i.e. the second cumulant, of exponential dispersion models. Let $Y \sim ED(\mu, \phi)$ having canonical parameter θ, then its expected value $E_{\theta,\phi}[Y]$ and its variance $Var_{\theta,\phi}[Y]$ can be expressed as functions of θ and ϕ in the following way

$$\mu = E_{\theta,\phi}[Y] = \left. \frac{dK_Y(t; \theta, \phi)}{dt} \right|_{t=0} = \frac{db(\theta)}{d\theta} = b'(\theta) \tag{1.4}$$

$$Var_{\theta,\phi}[Y] = \left. \frac{d^2 K_Y(t; \theta, \phi)}{dt^2} \right|_{t=0} = \frac{\phi}{m} \frac{d\mu}{d\theta} = \frac{\phi}{m} b''(\theta). \tag{1.5}$$

1.3.3 Unique correspondence between θ and μ

Since the second derivative of the cumulant generating function with respect to θ is equal to the variance, $\frac{d^2 K_Y(t;\theta,\phi)}{d\theta^2}$ must be positive. If the second derivative is positive,

then the first derivative of the CGF (i. e. the mean) must be a monotonic increasing function of θ. This implies that μ and θ are one-to-one functions of each other, hence all quantities derived up until now (the density function, the MGF and the CGF) can be equally expressed as functions of θ or μ, using the following parameterisation, where Ω is the parameter space of the mean μ.

$$\mu : \Theta \to \Omega \quad \mu = \mu(\theta) = b'(\theta) \tag{1.6}$$

$$\theta : \Omega \to \Theta \quad \theta = \theta(\mu) = [b'(\theta)]^{-1}\Big|_{\theta=\theta(\mu)}. \tag{1.7}$$

1.3.4 The variance function

Substituting the parameterisation (1.7) in (1.5), the variance of an EDM can be written as a function of the mean μ and the dispersion parameter ϕ.

$$Var[Y] = \frac{\phi}{m} b''(\theta(\mu)) = \frac{\phi}{m} v(\mu),$$

where

$$v(\mu) = b''(\theta(\mu)) = b''(\theta)\Big|_{\theta=\theta(\mu)}. \tag{1.8}$$

The function $v(\mu)$ is called the variance function of an EDM. If $v(\mu)$ is constant, the variance of the distribution varies independently of the mean: the only EDM for which this happens is the normal distribution. Otherwise, the variable's variance depends on the expected value through the relationship specified by the variance function.

It is possible to uniquely identify θ and $b(\theta)$ up to an additive constant, knowing only $v(\mu)$: this implies that EDMs are characterised by their variance function (Morris, 1983).

1.3.5 The unit deviance and the dispersion model form

Let Y be and EDM with mean μ and density function (1.1). Let us define $t(y, \mu) = \{\theta y - b(\theta)\}\Big|_{\theta=\theta(\mu)}$. Then, the unit deviance of Y is

$$d(y, \mu) = 2\{t(y, y) - t(y, \mu)\}. \tag{1.9}$$

It can be proven that $t(y, \mu)$ has a unique maximum in $\mu = y$ (Dunn & Smyth, 2018, page 219). This implies that $t(y, y) \geq t(y, \mu)\ \forall \mu \in \Omega$. Since the unit deviance is always

non-negative and is null only when $y = \mu$, it can be interpreted as a measure of distance between y and the mean μ.

The definition (1.9) assumes that S, the support of Y, is contained in the closure of Ω, the parameter space of the mean. However, there exist some EDMs for which this is not true. For those cases, the unit deviance definition can be generalised to

$$d(y, \mu) = 2\{\sup_{\mu \in \Omega}\{t(y, \mu)\} - t(y, \mu)\}. \tag{1.10}$$

Using the unit deviance, the density function of EDMs can be rewritten as

$$p_Y(y; \mu, \phi) = \widetilde{c}(y, \phi) \exp\left\{-\frac{d(y, \mu)}{2\phi/m}\right\} = \exp\left\{-\frac{d(y, \mu)}{2\phi/m} + c(y, \phi)\right\}. \tag{1.11}$$

This is called the dispersion model form of an EDM. The function $\widetilde{c}(y, \phi)$ coincides with $\widetilde{a}(y, \phi) \exp\{\frac{t(y,y)}{\phi/m}\}$, where $\widetilde{a}(y, \phi)$ is the normalising function defined in (1.1).

1.3.6 The saddlepoint approximation

Let $Y \sim ED(\mu, \phi)$ having density function $p_Y(y; \mu, \phi)$. Then, the saddlepoint approximation of $p_Y(y; \mu, \phi)$ is (Dunn & Smyth, 2018, page 223)

$$\widetilde{p}_Y(y; \mu, \phi) = \frac{1}{\sqrt{2\pi\phi\ v(y)}} \exp\left\{-\frac{d(y, \mu)}{2\phi/m}\right\} \to p_Y(y; \mu, \phi) \quad \text{as } \phi \to 0. \tag{1.12}$$

This result is particularly advantageous for EDMs whose normalising functions $\widetilde{c}(y, \phi)$ (1.1) and $\widetilde{a}(y, \phi)$ (1.11) cannot be written in closed form. Indeed, by comparing (1.12) with the density function in the dispersion model form (1.11), it is evident that using the saddlepoint approximation equals writing $\widetilde{c}(y, \phi) = \frac{1}{\sqrt{2\pi\phi v(y)}}$. Thus, a complicated function that is not available in closed form can be approximated by a simple analytic function.

The saddlepoint approximation is exact for the normal and the inverse Gaussian distributions.

Chapter 2

Generalised linear models

2.1 Introduction

As illustrated in the previous chapter, not all response variables Y can be interpreted as realisations of the normal distribution. Thus, the exponential dispersion family was introduced. A new class of models must be defined to perform statistical modelling using EDMs. To satisfy this need, generalised linear models (GLMs) were first introduced by Nelder & Wedderburn (1972).

2.2 Definitions

Let us consider a sample of n statistical units identified by the index $i = 1, \ldots, n$, a response variable Y and p covariates $x_j, \quad j = 1, \ldots, p$.

According to this notation, the following quantities can be defined:

y_i, value of the response Y measured on the i-th statistical unit;

x_{ij}, value of the j-th covariate measured on the i-th statistical unit;

$x_i = [x_{i1}, ...x_{ip}]$, p-variate row vector of the covariates measured on the i-th statistical unit;

$X = [x_1^\intercal, ..x_n^\intercal]^\intercal$, $n \times p$ model matrix.

The covariates are assumed to be quantitative, discrete or continuous, but factors with K levels can be easily converted into $K-1$ dummy variables of the following type:

$$x_{ik} = \begin{cases} 1 & \text{if the factor measured on the } i\text{-th statistical unit takes value} \\ & \text{on the } k\text{-th level} \\ 0 & \text{otherwise} \end{cases}$$

$i = 1, \ldots, n$ and $k = 2, \ldots, K$.

Like other regression models, a generalised linear model consists of a stochastic component, in which the probability distribution of the response variable is established, and a systematic component, that is wholly deterministic.

The systematic component is comprised of a linear predictor that, in most cases, is a linear combination of the unknown parameters $\beta = (\beta_1, ..., \beta_p)^\intercal$ with the covariates as coefficients.

$$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij} = x_i \beta, \text{ linear predictor of the } i\text{-th unit.}$$

However, it can also include interaction terms, offsets (additive components where the coefficient is known and requires no estimation), quadratic terms or, more rarely, other simple transformations of the covariates. It should be noted that η_i needs to be linear with respect to the regression parameters, not to the covariates.

The linear predictor is used to model a parameter of the distribution of the response variable Y_i, usually the expected value $E[Y_i]$.

In normal regression models, $\mu_i = E[Y_i]$ and η_i coincide with each other. However, typically that is not the case for other EDMs, since the domain of the mean Ω does not necessarily coincide with $\mathbb{R}$, but it is actually the interior of the convex hull of the support S of the response. This implies that there cannot be an identity link between the expectation $E[Y_i]$ of the response and the linear predictor η_i: in GLMs, they are defined as two distinct quantities that are linked through function $g : \Omega \to \mathbb{R}$, called a link function. In symbols

$$\eta_i = g(\mu_i).$$

The function $g(\cdot)$ must have domain Ω and range $\mathbb{R}$, be invertible, and preferably increasing, to facilitate the interpretation of the regression coefficients. Moreover, it needs to be differentiable to compute the score function for the parameter β.

For each EDM, there exists a privileged link function, known as canonical link function, which coincides with $\theta(\mu_i)$ (1.7). Using the canonical link function, the linear predictor corresponds to the natural parameter θ_i.

In GLMs, it's assumed that y_i for $i = 1, \ldots, n$ are independent realisations of an EDM having mean μ_i and dispersion parameter ϕ (Dunn & Smyth, 2018, page 230). In notation

$$Y_i \sim ED(\mu_i, \phi), \text{ independent} \quad i = 1, \ldots, n.$$

2.3 Inference

Unless specified differently, the discussion presented in this section follows Salvan et al. (2020, Chapter 2.3).

2.3.1 Loglikelihood, score function and information for β

Let $y = (y_1, \ldots, y_n)^\intercal$ be the n-variate vector of realisations of $ED(\mu_i, \phi)$ defined in the previous section. Then its likelihood and log-likelihood functions, using the (θ_i, ϕ) parameterisation, are

$$L^{\theta,\phi}(\theta_i, \phi) = \prod_{i=1}^{n} p_{Y_i}(y_i; \theta_i, \phi) = \prod_{i=1}^{n} \exp\left\{\frac{\theta_i y_i - b(\theta_i)}{\phi/m_i} + a(y_i, \phi)\right\}$$

$$l^{\theta,\phi}(\theta_i, \phi) = \sum_{i=1}^{n} \log\{L^{\theta,\phi}(\theta_i, \phi)\} = \sum_{i=1}^{n} \left\{\frac{\theta_i y_i - b(\theta_i)}{\phi/m_i} + a(y_i, \phi)\right\}.$$

They can be expressed as a function of the unknown regression parameters β, which will be the object of our estimation.

Using the canonical link function, which implies that $\theta_i = \eta_i = x_i\beta$, the log-likelihood can be rewritten as

$$l^{\beta,\phi}(\beta, \phi) = \sum_{i=1}^{n} \left\{\frac{x_i\beta y_i - b(x_i\beta)}{\phi/m_i} + a(y_i, \phi)\right\}. \tag{2.1}$$

Using a general link function $g(\cdot)$, which implies that $\theta_i = \theta(\mu_i) = \theta(g^{-1}(\eta_i)) = \theta(g^{-1}(x_i\beta))$, the log likelihood can be rewritten as

$$l^{\beta,\phi}(\beta, \phi) = \sum_{i=1}^{n} \left\{\frac{\theta(g^{-1}(x_i\beta))y_i - b(\theta(g^{-1}(x_i\beta)))}{\phi/m_i} + a(y_i, \phi)\right\}. \tag{2.2}$$

Since the parameter we are interested in estimating is β, while ϕ acts as a nuisance parameter (very much like the unknown common variance σ^2 in normal regression models), it's advantageous to split the score function into two parts, one that refers to β while the other to ϕ.

$$U(\beta, \phi)^\intercal = [U_\beta(\beta, \phi)^\intercal, U_\phi(\beta, \phi)^\intercal] = [\frac{\partial l^{\beta,\phi}(\beta, \phi)}{\partial \beta^\intercal}, \frac{\partial l^{\beta,\phi}(\beta, \phi)}{\partial \phi}]$$

$$= [\frac{\partial l^{\beta,\phi}(\beta, \phi)}{\partial \beta_1}, \ldots \frac{\partial l^{\beta,\phi}(\beta, \phi)}{\partial \beta_p}, \frac{\partial l^{\beta,\phi}(\beta, \phi)}{\partial \phi}] = [U_{\beta_1}, \ldots U_{\beta_p}, U_\phi].$$

The generic element of the block of the score function that refers to the regression parameters is

$$U_{\beta_j} = \frac{\partial l^{\beta,\phi}(\beta,\phi)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial l^{\beta,\phi}}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - b'(\theta_i)}{\phi/m_i}\frac{1}{v(\mu_i)}\frac{1}{g'(\mu_i)}\Big|_{\mu_i = g^{-1}(\eta_i)} x_{ij} \tag{2.3}$$

$j = 1, \ldots, p.$

Using the canonical link function, which implies that $\theta_i = \eta_i = x_i\beta$, the generic element U_{β_j} can be rewritten as

$$U_{\beta_j} = \sum_{i=1}^{n} \frac{\partial l^{\beta,\phi}}{\partial \theta_i}\frac{\partial \theta_i}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - b'(\theta_i)}{\phi/m_i} x_{ij}. \tag{2.4}$$

The element of the score function vector that refers to the dispersion parameter is

$$U_\phi = \frac{\partial l^{\theta,\phi}(\theta_i,\phi)}{\partial \phi} = \sum_{i=1}^{n} \left\{ \frac{b(\theta_i) - \theta_i y_i}{\phi^2/m_i} + a'(y_i,\phi) \right\}.$$

The calculation of the observed information matrix will be skipped. Instead, the Fisher information matrix is given below. As was done for the score vector, it is possible to split the information matrix into four blocks:

$$i(\beta,\phi) = E\begin{bmatrix} -\frac{\partial^2 l(\beta,\phi)}{\partial\beta\partial\beta^\intercal} & -\frac{\partial^2 l(\beta,\phi)}{\partial\beta\partial\phi} \\ -\frac{\partial^2 l(\beta,\phi)}{\partial\phi\partial\beta^\intercal} & -\frac{\partial^2 l(\beta,\phi)}{\partial\phi^2} \end{bmatrix} = \begin{bmatrix} -E[\frac{\partial U_\beta(\beta,\phi)}{\partial\beta^\intercal}] & -E[\frac{\partial U_\phi(\beta,\phi)}{\partial\beta^\intercal}] \\ -E[\frac{\partial U_\beta(\beta,\phi)}{\partial\phi}] & -E[\frac{\partial U_\phi(\beta,\phi)}{\partial\phi}] \end{bmatrix} = \begin{bmatrix} i_{\beta\beta} & i_{\beta\phi} \\ i_{\phi\beta} & i_{\phi\phi} \end{bmatrix}.$$

$i(\beta,\phi)$ is a block diagonal matrix, since the vectors $i_{\phi\beta}$ and $i_{\beta\phi}$ have null elements.

$$[i_{\phi\beta}]_j = [i_{\beta\phi}]_j = E[\frac{\partial U_{\beta_j}}{\partial\phi}] = E[\sum_{i=1}^{n}(\mu_i - Y_i)\frac{\partial\theta_i}{\partial\beta_j}\frac{m_i}{\phi^2}] = 0 \quad j = 1, \ldots, p.$$

This implies that the parameters (β,ϕ) are orthogonal: thus, the corresponding maximum likelihood estimators are asymptotically independent.

The generic element of the $i_{\beta\beta}$ block is

$$[i_{\beta\beta}]_{jk} = \sum_{i=1}^{n} \frac{m_i}{\phi}\frac{x_{ij}\ x_{ik}}{((g'(\mu_i)^2)\ v(\mu_i)}, \quad j,k = 1, \ldots, p. \tag{2.5}$$

2.3.2 Estimates of β and ϕ

With very few exceptions, the score equations $U_\beta(\beta,\phi) = 0$ do not have explicit solutions and need to be solved numerically. The most commonly used method for estimating the

regression coefficients is the iterative reweighted least squares algorithm (IRLS), which computes the estimate $\hat{\beta}$ without fixing a value for the dispersion parameter ϕ.

Using Newton's method on the loglikelihood function (2.4), with Fisher's information (2.5), the m-th step of the algorithm is

$$\hat{\beta}^{(m+1)} = \hat{\beta}^{(m)} + [i_{\beta\beta}(\hat{\beta}^{(m)})]^{-1} U_{\beta}(\hat{\beta}^{(m)}).$$

Let us define

$$W = \text{diag}(w_i), \text{ where } w_i = \frac{1}{((g'(\mu_i))^2 \ \phi/m_i v(\mu_i)}.$$

The iteration step can be rewritten as (for details see Salvan et al. (2020), page 86)

$$X^{\intercal} W X \ \hat{\beta}^{(m+1)} = X^{\intercal} W (X \ \hat{\beta}^{(m)} + u).$$

Where X is the model matrix and $u_i = (y_i - \mu_i) g'(\mu_i)$. This alternative formulation reduces the computational load because it is similar to the normal equation of generalised least squares $X^{\intercal}\Omega^{-1}X\beta = X^{\intercal}\Omega^{-1}Y$, where Ω is the known variance-covariance matrix of the response (Salvan et al., 2020, page 40). When the algorithm reaches its convergence, β is estimated as $\hat{\beta} = (X^{\intercal}\hat{W}X)^{-1}X^{\intercal}\hat{W}\hat{z}$, where $\hat{z}_i = x_i\hat{\beta} + (y_i - \hat{\mu}_i)g'(\hat{\mu}_i)$.

Even though, in this context, the dispersion parameter ϕ is not the object of interest, it needs to be estimated to perform statistical inference on the regression parameters β. For discrete EDMs, ϕ is fixed, thus no estimation is required. If Y is a continuous variable, ϕ is estimated using the method of moments, instead of the MLE, which is highly biased. The estimator of ϕ based on the method of moments is

$$\widetilde{\phi} = \frac{1}{n-p} \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{v(\hat{\mu}_i)}, \qquad \text{with } \hat{\mu}_i = g^{-1}(x_i\hat{\beta}) \ . \tag{2.6}$$

2.4 Randomised quantile residuals

Once a statistical model has been estimated, it is essential to assess how well the model describes the observed data. The most immediate way to evaluate the model's goodness of fit is by examining the discrepancies between the observed data and the values predicted by the model. In Statistics, the quantities that measure those differences are known as residuals. Residuals are also useful for identifying outliers and verifying whether the assumptions of the chosen model are satisfied. If the assumptions are not all met, the pattern that the residuals take on can suggest possible adjustments to improve the model's performance.

Several kinds of residuals can be defined: in normal regression models, the simplest and most commonly used are response residuals. Let y_i be the observed value of the response for the i-th unit and $\hat{\mu}_i$ be the predicted value of the response's mean e under the model for the i-th unit. Then, the response residual of the i-th unit is $r_i = y_i - \hat{\mu}_i$. Unfortunately, response residuals do not perform equally well in generalised linear models. If the response takes on a limited set of values, the plotted residuals exhibit unusual patterns and are far from being normally distributed.

Quantile residuals, first introduced by Cox & Snell (1968) but formally defined by Dunn & Smyth (1996), are designed to be distributed as a standard normal distribution and represent an alternative to the most commonly used types of residuals. This kind of residuals can be defined for any distribution: however, in this context, we limit ourselves to EDMs. Let

- $Y_i \sim ED(\mu_i, \phi)$ be a continuous random variable having cumulative distribution function $F_{Y_i}(y; \mu_i, \phi)$ and realisation y_i;
- $\hat{\mu}_i$ and $\hat{\phi}$ be the predicted value for the mean of Y_i and an estimate of the dispersion parameter under a correctly specified GLM.

Then, the corresponding quantile residual is defined as the random variable

$$R_i^Q = \Phi^{-1}(F_{Y_i}(Y_i; \hat{\mu}_i, \hat{\phi})) \dot{\sim} N(0, 1), \tag{2.7}$$

where $r_i^Q = \Phi^{-1}(F_{Y_i}(y_i; \hat{\mu}_i, \hat{\phi}))$ is the realisation of R_i^Q and $\Phi^{-1}(\cdot)$ is the quantile function of the standard normal distribution $N(0, 1)$. Since the cumulative probability function $F_{Y_i}(Y_i; \mu_i, \phi) \sim Unif(0, 1)$ and $\Phi^{-1}(Unif(0, 1)) \sim N(0, 1)$, quantile residuals are exactly distributed as a standard normal. If μ_i and ϕ are consistently estimated, then the distribution of $\Phi^{-1}(F_{Y_i}(Y_i; \hat{\mu}_i, \hat{\phi}))$ also converges to a standard normal.

If Y_i is a non-continuous EDM, a simple modification must be applied when defining quantile residuals so that they exhibit a continuous distribution. Otherwise, there would be masses of overlapping points in the residuals' plots, making the interpretation of such graphs nearly impossible (Dunn & Smyth, 1996). Let

- $Y_i \sim ED(\mu_i, \phi)$ be a discrete random variable having cumulative distribution function $F_{Y_i}(y; \mu_i, \phi)$ and realisation y_i,
- $\hat{\mu}_i$ and $\hat{\phi}$ be the predicted value for the mean and an estimate of the dispersion parameter under a correctly specified GLM,

- $b_i = F_{Y_i}(y_i; \hat{\mu}_i, \hat{\phi})$ be the value of the cumulative distribution function evaluated at y_i and $a_i = \lim_{\epsilon \to 0^-} F_{Y_i}(y_i + \epsilon; \hat{\mu}_i, \hat{\phi})$ be the value of the cumulative distribution function evaluated immediately before y_i,
- $U_i \sim Unif(a_i, b_i]$ having realisation u_i.

Then, the corresponding quantile residual is defined as the random variable

$$R_i^Q = \Phi^{-1}(U_i), \tag{2.8}$$

where $r_i^Q = \Phi^{-1}(u_i)$ is the realisation of R_i^Q and $\Phi^{-1}(\cdot)$ is the quantile function of the standard normal distribution $N(0, 1)$.

Because of the randomisation employed in defining quantile residuals for discrete distributions, it is advisable to compute this kind of residuals multiple times, as the realisations of the uniform variable U_i will vary. Only the patterns that remain consistent across multiple realisations should be considered when interpreting the plots (Dunn & Smyth, 1996).

TABLE 2.1: The most common EDMs in the (μ, ϕ) parameterisation and their support S, canonical parameter space Θ, parameter space of the mean Ω, function $b(\theta)$, parameterisations $\theta(\mu)$ and $\mu(\theta)$, variance function $v(\mu)$, unit deviance $d(y, \mu)$, constant m and dispersion parameter ϕ.

Model	S	Θ	Ω	$b(\theta)$	$\theta(\mu)$	$\mu(\theta)$	$v(\mu)$	$d(y,\mu)$	m	ϕ
$N(\mu,\phi)$	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	$\frac{\theta^2}{2}$	μ	θ	1	$(y-\mu)^2$	1	ϕ
$Ga(\mu,\phi)$	$\mathbb{R}^+$	$\mathbb{R}$	$\mathbb{R}^+$	$-\log(-\theta)$	$-\frac{1}{\mu}$	$-\frac{1}{\theta}$	μ^2	$2\{-\log\frac{y}{\mu}+\frac{y-\mu}{\mu}\}$	1	ϕ
$Pois(\mu)$	$\mathbb{N}_0$	$\mathbb{R}$	$\mathbb{R}^+$	e^θ	$\log(\mu)$	e^θ	μ	$2\{y\log\frac{y}{\mu}-(y-\mu)\}$	1	1
$\frac{Bin(m,\mu)}{m}$	$\{\ \frac{k}{m}\|k\in\mathbb{N}_0\}$	$\mathbb{R}$	$(0,1)$	$\log(1+e^\theta)$	$\log(\frac{\mu}{1-\mu})$	$\frac{\exp\{\theta\}}{1+\exp\{\theta\}}$	$\mu(1-\mu)$	$2\{y\log\frac{y}{\mu}+(1-y)\log\frac{1-y}{1-\mu}\}$	m	$\frac{1}{m}$
$NBin(k,\mu)-k$	$\mathbb{N}_0$	$\mathbb{R}^-$	$\mathbb{R}^+$	$-\log(1-e^\theta)$	$\log(\frac{\mu}{\mu+k})$	$\frac{k\exp\{\theta\}}{1-\exp\{\theta\}}$	$\mu+\frac{\mu^2}{k}$	$2\{y\log\frac{y}{\mu}-(y+k)\log\frac{y+k}{\mu+k}\}$	1	1
$IG(\mu,\phi)$	$\mathbb{R}^+$	$\mathbb{R}_{\leq 0}$	$\mathbb{R}^+$	$-\sqrt{-2\theta}$	$-\frac{1}{2\mu^2}$	$\sqrt{-\frac{1}{2\theta}}$	μ^3	$\frac{(y-\mu)^2}{\mu^2 y}$	1	ϕ

Chapter 3

Tweedie exponential dispersion models

3.1 Introduction

Upon closer examination of the summary Table 2.1, it can be noted that all of the described distributions, aside from the binomial and the shifted negative binomial, have a variance function $v(\mu)$ that is a power of the mean. To be more specific,

- if $Y \sim N(\mu, \phi)$ then $v(\mu) = \mu^0 = 1$,
- if $Y \sim Pois(\mu)$ then $v(\mu) = \mu^1 = \mu$,
- if $Y \sim Ga(\mu, \phi)$ then $v(\mu) = \mu^2$,
- if $Y \sim IG(\mu, \phi)$ then $v(\mu) = \mu^3$.

It is possible to define a new sub-family of EDMs characterised by a variance-to-mean power relationship.

Unless stated otherwise, the discussion presented in this chapter follows Dunn & Smyth (2018, Chapter 12), and Jorgensen (1997, Chapter 4).

3.2 Definitions

Let

- $Y \sim ED(\mu, \phi)$ be an EDM with mean μ, dispersion parameter ϕ and variance function $v(\mu)$,
- $\xi \in \mathbb{R} \setminus (0, 1)$ be a constant.

Then, Y is a Tweedie EDM if $v(\mu) = \mu^\xi$, in notation: $Y \sim TW_\xi(\mu, \phi)$. The density function of $TW_\xi(\mu, \phi)$ in the canonical and dispersion model form is

$$p_\xi(y; \mu, \phi) = \widetilde{a}_\xi(y, \phi) \exp\left\{\frac{\theta_\xi(\mu)y - b_\xi(\theta_\xi(\mu))}{\phi}\right\} = \widetilde{c}_\xi(y, \phi) \exp\left\{\frac{-d_\xi(y, \mu)}{2\phi}\right\} \tag{3.1}$$

The form taken by the functions $\theta_\xi(\mu)$, $b_\xi(\theta_\xi(\mu))$, $d_\xi(y, \mu)$ and $\widetilde{a}_\xi(y, \phi)$ will be specified in the following sections.

Since the binomial distribution does not belong to the Tweedie family, the weight m introduced in the definition (1.1) is assumed to equal 1.

3.3 Properties

3.3.1 Different types of Tweedie models and notable cases

For different values of ξ, Tweedie models have different support S, natural parameter space Θ and mean domain Ω. No Tweedie model is defined for $\xi \in (0, 1)$ (Jorgensen, 1997, page 132).

TABLE 3.1: Basic properties of Tweedie EDMs for different values of ξ.

ξ	S	Ω	Θ	Characteristics of the distribution
$\xi < 0$	$\mathbb{R}$	$\mathbb{R}^+$	$\mathbb{R}_{\geq 0}$	Continuous.
$\xi = 0$	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	Continuous, coincides with $N(\mu, \phi)$.
$\xi = 1$	$\{k\phi \mid k \in \mathbb{N}_0\}$	$\mathbb{R}^+$	$\mathbb{R}$	Discrete, corresponds with $\phi Pois(\frac{\mu}{\phi})$. For $\phi = 1$, it coincides with $Pois(\mu)$.
$\xi \in (1, 2)$	$\mathbb{R}_{\geq 0}$	$\mathbb{R}^+$	$\mathbb{R}^-$	Mixed, coincides with $CPG(\mu, \phi, \xi)$.
$\xi = 2$	$\mathbb{R}^+$	$\mathbb{R}^+$	$\mathbb{R}^-$	Continuous, coincides with $Ga(\mu, \phi)$.
$\xi > 2$	$\mathbb{R}^+$	$\mathbb{R}^+$	$\mathbb{R}_{\leq 0}$	Continuous. For $\xi = 3$, it coincides with $IG(\mu, \phi)$.

The notable cases of the normal, Poisson, gamma and inverse Gaussian distributions will not be analysed further: the discussion will primarily focus on the cases $\xi \in (1, 2)$, $\xi = 1$, $\xi > 2$ and, to a lesser extent, on $\xi < 0$.

3.3.2 The parameterisation $\mu_\xi(\theta)$ and $\theta_\xi(\mu)$

Tweedie models' parameterisation functions $\mu_\xi(\theta)$(1.7) and $\theta_\xi(\mu)$(1.8), can be determined starting from the variance-to-mean power relationship that characterises them and by applying the property $\frac{d\theta}{d\mu} = \frac{1}{v(\mu)}$ (Salvan et al., 2020, page 76).

$$\frac{d\mu}{d\theta} = v(\mu_\xi(\theta)) \ , \ v(\mu) = \mu^\xi \iff \mu'_\xi(\theta) = [\mu_\xi(\theta)]^\xi.$$

Setting the arbitrary constant to 0, the function $\mu_\xi(\cdot)$ that satisfies that relationship is

$$\mu_\xi(\theta) = \begin{cases} e^\theta & \text{if } \xi = 1 \\ [(1-\xi)\ \theta]^{\frac{1}{1-\xi}} & \text{if } \xi \neq 1 \end{cases} . \tag{3.2}$$

The parameterisation $\theta_\xi(\mu)$ can be obtained by inverting the function (3.2).

$$\theta_\xi(\mu) = \begin{cases} \log \mu & \text{if } \xi = 1 \\ \dfrac{\mu^{1-\xi}}{1-\xi} & \text{if } \xi \neq 1 \end{cases} . \tag{3.3}$$

3.3.3 The functions $b_\xi(\theta)$ and $K_\xi(t;\theta,\phi)$

The function $b_\xi(\theta)$ can be determined by substituting $\mu_\xi(\theta)$ in (1.4).

$$\frac{db_\xi(\theta)}{d\theta} = \mu_\xi(\theta) \Rightarrow b_\xi(\theta) = \int \mu_\xi(\theta) d\theta = \begin{cases} e^\theta & \text{if } \xi = 1 \\ -\log(-\theta) & \text{if } \xi = 2 \\ \dfrac{(1-\xi)^{\frac{2-\xi}{1-\xi}}}{2-\xi}\ \theta^{\frac{2-\xi}{1-\xi}} & \text{if } \xi \neq 1, 2 \end{cases} . \tag{3.4}$$

The function $b_\xi(\theta_\xi(\mu))$ takes a simpler form than $b_\xi(\theta)$ and can be useful in some contexts.

$$b_\xi(\theta_\xi(\mu)) = \begin{cases} \mu & \text{if } \xi = 1 \\ -\log(\frac{1}{\mu}) & \text{if } \xi = 2 \\ \dfrac{\mu^{2-\xi}}{2-\xi} & \text{if } \xi \neq 1, 2 \end{cases} . \tag{3.5}$$

Having obtained $b_\xi(\theta)$, the cumulant generating function of $TW_\xi(\mu, \phi)$, $K_\xi(t; \theta, \phi)$, can be derived by substituting (3.4) into (1.3).

$$K_\xi(t;\theta,\phi) = \begin{cases} \frac{e^\theta}{\phi}\ (e^{t\phi} - 1) & \text{if } \xi = 1 \\ -\frac{1}{\phi}\log(1 + \frac{t\phi}{\theta}) & \text{if } \xi = 2 \\ \frac{1}{\phi} b_\xi(\theta)[(1 + \frac{t\phi}{\theta})^{\frac{2-\xi}{1-\xi}} - 1] & \text{if } \xi \neq 1, 2 \end{cases} . \tag{3.6}$$

3.3.4 The unit deviance

Let $Y \sim TW_\xi(\mu, \phi)$, then the unit deviance of Y is

$$d_\xi(y,\mu) = \begin{cases} (y-\mu)^2 & \text{if } \xi = 0 \\ 2[y\log(\frac{y}{\mu}) - (y-\mu)] & \text{if } \xi = 1 \\ 2[y\log(\frac{\mu}{y}) + \frac{y}{\mu} - 1] & \text{if } \xi = 2 \\ 2[\frac{[\max\{y,0\}]^{2-\xi}}{(1-\xi)(2-\xi)} - \frac{y\mu^{1-\xi}}{1-\xi} + \frac{\mu^{2-\xi}}{2-\xi}] & \text{if } \xi \neq 0, 1, 2 \end{cases} . \tag{3.7}$$

For details on the proof for the cases $\xi \neq 0, 1, 2$, see Jorgensen (1997, page 133).

3.3.5 Scale transformations

It is a well-known fact that the normal and gamma distributions are closed under scale transformations. For those distributions, given a positive constant c, the following rescaling formulas apply (Jorgensen, 1997, page 128):

$$cN(\mu, \phi) = N(c\mu, c^2\phi)$$

$$cGa(\mu, \phi) = Ga(c\mu, \phi).$$

It can be proven that the Tweedie family as a whole is closed under scale transformations and that Tweedie models are the only EDMs that possess this property (Jorgensen, 1997, page 128). In other words,

- All Tweedie models are closed under scale transformations,
- All EDMs that are closed under scale transformations are Tweedie models.

There exists a generalised rescaling identity that applies to all Tweedie EDMs:

$$cTW_\xi(\mu, \phi) = TW_\xi(c\mu, c^{2-\xi}\phi). \tag{3.8}$$

For the sake of completeness, we will give the proof of (3.8) for $\xi = 1$ in Section 3.6 and for $\xi \in (1, 2)$ in Section 3.7.3. The case $\xi = 2$ will not be considered since we assume to know already that the gamma distribution is closed under scale transformations. Instead, the proof for the cases $\xi > 2$ and $\xi \leq 0$ is identical to the one for $\xi \in (1, 2)$ with $y > 0$.

Dunn & Smyth (2018, page 461) identified a useful relationship regarding the density function of a Tweedie variable and the one of its scale transformation: let $Y \sim TW_\xi(\mu, \phi)$ having density function $p_\xi(y; \mu, \phi)$ and c be a constant. Then

$$p_\xi(y; \mu, \phi) = c p_\xi(cy; c\mu, c^{2-\xi}\phi). \tag{3.9}$$

3.3.6 The Tweedie convergence theorem

Following Jorgensen (1997, pp. 148-149), it can be proven that exponential dispersion models having variance functions that behave asymptotically as powers of the mean, for large or small values of the expected value, converge in distribution to Tweedie models. Let

- $Y \sim ED(\mu, \phi)$, having variance function $v(\mu)$ such that $v(\mu) \sim c_0\mu^\xi$ when $\mu \to \infty$ or $\mu \to 0$, for $\xi \in \mathbb{R} \setminus (0, 1)$ and $c_0 \in \mathbb{R}^+$,
- $c \in \mathbb{R}^+$ be a constant such that $c\mu \in \Omega$, where Ω is the parameter space of μ.

Then

$$c^{-1}ED(c\mu, \phi c^{2-\xi}) \xrightarrow{d} TW_\xi(\mu, c_0\phi), \text{ as } c \to \infty \text{ or } c \to 0. \tag{3.10}$$

Numerous EDMs have variance functions that verify the assumptions of the Tweedie convergence theorem (see Table 2.1). Furthermore, most of the distributions commonly used in statistical modelling belong to the exponential dispersion family: this implies that Tweedie EDMs act as a point of convergence for plenty of models.

Practical applications of the Tweedie convergence theorem will be discussed in more detail in Section 3.7.5.

3.3.7 Link with Box-Cox transformations

In Section 1.1 it was shown that not all response variables Y are suitable to be modelled as realisations of the normal distribution, as not all data have support $\mathbb{R}$, a symmetric distribution and constant variance. Consequently, a new class of models was introduced in Section 2.2 to perform statistical modelling on non-normally distributed data.

An alternative, but less effective, approach is transforming the non-normally distributed response variable Y through a suitable function $h(Y)$, such that the transformed variable $Y^* = h(Y)$ approximately fits normality assumptions. Following this transformation, the normal regression model can be fit using Y^* as a response.

Power transformations having exponent $\lambda < 1$ or the logarithmic transformation $Y^* = \log(Y)$ are frequently employed to stabilise variances that increase with the mean or to symmetrise right-skewed distributions (Dunn & Smyth, 2018, pages 117-118).

Instead, power transformations having an exponent $\lambda > 1$ are used to stabilise variances that decrease with the mean or to symmetrise left-skewed distributions (Dunn & Smyth, 2018, pages 117-118). However, this form of asymmetry is less common than the previous one.

The further the exponent of the power transformation is from $\lambda = 1$, the more pronounced the effect of the transformation becomes.

The family of Box-Cox transformations (Box & Cox, 1964) generalises power transformations, for various values λ of the exponent, and the logarithmic transformation. Using the notation introduced in the previous paragraph, the Box-Cox family of transformations of the variable Y, parameterised by λ, are defined as

$$Y^* = h_\lambda(Y) = \begin{cases} \dfrac{Y^\lambda - 1}{\lambda} & \text{if} \quad \lambda \neq 0 \\ \log(Y) & \text{if} \quad \lambda = 0 \end{cases} \tag{3.11}$$

The numerical results for $\lambda \neq 0$ differ from the ones obtained using simple power transformations such as Y^λ, but the stabilising effect on the variance of the response remains consistent. Furthermore, this formulation has the advantage of being continuous for $\lambda \in \mathbb{R}$, since $(Y^\lambda - 1)/\lambda \to \log(Y)$ as $\lambda \to 0$.

Following Dunn & Smyth (2018, page 232), it can be demonstrated that there is a link between GLMs and scale-stabilising functions. Moreover, a unique correspondence with Tweedie models arises when the scale stabilising functions coincide with Box-Cox transformations. Let

- $Y \sim ED(\mu, \phi)$, having variance $Var[Y] = \phi v(\mu)$ and support S,
- $h : S \to \mathbb{R}$ be a differentiable function,
- $Y^* = h(Y)$ be the transformed variable through the function $h(\cdot)$, having variance $Var[Y^*]$.

Applying Taylor's expansion arrested to the first order of the function $h(Y)$ in $Y = \mu$, we obtain

$$Y^* = h(Y) \approx h(\mu) + h'(\mu)(Y - \mu)$$

Consequently, the variance of the transformed variable can be approximated as

$$Var[Y^*] \approx [h'(\mu)]^2]Var[Y] = [h'(\mu)]^2 \phi v(\mu)$$

This result suggests that $Var[Y^*]$ is roughly constant with respect to μ (in other words, it is stabilised) if $v(\mu) \propto [h'(\mu)]^{-2}$. Thus, fitting a normal regression model to the transformed variable $Y^* = h(Y)$ is approximately equivalent to fitting a GLM to Y with variance function $v(\mu) = [h'(\mu)]^{-2}$ and link function $\eta = h(\mu)$, where η is the linear predictor of the model. Therefore, if $h(Y)$ coincides with the Box-Cox transformations (3.11), the corresponding variance function is

$$v(\mu) = [h'(\mu)]^{-2} = \begin{cases} \mu^{2(1-\lambda)} & \text{if} \quad \lambda \neq 0 \\ \mu^2 & \text{if} \quad \lambda = 0 \end{cases}, \tag{3.12}$$

so that, in all cases, $v(\mu) = \mu^{2(1-\lambda)}$. The variance functions (3.12) are powers of the mean: this mean-variance relationship characterises Tweedie models. This implies that fitting a normal regression model having as a response variable the Box-Cox transformation, parameterised by λ, of Y, is approximately equivalent to fitting the Tweedie GLM on Y having power parameter $\xi = 2(1 - \lambda)$.

TABLE 3.2: Unique correspondence between Tweedie models and Box-Cox transformations.

Tweedie model	Transformation
$\xi < 0$	$Y^* = Y^\lambda$ with $\lambda > 1$
Normal: $\xi = 0$	$Y^* = Y$
Reproductive Poisson: $\xi = 1$	$Y^* = \sqrt{Y}$
Compound Poisson-gamma: $\xi \in (1, 2)$	$Y^* = Y^\lambda$ with $\lambda \in (0, \frac{1}{2})$
Gamma: $\xi = 2$	$Y^* = \log(Y)$ with $\lambda = 0$
$\xi > 2$	$Y^* = Y^\lambda$ with $\lambda < 0$
Inverse Gaussian: $\xi = 3$	$Y^* = 1/\sqrt{Y}$

The normal regression model on the Box-Cox transformed variable performs poorly in the presence of exact zeros (Dunn & Smyth, 2018, page 465). Consequently, this approach is recommended primarily for $\xi > 2$. Some comparisons of the two approaches on real data will be shown in Section 3.8.1.

3.4 The density function

Aside from the notable cases of the normal, Poisson, gamma and inverse Gaussian distributions, the density function of Tweedie EDMs does not have an explicit analytic expression. This complicates the use of these distributions in statistical modelling.

Even though the function $b_\xi(\theta)$ (3.4) and the unit deviance $d_\xi(y, \mu)$ (3.7) take simple forms, the functions $\widetilde{a}_\xi(y, \phi)$, for the canonical form of the density function (1.1), and $\widetilde{c}_\xi(y, \phi)$, for the dispersion form of the density function (1.11), cannot be written in closed form. In other words, they cannot be expressed using elementary functions, or combinations of them, through a finite number of arithmetic operations or function compositions. The functions $\widetilde{a}_\xi(y, \phi)$ and $\widetilde{c}_\xi(y, \phi)$ will be expressed as an infinite series, whose explicit sum is unavailable.

3.4.1 The saddlepoint approximation

Since the unit deviance has an explicit expression for all values of ξ, the saddlepoint approximation (1.12) can be applied. Let $Y \sim TW_\xi(\mu, \phi)$ with density function $p_\xi(y; \mu, \phi)$ (3.1). Then, the saddlepoint approximation (1.12) for $p_\xi(y; \mu, \phi)$ is

$$\widetilde{p}_\xi(y; \mu, \phi) = \frac{1}{\sqrt{2\pi\phi\ y^\xi}} \exp\left\{-\frac{d_\xi(y, u)}{2\phi}\right\} \to p_\xi(y; \mu, \phi) \quad \text{as } \phi \to 0 \tag{3.13}$$

The saddlpoint approximation for Tweedie models can be computed in R by the function `dtweedie.saddle` from the package `tweedie` (Dunn, 2017). However, this approximation performs poorly in data sets with exact zeros (Dunn & Smyth, 2001). Several more refined methods have been developed to evaluate the density function of Tweedie models more accurately. Each of these strategies performs better in different regions of the parameter space of ξ.

3.4.2 Infinite series expansion

As anticipated before, the functions $\widetilde{a}_\xi(y;\phi)$ of Tweedie densities can be expressed through an infinite series expansion (Dunn & Smyth, 2005):

$$\widetilde{a}_\xi(y;\phi) = \begin{cases} \frac{W(y;\phi,\xi)}{y} = \sum_{j=1}^{\infty} \frac{1}{y} W_j = \frac{1}{y} \sum_{j=1}^{\infty} \frac{(\xi-1)^{\alpha j} \phi^{j(\alpha-1)}}{j! y^{j\alpha} (2-\xi)^j \Gamma(-\alpha j)}, & \text{if } \xi \in (1,2) \\ \frac{V(y;\phi,\xi)}{\pi y} = \sum_{j=1}^{\infty} \frac{1}{\pi y} V_j = \frac{1}{\pi y} \sum_{j=1}^{\infty} \frac{\Gamma(1+\alpha j) \phi^{j(\alpha-1)} (\xi-1)^{\alpha j}}{\Gamma(1+j)(\xi-2)^j y^{\alpha j}} (-1)^j \sin(-j\pi\alpha), & \text{if } \xi > 2 \end{cases} \tag{3.14}$$

where $\alpha = \frac{\xi-2}{\xi-1}$. The formula for $\xi \in (1,2)$ will be proven in Section 3.7, while the expression for $\xi > 2$ arises from Fourier inversion of the characteristic function, after applying a Taylor series expansion (Dunn & Smyth, 2008). The case $\xi < 0$ is omitted due to its limited practical applications. However, further details may be found in Jorgensen (1997, page 137).

The most obvious approach to approximate the density function of Tweedie EDMs is to estimate the values that the series (3.14) take on. However, adding a large number of terms, starting from $j = 1$, until a chosen upper limit is not an advisable strategy to evaluate those infinite summations. Many of the terms take values close to zero: thus, they do not significantly contribute to the sum.

Dunn & Smyth (2005) propose an efficient strategy to approximate the series. To identify the most significant summands in terms of the index j, W_j and V_j are treated as continuous functions of j and are differentiated with respect to j, to determine local maxima. The terms of the series that are close, in both directions, to the local maxima are added until their contributions become negligible.

For $\xi \in (1,2)$, the terms of the series are always positive: thus, the summation converges rapidly. Dunn & Smyth (2005) demonstrate that, as the value of ξ approaches 2 and y increases, the number of terms necessary to achieve machine accuracy also rises. The level of accuracy does not depend on the magnitude of ϕ.

For $\xi > 2$, the terms of the series alternate sign and cancel each other out when added, hence a larger number of terms is required. To test the accuracy of the approximations, Dunn & Smyth (2005) apply their methods to the Tweedie models whose densities can be expressed in closed form, i.e. the inverse Gaussian distribution, which corresponds to $\xi = 3$. Then they compare the approximated version with the explicit expression to determine the accuracy. For the inverse Gaussian distribution, if $y > 0$, it is possible to achieve machine accuracy even with a relatively small number of terms. Instead, if y is close to 0, the relative error increases without bounds.

The denisty of Tweedie models evaluated utilising the series method can be computed in R by the function `dtweedie.series` from the package `tweedie` (Dunn, 2017).

3.4.3 Inversion of the cumulant generating function

Given a random variable, it is possible to express its density function as an integral of its characteristic function multiplied by the complex exponential transformation of the random variable.

Let Y be a continuous random variable having density function $p_Y(y)$ and characteristic function $C_Y(t) = E[e^{itY}]$ absolutely integrable. Then, Fourier's inversion theorem states (Pace & Salvan, 1997, page 76)

$$p_Y(y) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-ity} C_Y(t) dt \tag{3.15}$$

If the characteristic function of Y possesses an analytic expression while the density function does not, it is possible to evaluate the density function $p_Y(y)$ by solving the integral (3.15) numerically. The Formula (3.15) is only applicable to continuous random variables but, for $\xi \in (1, 2)$, Tweedie models exhibit mixed distributions. Thus, an adjustment in the approach is necessary: instead of considering the density of Y, Dunn & Smyth (2008) evaluate the density of $Y|Y > 0$, which is continuous.

Since $b_\xi(\theta)$ an $d_\xi(y, \mu)$ are known, the functions $\widetilde{a}_\xi(y, \phi)$ and $\widetilde{c}_\xi(y, \phi)$ are the object of the approximation. Consequently, there is no need to compute the integral (3.15) for all values of μ, since $\widetilde{a}_\xi(y, \phi)$ and $\widetilde{c}_\xi(y, \phi)$ can be derived from the Tweedie density function by putting $\mu = 1$ and $y = \mu$ respectively. Indeed, $p_\xi(y; 1, \phi) = \widetilde{a}_\xi(y, \phi)$ and $p_\xi(y; y, \phi) = \widetilde{c}_\xi(y, \phi)$.

Dunn & Smyth (2008) propose three different methodologies in increasing order of sophistication to compute the integral:

- The first method computes $\widetilde{a}_\xi(y, \phi)$ as $p_\xi(y; 1, \phi)$ by Fourier inversion and then substitutes it in (1.1).
- The second method uses the Tweedie rescaling identity (3.9) with $c = \frac{1}{\mu}$ to rescale μ to 1. The density function is obtained as

$$p_\xi(y; \mu, \phi) = \frac{1}{\mu} p_\xi(\frac{y}{\mu}; 1, \frac{\phi}{\mu^{2-\xi}}) = \widetilde{a}(\frac{y}{\mu}, \frac{\phi}{\mu^{2-\xi}}).$$

- The third method uses the Tweedie rescaling identity (3.9) with $c = \frac{1}{y}$ to rescale y to 1. The normalising function is obtained as

$$\widetilde{c}_\xi(y, \phi) = p_\xi(y; y, \phi) = \frac{f(1, 1, \frac{\phi}{y^{2-\xi}})}{y} = \frac{\widetilde{a}_\xi(1, \frac{\phi}{y^{2-\xi}})}{y}.$$

Dunn & Smyth (2008) show that, even though the three strategies are arithmetically equivalent, the third one is the most numerically accurate. Unlike the other two methods, both y and μ are fixed to 1, thus Fourier inversion is applied only for $y = \mu = 1$. Additionally, the relative error of the numerical integration is smaller: since $y = \mu$, the integral is calculated at the mean of the distribution, which is close to the mode, especially if the distribution is not exceptionally skewed. Since the calculation error in numerical integration is roughly constant, integrals with larger values have a much smaller relative error.

The denisty of Tweedie models evaluated utilising the inversion method can be computed in R by the function `dtweedie.inversion` from the package `tweedie` (Dunn, 2017).

3.5 Tweedie EDMs with $\xi < 0$

Tweedie models with $\xi < 0$ are continuous and can take on all real values.

This subset of Tweedie EDMs features a non-steep $b_\xi(\theta)$ function: consider an EDM having canonical parameter $\theta \in \Theta$. Let $\partial\Theta$ and $\overset{\circ}{\Theta}$ denote the boundary and the interior of Θ, respectively. Then, the function $b(\theta)$ is defined as steep if $(\widetilde{\theta} - \theta)\, b'(\alpha\theta + (1-\alpha)\widetilde{\theta}) \to \infty$ as $\alpha \to 0$, $\forall\, \widetilde{\theta} \in \partial\Theta$, $\theta \in \overset{\circ}{\Theta}$ (Wüthrich & Merz, 2023, page 37).

Steepness of the $b(\theta)$ function is a necessary and sufficient condition to verify the Banforff-Nielsen theorem (Wüthrich & Merz, 2023, page 37): this implies that Tweedie models with $\xi < 0$ are among the few EDMs where the parameter space of the mean Ω does not coincide with the interior of the convex hull of the support S: indeed, as shown in Table 2.1, $\Omega = \mathbb{R}^+$, while $S = \mathbb{R}$.

Because of their unusual characteristics, these models have limited real-world applications.

3.6 Tweedie EDMs with $\xi = 1$

The distribution $Y \sim TW_1(\mu, \phi)$ has support $S = \{k\phi \mid k \in \mathbb{N}_{\geq 0}\} = \{0, \phi, 2\phi ...\}$ and mean parameter space $\Omega = \mathbb{R}^+$.

It is possible to demonstrate that $Y \sim TW_1(\mu, \phi)$ coincides with $\phi Pois(\mu/\phi)$ and that its density function is

$$p_Y(y; \mu, \phi) = p_1(y; \mu, \phi) = \frac{e^{-\frac{\mu}{\phi}} \ \frac{\mu}{\phi}^{\frac{y}{\phi}}}{(\frac{y}{\phi})!}. \tag{3.16}$$

This distribution corresponds with the reproductive form of the Poisson distribution defined by Jorgensen (1997, page 72), while the conventional Poisson distribution coincides with the special case where ϕ is fixed to 1. In notation: $TW_1(\mu, 1) = Pois(\mu)$. For the sake of completeness, we give the proof of (3.16).

Proof. In the first instance, let us determine the probability function of $\phi Pois(\mu/\phi)$. Let $Y = \phi W$, where $W \sim Pois(\mu/\phi)$:

$$p_Y(y; \mu, \phi) = P_{\mu,\phi}(Y = y) = P_{\mu,\phi}(\phi W = y) = P_{\mu,\phi}(W = \frac{y}{\phi}) = \frac{e^{-\frac{\mu}{\phi}} \ \frac{\mu}{\phi}^{\frac{y}{\phi}}}{(\frac{y}{\phi})!}.$$

To demonstrate that $\phi Pois(\mu/\phi) = TW_1(\mu, \phi)$, we need to show that (3.16) can be rearranged as the density function of a Tweedie EDM (3.1), having $\xi = 1$. With some algebraic steps, we obtain

$$p_Y(y; \mu, \phi) = \frac{e^{-\frac{\mu}{\phi}} \ \frac{\mu}{\phi}^{\frac{y}{\phi}}}{(\frac{y}{\phi})!} = (\frac{1}{\phi})^{\frac{y}{\phi}} \frac{e^{\frac{-\mu}{\phi}} \mu^{\frac{y}{\phi}}}{(\frac{y}{\phi})!} = \frac{\phi^{\frac{-y}{\phi}}}{(\frac{y}{\phi})!} \ e^{\frac{-\mu}{\phi}} e^{\log\left\{\mu^{\frac{y}{\phi}}\right\}} = \frac{\phi^{\frac{-y}{\phi}}}{(\frac{y}{\phi})!} \exp\{\frac{\log(\mu)y}{\phi} - \frac{\mu}{\phi}\} =$$

$$= \frac{\phi^{\frac{-y}{\phi}}}{(\frac{y}{\phi})!} \exp\left\{\frac{\log(\mu)y - \mu}{\phi}\right\} = \widetilde{a}_1(y, \phi) \exp\left\{\frac{\theta_1(\mu)y - b_1(\theta_1(\mu))}{\phi}\right\} = p_1(y; \mu, \phi).$$

In this specific case, the function $\widetilde{a}_\xi(y, \phi)$ possesses an explicit analytic expression. □

As anticipated in Section 3.3.5, $\phi Pois(\mu/\phi)$ is closed under scale transformations. Let $Y \sim TW_1(\mu, \phi)$ and $Z = cY$, where c is a positive constant: then, $Z \sim TW_1(c\mu, c\phi)$ (Jorgensen, 1997, page 128).

Proof. To demonstrate the aforementioned property, we can show that the density of $cTW_1(\mu, \phi)$ can be rewritten as the density of $TW_1(c\mu, c\phi)$:

$$P_{\mu,\phi}(Z = z) = P_{\mu,\phi}(cY = z) = P_{\mu,\phi}(Y = \frac{z}{c}) = \frac{e^{-\frac{\mu}{\phi}} \ \frac{\mu}{\phi}^{\frac{z}{c\phi}}}{(\frac{z}{c\phi})!} = \frac{e^{-\frac{c\mu}{c\phi}} \ \frac{c\mu}{c\phi}^{\frac{z}{c\phi}}}{(\frac{z}{c\phi})!} = p_1(z, c\mu, c\phi).$$

□

It should be noted that the Poisson distribution, even though it belongs to the Tweedie family, is not closed under scale transformations: in fact $cPois(\mu) = cTW_1(\mu, 1) = TW_1(c\mu, c)$ is not a Poisson, since $\phi \neq 1$.

3.7 Tweedie EDMs with $\xi \in (1, 2)$

The distributions $TW_\xi(\mu, \phi)$ with $\xi \in (1, 2)$ have support $S = [0, +\infty)$, indicating that they can take value on all non-negative real numbers. For $Y > 0$, this sub-family of Tweedie EDMs behaves like a continuous distribution. However, Tweedie models with $\xi \in (1, 2)$ have a non-negative probability of taking a value of 0, resembling a discrete distribution. Random variables characterised by this dual nature are said to have a mixed distribution.

3.7.1 The link with the compound Poisson-gamma distribution

Consider

- $N \sim Pois(\lambda)$, where $\lambda \in \mathbb{R}^+$.
- N independent and identically distributed random variables $X_i \sim Ga^*(\alpha, \beta)$, for $i = 1, ...N$, where $\alpha \in \mathbb{R}^+$ is the shape parameter and $\beta \in \mathbb{R}^+$ is the scale parameter. The asterisk is used to distinguish this parameterisation from the one used previously with mean and dispersion parameter.
- A random variable Z defined as $Z = \begin{cases} \sum_{i=1}^{N} X_i & \text{if} \quad N \geq 1 \\ 0 & \text{if} \quad N = 0 \end{cases}$.

Then, the random variable Z follows a compound Poisson-gamma distribution, also known as Tweedie's compound Poisson or Poisson-gamma, with parameters (λ, α, β) (Jorgensen, 1997, page 140). In notation: $Z \sim CPG(\lambda, \alpha, \beta)$.

The CPG distribution can also be represented as a hierarchical model:

- fix a value $\widetilde{\lambda}$ for the parameter λ,
- generate a value n from the distribution $N \sim Pois(\widetilde{\lambda})$,
- fix two values $\widetilde{\alpha}$ and $\widetilde{\beta}$ for the two parameters of the gamma distribution α and β,
- generate a value z from the conditional distribution

$$Z|n = \begin{cases} \sum_{i=1}^{n} X_i & \text{if} \quad n \geq 1 \\ 0 & \text{if} \quad n = 0 \end{cases} \sim \begin{cases} Ga^*(n\widetilde{\alpha}, \widetilde{\beta}) & \text{if} \quad n \geq 1 \\ 0 & \text{if} \quad n = 0 \end{cases}.$$

Then, z is a realisation of a $CPG(\widetilde{\lambda}, \widetilde{\alpha}, \widetilde{\beta})$.

By comparing the moment generating functions of $Y \sim TW_\xi(\mu, \phi)$ with $\xi \in (1,2)$ and $Z \sim CPG(\lambda, \alpha, \beta)$, it can be proven that the two distributions coincide through the following mapping of the parameters (Jorgensen, 1997, page 140)

$$(\mu, \xi, \phi) \mapsto (\lambda, \alpha, \beta) = \begin{cases} \lambda = \dfrac{1}{\phi} \dfrac{\mu^{2-\xi}}{2-\xi} = \dfrac{b_\xi(\theta_\xi(\mu))}{\phi} \\ \alpha = \dfrac{2-\xi}{\xi-1} \\ \beta = -\dfrac{1}{\phi} \dfrac{\mu^{1-\xi}}{1-\xi} = -\dfrac{\theta_\xi(\mu)}{\phi} \end{cases} . \tag{3.17}$$

The alternative parameterisation with θ instead of μ takes on a simpler form:

$$(\theta, \xi, \phi) \mapsto (\lambda, \alpha, \beta) = \begin{cases} \lambda = \dfrac{b_\xi(\theta)}{\phi} \\ \alpha = \dfrac{2-\xi}{\xi-1} \\ \beta = -\dfrac{\theta}{\phi} \end{cases} . \tag{3.18}$$

Since the gamma shape parameter α must be positive, the representation of Tweedie models as a Poisson mixture of gamma random variables is valid only for $1 < \xi < 2$ (Smyth, 1996).

3.7.2 Density function

Following Jorgensen (1997, page 141), to determine the marginal density of Z, we first need to calculate the joint probability density $p_{Z,N}(z, n; \lambda, \alpha, \beta)$. Let $p_N(n; \lambda)$ be the marginal probability function of $N \sim Pois(\lambda)$ evaluated at $N = n$ and $p_{Z|N}(z|n; \alpha, \beta)$ be the conditional density function of $Z \sim Ga^*(n\alpha, \beta)$ evaluated at $Z = z > 0$, given $N = n$. Then, by applying the factorisation rule, we obtain

$$p_{Z,N}(z, n; \lambda, \alpha, \beta) = p_{Z|N}(z|n; \alpha, \beta)\ p_N(n; \lambda) = \begin{cases} \dfrac{\beta^{n\alpha} z^{n\alpha-1} e^{-\beta z}}{\Gamma(n\alpha)} \dfrac{e^{-\lambda}\lambda^n}{n!} & \text{if} \quad N \geq 1 \\ 1 \cdot e^{-\lambda} & \text{if} \quad N = 0 \end{cases}$$

$$= \begin{cases} \dfrac{\beta^{n\alpha}\ z^{n\alpha}\ \lambda^n}{z\ \Gamma(n\alpha)\ n!} \exp\{-\beta z - \lambda\} & \text{if} \quad N \geq 1 \\ e^{-\lambda} & \text{if} \quad N = 0 \end{cases} .$$

The marginal density $p_Z(z;\lambda,\alpha,\beta)$ can be determined using the total probability law. By substituting the parameters (λ,α,β) of the CPG distribution with the parameters (θ,ξ,ϕ) through the parameterisation (3.18), we obtain the density of a Tweedie EDM with $\xi\in(1,2)$, written in canonical form (1.1):

$$p_Z(z;\lambda,\alpha,\beta)=\sum_{n\in\mathbb{N}}p_{Z,N}(z,n;\lambda,\alpha,\beta)=\sum_{n\in\mathbb{N}}p_{Z,N}\left(z,n;\frac{b_\xi(\theta)}{\phi},\frac{2-\xi}{\xi-1},-\frac{\theta}{\phi}\right)$$

$$=\begin{cases}\frac{1}{z}\sum\limits_{n=1}^{+\infty}\frac{(\frac{-\theta\ z}{\phi})^{n\alpha}(\frac{b_\xi(\theta)}{\phi})^n}{\Gamma(n\alpha)\ n!}\exp\left\{\frac{\theta z-b_\xi(\theta)}{\phi}\right\} & \text{if}\quad N\geq 1\\ e^{-\lambda}=\exp\left\{-\frac{\mu^{2-\xi}}{\phi\ (2-\xi)}\right\} & \text{if}\quad N=0\end{cases},$$

where the normalising function is

$$\widetilde{a}_\xi(y,\phi)=\frac{1}{z}\sum_{n=1}^{+\infty}\frac{(\frac{-\theta\ z}{\phi})^{n\alpha}(\frac{b_\xi(\theta)}{\phi})^n}{\Gamma(n\alpha)\ n!}.$$

The probability of observing exact zeros is

$$P(Z=0)=e^{-\lambda}=\exp\left\{-\frac{\mu^{2-\xi}}{\phi\ (2-\xi)}\right\}.$$

3.7.3 Properties

The correspondence with the compound Poisson-gamma distribution can be utilised to derive the mean and the variance of Tweedie EDMs with $\xi\in(1,2)$. Let $N\sim Pois(\lambda)$ and $Z|N=n\sim Ga^*(n\alpha,\beta)$. Then the mean of $Z\sim CPG(\lambda,\alpha,\beta)$ can be determined by applying the law of total expectation:

$$E[Z]=E[E[Z|N]]=E[N\frac{\alpha}{\beta}]=\frac{\alpha}{\beta}E[N]=\lambda\alpha/\beta.$$

Instead, by applying the law of total variance, we obtain

$$\begin{aligned}\mathrm{Var}[Z]&=\mathrm{Var}[E[Z\mid N]]+E[\mathrm{Var}[Z\mid N]]=\mathrm{Var}\left[N\frac{\alpha}{\beta}\right]+E\left[N\frac{\alpha}{\beta^2}\right]\\&=\frac{\alpha^2}{\beta^2}\mathrm{Var}[N]+\frac{\alpha}{\beta^2}E[N]=\frac{\alpha^2}{\beta^2}\lambda+\frac{\alpha}{\beta^2}\lambda=\frac{\alpha\lambda}{\beta^2}(\alpha+1).\end{aligned}$$

As anticipated in Section 3.3.5, we provide below the proof of the Tweedie rescaling identity (3.8) for $\xi\in(1,2)$.

Proof. Let $Y \sim TW_\xi(\mu,\phi)$, having density function $p_\xi(y;\mu,\phi)$, and let $Z = cY$, having density function $p_Z(z;\mu,\phi,\xi)$. To prove the identity (3.8), without assuming closure under scale transformations, we can show that the density function $p_Z(z;\mu,\phi,\xi)$ of the scale transformed variable Z can be rewritten as the density function of a Tweedie variable, with power parameter ξ, mean $c\mu$ and dispersion parameter $c^{2-\xi}\phi$. In this initial part of the proof, we also suppose that $y > 0$ (or, equivalently, that $z > 0$). This ensures that $Y \sim TW_\xi(\mu,\phi)$ with $\xi \in (1,2)$ behaves like a continuous random variable. By applying the change of variable formula for continuous distributions, we obtain:

$$p_Z(z;\mu,\phi,\xi) = \frac{1}{c}p_\xi(\frac{z}{c};\mu,\phi) = \frac{1}{c}\widetilde{a}_\xi(\frac{z}{c},\phi)\exp\left\{\frac{\theta_\xi(\mu)\frac{z}{c} - b_\xi(\theta_\xi(\mu))}{\phi}\right\}.$$

We substitute the formulas (3.3) and (3.5) for the case $\xi \neq 1,2$ to $\theta_\xi(\mu)$ and $b_\xi(\theta_\xi(\mu))$ respectively. Hence,

$$p_Z(z;\mu,\phi,\xi) = \frac{\widetilde{a}_\xi(\frac{z}{c},\phi)}{c}\exp\left\{\frac{1}{\phi}\left\{\frac{\frac{z}{c}\mu^{1-\xi}}{1-\xi} + \frac{\mu^{2-\xi}}{2-\xi}\right\}\right\}.$$

And, with few algebraic steps,

$$p_Z(z;\mu,\phi,\xi) = \frac{\widetilde{a}_\xi(\frac{z}{c},\phi)}{c}\exp\left\{\frac{1}{\phi}\frac{c^{2-\xi}}{c^{2-\xi}}\left\{\frac{\frac{z}{c}\mu^{1-\xi}}{1-\xi} + \frac{\mu^{2-\xi}}{2-\xi}\right\}\right\}$$

$$= \frac{\widetilde{a}_\xi(\frac{z}{c},\phi)}{c}\exp\left\{\frac{1}{c^{2-\xi}\phi}\left\{\frac{z(c\mu)^{1-\xi}}{1-\xi} + \frac{(c\mu)^{2-\xi}}{2-\xi}\right\}\right\}.$$

Except for the function $\dfrac{\widetilde{a}_\xi(\frac{z}{c},\phi)}{c}$, this is the density of $TW_\xi(c\mu, c^{2-\xi}\phi)$, for $\xi \neq 1,2$. The normalising function requires special care because it does not possess an explicit analytic expression: to complete the proof, we need to show that $\dfrac{\widetilde{a}_\xi(\frac{z}{c},\phi)}{c} = \widetilde{a}_\xi(z, c^{2-\xi}\phi)$. By taking a closer look at the formulas of the normalising function for different values of ξ (3.14), it can be noted that $\widetilde{a}_\xi(y,\phi)$ depends on y and ϕ only through the quantity inside the summation $\dfrac{\phi^{j(\alpha-1)}}{y^{\alpha j+1}}$, where j is the index of the summation and $\alpha = \dfrac{\xi-2}{\xi-1}$. Therefore, to simplify calculations, we will work only on this factor instead of considering the whole function:

$$\frac{1}{c}\frac{\phi^{j(\alpha-1)}}{(\frac{z}{c})^{\alpha j+1}} = \frac{1}{c}\frac{\phi^{j(\alpha-1)}c^{\alpha j+1}}{z^{\alpha j+1}} = \frac{\phi^{j(\alpha-1)}c^{\alpha j}}{z^{\alpha j+1}} = \frac{(\phi c^{\frac{\alpha}{\alpha-1}})^{j(\alpha-1)}}{z^{\alpha j+1}}$$

where $\frac{\alpha}{\alpha - 1} = 2 - \xi$, which concludes our proof for $y > 0$.

Since it was assumed that $y > 0$ while the support of $TW_\xi(\mu, \phi)$ with $\xi \in (1, 2)$ is $\mathbb{R}_{\geq 0}$, we still need to prove that $Y \sim TW_\xi(\mu, \phi)$ with $\xi \in (1, 2)$ follows the identity (3.8) even when $y = 0$: for this value of its support, Tweedie models with $\xi \in (1, 2)$ behave like a discrete random variable. By applying the results of Section 3.7.2:

$$P(Z = 0) = P(cY = 0) = P(Y = 0) = \exp\left\{-\frac{\mu^{2-\xi}}{\phi\ (2 - \xi)}\right\} = \exp\left\{-\frac{c^{2-\xi}}{c^{2-\xi}}\frac{\mu^{2-\xi}}{\phi\ (2 - \xi)}\right\}$$

$$= \exp\left\{-\frac{(c\mu)^{2-\xi}}{c^{2-\xi}\phi\ (2 - \xi)}\right\}.$$

□

If we wanted to demonstrate closure with respect to scale transformations for the cases $\xi > 2$ and $\xi \leq 0$, we should follow an identical approach to that adopted for the case $\xi \in (1, 2)$ with $y > 0$, as the distribution is continuous for those values of the power parameter. Therefore, there is no need to distinguish between different values in the support.

3.7.4 Type of data modelled

$TW_\xi(\mu, \phi)$ with $\xi \in (1, 2)$ is the Tweedie sub-family most prevalent in practical applications: it is suitable for modelling time duration and total amounts of physical phenomena (Smyth, 1996). Variables that describe these types of measures are continuous and positive when the target phenomenon occurs. However, they have a non-null probability of taking a value of 0, which happens when the event of interest does not take place. According to this approach, the Poisson component of the CPG distribution counts the number of times the target phenomenon occurs, while the magnitude of the event on each occasion is modelled by the gamma component (Gilchrist & Drinkwater, 2000). Tweedie models with $\xi \in (1, 2)$ are also suitable for modelling data that show a variance that increases faster than the mean, but more slowly than the mean squared (Smyth, 1996).

Alternative models that deal with continuous data featuring an exact probability in zero require a special treatment of the null values: to use standard models, the data need to be transformed so that the zeros are removed. Otherwise, the exact zero class needs to be modelled separately. Instead, the compound Poisson-gamma distribution automatically deals with the exact zeros occurrences, because of its mixed nature (Foster & Bravington, 2013).

$TW_\xi(\mu, \phi)$ with $\xi \in (1,2)$ have been extensively used to model the total amount of rainfall in a certain area, in a specific time-frame. Some referencies are Hasan & Dunn (2011), Hasan & Dunn (2012), Hasan & Dunn (2015), Bonat & Kokonendji (2017), Dunn & Smyth (2018, page 466) and McKay Altman et al. (2023). In such applications, the variable that measures the total amount of rainfall takes on positive values when it rains, while it takes a value of 0 when it does not rain. Furthermore, the distribution is often right-skewed, since there are few days when the precipitations are abundant. These statistical models are used in weather forecasting, risk management and agricultural simulations (Hasan & Dunn, 2011).

Tweedie models with $\xi \in (1,2)$ are also commonly used in insurance claims data analyses (see Jorgensen & Smyth (2002),Quijano Xacur & Garrido (2015) and Fontaine et al. (2020)). Following the notation introduced in previous sections, the number of claims N within a specific time frame is modelled by the Poisson distribution, $N \sim Pois(\lambda)$, while the amount of each claim X_i is gamma distributed, $X_i \sim Ga^*(\alpha, \beta)$. The total insurance payout, i.e. the sum of the amounts of all claims in the time frame considered, follows a CPG distribution, $Y = \sum_{i=1}^{N} X_i \sim CPG(\lambda, \alpha, \beta)$ (Dunn & Smyth, 2018, page 463). Insurance claim data often present an excess of exact zeros, corresponding to insurance policies without claims, and display right-skewed distributions, since few claims have a really high monetary value (Fontaine et al., 2020).

The Tweedie distribution with $\xi \in (1,2)$ also finds application in ecological studies, where non-negative continuous data are frequent. Foster & Bravington (2013) employ the Tweedie distribution to model the combined weight of fish from a specific species captured in a trawl: the amount is equal to zero if the target species have not been caught.

Additional applications found in literature concern teenage alcohol consumption (Gilchrist & Drinkwater, 2000), the density of fine roots of apple trees (Dunn & Smyth, 2005) and the amount of time spent in gastroesophageal reflux by children during the night, in different sleeping positions (Smyth, 1996).

3.7.5 Taylor's power law

The power relationship between mean and variance that characterises Tweedie models has been observed in nature for many years.

Taylor (1961) noted that, in populations where individuals are not randomly distributed within their habitats but tend to aggregate, the variance of the population density is higher than the mean. Through the examination of samples from viruses,

macro-zooplanktons, worms, insects, ticks, fish, plants and mites populations, Taylor (1961) formulated the following rule, later known as Taylor's law.

Let Y be the variable that counts the number of units in a population, having mean μ and variance S^2. Then the variance S^2 is related to the mean μ through a power law:

$$S^2 = a\mu^b, \tag{3.19}$$

where a is a constant and b is an index of aggregation of the population, describing the clustering tendency of organisms. The majority of the observed b values fall within the $(1, 2)$ interval.

There is a clear correspondence between Taylor's power law (3.19) and the power relationship between mean and variance that characterises Tweedie models: the clustering constant b coincides with the Tweedie power parameter ξ, while the constant a corresponds to the dispersion parameter ϕ.

The variance-to-mean power relationship described by Taylor's law has been observed in several other systems.

- In human genetics, single nucleotide polymorphisms (SNPs), i.e. variations of a single nucleotide in a DNA sequence among individuals, form clusters on chromosome 7 that exhibit variance-mean power law (Kendal, 2004); Kendal (2003).
- In medicine, the number of cancer metastases developed in mice lungs after they were injected with cloned cancer cells follows Taylor's law with an exponent $b > 1$ (Fidler & Kripke, 1977); (Kendal & Frost, 1987).
- In signal processing and acoustics, $\frac{1}{f}$ noise, one of the most commonly observed signals in biological systems, having intensity $S(f) \propto \frac{1}{f^\gamma}$, where f is the frequency and $\gamma \in (0, 2)$, exhibits a variance-to-mean power law, with exponent $2 - \gamma$ (Jorgensen & Kendal, 2011b).

Other fields of application include physiology, regarding the description of regional blood flow (Bassingthwaighte et al., 1989), the number of houses built over the Tonami plain in Japan (Kendal, 1995), fractal theory (Jorgensen & Kendal, 2011b), the distribution of prime numbers (Jorgensen & Kendal, 2015) and the Mertens function in number theory (Jorgensen & Kendal, 2011a).

This recurring relationship has been explained through demographic and reproductive mechanisms, intraspecies behaviour and thermodynamics, yet its origin remains a topic of debate among scientists (Jorgensen & Kendal, 2011b). Indeed, such explanations do not apply to non-physical systems, thus they fail to justify the recurrence of

Taylor's law in number theory. Physical and biological mechanisms do not influence mathematics, but the reverse can occur.

Jorgensen & Kendal (2011a) propose an alternative and more elegant explanation: the Tweedie convergence theorem (3.10) introduced in Section 3.3.6 states that any EDM that asymptotically manifests a power mean-variance relationship converges in distribution to a Tweedie model. This theorem implies that any statistical model designed to produce Taylor's law must converge, by mathematical necessity, to a Tweedie EDM (Jorgensen & Kendal, 2011a). As illustrated in Section 3.3.6, numerous EDMs fit the assumptions of the Tweedie convergence theorem and most distributions commonly used in statistical modelling are EDMs: this justifies the persistence of Taylor's law across so many diverse systems.

3.8 Tweedie EDMs with $\xi > 2$

Tweedie models with $\xi > 2$ have support $S = \mathbb{R}^+$, mean domain $\Omega = \mathbb{R}^+$ and canonical parameter space $\Theta = \mathbb{R}_{\leq 0}$. They exhibit positive and continuous distributions which become progressively more right-skewed as the power parameter ξ increases.

When $\xi = 3$, the Tweedie model $TW_3(\mu, \phi)$ coincides with the inverse Gaussian distribution, whose characteristics are summarised in Table 2.1. This is the only case with $\xi > 2$ having a density function with an explicit analytic expression.

As shown in Section 3.3.7, given a response variable exhibiting a variance-to-mean power relationship with exponent $\xi > 2$, a simpler alternative to fitting a Tweedie model is estimating a normal regression model having response variable $Y^* = Y^{1-\frac{\xi}{2}}$. Specifically, this implies that fitting an inverse Gaussian GLM approximately corresponds to applying the transformation $Y^* = \frac{1}{\sqrt{Y}}$, while the commonly used inverse transformation $Y^* = \frac{1}{Y}$ corresponds with $TW_4(\mu, \phi)$. Nevertheless, the transformation approach is generally less effective than fitting a Tweedie model.

3.8.1 Type of data modelled

Responses that are positive, continuous and exhibit right-skewed distributions are typically adequately modelled by a gamma GLM. However, when the skewness becomes extreme, traditional models may no longer be able to properly represent the data (Dunn & Smyth, 2018, page 461).

Dunn & Smyth (2005) fit a Tweedie EDM with $\xi = 4$ to model the survival times of animals after exposure to three types of poisons and four types of treatments. The

value $\xi \approx 4$ was chosen using a method based on grouping, which will be introduced in Section 4.4.1. The same dataset was previously analysed by Box & Cox (1964), where they fit a normal regression model utilising the reciprocal transformation $\frac{1}{Y}$. Dunn & Smyth (2005) compared the fit of the Box-Cox and Tweedie models: they estimated both models for various values of ξ and computed the profile log-likelihoods. For all values of ξ considered, the Tweedie model exhibited a higher profile likelihood than its normal counterpart, indicating a better performance.

Bonat & Kokonendji (2017) employed a Tweedie EDM with $\xi \approx 2.6$ to model the earnings of 595 Australian individuals in 1982. Variables describing wages are known to be highly right-skewed since there are few high-earning subjects. They compared three different approaches to estimate ξ: one based on the maximum likelihood, one on the quasi-likelihood and the last on the pseudo-likelihood. All approaches performed well and provided coherent results. Finally, they compared the Tweedie model with the normal, gamma and inverse Gaussian GLMs: the maximised value of the log-likelihood of the Tweedie EDM was the highest, suggesting a better fit.

Chapter 4

Inference

4.1 Assumptions

Consider a sample of n statistical units identified by the index $i = 1, \ldots, n$, a response variable Y and p quantitative covariates xj, for $j = 1, \ldots, p$. Under the notation established in Section 2.2, the following quantities can be defined:

$Y_i \sim TW_\xi(\mu_i, \phi)$, response variable associated with the i-th unit, $i = 1, \ldots, n$;

It is assumed that $Y_1, \ldots Y_n$ are independent and Tweedie distributed, with common dispersion parameter ϕ and power parameter ξ, both usually unknown. Conversely, the mean μ_i varies with i.

y_i, value of the response variable measured on the i-th statistical unit, $i = 1, \ldots, n$;

$y^\intercal = [y_1, \ldots y_n]$, n-variate vector of independent realisations of Y_i, $i = 1, \ldots, n$;

x_{ij}, value of the j-th covariate measured on the i-th statistical unit, $i = 1, \ldots, n$, $j = 1, \ldots, p$;

$x_i = [x_{i1}, \ldots x_{ip}]$, p-variate row vector of the covariates measured on the i-th statistical unit, $i = 1, \ldots, n$;

$X = [x_1^\intercal, \ldots x_n^\intercal]^\intercal$, $n \times p$ model matrix;

$\beta^\intercal = [\beta_1, \ldots \beta_p]$, p-variate vector of the unknown regression parameters;

$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij} = x_i \beta$, linear predictor of the i-th statistical unit, $i = 1, \ldots, n$.

As shown in Table 3.2, for all $\xi \in \mathbb{R} \setminus [0, 1)$, Tweedie EDMs have mean domain $\Omega = \mathbb{R}^+$. Consequently, the most commonly used link function $g : \mathbb{R}^+ \to \mathbb{R}$ for Tweedie models is the logarithmic transformation (Dunn & Smyth, 2018, page 464);(Bonat & Kokonendji, 2017).

$E[Y_i] = \mu_i = g^{-1}(\eta_i) = \exp\{\eta_i\}$, mean of the response measured on the i-th unit, $i = 1, \ldots, n$.

The logarithmic transformation $\log(\mu_i)$ does not coincide with the canonical link function for Tweedie models (3.3), except for the case $\xi = 1$. However, it is preferred over $\theta_\xi(\mu_i)$ because it guarantees an easier interpretation of the regression parameters. Nevertheless, this link function should be avoided when $\xi = 0$, as the domain of $\log(\cdot)$ is $(0, +\infty)$, whereas the mean parameter space of the normal distribution is $\Omega = \mathbb{R}$.

The parameter we are primarily interested in estimating is β, while (ξ, ϕ) are not the main focus of the model. Consequently, it is convenient to partition the unknown parameters vector into two parts: one referring to the parameter of interest β, and the other to the nuisance parameters ξ and ϕ.

$\lambda = (\xi, \phi)$, vector of the nuisance parameters;

$\psi = (\beta, \xi, \phi) = (\beta, \lambda)$, whole vector of unknown parameters .

4.2 Maximum likelihood estimation

4.2.1 Likelihood, score function and information for (β, ξ, ϕ)

The discussion presented in this section follows the inferential framework for GLMs introduced in Chapter 2 and Bonat & Kokonendji (2017).

Under the notation introduced in Section 4.1, the likelihood function for the parameter vector $\psi = (\beta, \xi, \phi)$ is

$$L(\psi; y) = L(\beta, \lambda; y) = L(\beta, \xi, \phi; y) = \prod_{i=1}^{n} \widetilde{a}_\xi(y_i, \phi) \exp\left\{\frac{\theta_i y_i - b_\xi(\theta_i)}{\phi}\right\},$$

where $\theta_i = \theta_\xi(\mu_i) = \theta_\xi(g^{-1}(\eta_i)) = \theta_\xi(e^{\eta_i}) = \theta_\xi(e^{x_i\beta})$. The log-likelihood function for $\psi = (\beta, \xi, \phi)$ is

$$l(\psi; y) = l(\beta, \lambda; y) = \log(L(\psi; y)) = \sum_{i=1}^{n} \left\{\log(\widetilde{a}_\xi(y_i, \phi)) + \frac{\theta_i y_i - b_\xi(\theta_i)}{\phi}\right\}. \tag{4.1}$$

Similarly to what was done in Section 2.3, the score function can be partitioned into two parts, one that refers to the parameter of interest β, whereas the other to the nuisance parameters $\lambda = (\xi, \phi)$.

$$U^\intercal(\psi) = U^\intercal(\beta, \lambda) = [U_\beta^\intercal(\beta, \lambda), U_\lambda^\intercal(\beta, \lambda)] = [\frac{\partial l(\psi; y)}{\partial \beta^\intercal}, \frac{\partial l(\psi; y)}{\partial \lambda^\intercal}]$$

$$= [\frac{\partial l(\psi;y)}{\partial \beta_1}, \ldots, \frac{\partial l(\psi;y)}{\partial \beta_p}, \frac{\partial l(\psi;y)}{\partial \phi}, \frac{\partial l(\psi;y)}{\partial \xi}] = [U_{\beta_1}, \ldots, \ U_{\beta_p}, U_\phi, U_\xi].$$

The generic element U_{β_j} of the score function that refers to the regression parameters can be obtained from (2.3).

$$U_{\beta_j} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\phi} \frac{1}{\mu_i^{\xi}} \frac{x_{ij}}{\mu_i^{-1}} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)\mu_i x_{ij}}{\phi \mu_i^{\xi}} \quad j = 1, \ldots, p,$$

where $\mu_i = \exp\{\eta_i\} = \exp\{x_i\beta\}$. The two components of $U_\lambda(\beta, \lambda)$ are

$$U_\phi = \sum_{i=1}^{n} \left\{ \frac{\partial \log(\widetilde{a}_\xi(y_i, \phi))}{\partial \phi} - \frac{1}{\phi^2}(\theta_i y_i - b_\xi(\theta_i)) \right\}, \tag{4.2}$$

$$U_\xi = \sum_{i=1}^{n} \left\{ \frac{\partial \log(\widetilde{a}_\xi(y_i, \phi))}{\partial \xi} + \frac{1}{\phi} \left(\frac{\partial \theta_\xi(\mu_i)}{\partial \xi} y_i - \frac{\partial b_\xi(\theta_i)}{\partial \xi} \right) \right\}. \tag{4.3}$$

The partial derivatives $\dfrac{\partial \log(\widetilde{a}_\xi(y_i, \phi))}{\partial \phi}$ and $\dfrac{\partial \log(\widetilde{a}_\xi(y_i, \phi))}{\partial \xi}$ depend on the function $\widetilde{a}_\xi(y_i, \phi)$, which cannot be written in closed form, thus they do not possess an explicit analytic expression.

Under the (β, λ) parameterisation, the Fisher information matrix can be partitioned as:

$$i(\beta, \lambda) = E \begin{bmatrix} -\dfrac{\partial U_\beta(\beta,\lambda)}{\partial \beta^\intercal} & -\dfrac{\partial U_\lambda(\beta,\lambda)}{\partial \beta^\intercal} \\ -\dfrac{\partial U_\beta(\beta,\lambda)}{\partial \lambda^\intercal} & -\dfrac{\partial U_\lambda(\beta,\lambda)}{\partial \lambda^\intercal} \end{bmatrix} = \begin{bmatrix} i_{\beta\beta} & i_{\beta\lambda} \\ i_{\lambda\beta} & i_{\lambda\lambda} \end{bmatrix}. \tag{4.4}$$

The generic element of the $i_{\beta\beta}$ block can be obtained from (2.5)

$$[i_{\beta\beta}]_{jk} = \sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{\phi \mu_i^{\xi} \mu_i^{-2}} = \frac{x_{ij}x_{ik}}{\phi \mu_i^{\xi-2}} \quad j, k = 1, \ldots, p.$$

In Section 2.3, it was demonstrated that the dispersion parameter is orthogonal to the regression coefficients β and, consequently, it is also orthogonal to the mean μ_i and the canonical parameter θ_i. In the case of Tweedie models, it can be proven that β is orthogonal to the nuisance parameters $\lambda = (\xi, \phi)$, by showing that the information matrix (4.4) is a diagonal block matrix, as the components of the $i_{\beta\lambda} = i_{\lambda\beta}^\intercal$ blocks are all null.

$$[i_{\lambda\beta}]_{1j} = -E[\frac{\partial U_{\beta_j}}{\partial \phi}] = \frac{1}{\phi^2} \sum_{i=1}^{n} \frac{E[y_i - \mu_i]\mu_i x_{ij}}{\mu_i^{\xi}} = 0 \quad j = 1, \ldots, p.$$

$$[i_{\lambda\beta}]_{2j} = -E[\frac{\partial U_{\beta_j}}{\partial \xi}] = -\frac{1}{\phi} \sum_{i=1}^{n} E[y_i - \mu_i] x_{ij} \frac{\partial \mu_i^{1-\xi}}{\partial \xi} = 0 \quad j = 1, \ldots, p.$$

The $i_{\lambda\lambda}$ block of the information matrix (4.4) has four elements

$$i_{\lambda\lambda} = E\begin{bmatrix} -\dfrac{\partial U_\phi}{\partial \phi} & -\dfrac{\partial U_\xi}{\partial \phi} \\ -\dfrac{\partial U_\phi}{\partial \xi} & -\dfrac{\partial U_\xi}{\partial \xi} \end{bmatrix} = \begin{bmatrix} i_{\phi\phi} & i_{\phi\xi} \\ i_{\xi\phi} & i_{\xi\xi} \end{bmatrix}.$$

The elements of $i_{\lambda\lambda}$ depend on the partial derivatives $\dfrac{\partial^2 \widetilde{a}_\xi(y_i,\phi)}{\partial \phi^2}$, $\dfrac{\partial^2 \widetilde{a}_\xi(y_i,\phi)}{\partial \phi \partial \xi}$ and $\dfrac{\partial^2 \widetilde{a}_\xi(y_i,\phi)}{\partial \xi^2}$ that, just like the function $\widetilde{a}_\xi(y_i,\phi)$, cannot be written in closed form.

Dunn & Smyth (2018, page 470) strongly recommend the use of quantile residuals for Tweedie EDMs with $\xi \in (1,2)$ because of their mixed distribution.

4.3 Estimation of the parameters β and ϕ

In Chapter 2 it was stated that the regression coefficients β can be estimated using the iterative reweighted least squares algorithm (IRLS), without fixing a value for the nuisance parameter ϕ. The dispersion parameter can be subsequently estimated after fitting the appropriate GLM. Unfortunately, the same straightforward approach does not apply to Tweedie models: fitting a Tweedie GLM requires fixing a value for the power parameter ξ, as each possible value of ξ identifies a distinct exponential dispersion model and, consequently, a different GLM (Dunn & Smyth, 2018, page 465). In Sections 4.4.1 and 4.4.2, two methods for computing an estimate of ξ are illustrated.

Once an appropriate estimate $\hat{\xi}$ for the power parameter is given, a GLM can be fit, $\hat{\beta}$ can be computed using the IRLS algorithm and the dispersion parameter ϕ can also be estimated. Various estimators of ϕ are available.

As mentioned in Section 2.3.2, the maximum likelihood estimate (MLE) of ϕ is highly biased, however, it needs to be computed for conducting statistical inferences on the regression coefficients β and it will be used, in a slightly modified form, to compute an estimate for ξ (see Section 4.4.2). With few exceptions, $\hat{\phi}$ has no closed form and needs to be computed numerically.

Alternative estimators of the dispersion parameter have been defined: let $\hat{\mu}_i = e^{xi\hat{\beta}}$, $i = 1, \ldots, n$. Then, mean-deviance estimator of ϕ is defined as

$$\widetilde{\phi} = \frac{\sum_{i=1}^{n} d_{\hat{\xi}}(y_i, \hat{\mu}_i)}{n-p}, \tag{4.5}$$

where $d_{\hat{\xi}}(y_i, \hat{\mu}_i)$ is an estimate of the unit deviance for Tweedie models (3.7) and $p = \dim(\beta)$.

The Pearson estimate (2.6) for ϕ in Tweedie models is

$$\widetilde{\phi}^* = \frac{1}{n-p} \frac{\sum_{i=1}^{n} (y_i - \hat{\mu}_i)^2}{\hat{\mu}_i^{\hat{\xi}}}. \tag{4.6}$$

If $\xi = 0$ or $\xi = 3$, $\frac{n-p}{n}\widetilde{\phi}$ coincides with the MLE $\hat{\phi}$. In the other cases, it is advisable to use the Pearson estimate (4.6).

4.4 Estimation of the power parameter ξ

The parameter ξ functions as a model selector (Dunn & Smyth, 2018, page 465), since estimating a Tweedie GLM requires specifying a value for the power parameter. Consequently, unlike the dispersion parameter ϕ, ξ must be estimated before fitting the Tweedie GLM.

When exact zeros are present in the observed data, then necessarily $\xi \in (1, 2)$. However, the absence of exact zeros does not imply that $\xi > 2$: it is still possible that $1 < \xi < 2$ but, purely by chance, no occurrences of the variable being equal to 0 have been observed (Dunn & Smyth, 2018, page 465).

4.4.1 The estimation of ξ: a method based on grouping

Dunn & Smyth (2018, page 461) propose a simple method to obtain a rough estimate of the power parameter ξ. In the first place, the n units are divided into $K > 2$ groups $G_1, \ldots, G_K$, identified by the index $k = 1, \ldots, K$. For example, given a categorical covariate Z with K levels, the statistical units with identical values of Z could belong to the same group. Let

- n_k denote the size of the k-th group,
- $\overline{y}_k = \sum_{i \in G_k} \frac{y_i}{n_k}$ be the sample mean of the response in the k-th group,
- $\hat{\sigma}_k^2 = \sum_{i \in G_k} \frac{(y_i - \overline{y}_k)^2}{n_k}$ be the sample variance of the response in the k-th group, for $k = 1, \ldots, K$.

Since $Y_i \sim TW_\xi(\mu, \phi)$ for $i = 1, \ldots, n$, then $Var[Y_i] = \phi \mu_i^\xi$. By applying the logarithmic transformation to both sides, we obtain $\log(Var[Y_i]) = \log(\phi) + \xi \log(\mu_i)$: this reveals a linear relationship between the log-variance and the log-mean, where the logarithm of the dispersion parameter represents the intercept, while the power parameter ξ corresponds

to the slope coefficient. This suggests that the points $(\overline{y}_k, \hat{\sigma}_k^2)$ for $k = 1, \ldots, K$ should approximate a line on the plane.

Consider the normal regression model having the log-sample variances $\hat{\sigma}_k^2$, for $k = 1, \ldots, K$, as response variable and the log-sample means $\overline{y}_k$, for $k = 1, \ldots K$, as the covariate. The unknown regression coefficients are (β_1, β_2), where β_1 indicates the intercept and β_2 the slope of the line. Then ξ can be estimated as $\hat{\beta}_2$.

Even though this method is straightforward, it is not entirely reliable: the criterion applied to split the units and the number of groups formed are chosen arbitrarily. Indeed, different choices may lead to varying estimates of ξ (Dunn & Smyth, 2018, page 461).

Hence, a more rigorous approach to find the MLE of the Tweedie power parameter has been implemented in the function `tweedie.profile` from the R package `tweedie` (Dunn, 2017).

4.4.2 Inference about ξ

In this specific context, we are interested in drawing inferences solely on the power parameter ξ. Consequently, in this section, we will partially deviate from the notation established in Section 4.1: ξ, rather than β, will be the parameter of interest, whereas $\zeta = (\beta, \phi)$ will be regarded as a nuisance parameter. In such scenarios, the most commonly employed approach is the profile likelihood for the parameter of interest, as stated by Dunn & Smyth (2018, page 466).

4.4.3 Profile log-likelihood for ξ

Ideally, if the value of the nuisance parameter was known and equal to $\zeta_0 = (\beta_0, \phi_0)$, the log-likelihood could be expressed solely as a function of the power parameter: $l(\xi) = l(\beta_0, \xi, \phi_0)$ (Pace & Salvan, 2001, page 145). However, this typically is not the case: to obtain a log-likelihood function that depends only on ξ, (β, ϕ) must be estimated. Reasonably, the estimates of the nuisance parameters should depend on the value of ξ at which the function $l(\xi)$ is being evaluated. For this reason, they are known as constrained estimates: the constrained estimate $\hat{\zeta}_\xi = (\hat{\beta}_\xi, \hat{\phi}_\xi)$ of the parameter $\zeta = (\beta, \phi)$ given ξ is defined as

$$\hat{\zeta}_\xi = (\hat{\beta}_\xi, \hat{\phi}_\xi) \in \mathbb{R} \times \mathbb{R}^+ \mid l(\hat{\beta}_\xi, \xi, \hat{\phi}_\xi) \geq l(\beta, \xi, \phi) \; \forall \; (\beta, \phi) \in \mathbb{R} \times \mathbb{R}^+. \tag{4.7}$$

Essentially, $\hat{\zeta}_\xi = (\hat{\beta}_\xi, \hat{\phi}_\xi)$ is the value of the nuisance parameter that maximises the log-likelihood function when the power parameter is fixed at ξ.

Typically, $(\hat{\beta}_\xi, \hat{\phi}_\xi)$ would be the values of β and ϕ that satisfy $U_\beta = 0$, $U_\phi = 0$ when ξ is fixed (Pace & Salvan, 2001, page 146). However, $\hat{\beta}_\xi$ is computed using the iterative reweighted least squares algorithm that, as recalled in Section 2.3.2, does not depend on the value of the dispersion parameter, whereas $\hat{\phi}_\xi$ is computed as the value that maximises the log-likelihood when ξ is fixed and $\beta = \hat{\beta}_\xi$.

$$\hat{\phi}_\xi = \arg\max_{\phi} \{l(\hat{\beta}_\xi, \phi, \xi; y)\}. \tag{4.8}$$

The profile log-likelihood for ξ is defined as

$$l_P : \mathbb{R} \setminus (0,1) \to \mathbb{R} \quad l_P(\xi) = l(\hat{\beta}_\xi, \xi, \hat{\phi}_\xi). \tag{4.9}$$

The maximum likelihood estimate for ξ obtained through the profile log-likelihood is

$$\hat{\xi} \in \mathbb{R} \setminus (0,1) \mid l_P(\hat{\xi}) \geq l_P(\xi) \; \forall \; \xi \in \mathbb{R} \setminus (0,1). \tag{4.10}$$

The estimate of ξ obtained with the profile likelihood coincides with the MLE of ξ obtained by maximising the full log-likelihood $l(\beta, \xi, \phi)$ since the constrained estimate (4.7) associated with $\hat{\xi}$ is the MLE $(\hat{\beta}, \hat{\phi})$: thus, $(\hat{\beta}_{\hat{\xi}}, \hat{\xi}, \hat{\phi}_{\hat{\xi}}) = (\hat{\beta}, \hat{\xi}, \hat{\phi})$ (Pace & Salvan, 2001, page 147).

Under regularity conditions, the log-profile likelihood ratio $W_P(\xi)$ is an approximate pivotal quantity with asymptotic χ^2_1 distribution (Pace & Salvan, 2001, page 147); (Dunn & Smyth, 2018, page 467):

$$W_P(\xi) = 2[l(\hat{\psi}) - l_P(\xi)] = 2[l(\hat{\beta}, \hat{\xi}, \hat{\phi}) - l_P(\xi)] \dot{\sim} \chi^2_1. \tag{4.11}$$

Therefore, $W_P(\xi)$ can be used to define confidence intervals for ξ. The confidence interval (CI) for ξ with a nominal confidence level $1 - \alpha$, based on the profile log-likelihood ratio statistic is

$$\hat{\Xi}_{1-\alpha}(y) = \{\xi \in \mathbb{R} \setminus (0,1) \mid W_P(\xi) \leq \chi^2_{1;1-\alpha}\}. \tag{4.12}$$

Equivalently,

$$\hat{\Xi}_{1-\alpha}(y) = \{\xi \in \mathbb{R} \setminus (0,1) \mid l_P(\xi) \geq l(\hat{\beta}, \hat{\xi}, \hat{\phi}) - \frac{\chi^2_{1;1-\alpha}}{2}\}. \tag{4.13}$$

4.4.4 Computing the profile log-likelihood for ξ and related quantities

The profile log-likelihood function $l_P(\xi)$ (4.9), the constrained estimate $\hat{\zeta}_\xi = (\hat{\beta}_\xi, \hat{\phi}_\xi)$ (4.7), the maximum likelihood estimate $\hat{\psi} = (\hat{\beta}, \hat{\xi}, \hat{\phi})$ and the confidence interval $\hat{\Xi}_{1-\alpha}(y)$ (4.12) are not available in closed form and need to be computed numerically.

In this section, we will demonstrate how to implement the functions necessary to compute the quantities listed above, using the software R. The function `tweedie.profile` from the package `tweedie` (Dunn, 2017) served as the basis for the forthcoming script, even though some changes have been made. To improve the readability of the code, each computational step is handled by a separate function.

To ensure that the code functions correctly, it is necessary to install the packages `tweedie` (Dunn, 2017) and `statmod` (Smyth et al., 2006). The functions from the `tweedie` package that will be utilised in the following analysis are

- `dtweedie(`y`, `ξ`, `μ`, `ϕ`)` computes the density function of $TW_\xi(\mu, \phi)$ in the point y, by applying the optimal method for the chosen value of ξ among the ones described in Section 3.4;
- `rtweedie(`n`, `ξ`, `μ`, `ϕ`)` generates n independent values from a $TW_\xi(\mu, \phi)$;
- `tweedie.dev(`y`,`μ`,`ξ`)` computes the Tweedie unit deviance $d_\xi(y, \mu)$ of $Y \sim TW_\xi(\mu, \phi)$.

Instead, the `statmod` package is required to fit Tweedie models using the `glm` function: the command `glm(`$y \sim x$`, family=tweedie(var.power=`ξ`, link.power=0))` fits a Tweedie model with an intercept, having response y, covariate x, power parameter ξ and logarithmic link function.

```
library(tweedie)
library(statmod)
```

Listing 4.1: Installation of the packages

The function `ll.tw` computes the log-likelihood for Tweedie models starting from the density function `dtweedie` implemented in R from the package `tweedie` (Dunn, 2017).

```
ll.tw=function(beta,xi,phi,y,Xmm){
  #linear predictor
  eta<-Xmm%*%as.matrix(beta,ncol=1)
  #mean of the response
  mu=exp(eta)
  #loglikelihood
  value=sum(log(dtweedie(y,xi=xi,mu=mu,phi=phi)))
```

```
  return(value)}
```

LISTING 4.2: Tweedie loglikelihood $l(\beta, \xi, \phi; y)$

The function `f.beta.mle.cons` computes the constrained estimate of the regression parameter $\hat{\beta}_\xi$ (4.7) by fitting a Tweedie GLM given a value for the power parameter ξ. It distinguishes between the case $\xi \neq 0$, where the logarithmic link function can be utilised, and the case $\xi = 0$, which requires an identity link function.

```
f.beta.mle.cons=function(xi,y,dat){
  #if the response is not normal
  if (xi!=0){
    #tweedie glm with power parameter xi and logarithmic link function
    glm.fit<-glm(y~.,family=tweedie(var.power = xi,link.power =
   0),data=dat)
    beta.mle.cons=coef(glm.fit)}
  #if the response is normal
  if (xi==0){
    #linear normal regression model, identity link function
    glm.fit<-glm(y~.,data=dat)
    beta.mle.cons=coef(glm.fit)}
  return(beta.mle.cons)}
```

LISTING 4.3: constrained estimate of the regression coefficients $\hat{\beta}_\xi$

The function `f.phi.mle.cons` computes the constrained estimate of the dispersion parameter $\hat{\phi}_\xi$ (4.7). While several methods are available to compute $\hat{\phi}_\xi$, the MLE (4.8) is chosen as a default. To estimate ϕ, it is necessary to fix values both for ξ and for β, therefore this function should be called after `f.beta.mle.cons`.

```
f.phi.mle.cons=function(xi,beta,y,Xmm){
  #lower limit
  phi.lower=10^(-4)
  #upper limit: mean deviance estimator x 10
  eta=Xmm%*%as.matrix(beta,ncol=1)
  phi.upper=10*sum(tweedie.dev(y = y, mu = exp(eta), power =
   xi))/length(y)
  if (is.nan(phi.upper)){phi.upper=10^3}
  if (phi.upper==Inf){phi.upper=10^3}
  #call to the optimizer
  opt.phi.call<-optimize(ll.tw,interval=c(phi.lower,phi.upper),
                         xi=xi,beta=beta,y=y,Xmm=Xmm,maximum = T)
  #constrained estimate of phi
  phi.mle.cons=opt.phi.call$maximum
  return(phi.mle.cons)}
```

LISTING 4.4: constrained estimate of the dispersion parameter $\hat{\phi}_\xi$

The function `llp.tw` computes the profile likelihood for the power parameter $l_P(\xi)$. The implementation of `llp.tw` is identical to `ll.tw`; however, it has been included to differentiate between the profile likelihood $l_P(\xi)$ and the complete likelihood $l(\beta, \xi, \phi)$, therefore improving the readability of the code.

```
llp.tw=function(phi.mle.cons,beta.mle.cons,xi,y,Xmm)
   {ll.tw(beta.mle.cons,xi,phi.mle.cons,y,Xmm)}
```

LISTING 4.5: Profile log-likelihood $l_P(\xi)$

The function `Wp` computes the approximate pivotal quantity based on the profile log-likelihood for ξ (4.11). `Wp` has been included in the code, even though the main function will not call it: the same analysis conducted with the profile likelihood function can be performed utilising $W_P(\xi)$.

```
Wp=function(xi,phi.mle.cons,beta.mle.cons) {
  2*(ll.tw(beta.mle,xi.mle,phi.mle,y)-
       llp.tw(phi.mle.cons,beta.mle.cons,xi,y))}
```

LISTING 4.6: Pivotal quantity based on the profile log-likelihood $W_P(\xi)$

Since $l_P(\xi)$ is not available in closed form, it must be computed numerically. The `profiling` function, for a sequence of values of ξ, computes the constrained estimates $(\hat{\beta}_\xi, \hat{\phi}_\xi)$, the profile likelihood function $l_P(\xi)$ and stores them into vectors. If the user chooses to, it can print the computed values at each iteration. Oftentimes, the `glm` function raises errors or fails to converge for certain values of ξ: therefore, problematic values of the power parameter (which frequently correspond to larger values in the sequence) are removed.

The default option of the function `tweedie.profile` from the R package `tweedie` utilises a different approach: instead of considering a long sequence of values of ξ and computing the log-likelihood $l_P(\xi)$ for each value of the power parameter, it considers a shorter sequence of values and computes a smoothed estimate of $l_P(\xi)$.

```
profiling=function(xi.seq,verbose=F,y,Xmm,dat){
  min.xi=min(xi.seq)
  max.xi=max(xi.seq)
  n.seq=length(xi.seq)
  p=ncol(Xmm)
  #preallocation of vectors
  phi.seq=rep(NA,n.seq)
```

```
  beta.seq=matrix(NA,nrow=n.seq,ncol=p)
  Lp.seq<-rep(NA,n.seq)
  #for each value xi of the sequence
  for (i in 1:n.seq){
    #the value of xi is fixed
    xi=xi.seq[i]
    tryCatch(
    {#this code is run if glm does not raise an error
    #constrained estimate of beta
    beta.seq[i,]=f.beta.mle.cons(xi,y=y,dat)
    #constrained estimate of phi
    phi.seq[i]=f.phi.mle.cons(xi=xi,beta=beta.seq[i,],y=y,Xmm)
    #profile loglikelihood
    Lp.seq[i]=llp.tw(phi.seq[i],beta.seq[i,],xi,y=y,Xmm)
    #if verbose==T, prints the computed values
    if (verbose==T){cat('\n', ' Iteration n. ',i, ' xi=',xi.seq[i], '
   phi=', phi.seq[i], ' beta=',beta.seq[i,], ' Profile ll=',
   Lp.seq[i],'\n')}},
    #this code is run f glm raises an error
    error = function(e) {
      if (verbose==T){cat("\n glm raised an error in iteration", i,
   "for xi =", xi.seq[i], ":", conditionMessage(e))}})}
  #removing the values that caused the error
  #indices of the values of xi the caused the error
  idx=as.numeric(which(apply(beta.seq, 1, function(row)
   all(is.na(row)))))
  #if glm raised an error at least once
  if (length(idx)!=0){
    #sequences without the problematic values
    xi.seq.new=xi.seq[-idx]
    beta.seq=beta.seq[-idx,]
    phi.seq=phi.seq[-idx]
    Lp.seq=Lp.seq[-idx]
    #new lenght of the sequences
    n.seq.new=length(xi.seq.new)
  }
  #if glm never raised an error
  if (length(idx)==0){
    #the length of the sequence stays the same
    n.seq.new=length(xi.seq)
    #the sequence of values of xi stays the same
    xi.seq.new=xi.seq}
  #returns a list of the computed values
returned_list=list(Lp.seq=Lp.seq,xi.seq=xi.seq.new,
```

```
    beta.seq=beta.seq,phi.seq=phi.seq,n.seq.new=n.seq.new)
  return(returned_list)}
```

LISTING 4.7: Profiling function

The function `mle.tweedie` efficiently computes the maximum likelihood estimate of (β, ξ, ϕ) by utilising the calculations completed by the profiling function. Instead of directly maximising the complete log-likelihood $l(\beta, \xi, \phi)$, it identifies the value of ξ in the sequence chosen by the user that maximises the profile log-likelihood $l_P(\xi)$, along with the constrained estimates $(\hat{\beta}_\xi, \hat{\phi}_\xi)$ associated to it. As illustrated in Section 4.4.3, these values coincide with the MLE $(\hat{\beta}, \hat{\xi}, \hat{\phi})$. This implies that the sequence of values of ξ must be sufficiently dense: for example, 100 elements is an appropriate number.

```
mle.estimate=function(Lp.seq,xi.seq,beta.seq,phi.seq){
  #index that maximises the loglikelihood
  L.max.idx<-which.max(Lp.seq[which(is.finite(Lp.seq))])
  #maximised value of the loglikelihood
  L.max<-Lp.seq[L.max.idx]
  #mle of xi
  xi.mle<-xi.seq[L.max.idx]
  #mle of beta
  beta.mle<-beta.seq[L.max.idx,]
  #mle of phi
  phi.mle<-phi.seq[L.max.idx]
  returned_list=list(L.max.idx=L.max.idx,L.max=L.max,xi.mle=xi.mle,
    beta.mle=beta.mle,phi.mle=phi.mle)
  return(returned_list)
}
```

LISTING 4.8: Maximum likelihood estimation of (β, ξ, ϕ)

The function `CI.tweedie` computes the confidence interval for ξ based on the profile log-likelihood (4.13) adopting a different approach from the one employed by Dunn (2017) in the function `tweedie.profile`. The function `CI.tweedie` aims to determine the bounds (ξ_L, ξ_U) of the confidence interval $\hat{\Xi}_{1-\alpha}(y)$: from a graphical point of view, they correspond to the intersection points between the profile log-likelihood and its quantile line $q_{1-\alpha} = l(\hat{\beta}, \hat{\xi}, \hat{\phi}) - \frac{\chi^2_{1;1-\alpha}}{2}$ (see Figure 4.1). In notation,

$$(\xi_L, \xi_U) = \{\xi \mid l_P(\xi) = q_{1-\alpha}\} = \{\xi \mid l_P(\xi) = l(\hat{\beta}, \hat{\xi}, \hat{\phi}) - \frac{\chi^2_{1;1-\alpha}}{2}\}.$$

This task can be approached as an optimisation problem: the previous expression can be rewritten as

$$(\xi_L, \xi_U) = \{\xi \ : \ |l_P(\xi) - q_{1-\alpha}| = 0\} = \{\xi \mid f_{OBJ}(\xi; \hat{\beta}, \hat{\xi}, \hat{\phi}) = 0\},$$

where $f_{OBJ}(\xi; \hat{\beta}, \hat{\xi}, \hat{\phi}) = |l_P(\xi) - q_{1-\alpha}|$ is a positive valued function. Therefore, ξ_L and ξ_U can be obtained as

$$(\xi_L, \xi_U) = (\underset{\xi<\hat{\xi}}{\arg\min} \{f_{OBJ}(\xi; \hat{\beta}, \hat{\xi}, \hat{\phi})\} \ , \ \underset{\xi>\hat{\xi}}{\arg\min} \{f_{OBJ}(\xi; \hat{\beta}, \hat{\xi}, \hat{\phi})\}).$$

If `min.xi` and `max.xi`, the minimum and maximum values of the power parameter given as input to the function `CI.tweedie`, are too close to each other, the confidence interval computed by the function might not be reliable. More precisely, if `min.xi`, `max.xi` $\in \hat{\Xi}_{1-\alpha}(y)$, the bounds of the computed confidence interval $(\hat{\xi}_L, \hat{\xi}_U)$ will coincide with `min.xi` and `max.xi`. To avoid this issue, the interval in which the objective function is optimised is larger than (`min.xi`, `max.xi`). However, a warning message will be printed if $(\hat{\xi}_L, \hat{\xi}_U)$ are still too close to `min.xi, max.xi`.

`CI.tweedie` only considers the values of ξ in the sequence such that the `glm` function converges when computing the constrained estimate $\hat{\beta}_\xi$: as mentioned previously, problematic values of the power parameter frequently correspond to larger values in the sequence.

To summarise, the lower limit $\hat{\xi}_L$ of the computed confidence interval is inaccurate when the user-selected value of `min.xi` is too high and falls within the CI we are trying to compute: i.e. `min.xi` $\in \hat{\Xi}_{1-\alpha}(y)$. The most straightforward solution to this issue is to decrease the value of `min.xi` until an accurate lower limit is computed. However, values of `min.xi` excessively close to 1 should be avoided, since $l_P(\xi)$ might develop a spurious local maxima at $\xi = 1$ in some circumstances (Dunn & Smyth, 2005). Similarly, the computed upper limit $\hat{\xi}_U$ is inaccurate when the user-selected value of `max.xi` is too low and falls within the CI we are trying to compute. In this case, the value of `max.xi` can be increased as needed since the parameter space of ξ is unbounded above. Nevertheless, another issue may arise. As previously mentioned, the `glm` function occasionally fails to converge for higher values of ξ in the sequence: if some of these values belong to the confidence interval $\hat{\Xi}_{1-\alpha}(y)$, then the computed upper limit will also be unreliable. This issue is more challenging to solve and mostly occurs when covariates are involved, since `glm` has to fit more complex models. However, in some cases, adjusting the user-selected sequence of values of ξ can fix the problem.

```
CI.tweedie=function
```

```
    (alpha=0.05,xi.mle,min.xi,max.xi,beta.mle,phi.mle,y,Xmm,dat)
{#objective function
  f.obj=function(xi,dat){
    beta.mle.cons=rep(-10^4,ncol(Xmm))
    tryCatch(
      {#this code is run if glm does not raise an error
        #constrained estimate of beta
        beta.mle.cons=f.beta.mle.cons(xi,y=y,dat=dat)})
  #constrained estimate of xi
  phi.mle.cons=f.phi.mle.cons(xi=xi,beta=beta.mle.cons,y=y,Xmm=Xmm)
  #quantile of the profile loglikelihood
  quantile.level<-
    ll.tw(beta.mle,xi.mle,phi.mle,y=y,Xmm=Xmm)-(qchisq(1-alpha,1)/2)
  #value of the objective function
  value<-abs
    (llp.tw(phi.mle.cons,beta.mle.cons,xi,y=y,Xmm=Xmm)-quantile.level)
  return(value)}
  #lower limit of the CI
  xi.lower=NA
  xi.lower=tryCatch(optimize(f.obj,interval=

   c(min.xi-xi.mle/1000,xi.mle),dat=dat)$minimum)
  #upper limit of the CI
  xi.upper=NA
  xi.upper=tryCatch(optimize(
    f.obj,interval=c(xi.mle,max.xi+xi.mle/1000),dat=dat)$minimum)
  #checking if the limits of the confidence interval are reliable
  if (round(xi.lower,1)==round(min.xi-xi.mle/1000,1)){cat('The lower
   limit of the confidence interval might not be reliable, the value
   of min.xi=',min.xi,' is too high', '\n')}
  if (round(xi.upper,1)==round(max.xi+xi.mle/1000,1)){cat('The upper
   limit of the confidence interval might not be reliable, the value
   of max.xi=',max.xi, ' is too low', '\n')}
  return(c(xi.lower,xi.upper))}
```

LISTING 4.9: Confidence interval (C.I.) based on the profile log-likelihood for ξ

The function `main.tweedie` calls all the previously defined functions: the user inputs a minimum, `min.xi`, and a maximum, `max.xi`, value for ξ to compute a sequence of values for the power parameter: only values of ξ such that $\xi \geq 1$ can be included into the sequence. The minimum and the maximum values in the sequence should be sufficiently spaced apart, otherwise the confidence interval computed by the function will not be reliable. The choice of the sequence of ξ values given as input can influence the

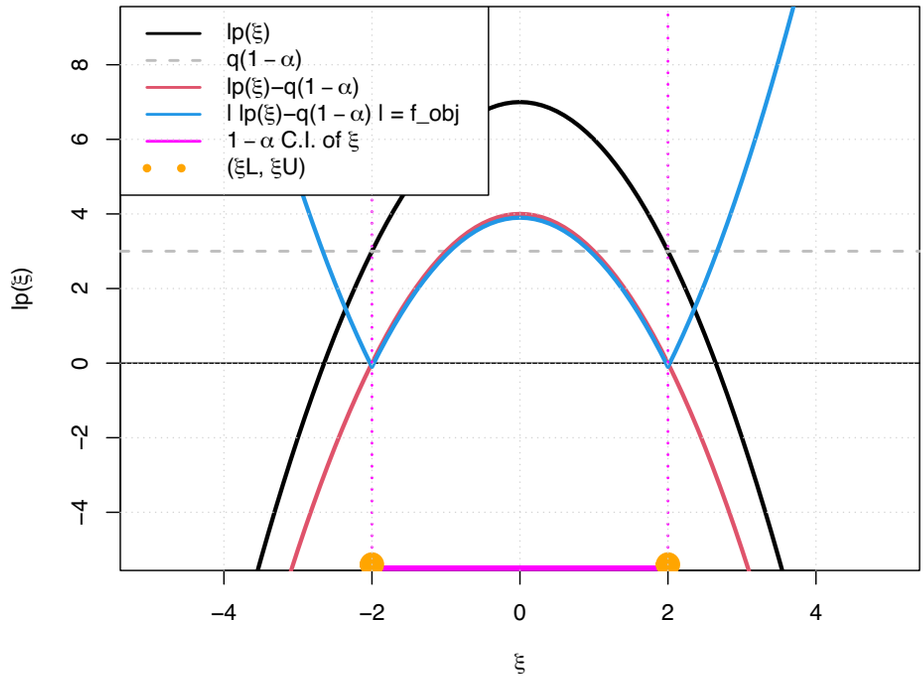

FIGURE 4.1: Graphical representation of the method by which the `ci.tweedie` function computes the bounds (ξ_L, ξ_U) of the confidence interval $\hat{\Xi}_{1-\alpha}(y)$. The function represented in the plot is not a real loglikelihood: simpler functions have been used to make the graph easier to interpret.

algorithm's performance: the ideal sequence is sufficiently dense, contains the confidence interval $\hat{\Xi}_{1-\alpha}(y)$ and is slightly larger than it. It is recommended to call the function multiple times while adjusting the sequence of values of ξ to obtain the best results. Then, `main.tweedie` computes the sequences of $(\hat{\beta}_\xi, \hat{\phi}_\xi)$ and $l_P(\xi)$ evaluated at the values of ξ chosen by the user, the MLE $(\hat{\beta}, \hat{\xi}, \hat{\phi})$ and a confidence interval $\hat{\Xi}_{1-\alpha}(y)$. If specified by the user, `main.tweedie` plots a graph of the values $(\xi, l_P(\xi))$, the location of maximum likelihood $\hat{\xi}$, the limits of the confidence interval $\hat{\Xi}_{1-\alpha}(y)$ and the $1-\alpha$ quantile line for the profile loglikelihood $q_{1-\alpha} = l(\hat{\beta}, \hat{\xi}, \hat{\phi}) - \frac{\chi^2_{1;1-\alpha}}{2}$ (see Figure 4.2).

```
main.tweedie=function(min.xi,max.xi,n.seq,
    verbose=F,do.plot=F,alpha=0.05,y,X=NULL){
  #dataset
  #if there are no covariates
  if (is.null(X)){dat=data.frame(y=y)}
  #if there are covariates
  if (!is.null(X)) {dat=data.frame(X,y=y)}
  #obtaining the model matrix
  Xmm=model.matrix(lm(y~.,data=dat))
```

```
p=ncol(Xmm)
#sequence of values of xi
xi.seq=seq(min.xi,max.xi,length=n.seq)
xi.seq =xi.seq[xi.seq!= 2]
N=length(y)
#computes the sequences
if (verbose==T){profile=profiling(xi.seq,
 verbose=T,y=y,Xmm=Xmm,dat=dat)} else {profile=profiling(xi.seq,
 verbose=F,y=y,Xmm=Xmm,dat=dat)}
#if some values are excluded from the sequences
if (profile$n.seq.new!=n.seq){cat('\n The last
 ',n.seq-profile$n.seq.new, ' values in the sequence were excluded
 since the glm function did not converge \n')}
#computes the maximum likelihood estimate
mle=mle.estimate(
  profile$Lp.seq,profile$xi.seq,profile$beta.seq,profile$phi.seq)
#quantile of level 1-alpha of the profile loglikelihood
quantile.line=ll.tw(
  mle$beta.mle,mle$xi.mle,mle$phi.mle,y,Xmm)-qchisq(1-alpha,1)/2
#computes confidence interval for xi
CI=CI.tweedie(alpha,mle$xi.mle,min.xi,max(max(profile$xi.seq)-
  10^(-2),mle$xi.mle),mle$beta.mle,mle$phi.mle,y,Xmm,dat)
if (min(profile$Lp.seq)>=quantile.line) {cat('\n The confidence
 interval is not reliable: decrease the value of min.xi=
 ',min.xi,', increase the value of max.xi=', max.xi)}
#warns the user if exact zeros are present in the sample but the
 MLE of xi is >2
if (mle$xi.mle>=2 & any(y==0)){cat('\n The computed MLE of xi is >
 2, but exact zeros are present in the sample')}
if (do.plot==T){
  #plots profile loglikelihood
  plot(profile$xi.seq,profile$Lp.seq,type='l',lwd=2,main=
      expression(paste('Profile log-likelihood for the Tweedie
 parameter ',xi)),xlab=expression(xi),ylab='lp')
  grid()
  #plots the mle of xi
  points(mle$xi.mle,mle$L.max,pch=19,col=2,lwd=2)
  abline(v=mle$xi.mle,col=2,lty=3,lwd=2)
  #plots the quantile line of the loglikelihood
  abline(h=quantile.line,col='grey60',lty=2)
  #plots confidence interval for xi, if it has been computed
  if (min(profile$Lp.seq)<quantile.line){abline(
    v=c(CI[1],CI[2]),col=4,lwd=2,lty=2)}
  #adding a legend
```

```
    legend('bottomright',legend=expression('profile
   loglik',paste('M.L.E. of ',xi),paste(1-alpha,' quantile
   line'),paste(1-alpha,' C.I. of
   ',xi)),col=c(1,2,'grey60',4),lty=c(1,3,2,2),lwd=2)}
  returning_list=list(xi.mle=mle$xi.mle,xi.lower=CI[1],xi.upper=CI[2],
    L.max=mle$L.max,Lp.seq=profile$Lp.seq,n.seq.new=profile$n.seq.new)
  return(returning_list)}
```

Listing 4.10: Main function

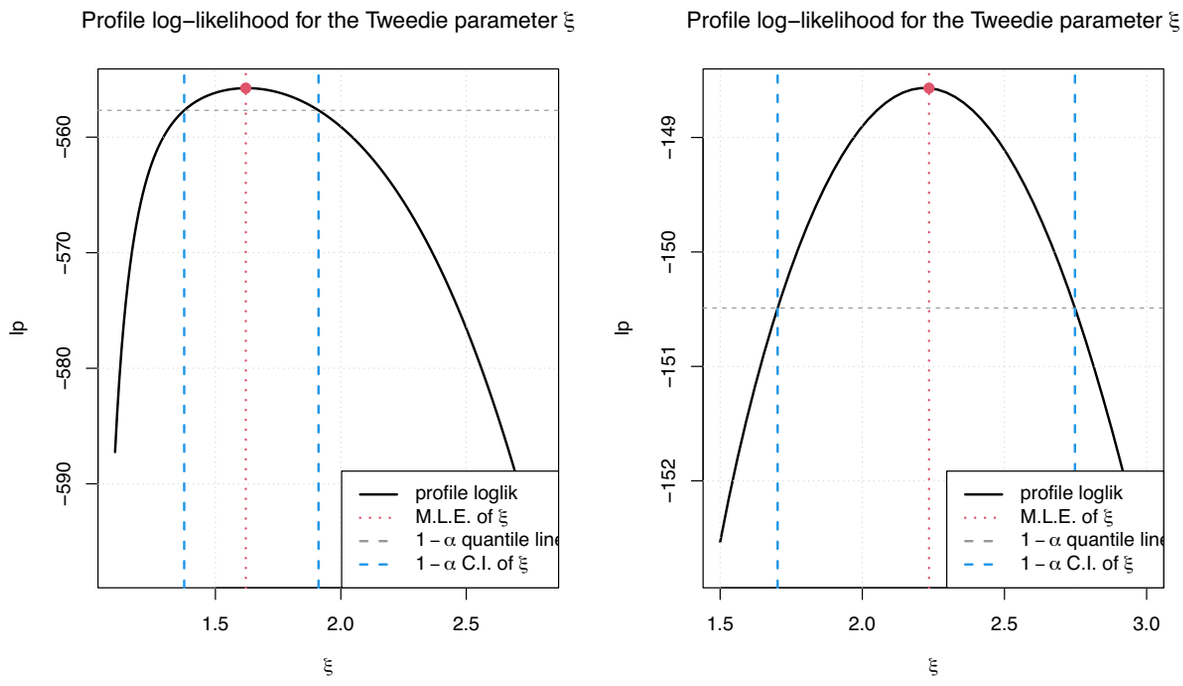

Figure 4.2: Graph plotted by the main function for two examples with generated data without covariates where true values of ξ are 1.8 and 2.4 respectively.

Chapter 5

Simulation studies

In Statistics, simulation studies are commonly utilised to evaluate the properties of inference procedures. In this context, our objective is to examine the performance of the maximum likelihood estimator (MLE) of ξ, as computed by the function `main.tweedie` from Section 4.10. Under the notation introduced in Section 4.1, we consider two different scenarios:

- Model without covariates: the response variable y is a realisation of the independent and identically distributed random variables $Y = [Y_1, \ldots, Y_n]$, where $Y_i \sim TW_\xi(\mu_i, \phi)$, which all share a common mean $\mu_i = e^\beta, \quad i = 1, \ldots, n$. In our experiment, the intercept coefficient is fixed at $\beta = -0.5$ and the mean vector is computed as $\mu^\intercal = [e^{-0.5}, \ldots, e^{-0.5}]$.
- Model with covariates: the response variable y is a realisation of the independent (but not identically distributed) random variables $Y = [Y_1, \ldots, Y_n]$, where $Y_i \sim TW_\xi(\mu_i, \phi)$, where the mean $\mu_i = e^{x_i\beta}$ varies with the index i. Before the simulation is initiated, the values of the predictor variables are generated. Two covariates are considered: a real-valued quantitative variable $x_1^\intercal = [x_{11}, \ldots, x_{n1}]$ (which is the realisation of a standard normal $N(0, 1)$) and a dummy variable $x_2^\intercal = [x_{12}, \ldots, x_{n2}]$ (which is the realisation of a Bernoulli $Ber(0.4)$). An intercept is included, $x_0 = 1_n$. Subsequently, the regression coefficients are fixed at $\beta = (2, -2, -1)$ and the mean vector $\mu^\intercal = [\mu_1, \ldots, \mu_n]$ is computed as $\mu = X\beta$, where $X = [x_0, x_1, x_2]$ is the model matrix. This implies that the model matrix X and the mean vector μ are the same for all simulations.

In both cases, the dispersion coefficient is fixed at $\phi = 0.5$. Reasonably, the estimator's performance should be influenced by the true (and usually unknown, outside of simulation studies) value of ξ and by the sample size. Consequently, we consider six different

values of the power parameter, $\xi_{seq} = (\xi_1, \ldots, \xi_6) = (1.2, 1.5, 1.8, 2, 2.3, 2.6)$, and two different sample sizes, $n_{seq} = (n_1, n_2) = (30, 100)$. Let

- $\hat{\xi}(Y)$ be the maximum likelihood estimator of ξ, as computed by the function `main.tweedie`,
- $\hat{\Xi}_{1-\alpha}(Y)$ be the confidence interval with nominal level $1-\alpha$ for ξ (4.13), here expressed as a function of the random vector Y, instead of the data y.

Then, we are interested in examining the following properties of $\hat{\xi}(Y)$ and $\hat{\Xi}_{1-\alpha}(Y)$.

- The bias of the estimator $B = E_\xi[\hat{\xi}(Y)] - \xi$,
- The square root of the mean squared error of the estimator $RMSE = \sqrt{E_\xi[(\hat{\xi}(Y) - \xi)^2]}$,
- The probability of underestimating the power parameter $PU = P_\xi(\hat{\xi}(Y) < \xi)$,
- The actual coverage of the confidence interval $C_{true} = P_\xi(\xi \in \hat{\Xi}_{1-\alpha}(Y))$.

Let us consider the case without covariates: let $\phi = 0.5$ be the dispersion parameter, $\beta = -0.5$ be the sole regression coefficient and $1 - \alpha = 0.95$ be the nominal coverage of the confidence intervals. For each pair of values (ξ_k, n_j), where $k \in \{1, \ldots, 6\}$, $j \in \{1, 2\}$, the simulation is run $Nsim = 10^4$ times. Then, the generating mechanism is as follows:

- The mean vector is computed as $\mu_j^\intercal = [\mu_{j1}, \ldots, \mu_{jn_j}]$, where $\mu_{ji} = e^\beta$, $i = 1, \ldots, n_j$.
- N_{sim} samples $y_1^*, \ldots, y_{N_{sim}}^*$ are generated, using the `rtweedie` function, from $Y = [Y_1, \ldots, Y_{n_j}]$, where $Y_i \sim TW_{\xi_k}(\mu_{ji}, \phi)$.
- The estimates of the power parameter in each generated sample $\hat{\xi}(y_1^*), \ldots, \hat{\xi}(y_{N_{sim}}^*)$ are computed using the function `main.tweedie`.
- The confidence intervals for the power parameter in each generated sample $\Xi_{1-\alpha}(y_1^*), \ldots, \Xi_{1-\alpha}(y_{N_{sim}}^*)$ are computed using the function `main.tweedie`.
- The bias is estimated as $\hat{B} = \sum_{w=1}^{N_{sim}} \frac{\xi_k - \hat{\xi}(y_w^*)}{N_{sim}}$.
- The square root of the mean squared error is estimated as $\widehat{RMSE} = \sqrt{\sum_{w=1}^{N_{sim}} \frac{(\xi_k - \hat{\xi}(y_w^*))^2}{N_{sim}}}$.
- The probability of underestimating the power parameter is estimated as $\widehat{PU} = \sum_{w=1}^{N_{sim}} \frac{I(\hat{\xi}(y_w^*) < \xi_k)}{N_{sim}}$, where $I(\hat{\xi}(y_w^*) < \xi_k) = 1$ if $\hat{\xi}(y_w^*) < \xi_k$ and is 0 otherwise,

- The actual coverage of the confidence interval is estimated as $\hat{C}_{true} = \sum_{w=1}^{N_{sim}} \frac{I(\xi_k \in \hat{\Xi}_{1-\alpha}(y_w^*))}{N_{sim}}$, where $I(\xi_k \in \hat{\Xi}_{1-\alpha}(y_w^*)) = 1$ if $\xi_k \in \hat{\Xi}_{1-\alpha}(y_w^*)$ and is 0 otherwise.
- The number of times the function `main.tweedie` failed to converge is also recorded, along with the number of times the computed CI's limits were inaccurate. As illustrated in Section 4.4.4, the function `main.tweedie` computes unreliable limits of the confidence interval when `min.xi` and `max.xi`, the limits of the user-selected sequence of values of ξ given as input to the `main.tweedie` function, are not sufficiently spaced apart, or when the `glm` function fails to converge multiple times.

The case with covariates follows a similar procedure, except that $\beta = (-2, 2, 1)$ and $\mu_{ji} = \exp\{-2 + 2x_{i1} + x_{i2}\}$, $i = 1, \ldots, n_j$ and $j = 1, 2$. It is important to note that the iterations where the function `main.tweedie` failed to converge are excluded from the results, since `main.tweedie` failed to compute the MLE $\hat{\xi}(y)$ and the CI $\hat{\Xi}_{1-\alpha}(y)$. On the other hand, the cases where the function converged but computed an inaccurate confidence interval are included in the results, since this issue might occur frequently, particularly for small values of ξ, and contribute to calculating the true coverage C_{true}. The code used to perform the simulations can be found in the appendix. In the following, we provide the results of the simulations rounded to the fourth significant digit.

TABLE 5.1: Simulation results for the case without covariates. For each value of ξ considered and for two different sample sizes, the table contains an estimate of the bias B, of the mean squared error's square root $RMSE$ and of the probability of underestimating the power parameter PU. It also presents an estimate of the true coverage C_{true} of the confidence interval with nominal level 0.95, the proportion of failed iterations $\%ERR$ and the percentage of accurate lower and upper limits of the CI, $\%\xi_L$ and $\%\xi_U$, respectively.

ξ	n	$\hat{B}$	$\widehat{RMSE}$	$\widehat{PU}$	$\widehat{C}_{true}$	$\%ERR$	$\%\ \xi_L$	$\%\ \xi_U$
1.2	30	-0.02869	0.09905	0.6383	0.8967	0.0000	58.52	99.79
1.5	30	0.05852	0.3092	0.4800	0.9000	0.0000	94.38	97.90
1.8	30	0.1954	0.4600	0.3872	0.9611	0.0000	87.04	97.71
2.0	30	0.1663	0.5584	0.4070	0.9620	0.0000	76.43	95.17
2.3	30	0.1414	0.7113	0.4422	0.9555	0.0000	65.82	94.82
2.6	30	0.05869	0.9451	0.4681	0.9429	0.0000	83.87	87.37
1.2	100	-0.00902	0.04880	0.5753	0.9405	0.0000	98.15	100 .0
1.5	100	-0.00296	0.08516	0.5020	0.9448	0.0000	100.0	99.51
1.8	100	0.07546	0.1901	0.3591	0.9537	0.0000	100.0	99.90
2.0	100	0.05947	0.2504	0.4329	0.9535	0.0000	99.70	99.66
2.3	100	0.05156	0.3318	0.4430	0.9480	0.0000	98.32	99.84
2.6	100	0.04555	0.4048	0.4461	0.9458	0.0000	99.38	100.0

Table 5.1 presents the simulation's results for the scenario without covariates. None of the iterations failed, which implies that the obtained results are credible and the `main.tweedie` function performs effectively. The estimator $\hat{\xi}(Y)$ appears to be biased since it tends to underestimate when ξ is small and overestimate otherwise.

For $n = 100$, the upper and lower limits of the computed confidence intervals are accurate in the vast majority of replications. However, for $n = 30$, issues arose in computing the lower limit of CIs. For $\xi = 1.2$, the lower limit is accurate only in 58% of the cases. This issue likely arises from the sequence of ξ values given as input to the `main.tweedie` function: the minimum value in the sequence, `min.xi`, had to be slightly greater than 1 since the profile log-likelihood $l_P(\xi)$ might develop a spurious local maximum at $\xi = 1$ in some circumstances (Dunn & Smyth, 2005). Subsequently, since the true value of ξ is small, the true lower limit of the confidence interval was likely lower than `min.xi`, resulting in smaller computed confidence intervals, having coverage remarkably below the nominal level. However, $\%\xi_L$ is also quite low for $\xi = 1.8, 2, 2.3, 2.6$, even though `min.xi` was set to $1.1, 1.1, 1.1$ and 1.05 respectively. When $n = 100$, $\hat{C}_{true}$ approaches the nominal value 0.95 in all scenarios, indicating that the approximating χ^2_1 distribution becomes more accurate as n increases.

TABLE 5.2: Simulation results for the case with covariates. For each value of ξ considered and for two different sample sizes, the table contains an estimate of the bias B, the squared root of the mean squared error $RMSE$ and the probability of underestimating the power parameter PU of the maximum likelihood estimator $\hat{\xi}(Y)$. It also presents an estimate of the true coverage of the confidence interval C_{true} with nominal level 0.95, the proportion of failed iterations $\%ERR$ and the percentage of accurate computed values of the lower and upper limits of the confidence interval, $\%\xi_L$ and $\%\xi_U$, respectively.

ξ	n	$\hat{B}$	$\widehat{RMSE}$	$\widehat{PU}$	$\widehat{C}_{true}$	$\%ERR$	$\%\ \xi_L$	$\%\ \xi_U$
1.2	30	-0.05102	0.09984	0.6911	0.8406	0.0000	50.37	99.99
1.5	30	-0.03275	0.09638	0.5985	0.9278	0.0000	99.62	99.94
1.8	30	0.03571	0.1273	0.4223	0.9377	0.0106	98.94	97.95
2.0	30	0.03064	0.1698	0.4747	0.9314	0.0003	99.97	98.25
2.3	30	0.05355	0.2049	0.4254	0.9193	0.0002	99.50	95.80
2.6	30	0.08168	0.2376	0.3674	0.9001	0.0005	99.45	97.52
1.2	100	-0.01329	0.04606	0.5962	0.9328	0.0000	97.76	100.0
1.5	100	-0.01051	0.04773	0.5799	0.9404	0.0000	100.0	100.0
1.8	100	0.01463	0.06652	0.4193	0.9431	0.0246	97.54	97.54
2.0	100	0.007350	0.08988	0.4839	0.9447	0.0000	100.0	100.0
2.3	100	0.01227	0.1077	0.4619	0.9391	0.0005	99.95	99.91
2.6	100	0.01847	0.1259	0.4561	0.9259	0.0023	99.51	99.60

Table 5.2 shows the results for the case with covariates. Very few iterations failed, suggesting that the obtained results are reliable. These simulations confirm that $\hat{\xi}(Y)$ is biased since it tends to underestimate when ξ is small and overestimate otherwise. The upper and lower limits of the computed confidence intervals are generally accurate. However, when $\xi = 1.2$ and $n = 30$, the lower limit is accurate only in half of the cases, similar to the case without covariates. In the other scenarios, the estimated true coverage of $\hat{\Xi}_{1-\alpha}(Y)$ was also below the nominal value 0.95 when $n = 30$. However, when $n = 100$, $\hat{C}_{true}$ approaches the nominal value 0.95, confirming that the approximating χ^2_1 distribution becomes more accurate as n increases.

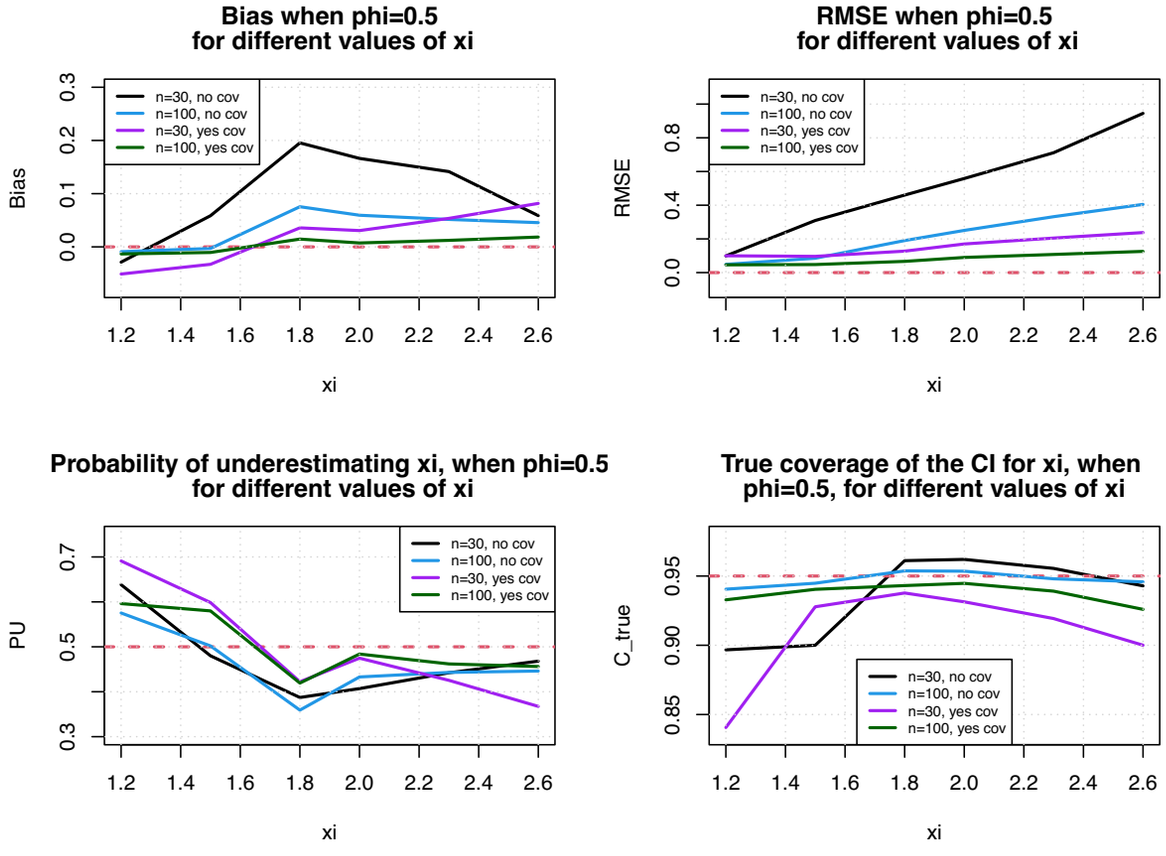

FIGURE 5.1: Graphical representation of the simulations' results.

Figure 5.1 summarises the results of all simulations. The estimator $\hat{\xi}(Y)$ performs better with large sample sizes and when covariates are included in the model, despite the presence of more nuisance parameters. The bias approaches 0 for large sample sizes, confirming that $\hat{\xi}(Y)$ is asymptotically unbiased. In all considered scenarios, when ξ is small, the bias is negative, resulting in a high probability of underestimation. Conversely, as ξ increases, the bias becomes positive and the probability of underestimation decreases below the target value of 0.5.

In all cases, the true coverage of the confidence interval approaches 0.95 for intermediate values of the power parameter, whereas $\hat{C}_{true}$ is lower for more extreme values of ξ. The case with covariates and $n = 30$ returns confidence intervals with true coverage furthest from the nominal value. Thus, the inclusion of covariates seems detrimental in achieving coverage close to the nominal one.

Conclusion

While the initial chapters focus on topics that have already been extensively researched, little has been written on the properties and performance of the maximum likelihood estimator of the power parameter ξ and its confidence interval based on the profile log-likelihood, at least to our knowledge. The discussions presented in Chapter 4 and Chapter 5 are not exhaustive. More detailed simulations are necessary, considering different types of covariates, additional sample sizes and larger values of ξ. Furthermore, in all simulations presented in Chapter 5, the dispersion parameter ϕ was fixed at 0.5: varying this parameter could lead to interesting results. Since the function `tweedie.profile`, unlike `main.tweedie`, is implemented in R and widely accessible, conducting similar simulations to the ones presented in Chapter 5 while using that function could be beneficial.

The function `main.tweedie` can be improved: it is not optimised and sometimes computes unreliable confidence intervals. Repeated trials have shown that the performance of the `main.tweedie` function depends on the sequence of values of ξ given as input: identifying optimal criteria for this sequence and setting it as the default could improve the function's performance, especially in simulations.

However, it is important to note that, in practical applications, the function `main.tweedie` might perform better. As discussed in Section 4.4.4, calling the function multiple times while adjusting the sequence of values of ξ given as input is recommended to achieve optimal performance. This was not possible during the simulations, as they were automated and the sequences of ξ values had been fixed priorly.

Appendix

```
library(tweedie)
library(statmod)

ll.tw=function(beta,xi,phi,y,Xmm){
  #linear predictor
  eta<-Xmm%*%as.matrix(beta,ncol=1)
  #mean of the response
  mu=exp(eta)
  #loglikelihood
  value=sum(log(dtweedie(y,xi=xi,mu=mu,phi=phi)))
  return(value)}

f.beta.mle.cons=function(xi,y,dat){
  #if the response is not normal
  if (xi!=0){
    #tweedie glm with power parameter xi and logarithmic link function
    glm.fit<-glm(y~.,family=tweedie(var.power = xi,link.power =
   0),data=dat)
    beta.mle.cons=coef(glm.fit)}
  #if the response is normal
  if (xi==0){
    #linear normal regression model, identity link function
    glm.fit<-glm(y~.,data=dat)
    beta.mle.cons=coef(glm.fit)}
  return(beta.mle.cons)}

f.phi.mle.cons=function(xi,beta,y,Xmm){
  #lower limit
  phi.lower=10^(-4)
  #upper limit: mean deviance estimator x 10
  eta=Xmm%*%as.matrix(beta,ncol=1)
  phi.upper=10*sum(tweedie.dev(y = y, mu = exp(eta), power =
   xi))/length(y)
  if (is.nan(phi.upper)){phi.upper=10^3}
```

```
  if (phi.upper==Inf){phi.upper=10^3}
  #call to the optimiser
  opt.phi.call<-optimize(ll.tw,interval=c(phi.lower,phi.upper),
                         xi=xi,beta=beta,y=y,Xmm=Xmm,maximum = T)
  #constrained estimate of phi
  phi.mle.cons=opt.phi.call$maximum
  return(phi.mle.cons)}

llp.tw=function(phi.mle.cons,beta.mle.cons,xi,y,Xmm)
   {ll.tw(beta.mle.cons,xi,phi.mle.cons,y,Xmm)}

profiling=function(xi.seq,verbose=F,y,Xmm,dat){
  min.xi=min(xi.seq)
  max.xi=max(xi.seq)
  n.seq=length(xi.seq)
  p=ncol(Xmm)
  #preallocation of vectors
  phi.seq=rep(NA,n.seq)
  beta.seq=matrix(NA,nrow=n.seq,ncol=p)
  Lp.seq<-rep(NA,n.seq)
  #for each value xi of the sequence
  for (i in 1:n.seq){
    #the value of xi is fixed
    xi=xi.seq[i]
    tryCatch(
    {#this code is run if glm does not raise an error
    #constrained estimate of beta
    beta.seq[i,]=f.beta.mle.cons(xi,y=y,dat)
    #constrained estimate of phi
    phi.seq[i]=f.phi.mle.cons(xi=xi,beta=beta.seq[i,],y=y,Xmm)
    #profile loglikelihood
    Lp.seq[i]=llp.tw(phi.seq[i],beta.seq[i,],xi,y=y,Xmm)
    },
    #this code is run f glm raises an error
    error = function(e) {
      if (verbose==T){cat("\n glm raised an error in iteration", i,
   "for xi =", xi.seq[i], ":", conditionMessage(e))}})}
  #removing the values that caused the error
  #indices of the values of xi the caused the error
  idx=as.numeric(which(apply(beta.seq, 1, function(row)
   all(is.na(row)))))
  #if glm raised an error at least once
  if (length(idx)!=0){
    #sequences without the problematic values
```

```
    xi.seq.new=xi.seq[-idx]
    beta.seq=beta.seq[-idx,]
    phi.seq=phi.seq[-idx]
    Lp.seq=Lp.seq[-idx]
    #new lenght of the sequences
    n.seq.new=length(xi.seq.new)
  }
  #if glm never raised an error
  if (length(idx)==0){
    #the length of the sequence stays the same
    n.seq.new=length(xi.seq)
    #the sequence of values of xi stays the same
    xi.seq.new=xi.seq}
  #returns a list of the computed values
returned_list=list(Lp.seq=Lp.seq,xi.seq=xi.seq.new,
    beta.seq=beta.seq,phi.seq=phi.seq,n.seq.new=n.seq.new)
  return(returned_list)}

mle.estimate=function(Lp.seq,xi.seq,beta.seq,phi.seq){
  #index that maximises the loglikelihood
  L.max.idx<-which.max(Lp.seq[which(is.finite(Lp.seq))])
  #maximised value of the loglikelihood
  L.max<-Lp.seq[L.max.idx]
  #mle of xi
  xi.mle<-xi.seq[L.max.idx]
  #mle of beta
  beta.mle<-beta.seq[L.max.idx,]
  #mle of phi
  phi.mle<-phi.seq[L.max.idx]
  returned_list=list(L.max.idx=L.max.idx,L.max=L.max,xi.mle=xi.mle,
    beta.mle=beta.mle,phi.mle=phi.mle)
  return(returned_list)
}

CI.tweedie=function
    (alpha=0.05,xi.mle,min.xi,max.xi,beta.mle,phi.mle,y,Xmm,dat)
{#objective function
  f.obj=function(xi,dat){
    beta.mle.cons=rep(-10^4,ncol(Xmm))
    tryCatch(
      {#this code is run if glm does not raise an error
        #constrained estimate of beta
        beta.mle.cons=f.beta.mle.cons(xi,y=y,dat=dat)})
  #constrained estimate of xi
```

```
  phi.mle.cons=f.phi.mle.cons(xi=xi,beta=beta.mle.cons,y=y,Xmm=Xmm)
  #quantile of the profile loglikelihood
  quantile.level<-
    ll.tw(beta.mle,xi.mle,phi.mle,y=y,Xmm=Xmm)-(qchisq(1-alpha,1)/2)
  #value of the objective function
  value<-abs
    (llp.tw(phi.mle.cons,beta.mle.cons,xi,y=y,Xmm=Xmm)-quantile.level)
  return(value)}
  #lower limit of the CI
  xi.lower=NA
  xi.lower=tryCatch(optimize(f.obj,interval=

   c(min.xi-xi.mle/1000,xi.mle),dat=dat)$minimum)
  #upper limit of the CI
  xi.upper=NA
  xi.upper=tryCatch(optimize(
    f.obj,interval=c(xi.mle,max.xi+xi.mle/1000),dat=dat)$minimum)
  #checking if the limits of the confidence interval are reliable
  if (round(xi.lower,1)==round(min.xi-xi.mle/1000,1)){lower.ok=0}
  if (round(xi.upper,1)==round(max.xi+xi.mle/1000,1)){upper.ok=0}
  return(c(xi.lower,xi.upper,lower.ok,upper.ok))}

main.tweedie=function(min.xi,max.xi,n.seq,
   verbose=F,do.plot=F,alpha=0.05,y,X=NULL){
  #dataset
  #if there are no covariates
  if (is.null(X)){dat=data.frame(y=y)}
  #if there are covariates
  if (!is.null(X)) {dat=data.frame(X,y=y)}
  #obtaining the model matrix
  Xmm=model.matrix(lm(y~.,data=dat))
  p=ncol(Xmm)
  #sequence of values of xi
  xi.seq=seq(min.xi,max.xi,length=n.seq)
  xi.seq =xi.seq[xi.seq!= 2]
  N=length(y)
  #computes the sequences
  if (verbose==T){profile=profiling(xi.seq,
   verbose=T,y=y,Xmm=Xmm,dat=dat)} else {profile=profiling(xi.seq,
   verbose=F,y=y,Xmm=Xmm,dat=dat)}
  #computes the maximum likelihood estimate
  mle=mle.estimate(
    profile$Lp.seq,profile$xi.seq,profile$beta.seq,profile$phi.seq)
  #quantile of level 1-alpha of the profile loglikelihood
```

```
  quantile.line=ll.tw(
    mle$beta.mle,mle$xi.mle,mle$phi.mle,y,Xmm)-qchisq(1-alpha,1)/2
  #computes confidence interval for xi
  CI=CI.tweedie(alpha,mle$xi.mle,min.xi,max(max(profile$xi.seq)-
    10^(-2),mle$xi.mle),mle$beta.mle,mle$phi.mle,y,Xmm,dat)
  if (do.plot==T){
    #plots profile loglikelihood
    plot(profile$xi.seq,profile$Lp.seq,type='l',lwd=2,main=
        expression(paste('Profile log-likelihood for the Tweedie
   parameter ',xi)),xlab=expression(xi),ylab='lp')
    grid()
    #plots the mle of xi
    points(mle$xi.mle,mle$L.max,pch=19,col=2,lwd=2)
    abline(v=mle$xi.mle,col=2,lty=3,lwd=2)
    #plots the quantile line of the loglikelihood
    abline(h=quantile.line,col='grey60',lty=2)
    #plots confidence interval for xi, if it has been computed
    if (min(profile$Lp.seq)<quantile.line){abline(
      v=c(CI[1],CI[2]),col=4,lwd=2,lty=2)}
    #adding a legend
    legend('bottomright',legend=expression('profile
   loglik',paste('M.L.E. of ',xi),paste(1-alpha,' quantile
   line'),paste(1-alpha,' C.I. of
   ',xi)),col=c(1,2,'grey60',4),lty=c(1,3,2,2),lwd=2)}
  returning_list=list(xi.mle=mle$xi.mle,xi.lower=CI[1],xi.upper=CI[2],
    L.max=mle$L.max,Lp.seq=profile$Lp.seq,n.seq.new=profile$n.seq.new)
  return(returning_list)}
```

LISTING .1: Functions from Chapter 4 that compute the MLE and CI of ξ, slightly modified to improve efficiency.

```
#WITHOUT COVARIATES
library(parallel)
#number of elements in the sequence of values of xi
n.seq=100
#true xi values
xi.true.values=c(1.2, 1.5, 1.8, 2.0, 2.3, 2.6)
#SAMPLE SIZE: 100
N=100
#phi
phi.true=0.5
#true xi values
xi.true.values=c(1.2, 1.5, 1.8, 2.0, 2.3, 2.6)
min.xi.values=c(1.01, 1.10, 1.1, 1.1, 1.1, 1.1)
```

```
max.xi.values=c(1.7, 2, 3.2, 3.5, 4.2, 4.9)
#if N=30, the limits of the sequence would be
#min.xi.values=c(1.01, 1.01, 1.1, 1.1, 1.1,1.05)
#max.xi.values=c(1.75, 3.5, 4.5, 4.7, 5.5, 5.7)
#matrix of results
results100=data.frame(xi.true=xi.true.values,sample.size=rep(100,6),
    bias=NA,rmse=NA,underest=NA,coverage=NA,n.errors=NA,
    reliable.lower=NA,reliable.upper=NA)
results100
#value of the regression coefficient
beta.true=-0.5
#linear predictor
eta.true<-rep(beta.true,N)
#mean vector
mu.true=exp(eta.true)
#preallocation of vectors
min.xi=0
max.xi=0
main.values.mc=rep(NA,Nsim)
xi.values.mc=rep(NA,Nsim)
CI.values.mc=matrix(0,nrow=2,ncol=Nsim)
#SIMULATION
#vaue of xi chosen
xi.idx=1
xi.true=xi.true.values[xi.idx]
min.xi=min.xi.values[xi.idx]
max.xi=max.xi.values[xi.idx]
i=xi.idx
cat('\n True xi: ', xi.true, ', Sample size: ',N, 'Min xi: ', min.xi,
   'Max:xi ', max.xi, '\n')
#Simulation function
simulation_function=function(seed){
tryCatch({main.tweedie(min.xi,max.xi,n.seq,do.plot=F, alpha=0.05,
    y=rtweedie(N,xi=xi.true,mu=as.numeric(mu.true),phi=phi.true))},
                                              error = function(e)
   {message("Caught an error:",e$message)
                                              return(rep(NA,5))})}
Nsim=10^4
#chosen seeds
seeds <- 2026 + 1:Nsim
#Parallelisation
#exporting the necessary variables out of the global environment
tutte_var <- ls(envir = .GlobalEnv)
#using all cores except one
```

```
numCores <- detectCores() - 1
cl <- makeCluster(numCores)
#exporting the necessary libraries
clusterEvalQ(cl, {
  library(tweedie)
  library(statmod)
})
clusterExport(cl, varlist = tutte_var)
#simulation with parLapply
start=Sys.time()
main.cal1=parLapply(cl,seeds,simulation_function)
end=Sys.time()
end-start
#storing the results into a matrix
main.values.mc<-do.call(cbind, main.call)
vec=as.numeric(main.values.mc)
main.values.mc.mat=matrix(vec,ncol=5,nrow=10000,byrow=T )
View(main.values.mc.mat)
```

LISTING .2: Example of simulation for the case without covariates. The calculations have been parallelised using the library `parallel` (Ripley et al., 2011). In this case, the sample size is $n = 100$ and the true value of the power parameter is $\xi = 1.2$. The fixed seeds for each simulation iteration are $2026, 2027, \ldots, 12026$. It is advisable to save the results of each simulation in a file.

```
#WITH COVARIATES
library(parallel)
#number of simulations
Nsim=10^4
#number of elements in the sequence of values of xi
n.seq=100
#SAMPLE SIZE: 30
N=30
#phi
phi.true=0.5
#true xi values
xi.true.values=c(1.2, 1.5, 1.8, 2.0, 2.3, 2.6)
min.xi.values=c(1.01, 1.10, 1.20, 1.30, 1.60, 1.80)
max.xi.values=c(1.7, 1.9, 2.7, 2.9, 3.2, 3.7)
#if N=100, the limits of the sequences would be
# min.xi.values=c(1.00, 1.10, 1.30, 1.60, 1.75, 2.00)
# max.xi.values=c(1.55, 1.80, 2.30, 2.50, 2.90, 3.20)
#matrix of results
results30=data.frame(xi.true=xi.true.values,sample.size=rep(30,6),
```

```
    bias=NA,mse=NA,underest=NA,coverage=NA,n.errors=NA,
    reliable.lower=NA,reliable.upper=NA)
results30
#generating the covariates
set.seed(2024)
x1=rnorm(N)
set.seed(2024)
x2=rbinom(N,1,0.4)
X=data.frame(x1=x1,x2=x2)
head(X)
#true value of the regression coefficients
p=ncol(X)+1
beta.true<-c(-2,1,2)
dat=data.frame(X,y=1:N)
#obtaining the model matrix
Xmm=model.matrix(lm(y~.,data=dat))
#linear predictor
eta.true<-Xmm%*%as.matrix(beta.true,ncol=1)
#mean vector
mu.true=exp(eta.true)
#preallocation of vectors
min.xi=0
max.xi=0
main.values.mc=rep(NA,Nsim)
xi.values.mc=rep(NA,Nsim)
CI.values.mc=matrix(0,nrow=2,ncol=Nsim)
#SIMULATION
#vaue of xi chosen
xi.idx=6
xi.true=xi.true.values[xi.idx]
min.xi=min.xi.values[xi.idx]
max.xi=max.xi.values[xi.idx]
i=xi.idx
cat('\n True xi: ', xi.true, ', Sample size: ',N, 'Min xi: ', min.xi,
   'Max:xi ', max.xi, '\n')
simulation_function=function(seed){tryCatch(
    {main.tweedie(min.xi,max.xi,n.seq,do.plot=F,alpha=0.05,verbose=F,

   y=rtweedie(N,xi=xi.true,mu=as.numeric(mu.true),phi=phi.true),X=X)},
                                    error = function(e)
   {message("Caught an error: ", e$message)
                                                return(rep(NA,5))})}
#number of replications
Nsim=10^4
```

```
seeds <- 2030 + 1:Nsim
#parallelising
#all variables from the global enviroment
tutte_var <- ls(envir = .GlobalEnv)
#all cores except one
numCores <- detectCores() - 1
cl <- makeCluster(numCores)
#export all necessary libraries
clusterEvalQ(cl, {
  library(tweedie)
  library(statmod)
})
clusterExport(cl, varlist = tutte_var)
#simulation with parLapply
start=Sys.time()
main.call=parLapply(cl,seeds,simulation_function)
end=Sys.time()
end-start
#storing results into a matrix
main.values.mc<-do.call(cbind, main.call)
vec=as.numeric(main.values.mc)
class(vec)
main.values.mc.mat=matrix(vec,ncol=5,nrow=10000,byrow=T )
View(main.values.mc.mat)
```

LISTING .3: Example of simulation for the case with covariates. The calculations have been parallelised using the library `parallel` (Ripley et al., 2011). In this case, the sample size is $n = 30$ and the true value of the power parameter is $\xi = 1.2$. The fixed seeds for each simulation iteration are $2030, 2031, \ldots, 12030$. It is advisable to save the results of each simulation in a file.

```
#reading the results of the simulations that have been previously
   saved

#no covariates
xi12n30=read.table('xi12n30',header=T)
xi15n30=read.table('xi15n30',header=T)
xi18n30=read.table('xi18n30',header=T)
xi2n30=read.table('xi2n30',header=T)
xi23n30=read.table('xi23n30',header=T)
xi26n30=read.table('xi26n30',header=T)
xi12n100=read.table('xi12n100',header=T)
xi15n100=read.table('xi15n100',header=T)
```

```
xi18n100=read.table('xi18n100',header=T)
xi2n100=read.table('xi2n100',header=T)
xi23n100=read.table('xi23n100',header=T)
xi26n100=read.table('xi26n100',header=T)
sim=as.matrix(cbind(xi12n30,xi15n30,xi18n30,xi2n30,xi23n30,
    xi26n30,xi12n100,xi15n100,xi18n100,xi2n100,xi23n100,xi26n100))
#with covariates
xi12n30cov=read.table('xi12n30cov',header=T)
xi15n30cov=read.table('xi15n30cov',header=T)
xi18n30cov=read.table('xi18n30cov',header=T)
xi2n30cov=read.table('xi2n30cov',header=T)
xi23n30cov=read.table('xi23n30cov',header=T)
xi26n30cov=read.table('xi26n30cov',header=T)
xi12n100cov=read.table('xi12n100cov',header=T)
xi15n100cov=read.table('xi15n100cov',header=T)
xi18n100cov=read.table('xi18n100cov',header=T)
xi2n100cov=read.table('xi2n100cov',header=T)
xi23n100cov=read.table('xi23n100cov',header=T)
xi26n100cov=read.table('xi26n100cov',header=T)
sim_cov=as.matrix(cbind(xi12n30cov,xi15n30cov,xi18n30cov,xi2n30cov,
    xi23n30cov,xi26n30cov,xi12n100cov,xi15n100cov,xi18n100cov,
    xi2n100cov,xi23n100cov,xi26n100cov))

xi.true.values=rep(c(1.2, 1.5, 1.8, 2.0, 2.3, 2.6),2)
#matrix of results
#without covariates
results=data.frame(xi.true=xi.true.values,sample.size=c(rep(30,6),
   rep(100,6)),bias=NA,rmse=NA,underest=NA,coverage=NA,n.errors=NA,
reliable.lower=NA,reliable.upper=NA)

#with covariates
results_cov=data.frame(xi.true=xi.true.values,sample.size=c(rep(30,6),
rep(100,6)),bias=NA,rmse=NA,underest=NA,coverage=NA,n.errors=NA,
reliable.lower=NA,reliable.upper=NA)
results_cov

#properties we are interested in evalueating
#probability of underestimating
prob.underest=function(xi.values.mc,xi.true){xi.true>xi.values.mc}
#bias
bias=function(xi.values.mc,xi.true){xi.values.mc-xi.true}
#squared error
se=function(xi.values.mc,xi.true){(xi.values.mc-xi.true)^2}
#true coverage
```

```
true.coverage=function(CI.values.mc,xi.true){xi.true>CI.values.mc[1,]
   & xi.true<CI.values.mc[2,]}

#without covariates
for (i in 1:12){
  xi.true=xi.true.values[i]
  SIMi=sim[,(5*(i-1)+1):(5*(i-1)+5)]
  #indices of the values of xithate caused the error
  idx=as.numeric(which(apply(SIMi,1, function(row) all(is.na(row)))))
  if (length(idx)!=0){
    #removing the values that caused the error
    SIMi<-as.matrix(SIMi[-idx,],nrow=5)
  }
  #empirical distribution of the MLE of xi
  xi.values.mc<-as.numeric(SIMi[,1])
  #empirical distribution of the CI of xi
  CI.values.mc<-rbind(SIMi[,2],SIMi[,3])
  #number of times the lower bound of the CI was not reliable
  results[i,8]=sum(SIMi[,4])/10^4
  #number of times the upper bound of the CI was not reliable
  results[i,9]=sum(SIMi[,5])/10^4
  #bias
  bias.values.mc<-bias(xi.values.mc,xi.true)
  results[i,3]<-mean(bias.values.mc)
  #mean squared error
  se.values.mc<-se(xi.values.mc,xi.true)
  results[i,4]<-sqrt(mean(se.values.mc))
  #prob of underestimating
  prob.underest.values.mc<-prob.underest(xi.values.mc,xi.true)
  results[i,5]<-mean(prob.underest.values.mc)
  #true coverage
  true.coverage.values.mc<-true.coverage(CI.values.mc,xi.true)
  results[i,6]<-mean(true.coverage.values.mc)
  #number of failed iterations
  results[i,7]<-length(idx)
}
results

#with covariates
for (i in 1:12){
  xi.true=xi.true.values[i]
  SIMi=sim_cov[,(5*(i-1)+1):(5*(i-1)+5)]
  #indices of the values of xi the caused the error
  idx=as.numeric(which(apply(SIMi,1, function(row) all(is.na(row)))))
```

```
  if (length(idx)!=0){
    #removing the values that caused the error
    SIMi<-as.matrix(SIMi[-idx,],nrow=5)
  }
  #empirical distribution of the MLE of xi
  xi.values.mc<-as.numeric(SIMi[,1])
  #empirical distribution of the CI of xi
  CI.values.mc<-rbind(SIMi[,2],SIMi[,3])
  #number of times the lower bound of the CI was not reliable
  results_cov[i,8]=sum(SIMi[,4])/10^4
  #number of times the upper bound of the CI was not reliable
  results_cov[i,9]=sum(SIMi[,5])/10^4
  #bias
  bias.values.mc<-bias(xi.values.mc,xi.true)
  results_cov[i,3]<-mean(bias.values.mc)
  #mean squared error
  se.values.mc<-se(xi.values.mc,xi.true)
  results_cov[i,4]<-sqrt(mean(se.values.mc))
  #prob of underestimating
  prob.underest.values.mc<-prob.underest(xi.values.mc,xi.true)
  results_cov[i,5]<-mean(prob.underest.values.mc)
  #true coverage
  true.coverage.values.mc<-true.coverage(CI.values.mc,xi.true)
  results_cov[i,6]<-mean(true.coverage.values.mc)
  #number of failed iterations
  results_cov[i,7]<-length(idx)/10^4
}
results_cov
```

Listing .4: The files with the simulations' results that have been previously saved are used to compute an estimate of the bias, RMSE, probability of underestimating and true coverage, for all of the scenarios considered.

Bibliography

Bassingthwaighte, J., King, R. & Roger, S. (1989). Fractal nature of regional myocardial blood flow heterogeneity. *Circulation Research* **81**, 066112.

Bonat, H. & Kokonendji, C. (2017). Flexible Tweedie regression models for continuous data. *Journal of Statistical Computation and Simulation* **87**, 2138–2152.

Box, G. E. & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **26**, 211–243.

Cox, D. R. & Snell, E. J. (1968). A general definition of residuals. *Journal of the Royal Statistical Society: Series B (Methodological)* **30**, 248–265.

Dunn, P. & Smyth, G. (2005). Series evaluation of Tweedie exponential dispersion model densities. *Statistics and Computing* **15**, 267–280.

Dunn, P. K. (2017). *tweedie: Tweedie exponential family models*. R package version 2.3.5.

Dunn, P. K. & Smyth, G. K. (1996). Randomized quantile residuals. *Journal of Computational and Graphical Statistics* **5**, 236–244.

Dunn, P. K. & Smyth, G. K. (2001). Tweedie family densities: Methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling, Odense, Denmark* , pp. 2–6.

Dunn, P. K. & Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing* **18**, 73–86.

Dunn, P. K. & Smyth, G. K. (2018). *Generalized Linear Models with Examples in R*. Springer.

Fidler, I. J. & Kripke, M. L. (1977). Metastasis results from preexisting variant cells within a malignant tumour. *Science* **197**, 893–895.

Fontaine, S., Yang, Y., Qian, W., Gu, Y. & B., F. (2020). A unified approach to sparse Tweedie modelling of multisource insurance claim data. *Technometrics* **62**, 339–356.

Foster, S. D. & Bravington, M. V. (2013). A Poisson–gamma model for analysis of ecological non-negative continuous data. *Environmental and Ecological Statistics* **5**, 181–202.

Gilchrist, A. & Drinkwater, D. (2000). The use of the Tweedie distribution in statistical modelling. *Proceedings in Computational Statistics ; 14th Symposium Held in Utrecht, The Netherlands* , pp. 313–318.

Hasan, M. M. & Dunn, P. K. (2011). Two Tweedie distributions that are near-optimal for modelling monthly rainfall in Australia. *International Journal of Climatology* **31**, 1389–1397.

Hasan, M. M. & Dunn, P. K. (2012). Understanding the effect of climatology on monthly rainfall amounts in Australia using Tweedie glms. *International Journal of Climatology* **32**, 1006–1017.

Hasan, M. M. & Dunn, P. K. (2015). Seasonal rainfall totals of Australian stations can be modelled with distributions from the Tweedie family. *International Journal of Climatology* **35**, pp. 3093–3101.

Jorgensen, B. (1997). *The Theory of Dispersion Models.* Chapman & Hall.

Jorgensen, B. & Kendal, W. (2011a). Taylor's power law and fluctuation scaling explained by a central-limit-like convergence. *Physical review E* **83**.

Jorgensen, B. & Kendal, W. (2011b). Tweedie convergence: A mathematical basis for Taylor's power law, noise, and multifractality. *Physical review E* **84**, 066120.

Jorgensen, B. & Kendal, W. (2015). A scale invariant distribution of the prime numbers. *Computation* **3**, 528–540.

Jorgensen, B. & Smyth, G. K. (2002). Fitting Tweedie's compound Poisson model to insurance claims data: dispersion modelling. *ASTIN Bulletin: The Journal of the IAA* **32**, 143–157.

Kendal, W. (2003). An exponential dispersion model for the distribution of human single nucleotide polymorphisms. *Molecular Biology and Evolution,* **20**, 579–590.

Kendal, W. S. (1995). A probabilistic model for the variance to mean power law in ecology. ecological modelling. *Ecological Modelling* **80**, 293–297.

Kendal, W. S. (2004). A scale invariant clustering of genes on human chromosome 7. *BMC Evolutionary Biology* **81**, 066112.

Kendal, W. S. & Frost, P. (1987). Experimental metastasis: a novel application of the variance-to-mean power function. *Journal of the National Cancer Institute* **79**, 1113–1115.

McKay Altman, R., Harari, O., Moisseeva, N. & Steyn, D. (2023). Statistical modelling of the annual rainfall pattern in Guanacaste, Costa Rica. *Water* **15**, 700.

Morris, C. N. (1983). Natural exponential families with quadratic variance functions: statistical theory. *The Annals of Statistics II* , 515–529.

Nelder, J. A. & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society A* **135**, 370–384.

Pace, L. & Salvan, A. (1997). *Principles of Statistical Inference: From a Neo-Fisherian Perspective*. World Scientific Pub Co Inc.

Pace, L. & Salvan, A. (2001). *Introduzione alla Statistica. Vol. 2: Inferenza, Verosimiglianza, Modelli*. CEDAM.

Quijano Xacur, O. A. & Garrido, G. (2015). Generalised linear models for aggregate claims: to Tweedie or not? *European Actuarial Journal* **32**, 143–157.

Ripley, B., Tierney, L. & S., U. (2011). *parallel:Support for Parallel Computation*. R package version 2.14.0.

Salvan, A., Sartori, N. & Pace, L. (2020). *Modelli Lineari Generalizzati*. Springer.

Smyth, G. (1996). Regression analysis of quantity data with exact zeros. *In Proceedings of the Second Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management, Gold Coast, Australia* , pp. 17–19.

Smyth, G., Chen, L., Hu, Y., Dunn, P., Phipson, B. & Chen, Y. (2006). *statmod: Statistical Modeling*. R package version 1.5.0.

Taylor, R. (1961). Aggregation, variance and the mean. *Nature* **189**, 1732–735.

Tweedie, M. C. (1984). An index which distinguishes between some important exponential families. *In Statistics: Applications and new directions: Proc. Indian Statistical Institute Golden Jubilee International Conference* **579**, 579–604.

Wüthrich, M. V. & Merz, M. (2023). *Statistical Foundations of Actuarial Learning and its Applications*. Springer Nature.