

Research Article

New Link Functions for Distribution-Specific Quantile Regression Based on Vector Generalized Linear and Additive Models

V. F. Miranda-Soberanis ¹ and T. W. Yee ²

¹*School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology, New Zealand*

²*Department of Statistics, University of Auckland, New Zealand*

Correspondence should be addressed to V. F. Miranda-Soberanis; victor.miranda@aut.ac.nz

Received 7 December 2018; Accepted 22 January 2019; Published 7 May 2019

Guest Editor: Rahim Alhamzawi

Copyright © 2019 V. F. Miranda-Soberanis and T. W. Yee. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the usual quantile regression setting, the distribution of the response given the explanatory variables is unspecified. In this work, the distribution is specified and we introduce new link functions to directly model specified quantiles of seven 1-parameter continuous distributions. Using the vector generalized linear and additive model (VGLM/VGAM) framework, we transform certain prespecified quantiles to become linear or additive predictors. Our parametric quantile regression approach adopts VGLMs/VGAMs because they can handle multiple linear predictors and encompass many distributions beyond the exponential family. Coupled with the ability to fit smoothers, the underlying strong assumption of the distribution can be relaxed so as to offer a semiparametric-type analysis. By allowing multiple linear and additive predictors simultaneously, the quantile crossing problem can be avoided by enforcing parallelism constraint matrices. This article gives details of a software implementation called the VGAMextra package for R. Both the data and recently developed software used in this paper are freely downloadable from the internet.

1. Introduction

1.1. Background. Much of modern regression analysis for estimating conditional quantile functions may be viewed as starting from Koenker and Bassett [1], who offered a systematic strategy for examining how covariates influence the entire response distribution. The fundamental idea is based on the linear specification of the τ th quantile function $Q_y(\tau | \mathbf{x}) = \beta_\tau^T \mathbf{x}$ and finding $\beta_\tau \in \mathbb{R}^p$ that solves the optimization problem

$$\min_{\beta_\tau \in \mathbb{R}^p} \sum \rho_\tau(y_i - \beta_\tau^T \mathbf{x}_i), \quad (1)$$

for independent and identically distributed (i.i.d.) observations from a family of linear quantile regression models, say $y_i = \beta_\tau^T \mathbf{x}_i + \varepsilon_{i,\tau}$, $i = 1, \dots, n$. Equation (1) can be reformulated as a linear programming problem using the piecewise linear

function $\rho_\tau(u) = u \cdot [\tau - I(u < 0)]$ for $\tau \in (0, 1)$. More details can be found in Koenker [2].

In the spirit of quantile regression, the conditional distribution $Y | \mathbf{x}$ is usually unspecified, although it relies on normal-based asymptotic theory that is used for inference, whilst the assumption of homoskedasticity of the error terms $\varepsilon_{i,\tau}$ is dropped. In this paper we use an alternative approach of conditional-quantile regression based on assuming a pre-specified distribution for the response. Parametric quantile regression has some advantages over many nonparametric approaches, including overcoming the quantile crossing problem. Two examples are Noufaily and Jones [3] which is based on the generalized gamma distribution and generalized additive models for location, scale, and shape (GAMLSS; [4]). Further examples are the LMS-BCN method involving the standard normal distribution and a three-parameter Box-Cox transformation [5] and the classical method of

TABLE 1: Some VGLM/VGAM link functions. The 4th row is implemented in VGAMextra.

Functions	Links $g_j(\theta_j)$	Domain of θ_j	Link names
<code>loge()</code>	$\log \theta_j$	$(0, \infty)$	Logarithmic
<code>cloglog()</code>	$\log(-\log(1 - \theta_j))$	$(0, 1)$	Complementary log-log
<code>logit()</code>	$\log \frac{\theta_j}{1 - \theta_j}$	$(0, 1)$	Logit
<code>logffMeanlink()</code> [*]	$\text{logit}(\theta_j) - \text{cloglog}(\theta_j)$	$(0, 1)$	logffMeanlink
<code>rhobit()</code>	$\log((1 + \theta_j)/(1 - \theta_j))$	$(-1, 1)$	rhobit

^{*}This is the VGLM-link for the mean function of the logarithmic distribution.

quantile regression based on the asymmetric Laplace distribution (ALD).

Our approach uses the vector generalized linear and additive model (VGLM/VGAM; [6, 7]) framework. We develop new link functions, \mathcal{G} , for the quantile regression model

$$Y | \mathbf{x} \sim \mathcal{F}(\mathbf{x}; \theta), \quad (2)$$

$$\eta_{\tau}(\theta) = \mathcal{G}(Q_y(\tau | \mathbf{x}, \theta)), \quad (3)$$

for a vector of quantiles $\boldsymbol{\tau} = (\tau_1, \dots, \tau_L)^T$. Our methodology relies on the prespecification of the distribution \mathcal{F} . We will also show that the quantile crossing problem can be overcome by this modelling framework. Equations (2)–(3) state that the conditional distribution of the response at a given value of \mathbf{x} has a distribution involving a parameter θ and that the transformed quantile of the distribution becomes a linear predictor of the form (5). This can be achieved by defining link functions that connect (3) to (5). The reason for the linear predictors is that generalized linear modelling [8] is a very well-established method for regression modelling. GLMs are estimated by iteratively reweighted least squares (IRLS) and Fisher scoring, and this algorithm is also adopted by VGLMs and VGAMs.

The method presented in this paper differs from conventional quantile regression [1] in that we assume \mathcal{F} is known whereas the conventional case does not but use an empirical method instead to obtain the quantiles ξ_{τ} : the expectation of the check function $\rho_{\tau}(u)$ results in the property $\tau = F(\xi_{\tau})$ which defines the τ -quantile (F is the cumulative distribution function (CDF) of \mathcal{F}). In this paper we consider the \mathcal{F} s listed in Table 2.

1.2. VGLMs and VGAMs. VGLMs/VGAMs provide the engine and overall modelling framework in this work—the VGAM R package described below fits over 150 models and distributions—therefore we only sketch the details here. VGLMs are defined in terms of M linear predictors, $\boldsymbol{\eta} = (\eta_1, \dots, \eta_M)^T$, as any statistical model for which the conditional density of \mathbf{y} given a d -dimensional vector of explanatory variables, $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$ has the form

$$\mathcal{F}(\mathbf{y} | \mathbf{x}; \mathbf{B}) = h(\mathbf{y}, \eta_1, \dots, \eta_M; \mathbf{x}), \quad (4)$$

for some known function $h(\cdot)$, with $\mathbf{B} = (\boldsymbol{\beta}_1 \ \boldsymbol{\beta}_2 \ \dots \ \boldsymbol{\beta}_M)$, a $d \times M$ matrix of unknown regression coefficients. Ordinarily, $x_1 \equiv 1$ for an intercept.

In general, the η_j of VGLMs may be applied directly to the M parameters, θ_j , of any distribution, transformed if necessary, as the j th linear predictor

$$\eta_j = g_j(\theta_j) = \eta_j(\mathbf{x}) = \boldsymbol{\beta}_j^T \mathbf{x} = \sum_{k=1}^d \beta_{(j)k} x_k, \quad (5)$$

$$j = 1, \dots, M,$$

where g_j is a VGLM-parameter link function, as in Table 1 (see [6] for further choices) and $\beta_{(j)k}$ is the k th element of $\boldsymbol{\beta}_j$. Prior to this work the θ_j were ‘raw’ parameters such as location, scale, and shape parameters; however, in this present work we define them to be quantiles or a very simple function of quantiles.

In matrix form one can write $\boldsymbol{\eta} = \boldsymbol{\eta}(\mathbf{x}) =$

$$\begin{pmatrix} \eta_1(\mathbf{x}) \\ \vdots \\ \eta_M(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta}_1^T \mathbf{x} \\ \vdots \\ \boldsymbol{\beta}_M^T \mathbf{x} \end{pmatrix} = \begin{pmatrix} \beta_{(1)1} & \dots & \beta_{(1)p} \\ \vdots & \ddots & \vdots \\ \beta_{(M)1} & \dots & \beta_{(M)p} \end{pmatrix} \mathbf{x} \quad (6)$$

$$= \sum_{k=1}^d \boldsymbol{\beta}_{(k)} x_k = \mathbf{B}^T \mathbf{x},$$

where $\boldsymbol{\beta}_{(k)} = (\beta_{(1)k}, \beta_{(2)k}, \dots, \beta_{(M)k})^T$, $k = 1, \dots, d$. Sometimes, for some j , it may be required to model η_j as *intercept-only*, that is, $\eta_j = \beta_{(j)1}$, and $\beta_{(j)k} \equiv 0$ for $k = 2, \dots, d$.

VGAMs are a nonparametric extension of VGLMs, that is, (6) is generalized to

$$\boldsymbol{\eta}(\mathbf{x}) = \boldsymbol{\beta}_{(1)} + \sum_{k=2}^d \mathbf{f}_k^*(x_k) = \mathbf{H}_1 \boldsymbol{\beta}_{(1)}^* + \sum_{k=2}^d \mathbf{H}_k \mathbf{f}_k^*(x_k) \quad (7)$$

with $\mathbf{f}_k^*(x_k) = (f_{(1)k}^*(x_k), \dots, f_{(M)k}^*(x_k))^T$. Usually the component functions are estimated by splines. Here, $\mathbf{H}_1, \dots, \mathbf{H}_d$ are known full-column rank constraint matrices, and $\boldsymbol{\beta}_{(1)}$ is a vector of unknown intercepts. With no constraints at all, $\mathbf{H}_1 = \dots = \mathbf{H}_d = \mathbf{I}_M$ (the order- M identity matrix). For VGLMs, the \mathbf{f}_k are linear so that, cf. (6),

$$\mathbf{B} = (\mathbf{H}_1 \boldsymbol{\beta}_{(1)}^* \mid \mathbf{H}_2 \boldsymbol{\beta}_{(2)}^* \mid \dots \mid \mathbf{H}_d \boldsymbol{\beta}_{(d)}^*). \quad (8)$$

The \mathbf{H}_k can enforce a wide range of linear constraints such as parallelism and exchangeability.

TABLE 2: New link functions for the quantiles of some 1-parameter distributions. The selected \mathcal{G} function is also shown.

Distribution	θ	Support of y	Quantile function ξ_τ	Function \mathcal{G}	Quantile link $[\eta(\theta; \tau)]$
Exponential	λ	$(0, \infty)$	$-\frac{1}{\lambda} \log(1 - \tau)$	log-link	$\log \log[(1 - \tau)^{-1/\theta}]$
Benini	s	(y_0, ∞)	$y_0 \exp\left(\sqrt{\frac{-\log(1 - \tau)}{s}}\right)$	log-link	$\log y_0 + \sqrt{\log[(1 - \tau)^{-1/\theta}]}$
Rayleigh	b	$(0, \infty)$	$b\sqrt{-2 \log(1 - \tau)}$	log-link	$\log \theta + \frac{1}{2} \log \log[(1 - \tau)^{-2}]$
Gamma	s	$(0, \infty)$	No closed-form	log-link	$\log \text{qgamma}(\tau, \text{shape} = \theta)$
Maxwell [†]	a	$(0, \infty)$	$\sqrt{\frac{2}{a}} \cdot \text{qgamma}(\tau, 1.5)$	log-link	$\frac{1}{2} \log\left(\frac{2 \text{qgamma}(\tau, 1.5)}{\theta}\right)$
Topp-Leone [‡]	s	$(0, 1)$	$1 - \sqrt{1 - \tau^{1/s}}$	logit	$\text{logit}(1 - \sqrt{1 - \tau^{1/\theta}})$
1-par Normal [‡]	σ	\mathbb{R}	$\mu_0 \pm \sqrt{2 \cdot \theta^2 \kappa(\tau)}$	identity	$\mu_0 \pm \sqrt{2 \cdot \theta^2 \kappa(\tau)}$

[†]qgamma() is the quantile function of the standard gamma distribution in R.

[‡]logit(θ) = $\log(\theta/(1 - \theta))$.

[‡] $\kappa(\tau) = \text{erf}^{-1}(2\tau - 1)$, with erf() denoting the error function.

1.3. Estimation. VGLMs are estimated by maximum likelihood performed by IRLS using the expected information. The VGLM log-likelihood is given by

$$\ell(\boldsymbol{\eta}) = \sum_{i=1}^n w_i \ell_i\{\eta_1(\mathbf{x}_i), \dots, \eta_M(\mathbf{x}_i)\}, \quad (9)$$

for known fixed positive prior weights w_i , and a Newton-like algorithm for maximizing (9) has the form $\boldsymbol{\beta}^{(a)} = \boldsymbol{\beta}^{(a-1)} + \mathcal{F}(\boldsymbol{\beta}^{(a-1)})^{-1} \mathbf{U}(\boldsymbol{\beta}^{(a-1)})$, where \mathcal{F} is the overall expected information matrix (EIM), \mathbf{U} is the score vector, and a is the iteration number. The vector $\boldsymbol{\beta}^{(a)}$ is obtained as the solution of the generalized least squares problem $\boldsymbol{\beta}^{(a)} = \arg \min_{\boldsymbol{\beta}} \text{RSS}$, where the quantity minimized at each IRLS iteration is the weighted (or residual) sum of squares, $\text{RSS} =$

$$\sum_{i=1}^n w_i \{\mathbf{z}_i^{(a-1)} - \boldsymbol{\eta}_i^{(a-1)}\}^T \mathbf{W}_i^{(a-1)} \{\mathbf{z}_i^{(a-1)} - \boldsymbol{\eta}_i^{(a-1)}\}. \quad (10)$$

The $(M \times M)$ \mathbf{W}_i are known as the working weight matrices and they have (j, k) th element given by

$$[\mathbf{W}_i]_{j,k} = -w_i \mathbb{E} \left(\frac{\partial^2 \ell_i}{\partial \eta_j \partial \eta_k} \right). \quad (11)$$

The use of individual EIMs instead of observed information matrices means that Fisher scoring is used rather than the Newton-Raphson algorithm.

VGAMs are also estimated by IRLS, where the difference with respect to VGLMs is that a vector additive model is now fitted to the pseudo-response \mathbf{z}_i with explanatory variables \mathbf{x}_i and working weight matrices \mathbf{W}_i at each IRLS iteration. Two approaches are currently used by VGAM to estimate the component functions \mathbf{f}^* : regression splines and vector smoothing methods with vector backfitting. Rudimentary P-splines [9] are almost operational, albeit this work is not yet complete. Compared to VGLMs, the VGAM log-likelihood

includes a penalty if used with vector smoothing splines. In VGAM the objective function maximized is

$$\ell_i\{\eta_1(\mathbf{x}_i), \dots, \eta_M(\mathbf{x}_i)\} - \frac{1}{2} \sum_{k=1}^d \sum_{j=1}^{\text{ncol}(\mathbf{H}_k)} \lambda_{(j)k} \int_{a_k}^{b_k} \{f_{(j)k}^{*''}(t)\}^2 dt. \quad (12)$$

Here, the $\lambda_{(j)k}$ are nonnegative smoothing parameters, and $a_k \leq x_{ik} \leq b_k$ are endpoints covering the values of each covariate. The basic penalty approach adopted here is described in Green and Silverman Green and Silverman [10].

2. Methodology

Let $\mathcal{F}(\boldsymbol{\eta}; y, \mathbf{x})$ be a 1-parameter statistical model as in (4) parametrized by $\theta \in \Theta \subset \mathbb{R}$ for some parameter space Θ residing in $(-\infty, \infty)$. Also let $Q_y(\tau | \mathbf{x})$ be the corresponding quantile function with $\tau \in (0, 1)$. Crucially, note that (5) handles suitable transformations of θ in the linear predictor by parameter link functions. In contrast our proposal focusses on directly modelling $Q_y(\tau | \mathbf{x}, \theta)$ via a smooth and one-to-one function \mathcal{G} , in the form of

$$\eta_\tau = \mathcal{G}(Q_y(\tau | \mathbf{x}, \theta)) = \mathcal{G}^*(\theta | \mathbf{x}, \boldsymbol{\tau}), \quad (13)$$

which is to be incorporated in the VGLM/VGAM log-likelihood, namely, (9) and (12). Here, $\boldsymbol{\tau} = (\tau_1, \dots, \tau_L)^T$ is a prespecified vector of quantiles of interest. Examples of (13) are $\log \xi_\tau$, $\text{logit} \xi_\tau$, and ξ_τ .

Equation (13) is central to this work. It allows modelling choices via \mathcal{G} for the quantile function Q_y , and it represents a new modification to the VGLM/VGAM framework. Note that \mathcal{G} resembles a *link function* within the VGLM/VGAM framework as in Table 1. Two notes: first, without any loss of generality, (13) can be seen (strictly) as a function of θ since the quantiles $\boldsymbol{\tau}$ and the covariates \mathbf{x} are known. Secondly, \mathcal{G}^* is monotonic and one-to-one, as a result of the composite of \mathcal{G} and Q_y which also hold such properties. However, during the fitting process, the IRLS algorithm internally requires the inverse of \mathcal{G}^* . Working with 1-parameter distribution

at this stage eases the implementation via Fisher scoring because the inverse $(\mathcal{G}^*)^{-1}$ can be derived manually and then incorporated in the IRLS algorithm. In a few cases, the inverse of \mathcal{G}^* does not have a closed form, such as the 1-parameter gamma distribution, and an alternative iterative method is employed to approximate $(\mathcal{G}^*)^{-1}$. To achieve this efficiently, two choices are available. These are (a) `newtonRaphson.basic()` from `VGAMextra` and (b) `VGAM::bisection.basic()`, two *vectorized* implementations of the well-known Newton–Raphson and bisection algorithms, to solve the roots of a real-valued function in a given interval (a, b) . Further details are given in Section 2.2.

One advantage of this work is that the VGLM/VGAM framework can circumvent the quantile crossing problem (e.g., [2, 11], Sect. 2.5) by choosing $\mathbf{H}_1 = \mathbf{I}_M$ and $\mathbf{H}_2 = \mathbf{H}_3 = \dots = \mathbf{H}_d = \mathbf{1}_M$ (an M -vector of ones). Under this parallelism assumption the method borrows strength across the entire data set so that the additive models for η_j with respect to x_2, \dots, x_d are parallel. Each family function in Tables 2 and 3 has a `parallel` argument which is `FALSE` by default. Using the syntax of `VGAM` based on Chambers and Hastie [12], setting `parallel = TRUE` (or `parallel = FALSE ~ 1`) results in $\mathbf{H}_1 = \mathbf{I}_M$ and $\mathbf{H}_2 = \mathbf{H}_3 = \dots = \mathbf{1}_M$; i.e., it is false for only the intercept.

It is noted that for some distributions such as the exponential and Maxwell the η_j are naturally parallel with respect to x_2, \dots, x_d because $\log \xi_\tau$ has the form $h_1(\tau) + h_2(x_2, \dots, x_d)$. If this is so then only the intercepts will change as a function of τ and the MLEs for h_2 are the same. Other distributions such as the 1-parameter gamma do not possess this property, and then it is necessary to constrain $\mathbf{H}_2 = \dots = \mathbf{H}_d = \mathbf{1}_M$ to avoid the quantile crossing problem.

2.1. Two Derivations. Ideally the link transforms the support of Y to \mathbb{R} because η_j should be unbounded. The three most common cases are as follows. For $Y \in (0, \infty)$ a log link is recommended, for $Y \in (0, 1)$ a logit link is a good choice, and $Y \in (-\infty, \infty)$ means that an identity link is natural. These cases have been implemented for seven 1-parameter distributions. The selection of the function \mathcal{G} for each Q is shown in the 5th column of Table 2, whilst the resulting quantile links as functions of θ are shown in the last column.

We now describe the quantile links for the exponential and the Topp–Leone distributions as examples. Firstly, for $Y \sim \text{exponential}(\theta)$, with a rate parameter $\theta > 0$, the density and CDF are given by $f(y; \theta) = \theta e^{-\theta y}$ and $F(y; \theta) = 1 - e^{-\theta y}$. With a slight change in notation, the quantile function is given by F^{-1} , i.e.,

$$Q_Y(\tau; \theta) = -\frac{1}{\theta} \log(1 - \tau), \quad (14)$$

which lies in $(0, \infty)$ regardless of the values of τ and θ . Given that values of τ are known (prespecified by the user), (14) becomes a function of θ . Thus, the new quantile link for the exponential distribution as shown in Table 2 is simply

obtained by taking \mathcal{G} as the logarithmic transformation, as follows:

$$\begin{aligned} \eta(\theta; \tau) &= \log \left[-\frac{1}{\theta} \log(1 - \tau) \right] \\ &= \log \log \left[(1 - \tau)^{-1/\theta} \right]. \end{aligned} \quad (15)$$

This quantile link has been implemented in `VGAMextra` via the function `expQlink()`, as shown in Table 3. Its inverse (denoted as $\theta(\eta; \tau)$) can be manually obtained from the inverse of (15). Note that the corresponding family function (`exponential()`) implemented in `VGAM` includes a (known) location parameter A , which gives the density $f(y; \theta) = \theta e^{-\theta(y-A)}$. By default $A = 0$, and it is handled by the argument `location`.

Secondly, consider the Topp–Leone distribution $Y \sim \text{Topp} - \text{Leone}(s)$ whose support is $(0, 1)$ and

$$Q_Y(s; \tau_j) = 1 - \sqrt{1 - \tau_j^{1/s}}, \quad (16)$$

with $0 < \tau_j < 1$. Here, $\theta = s$. To verify this restriction note that $1/s > 1$, for any shape parameter $s \in (0, 1)$, and hence for any $\tau_j \in (0, 1)$,

$$\begin{aligned} 0 < \tau_j^{1/s} < 1 &\iff \\ 0 < 1 - \tau_j^{1/s} < 1 &\iff \\ 0 < 1 - \sqrt{1 - \tau_j^{1/s}} < 1. \end{aligned} \quad (17)$$

Thus, to allow the quantile function to be modelled by covariates, we take the logit transformation as \mathcal{G} . The resulting quantile link for this distribution is simply $\eta(s; \tau) = \log Q_Y(s; \tau)$, shown in Table 2. The distribution has CDF $F(y; s) = [y \cdot (2 - y)]^s$ for $0 < y < 1$, and density $f(y; s) = 2s(1 - y) \cdot [y(2 - y)]^{s-1}$. The quantile function derives from solving the equation $\tau_0 = F(y; s) = [y \cdot (2 - y)]^s$, for $0 < \tau_0 < 1$, which leads to the quadratic equation $y^2 - 2y + \tau_0^{1/s} = 0$. The solution must lie in $(0, 1)$ and is in fact (16), as a function of s . The family function `topple()` from `VGAM` estimates s , where the default link is $\eta(s) = \text{logit}(s)$.

2.2. Software Implementation. For practical use by others, we have implemented seven VGLM–quantile links, η_τ in the R package `VGAMextra`. They are summarized in Table 2. The package `VGAM` is a requirement of `VGAMextra` because the modelling functions `vglm()` and `vgam()`, and all but the last family function of Table 2, reside there. For this paper `VGAMextra` 0.0-2 and `VGAM` 1.1-0 or later are required; they are available at www.stat.auckland.ac.nz/~vmir178 and www.stat.auckland.ac.nz/~yee/VGAM/prerelease/ whilst older versions of both are available on CRAN (<http://CRAN.R-project.org>).

One special case is `gamma1Qlink()`, for the 1-parameter (shape) gamma distribution, defined as

$$\eta(\theta; \tau) = \log \text{gamma}(\tau, \text{shape} = \theta), \quad (18)$$

TABLE 3: Inverse of the quantile links and names in VGAMextra. “Approximate” means that Newton–Raphson or bisection is used to approximate the inverse. All family functions except for `normal1sdff()`, which is in VGAMextra, are in VGAM.

Distribution	θ	Inverse $L[\theta(\eta; \boldsymbol{\tau})]$	Family function	Link in VGAMextra
Exponential	λ	$\frac{-\log(1-\boldsymbol{\tau})}{e^\eta}$	<code>exponential()</code>	<code>expQlink()</code>
Benini	s	$\frac{\log(1-\boldsymbol{\tau})}{(\eta - \log y_0)^2}$	<code>benini1()</code>	<code>benini1Qlink()</code>
Rayleigh	b	$\frac{\exp(\eta)}{\sqrt{-2\log(1-\boldsymbol{\tau})}}$	<code>rayleigh()</code>	<code>rayleighQlink()</code>
Gamma	s	Approximate	<code>gamma1()</code>	<code>gamma1Qlink()</code>
Maxwell	a	$\frac{2\text{qgamma}(\boldsymbol{\tau}, 1.5)}{\exp(2\eta)}$	<code>maxwell()</code>	<code>maxwellQlink()</code>
Topp–Leone [†]	s	$\frac{\log \boldsymbol{\tau}}{\log\{1 - [1 - \text{logit}^{-1}(\eta)]^2\}}$	<code>topple()</code>	<code>toppleQlink()</code>
$N(\mu = 0, \sigma)$	σ	$ (\eta - \mu_0)/\sqrt{2} \cdot \kappa(\boldsymbol{\tau}) $	<code>normal1sdff()</code>	<code>normal1sdQlink()</code>

[†] `logit-1()` denotes the inverse of the `logit()` transformation.

whose primary arguments are $\boldsymbol{\tau}$ and θ . Its inverse (Table 3) does not admit a closed form and it is approximated by the function `VGAM::newtonRaphson.basic()`, a *vectorized* implementation of the Newton–Raphson algorithm. Almost all implementations elsewhere of this are for a scalar argument, but we operate on vectors of length n . It works as follows. Our data is effectively $\mathbf{y}_i = \{y_i, \mathbf{x}_{i,d}\}$, $i = 1, \dots, n$, whilst the quantiles of interest, $\boldsymbol{\tau}$ or \mathbf{p} , must be entered by the user. The shape parameter θ is estimated by IRLS and therefore it is available at each iteration. Thus, for each $\eta(\theta; \mathbf{p})_0$, the ‘inverse’ is given by the root, θ , of the function

$$f(\theta; \mathbf{p}, \eta) = \eta(\theta; \mathbf{p})_0 - \log \text{gamma}(\boldsymbol{\tau}, \text{shape} = \theta). \quad (19)$$

Finally, the inverse of all the VGML–quantile links is shown in Table 3, as well as the name of the corresponding implementation in VGAMextra. The inverse–links are required at different stages of the IRLS by Fisher scoring, which internally switches between $\eta(\theta; \boldsymbol{\tau})$ (namely, Table 2) and $\theta(\eta; \boldsymbol{\tau})$ (namely, Table 3). Specifically, the algorithm requires the score vector and the EIMs at each

IRLS iteration, which are given by the following chain–rule formulas:

$$\begin{aligned} \frac{\partial \ell}{\partial \eta} &= \frac{\partial \ell}{\partial \theta} \cdot \frac{\partial \theta}{\partial \eta}, \\ -\mathbb{E} \left[\frac{\partial^2 \ell}{\partial \eta^2} \right] &= -\mathbb{E} \left[\frac{\partial^2 \ell}{\partial \theta^2} \right] \left(\frac{\partial \theta}{\partial \eta} \right)^2. \end{aligned} \quad (20)$$

Internally, the functions utilized to compute the inverse are `VGAM::eta2theta()` or `VGAM::theta2eta()`. The VGAMextra Manual and Miranda-Soberanis [13] give further details about the derivation of the quantile links, whilst Yee [6] describes in the IRLS and Fisher scoring algorithms for estimating VGLMs and VGAMs. Complements at the second author’s homepage give additional details on link functions.

2.3. Software Use. For the user, this methodology runs as usual by calling the modelling functions `VGAM::vglm()` and `VGAM::vgam()`, except for two modifications that are described below.

To start with, we give the following output that shows the central arguments handled by `VGAM::vglm()`:

```
> library("VGAM")
> args(vglm)

function (formula, family = stop("argument 'family' needs to be assigned"),
  data = list(), weights = NULL, subset = NULL, na.action = na.fail,
  etastart = NULL, mustart = NULL, coefstart = NULL, control = vglm.control(...),
  offset = NULL, method = "vglm.fit", model = FALSE, x.arg = TRUE,
  y.arg = TRUE, contrasts = NULL, constraints = NULL, extra = list(),
  form2 = NULL, qr.arg = TRUE, smart = TRUE, ...)
NULL
```

The first adjustment takes place with the argument `formula`, a symbolic description of the model to be fit. Usually, an expression like $y \sim x_2 + x_3$ should suffice

for a response y and covariates x_2 and x_3 . This effectively works for univariate and even for multiple responses say y_1 , y_2 , and y_3 , where the only change is to set

TABLE 4: Arguments handled by the function `VGAMextra::Q.reg()`.

Argument	Description
<code>y</code>	Numeric, a vector or a matrix. It is the response or dependent variable in the formula of the model to be fitted.
<code>pvector</code>	A prototype vector. Entries are the conditional p-quantiles in the fitting process.
<code>length.arg</code>	A unit-length positive integer. It is the number of p-quantiles to be modelled.

`cbind(y1, y2, y3) ~ x2 + x3`. Here, the right hand side (RHS) of the formula is applied to each linear predictor.

For quantile modelling using VGLMs and VGAMs, `Q.reg()` must be incorporated in the formula, whose arguments are shown in Table 4. For a given set of quantiles of interest, entered through $\tau = (\tau_1, \dots, \tau_L)^T$, `Q.reg()` replicates the response matrix \mathbf{Y} into $NOS \cdot \dim(\tau)$ columns, where NOS denotes the number of columns of \mathbf{Y} . Then, the

```
> p.quantiles <- c(0.25, 0.5, 0.75) ## Quantiles of interest
> fit.quantile <-
  vglm(Q.reg(y, pvector = p.quantile) ~ x2 + x3,
       family = rayleigh(lscale = rayleighQlink(p = p.quantile)),
       data = mydata)
```

Further fitting variants can be incorporated here, e.g., categorical covariates and the use of smoothers such as regression splines. These and a few other features are illustrated in the following section.

3. Examples

3.1. Maxwell Data. We use simulation to generate $n = 200$ random variates from a Maxwell distribution whose rate parameter is a function of a single covariate x_2 . To account for a nonlinear trend in the dataset, additive models with cubic smoothing splines appear to be a better choice over linear schemes such as with VGLMs. In this example we perform the following steps to confirm the performance of the methodology.

- (1) Generate random deviates from the Maxwell distribution.
- (2) Run conditional VGAM-quantile modelling using `maxwellQlink()` based on the VGAM family function `VGAM::maxwell()`, which estimates the Maxwell distribution by Fisher scoring.

RHS of the formula applies to every set of columns according to the number of quantiles of interest. Ordinarily the response is a vector so that $NOS = 1$ and $L = M$.

As an example, suppose that we have two responses Y_1 and Y_2 sampled from a prespecified distribution \mathcal{F} , as per Table 3, and the quantiles of interest are $p = (0.25, 0.50, 0.75)^T$. Then `Q.reg(cbind(Y1, Y2), pvector = p)` will return a matrix with six columns, with the first three columns being Y_1 , one for each quantile, and similarly the last three columns equal Y_2 . Thus `vglm()` handles this model as a multiple responses fit.

The second adjustment is related to the argument `family`, a function that describes the statistical model to be fitted. Each family has at least one argument for the link functions to be used in the fitting process (the name changes from family to family). For example, for `VGAM::exponential()` this is called `link`, whilst for the family function `VGAM::benini1()` (see the third column of Table 3), it is called `lshape`. When VGLM-quantile modelling is to be performed, the corresponding link (last column of Table 3) must be entered into the family accordingly. All the quantile links manage the same arguments, including `p`, the vector of quantiles, except by `benini1Qlink()` which has the additional argument `y0`.

With both modifications, a typical call has the following form:

- (3) Perform ordinary quantile regression using `VGAM::alaplace1()` that estimates the 1-parameter ALD by Fisher scoring. Here, the special argument `tau` will be employed.
- (4) Plot the artificial data with the estimated quantile functions, $Q_y(\tau \mid \mathbf{x}, \hat{\theta})$ (from (2)), and the estimated quantile curves (from (3)) superimposed.

We will consider the quantiles 25%, 50%, and 75% for simplicity, so that $\tau = (1/4, 1/2, 3/4)^T$.

Regarding (1), the data is generated by `VGAM::rmaxwell()`, which gives random deviates from the Maxwell distribution whose density is $f(y; a) = \sqrt{2/\pi} a^{3/2} y^2 \exp(-ay^2/2)$. We use the rate function

$$a = \exp \left\{ 2 - \frac{6 \sin(2x_{i2} - 1/5)}{(x_{i2} + 1/2)^2} \right\}, \quad (21)$$

where $X_{i2} \stackrel{\text{i.i.d.}}{\sim} \text{Unif}(0, 1)$, $i = 1, \dots, n$. The following code chunk sets things up and the dataset is saved as `maxdata`.

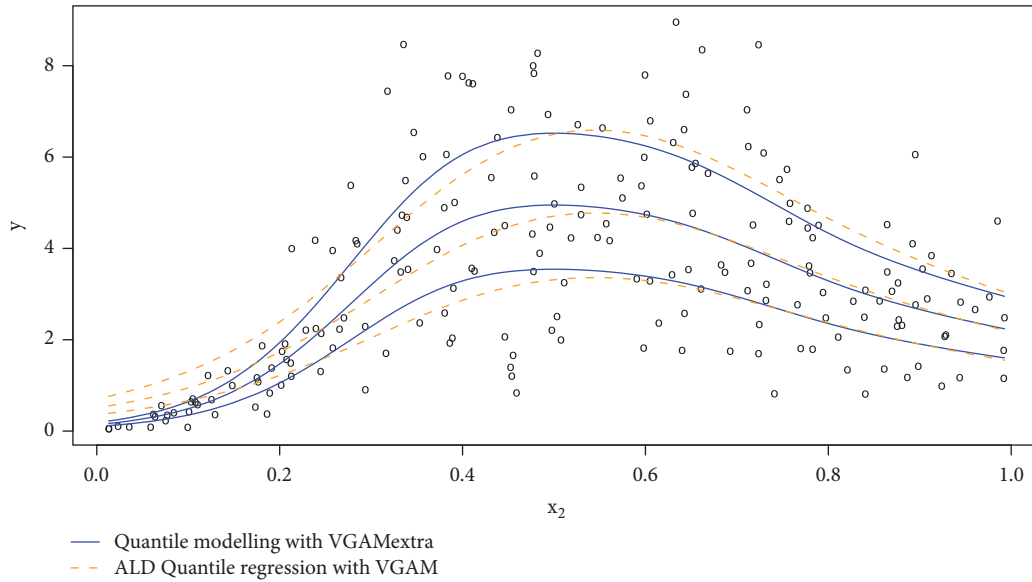


FIGURE 1: Simulated Maxwell data (21) including (a) the fitted quantile functions from `fit.Qmodelling` (VGAMextra) and (b) the fitted quantile curves from `fit.Qregression` (VGAM). The quantile curves in both cases derive from vector smoothing spline fits.

```
set.seed(1)
maxdata <- data.frame(x2 = sort(runif(n <- 200))) # Sorted for plotting
ratefun <- function(x) exp(2 - 6 * sin(2 * x - 0.2) / (x + 0.5)^2)
# Generate the data:
maxdata <- transform(maxdata, y = rmaxwell(n, rate = ratefun(x2)))
my.tau <- c(0.25, 0.50, 0.75) # Use these quantiles
```

The following code chunk performs steps (2) and (3). Note the fitting of additive models via `VGAM::vgam()` with smooth terms defined by `VGAM::s()` where x_2 is

to be smoothed. To compare both fits, they are saved in `fit.Qmodelling` (from (2)) and `fit.Qregression` (from (3)).

```
### Quantile modelling with 'maxwellQlink()'
mydof <- 4 # Effective degrees of freedom of the smoothing spline
fit.Qmodelling <-
  vgam(Q.reg(y, pvector = my.tau) ~ s(x2, df = mydof), data = maxdata,
        maxwell(link = maxwellQlink(p = my.tau),
                  type.fitted = "Qlink"))

### Quantile regression with alaplace2() from VGAM
fit.Qregression <- vgam(y ~ s(x2, df = mydof), alaplace1(tau = my.tau,
                  llocation = "loge", parallel.locat = TRUE),
                      data = maxdata, maxit = 100)
```

Figure 1 shows the simulated data, the estimated quantile functions, and the fitted quantile curves from `fit.Qmodelling` and `fit.Qregression`, obtained from vector smoothing spline fits [14]. The results are similar for $x_2 > 0.3$, but our present work performs better at the bottom LHS tail. The data coverage from each modelling framework

is summarized in Table 5. Once again our work outperforms the ALD method.

We conclude with a few remarks.

- (1) The argument `p` is available for *all* quantile links in Table 3 and not only for `maxwellQlink()`.

It can be assigned any vector of percentile values.

- (2) Under the conditional VGAM-quantile modelling framework, the arguments to handle the parallelism assumption such as the arguments `parallel.locat` and `parallel.scale` in family functions are no longer required. This is internally managed by the new quantile links rather than being managed by the family function.

- (3) If `fit` is a Qlink fit, then `fitted(fit)` returns the fitted quantiles. This is in the form of a $n \times (L \cdot \text{NOS})$ matrix. Similarly, `predict(fit)` returns a $n \times M$ matrix where the i th row is η_i^T .

3.2. *Comparison with the Quantreg Package.* For checking purposes, the results are compared with `quantreg` too. Figure 2 gives the results based on the following code.

```
> qrfit25 <- rq(y ~ bs(x2), data = maxdata, tau = 0.25)
> qrfit50 <- rq(y ~ bs(x2), data = maxdata, tau = 0.50)
> qrfit75 <- rq(y ~ bs(x2), data = maxdata, tau = 0.75)

> mylwd <- 1.5
> plot(y ~ x2, maxdata, main = "", xlab = expression(x[2]), las = 1,
       ylab = "y", cex = 0.85)
> with(maxdata, matlines(x2, fitted(fit.Qmodelling), col = "blue",
                        lwd = mylwd, lty = 1))
> lines(predict(qrfit25) ~ x2, maxdata, lty = 2, lwd = mylwd^2,
       col = "orange")
> lines(predict(qrfit50) ~ x2, maxdata, lty = 2, lwd = mylwd^2,
       col = "orange")
> lines(predict(qrfit75) ~ x2, maxdata, lty = 2, lwd = mylwd^2,
       col = "orange")
> legend("topleft", lwd = c(mylwd, mylwd^2),
       c("Quantile modelling with VGAMextra",
         "Quantile regression with quantreg"),
       cex = 0.95, col = c("blue", "orange"), lty = 1:2)
```

The results should be the similar to Section 3.1 because the ALD and the classical quantile regression method are essentially the same. It can be seen that the bottom LHS corner is not modelled well with `quantreg` either. Once again our method performs best, which is not surprising given the strong distributional assumption.

3.3. *Exponential Data.* Feigl and Zelen [15] fit an exponential distribution to a data set comprising the time to death

(in weeks) and white blood cell counts for two groups of leukaemia patients, and a binary variable for AG-positive and AG-negative. The two groups were not created by random allocation. The variable AG is the morphological variable, the AG factor; a numeric vector where 1 means AG-positive and 2 means AG-negative. We create AG01 which is $AG - 1$. We take the log of the white blood cell count (WBC) because it is very highly skewed. The data are found in `GLMsData` on CRAN, which supports Dunn and Smyth [16].

```
> leukwbc <- transform(leukwbc, AG01 = as.factor(AG - 1), # Creating AG01
                      logWBC = log(WBC) - 9) # Symmetric and centred
> tauvec <- c(25, 75) / 100 # Quantiles of interest
> expfit1 <- vglm(Q.reg(Time, tauvec) ~ logWBC + AG01, data = leukwbc,
                 exponential(expQlink(p = tauvec), ishrink = 0.05))
> coef(expfit1, matrix = TRUE)

               expQlink(rate1; 0.25) expQlink(rate2; 0.75)
(Intercept)          2.84757          4.42010
logWBC              -0.30442          -0.30442
AG011               -1.01765          -1.01765
```

One benefit of quantile modelling with VGLMs is that it easily allows comparisons of the effect of AG01 or any other

indicator variable, at different quantiles. First note that, for AG-positive patients with $\log WBC = 9$, the 25% percentile for

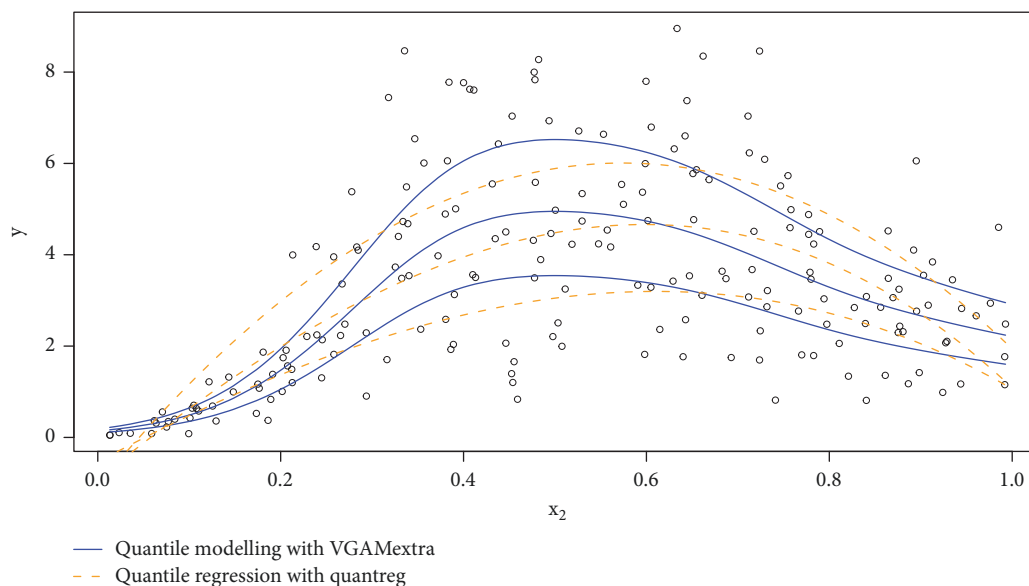


FIGURE 2: Simulated Maxwell data (21) including (a) the fitted quantile functions from `fit.Qmodelling` (VGAMextra) and (b) the fitted quantile curves from `qrfit50`, etc. from `quantreg`.

time to death is around $\exp 2.848 \approx 17.25$ weeks, whilst the 75% percentile is about $\exp 4.42 \approx 83.1$ weeks. Secondly, the coefficient of AG011 measures the influence of the AG factor on the time to death. Keeping the levels of WBC constant, for patients at either the 25% or 75% percentiles, the time to death for AG–negatives compared to AG–positives is multiplicative by a factor of $\exp(-1.02) \approx 0.361$, i.e., a 63.9% reduction in lifetime.

For further illustration's sake, we fit a 1-parameter gamma distribution to these data and interpret the results. Unlike the Maxwell and exponential distributions, where simple mathematics shows that different quantiles are parallel because their logarithm is additive with respect to τ , the 1-parameter gamma does not possess this property.

```
> # Fit a gamma model
> gamma1fit <- vglm(Q.reg(Time, tauvec) ~ logWBC + AG01, data = leukwbc,
  gamma1(gamma1Qlink(p = tauvec),
    parallel = FALSE ~ AG01))
> coef(gamma1fit, matrix = TRUE)

      gamma1Qlink(shape1; 0.25) gamma1Qlink(shape2; 0.75)
(Intercept)           3.67717           3.86380
logWBC                -0.58688           -0.58688
AG011                 -1.46180           -1.26843
```

Here, keeping the level of WBC constant, for patients at the 25% percentile, the time to death for AG–negatives compared to AG–positives is multiplicative by a factor of $\exp(-1.46) \approx 0.232$, i.e., a 76.8% reduction. In comparison, for patients at the 75% percentile, the time to death for AG–negatives compared to AG–positives is multiplicative by a factor of

$\exp(-1.27) \approx 0.281$, i.e., a 71.9% reduction. This suggests that the effect of AG is greater for more severe cases than those who live longer in general.

Finally, just to check, we obtain the constraint matrices for each predictor:

```
> constraints(gamma1fit, matrix = TRUE)

      (Intercept):1 (Intercept):2 logWBC AG011:1 AG011:2
gamma1Qlink(shape1; 0.25)      1      0      1      1      0
gamma1Qlink(shape2; 0.75)      0      1      1      0      1
```

TABLE 5: Empirical data coverage from quantile-modelling using VGAMextra (QM-VGAMextra) and quantile-regression from VGAM (QR-VGAM), after fitting (21).

τ_j	QM-VGAMextra coverage	QR-VGAM coverage
25%	26%	28.5%
50%	50%	54%
75%	73.5%	78.5%

There is a parallelism assumption made for logWBC but not for any of the other explanatory variables.

4. Discussion and Future Work

This work in parametric quantile regression is blighted by the strong assumption of the assumed distribution. In theory, this might be ameliorated somewhat by implementing as many distributions as possible. Some of the distributions

```
> fitted(maxwellfit, percentile = 20, type.fitted = "quantile")
```

worked like many other VGAM models. The difficulty here is that the @linkinv S4 slot of a VGAM family function has eta as an argument, and in our implementation this could only possibly be created by supplying the new percentiles to predict() beforehand.

Another minor deficiency in our software implementation is that the response vector is replicated $\dim(\tau)$ times so that is a form of recycling. Possibly this could be avoided because the memory requirement might be excessive when either $\dim(\tau)$ or n are very large.

At present, the VGAM framework has infrastructure to afford 1-parameter quantile links. For quantile functions depending on 2 or more parameter, such as the two-parameter gamma distribution, the quantiles will be bivariate functions whose inverse would probably not admit a closed form. Nevertheless, future work includes being able to write links for two-parameter distributions, of which the normal distribution would be the most important. For this, the methodology behind Yee and Miranda-Soberanis [17] could be employed; they solve a decades-old problem implementing the two-parameter canonical link function $\log(\mu/(\mu + k))$ of the negative binomial distribution. We have already commenced work in this direction, e.g., with the 2-parameter gamma distribution.

Data Availability

The data used to support the findings of this study are available in the Supplementary Materials.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

listed in Table 2 have real applications, for example, in kinetic-molecular theory the speed of individual molecules of idealized gases follows the Maxwell distribution and the average kinetic speed is directly related to Kelvin temperature. In experiments that do not satisfy the various postulates made (such as the effects of the container) one might model the median particle speed with \mathbf{x} comprising temperature and other covariates such as volume of the container and density of the container walls. Different forms of gases, such as plasmas and rarefied gases, could be modelled as such too. Another example is the Rayleigh distribution which is similar to the Maxwell distribution. In two-dimensions, and in applications of magnetic resonance imaging (MRI), complex images are often viewed in terms of the background data, which is Rayleigh distributed. Nonstandard background information could be included in \mathbf{x} and their effects on the distribution examined.

In the current software implementation there are limitations due to its internal design. For example, it would be good if

Acknowledgments

VM's work was supported by University of Auckland Doctoral Scholarship.

Supplementary Materials

The file is a text file that comprises R commands that can be copied into R. The file contains information about what R packages need to be installed and loaded to run properly. There are no copyright issues. Some of the data is contained in the R packages, and some of the data is simulated data. (*Supplementary Materials*)

References

- [1] R. Koenker and G. Bassett Jr., "Regression quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978.
- [2] R. Koenker, *Quantile Regression*, vol. 38, Cambridge University Press, New York, USA, 2005.
- [3] A. Noufaily and M. C. Jones, "Parametric quantile regression based on the generalized gamma distribution," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 62, no. 5, pp. 723–740, 2013.
- [4] R. A. Rigby and D. M. Stasinopoulos, "Generalized additive models for location, scale and shape," *Journal of Applied Statistics*, vol. 54, no. 3, pp. 507–554, 2005.
- [5] T. J. Cole and P. J. Green, "Smoothing reference centile curves: the LMS methods and penalized likelihood," *Statistics in Medicine*, vol. 110, no. 5, pp. 1305–1319, 1992.
- [6] T. W. Yee, *Vector Generalized Linear and Additive Models with an Implementation in R*, Springer, New York, USA, 2015.

- [7] T. W. Yee and T. J. Hastie, “Reduced-rank vector generalized linear models,” *Statistical Modelling. An International Journal*, vol. 3, no. 1, pp. 15–41, 2003.
- [8] J. Nelder and R. Wedderburn, “Generalized linear models,” *Journal of the Royal Statistical Society*, vol. 1350, no. 3, pp. 370–384, 1972.
- [9] P. H. C. Eilers and B. D. Marx, “Flexible smoothing with B-splines and penalties,” *Statistical Science*, vol. 11, no. 2, pp. 89–121, 1996.
- [10] P. J. Green and B. W. Silverman, *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman & Hall, London, UK, 1994.
- [11] X. He, “Quantile Curves without Crossing,” *The American Statistician*, vol. 51, no. 2, pp. 186–192, 1997.
- [12] J. M. Chambers and T. J. Hastie, *Statistical Models in S*, Chapman & Hall, New York, USA, 1993.
- [13] V. Miranda-Soberanis, *Vector Generalized Linear Time Series Models with an Implementation in R [Ph.D. thesis]*, Department of Statistics, University of Auckland, New Zealand, 2018.
- [14] T. W. Yee and C. J. Wild, “Vector generalized additive models,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 3, pp. 481–493, 1996.
- [15] P. Feigl and M. Zelen, “Estimation of exponential survival probabilities with concomitant information,” *Biometrics*, vol. 21, no. 4, pp. 826–838, 1965.
- [16] P. K. Dunn and G. K. Smyth, *Generalized Linear Models with Examples in R*, Springer, Springer, New York, 2018.
- [17] T. W. Yee and V. F. Miranda-Soberanis, “Vector generalized linear models and negative binomial regression,” *Australian & New Zealand Journal of Statistics*, 2018.

