Online Supplement to

Two–parameter Link Functions, With Applications to Negative Binomial, Weibull and Quantile Regression

By V. F. Miranda–Soberanis, and Thomas W. Yee.

Contents

A. NB Canonical Link - Example 1.

A.1. Supplementary material for Example 1, viz. Section 5.1.1.

This section gives the code to generate Table 2 and Table 3, adapted from the COUNT documentation. Here $\boldsymbol{\beta}_1 = (-3, 1.25, 0.1)^T$, $k = e^7 \approx 1100$, and x_2 and $x_3 \sim \text{Unif}(0, 1)$ independently. The following code chunk sets this up.

```
> ## Generating the data, stored in the data frame 'ndata'.
> library("COUNT")
> n <- 1000
> set.seed(1)
> x2 <- runif(n)
> x3 <- runif(n)
> alpha <- 1 / exp(7)  # value of alpha = 1 / size
> xb <- -3 + 1.25 * x2 + 0.1 * x3
> mu <- 1 / ((exp(-xb) - 1) * alpha)
> prob <- 1 / (1 + alpha * mu)
> ndata <-
    data.frame(y = rnbinom(n, size=1/alpha, prob = prob), x2, x3)
```

The data is stored in `ndata`. Next, we fit two models, with `VGAM::vglm()` and `COUNT::ml.nbc()` stored in `fit1` and `fit2` respectively. The estimated coefficients, Std. errors and z values are displayed (viz. Table 2). Results from `COUNT:ml.nbc()` shows the instability referred to.

```
> library("VGAM")
> #'fit1' from vglm(), and the estimated coefficients.
> fit1 <- vglm(y ~ x2 + x3, negbinomial(lmu = "nbcanlink"), ndata)
> mymat1 <- summary(fit1)@coef3
>
> ## TABLE 2 LEFT - HAND SIDE (LHS) ##
> round(mymat1[-2, -4], 3)

              Estimate Std. Error z value
(Intercept):1   -2.680      0.218 -12.290
x2               1.204      0.041  29.321
x3               0.097      0.009  10.658

> # 'fit2' from ml.nbc(), and estimated coefficients + CL + UL
> fit2 <- ml.nbc(y ~ x2 + x3, data = ndata)

Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
```

```
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced
Warning in dnbinom(y, size = r.hat, prob = p.hat, log = TRUE): NaNs
produced

> mymat3 <- matrix(round(c(fit2$Estimate[-4], fit2$SE[-4],
                           fit2$Z[-4]), 3), nrow = 3, ncol = 3,
                  byrow = FALSE)
> rownames(mymat3) <- c("(Intercept):1", "x2", "x3")
> colnames(mymat3) <- c("Estimate", "Std. Error", "z value")
>
> ## TABLE 2 RIGHT - HAND SIDE (RHS) ##
> mymat3

              Estimate Std. Error z value
(Intercept):1   -1.511      0.009 -175.72
x2               0.871      0.008  107.37
x3               0.051      0.007    7.24
```

Finally, the approximate 95% Wald confidence intervals for the regression coefficients, viz. Table 3, are obtained as follows:

```
> mymat <- mymat1[c(1, 3, 4, 2), 1 , drop = FALSE]
> ULimVGAM <- c(mymat + 1.96 * mymat1[ c(1, 3, 4, 2), 2,
                                       drop = TRUE])
> LLimVGAM <- c(mymat - 1.96 * mymat1[ c(1, 3, 4, 2), 2,
                                       drop = TRUE])
>
```

```
> mymat3 <- data.frame(mymat3)
> mymat3[4, ] <- c(fit2[4, c(1, 2, 3)])
> ULimCOUNT <- c(mymat3[, 1, drop = TRUE] +
                  1.96 * mymat3[, 2, drop = TRUE])
> LLimCOUNT <- c(mymat3[, 1, drop = TRUE] -
                  1.96 * mymat3[, 2, drop = TRUE])
> mymat <- matrix(c(LLimVGAM, ULimVGAM, LLimCOUNT, ULimCOUNT),
                 nrow = 4, ncol = 4, byrow = FALSE)
> mymat[4, c(4, 3)] <- 1/mymat[4, c(4, 3)]
> mymat[4, c(1, 2)] <- exp(mymat[4, c(1, 2)])
> colnames(mymat) <- c("2.5%", "97.5%", "2.5%", "97.5%")
> rownames(mymat) <- c("(Intercept):1", "x2", "x3", "k")
>
>
> ## TABLE 3 ##
> round(mymat, 3)

                 2.5%    97.5%    2.5%   97.5%
(Intercept):1  -3.107   -2.253  -1.529  -1.493
x2              1.123    1.284   0.855   0.887
x3              0.079    0.115   0.037   0.065
k             503.394 1230.032 220.375 214.945
```

B. NB Canonical Link - Example 2.

B.1. Supplementary material for Example 2, viz. Section 5.1.2

R code used for the analysis presented in Section 5.1.2.

```
# Fit the four models ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

data("trap0", package = "VGAMdata")
use.df <- trap0

trapo.nb1 <-
    vglm(BMTW ~ bs(doy), negbinomial(parallel = TRUE, zero = NULL),
         data = use.df, trace = FALSE)

trapo.nb2 <-
    vglm(BMTW ~ bs(doy), negbinomial, data = use.df, trace = FALSE)

trapo.nbh <-
    vglm(BMTW ~ bs(doy), negbinomial(zero = NULL),
         data = use.df, trace = FALSE)

trapo.nbc <-
    vglm(BMTW ~ bs(doy), negbinomial(lmu = "nbcanlink", zero = NULL),
         data = use.df, trace = FALSE)

all.wffci <- c('nb1' = AIC(trapo.nb1),
               'nb2' = AIC(trapo.nb2),
               'nbh' = AIC(trapo.nbh),
               'nbc' = AIC(trapo.nbc))

# For the table
# round(sort(all.wffci) , 2)
```

C. Weibull quantile modelling - Example 3

Section 5.2 compares distribution–specific quantile modelling using using `VGAMextra::weibullQlink()` with quantile curves produced by a BCCG regression with GAMLSS. The code is given in this section. We use `weibullMlink()`, the link for the mean of the 2–parameter Weibull distribution given by $\texttt{weibullMlink}(\beta; \alpha) = \log\left[\beta \cdot \Gamma(1 + 1/\alpha)\right]$.

```
library("VGAMextra")
library("gamlss")

## Fitting Model (5.1) with GAMLSS - BCCG distribution
data(rent99)     ## 'rent99' is the Munich rental guide data
rent99[, "area"] <- as.numeric(rent99[, "area"])
fitgamlss <- gamlss(rent ~ s(area),
                sigma.formula = ~  1, nu.formula = ~1,
                family = BCCG(mu.link = "log"), data = rent99,
                trace = FALSE)

## Fitting Model (5.1) with VGAM and VGAMextra - Weibull dist.
fitvgam <- vgam(rent ~ s(area),
                 family = weibullRff(link1 = "weibullMlink"),
                 data = rent99, trace = FALSE)
## AICs and BICs - Table 6.
rbind(AIC(fitgamlss), AIC(fitvgam))
rbind(BIC(fitgamlss), BIC(fitvgam))


## -- FIGURE 2 -- QQplots Weibull vs BCCG ##
par(mfrow = c(1, 2))
par(mar = c(4.0, 4.0, 1.5, 0.5) + 0.1)

## QQplot- Weibull
scores <- rent99$rent
w_scores = qweibull(ppoints(length(scores)),
                    shape = loglink(predictors(fitvgam)[, 2],
                                    inverse = TRUE),
                    scale = weibullMlink(predictors(fitvgam)[, 1],
         shape = loglink(predictors(fitvgam)[, 2], inverse = TRUE),
                                        inverse = TRUE) )

qqplot(x = w_scores, y = scores, pch = 20, cex = 0.75,
       ylab = "Sample Quantiles", xlab = "Theoretical Quantiles",
       main = "")
#title("Weibull QQ-Plot")
```

```r
qqline(scores, distribution = function(p)  {
  qweibull(p,  shape = loglink(predictors(fitvgam)[, 2],
                                    inverse = TRUE),
           scale = weibullMlink(predictors(fitvgam)[, 1],
         shape = loglink(predictors(fitvgam)[, 2], inverse = TRUE),
                                    inverse = TRUE) )
}
,
col="red",lwd = 2,lty = 2, probs = c(0.0090, 0.011817))
mtext("(a)", line = 2.5, outer = FALSE, at = 2000, side = 2,
      cex = 1.5, las = 2)

## QQ plots - GAMLSS
w_scores <- qBCCG(ppoints(length(scores)),
                  mu = fitted(fitgamlss),
                  sigma = exp(fitgamlss$sigma.coefficients),
                  nu = exp(fitgamlss$nu.coefficients) )

qqplot(x = w_scores, y = scores, pch = 20, cex = 0.75,
       ylab = "Sample Quantiles", xlab = "Theoretical Quantiles",
       main = "" )
qqline(scores,
       distribution = function(p) {

         qBCCG(p, mu = fitted(fitgamlss),
               sigma = exp(fitgamlss$sigma.coefficients),
               nu = exp(fitgamlss$nu.coefficients) )
       },
       col="red",lwd = 2,lty = 2, probs = c(0.25, 0.32))
mtext("(b)", line = 2.5, outer = FALSE, at = 2000, side = 2,
      cex = 1.5, las = 2)
## End Figure 2 ##


## FIGURE 3 - Quantile curves GAMLSS (BCCG) vs VGAMextra (Weibull)
par(mar = c(4.0, 4.0, 1.5, 0.5) + 0.15)
plot(rent ~ area, pch = 19, data = rent99, cex = 0.12,
     xlab = "area", col= "black",
     ylab = "rent", main = 'Weibull and BCCG quantile curves.')

### Weibull quantile modelling - weibullQlink().
tau <- c(0.05, 0.25, 0.50, 0.75, 0.95)
fittau <- vgam(Q.reg(rent, length = 5)~ s(area),
               family = weibullRff(link1 = "weibullQlink",
                           percentile = tau* 100, zero = "shape"),
```

```
              data = rent99, trace = FALSE)

### Quantile curves from GAMLSS ###
nu <- predict(fitgamlss, what = "nu")
sigma <- exp(predict(fitgamlss, what = "sigma"))
mu <- exp(predict(fitgamlss, what = "mu"))

tau <- 5/100
pred50 <- mu * qBCCG(p = tau, mu = 1, sigma = sigma, nu = nu)
pc5 <- 100* length(pred50[pred50 > rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, pred50)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "solid", col = "red")

tau <- 25/100
pred50 <- mu * qBCCG(p = tau, mu = 1, sigma = sigma, nu = nu)
pc25 <- 100* length(pred50[pred50 > rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, pred50)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "solid", col = "red")

tau <- 50/100
pred50 <- mu * qBCCG(p = tau, mu = 1, sigma = sigma, nu = nu)
pc50 <- 100* length(pred50[pred50 > rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, pred50)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "solid", col = "red")

tau <- 75/100
pred50 <- mu * qBCCG(p = tau, mu = 1, sigma = sigma, nu = nu)
pc75 <- 100* length(pred50[pred50 > rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, pred50)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "solid", col = "red")

tau <- 95/100
pred50 <- mu * qBCCG(p = tau, mu = 1, sigma = sigma, nu = nu)
pc95 <- 100* length(pred50[pred50 > rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, pred50)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
```

```r
        lty = "solid", col = "red")

## Quantile curves from VGAMextra - Weibull quantile modelling
predvgam <- fitted(fittau)[, 1]
pc5v <- 100* length(predvgam[predvgam >
                              rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, predvgam)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.5,
        lty = "dashed", col = "blue")

predvgam <- fitted(fittau)[, 2]
pc25v <- 100* length(predvgam[predvgam > rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, predvgam)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "dashed", col = "blue")

predvgam <- fitted(fittau)[, 3]
pc50v <- 100* length(predvgam[predvgam >
                               rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, predvgam)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "dashed", col = "blue")

predvgam <- fitted(fittau)[, 4]
pc75v <- 100* length(predvgam[predvgam >
                               rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, predvgam)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "dashed", col = "blue")

predvgam <- fitted(fittau)[, 5]
pc95v <- 100* length(predvgam[predvgam >
                               rent99$rent])/dim(rent99)[1]
pred50 <- data.frame(area = rent99$area, predvgam)
pred50 <- pred50[order(pred50[, 1]), ]
matplot(pred50[, 1], pred50[, 2], type = "l", add = T, lwd = 1.25,
        lty = "dashed", col = "blue")

legend("topleft", c("VGAMextra - WeibullQlink",  "GAMLSS - BCCG"),
       lwd = rep(2, 1.25), lty = c("dashed", "solid"),
       col = c("blue", "red"))
```

D. Quantile Modelling with `uninormalQlink()` - Example 4, Section 5.3.1

The data set *BWeights.csv* is a genuine copy Sashelp.BWeights, a supplementary file in the SAS® software with the random sample ($n = 50,000$) of the June 1997 Detailed Natality Data referred to in Section 5.3.1. The SAS® code to generate this CSV file is the following:

```
PROC EXPORT data = sashelp.bweight
outfile = " /folders/myfolders/bweight.csv" dbms = CSV
run;
```

BWeights.csv spans ten variables as described in Table A.4 (taken from Koenker and Hallock 2001).

Table A.4. Variables in the data set *BWeights.csv*

Variable (Column name)	Description
`Weight`	(continuous) Infant birth weight.
`Black`	(binary) 1 = Black mother, 0 = White mother.
`Married`	(binary) 1 = Married mother, 0 = Unmarried.
`Boy`	(binary) 1 = Baby boy, 0 = Girl.
`MomAge`	(continuous) Mother's age at birth centered at its median.
`MomSmoke`	(binary) 1 = Smoking mother, 0 = Nonsmoker.
`CigsPerDay`	(numeric–integer) Cigarettes per day.
`MomWtGain`	(continuous) Mother's weight gain during pregnancy centered at its median.
`Visit`	(categorical) Prenatal visit during pregnancy. 0 = No visit, 1 = First trimester, 2 = Second trimester, 3 = Last trimester.
`MomEdLevel`	(categorical) Mother's education level at birth. 0 = Less than high school, 1 = High school, 2 = Some college, 3 = College graduate.

The code below gives the trace output and the model summary when estimating (4.11) using `uninormalQlink()`. The fitted model is saved as `fit.qr.mod` . The data is formatted according to Table A.4.

```
## Saving and formatting 'qrdata'.
## 'CigsParDay' read okay.
qrdata <- read.csv("bweight.csv")
qrdata$Black <- as.factor(qrdata$Black)
qrdata$Married <- as.factor(qrdata$Married)
qrdata$Boy <- as.factor(qrdata$Boy)
qrdata$MomSmoke <- as.factor(qrdata$MomSmoke)
qrdata$MomEdLevel <- as.factor(qrdata$MomEdLevel)
qrdata$Weight <- as.numeric(qrdata$Weight)
qrdata$MomWtGain <- as.numeric(qrdata$MomWtGain)
qrdata$Visit <- as.numeric(qrdata$Visit)
```

```
> ## The first six observations
> head(qrdata)
>
> ### VGLM quantile modelling from Section 5.3.1-Normal distribution
> ## PERCENTILES of interest.
> mytau <- 100  *  c(0.05,  0.25, 0.50, 0.75, 0.95)
> ## VGLM quantile modelling - eta_1 is handled by 'uninormalQlink()'.
> fit.qr.mod <-
    vglm(Q.reg(Weight, pvector = mytau) ~ Black + Married + Boy +
               Visit + MomEdLevel + MomSmoke + CigsPerDay + MomAge +
               MomWtGain + I(MomAge^2) + I(MomWtGain^2),
         uninormalff(link1 = "uninormalQlink", lsd = "loglink",
                     zero = NULL, percentile = mytau),
         trace =FALSE, data = qrdata)
> ## Uncomment the following the see the model summary
> ## summary(fit.qr.mod)
```

The following code chunk generates Figure 4.

```
> ## FIGURE 4 - R CODE
> par(mfrow = c(2, 2))
>
> mymat <- coef(fit.qr.mod, mat = TRUE)
> BlackCoef <- mymat["Black1", c(TRUE, FALSE)]
> Bcqr <- coef(fit.quantreg)[2, ]
> plot(mytau, BlackCoef, pch = 17, col = "black",
       ylab = "Black Mother", cex = 1.35,
       xlab = "Percentile (%)", lwd = 4, ylim = c(-300, 100),
       xlim = c(0, 100), xaxt = "n")
> axis(side = 1, at = c(5, 25, 50, 75, 95))
> points(100*pretau, Bcqr, lwd = 3, pch = 17, cex = 1.35, col = "orange")
> lines(mytau, BlackCoef, col = "black", lty = 1, lwd = 1.5)
> lines(100*pretau, Bcqr, col = "orange", lty = "dashed", lwd = 1.5)
```

```
> abline(h = 0, lty = 2)
> legend("topleft",
         legend = c("uninormalQlink() - Miranda & Yee (2022)",
                    "QR - Koenker & Basset (1978)"),
         lwd = 1.5, col = c("black", "orange"), cex = 0.75,
         pch = c(17, 17),  lty = c("solid", "dashed"))
>
>
>
>
>
> cpdqr <- coef(fit.quantreg)[10, ]
> CigpdCoef <- mymat["nCigsPerDay1", c(TRUE, FALSE)]
> plot(mytau, CigpdCoef, pch = 17, col = "black",
       ylab = "Cigarettes per day", cex =1.35,
       xlab = "Percentile (%)", lwd = 4, ylim = c(-300, 100),
       xlim =c(0, 100), xaxt = "n")
> axis(side = 1, at = c(5, 25, 50, 75, 95))
> points(mytau, cpdqr, lwd = 3, pch = 17, cex = 1.35, col = "orange")
> lines(mytau, cpdqr, col = "orange", lty = "dashed", lwd = 1.5)
> lines(mytau, CigpdCoef, col = "black", lty = 1, lwd = 1.5)
> abline(h = 0, lty = 2)
> legend("topleft",
         legend = c("uninormalQlink() - Miranda & Yee (2022)",
                    "QR - Koenker & Basset (1978)"),
         lwd = 1.5, col = c("black", "orange"), cex = 0.75,
         pch = c(17, 17),  lty = c("solid", "dashed"))
>
>
>
>
>
> MomAge <- mymat["MomAge", c(TRUE, FALSE)]
> mageqr <- coef(fit.quantreg)[11, ]
> plot(mytau, MomAge, pch = 21, col = "black",
       ylab = "Mom's Age",
       xlab = "Percentile (%)", ylim =c(-1, 30),
       xlim = c(0, 100), lwd = 4, xaxt = "n")
> axis(side = 1, at = c(5, 25, 50, 75, 95))
> points(mytau, mageqr, lwd = 3, pch = 17, cex = 1.35, col = "orange")
> lines(mytau, MomAge, col = "black", lty = 1, lwd = 1.5)
> lines(mytau, mageqr, col = "orange", lty = "dashed", lwd = 1.5)
> abline(h = 0, lty = 2)
> legend("topleft",
         legend = c("uninormalQlink() - Miranda & Yee (2022)",
```

```
                     "QR - Koenker & Basset (1978)"),
         lwd = 1.5, col = c("black", "orange"), cex = 0.75,
         pch = c(17, 17),  lty = c("solid", "dashed"))
>
>
>
>
> MomwtCoef <- mymat["MomWtGain", c(TRUE, FALSE)]
> momwtqr <- coef(fit.quantreg)[12, ]
> plot(mytau, MomwtCoef, pch = 21, col = "black",
       ylab = "Mom weight gain",
       xlab = "Percentile (%)", lwd = 4, xlim = c(0, 100),
       ylim = c(-1, 30), xaxt = "n")
> axis(side = 1, at = c(5, 25, 50, 75, 95))
> points(mytau, momwtqr, lwd = 3, pch = 17, cex = 1.35, col = "orange")
> lines(mytau, MomwtCoef, col = "black", lty = 1, lwd = 1.5)
> lines(mytau, momwtqr, col = "orange", lty = "dashed", lwd = 1.5)
> abline(h = 0, lty = 2)
> legend("topleft",
         legend = c("uninormalQlink() - Miranda & Yee (2022)",
                     "QR - Koenker & Basset (1978)"),
          lwd = 1.5, col = c("black", "orange"), cex = 0.75,
         pch = c(17, 17),  lty = c("solid", "dashed"))
```