Fisher scoring for some univariate discrete distributions

Thomas Yee

University of Auckland

26 August 2010

t.yee@auckland.ac.nz
http://www.stat.auckland.ac.nz/~yee

Outline of This Talk

Introduction to VGLMs and VGAMs I

The VGAM package for R implements several large classes of regression models of which *vector generalized linear and additive models* (VGLMs/VGAMs) are most commonly used.

The primary key words are

- *iteratively reweighted least squares (IRLS),*
- *maximum likelihood estimation,*
- *Fisher scoring,*
- *additive models.*

Other concepts are

- *reduced-rank regression,*
- *constrained ordination,*
- *vector smoothing.*

Introduction to VGLMs and VGAMs II

Basically . . .

VGLMs model each parameter, transformed if necessary, as a linear combination of the explanatory variables. That is,

$$g_j(\theta_j) \;=\; \eta_j \;=\; \boldsymbol{\beta}_j^T \mathbf{x} \;=\; \beta_{(j)1}\, x_1 + \cdots + \beta_{(j)p}\, x_p \tag{1}$$

where g_j is a *parameter link function*.

VGAMs extend (1) to

$$g_j(\theta_j) \;=\; \eta_j \;=\; f_{(j)1}(x_1) + \cdots + f_{(j)p}(x_p) \tag{2}$$

i.e., an *additive model* for each parameter. Estimated by smoothers, this is a data-driven approach.

Introduction to VGLMs and VGAMs III

Example: negative binomial

Y has a probability function that can be written as

$$P(Y = y; \mu, k) = \binom{y + k - 1}{y} \left(\frac{\mu}{\mu + k}\right)^y \left(\frac{k}{k + \mu}\right)^k$$

where $y = 0, 1, 2, \ldots$. Parameters $\mu > 0$ and $k > 0$.

VGAM can fit

$$\begin{aligned} \log \mu &= \eta_1 = \boldsymbol{\beta}_1^T \mathbf{x}, \\ \log k &= \eta_2 = \boldsymbol{\beta}_2^T \mathbf{x}, \quad \text{with} \end{aligned}$$

`vglm(y ~ x2 + x3 + ··· + xp, negbinomial(zero = NULL))`

Introduction to VGLMs and VGAMs IV

The framework extends GLMs and GAMs in three main ways:

(i) **y** not restricted to the exponential family,

(ii) multivariate responses **y** and/or *linear/additive predictors* $\boldsymbol{\eta}$ are handled,

(iii) η_j need not be a function of a mean μ: $\eta_j = g_j(\theta_j)$ for *any parameter* θ_j.

This formulation is deliberately general so that it encompasses as many distributions and models as possible. We wish to be limited only by the assumption that the regression coefficients enter through a set of linear or additive predictors.

Given the covariates, the conditional distribution of the response is intended to be completely general. *More general* $\Longrightarrow$ *more useful.*

Introduction to VGLMs and VGAMs V

The scope of VGAM is very broad; it potentially covers

- univariate and multivariate distributions,
- categorical data analysis,
- quantile and expectile regression,
- time series,
- survival analysis,
- mixture models,
- extreme value analysis,
- nonlinear regression,
- reduced-rank regression,
- ordination, ...

It conveys GLM/GAM-type modelling to a much broader range of models.

Introduction to VGLMs and VGAMs VI

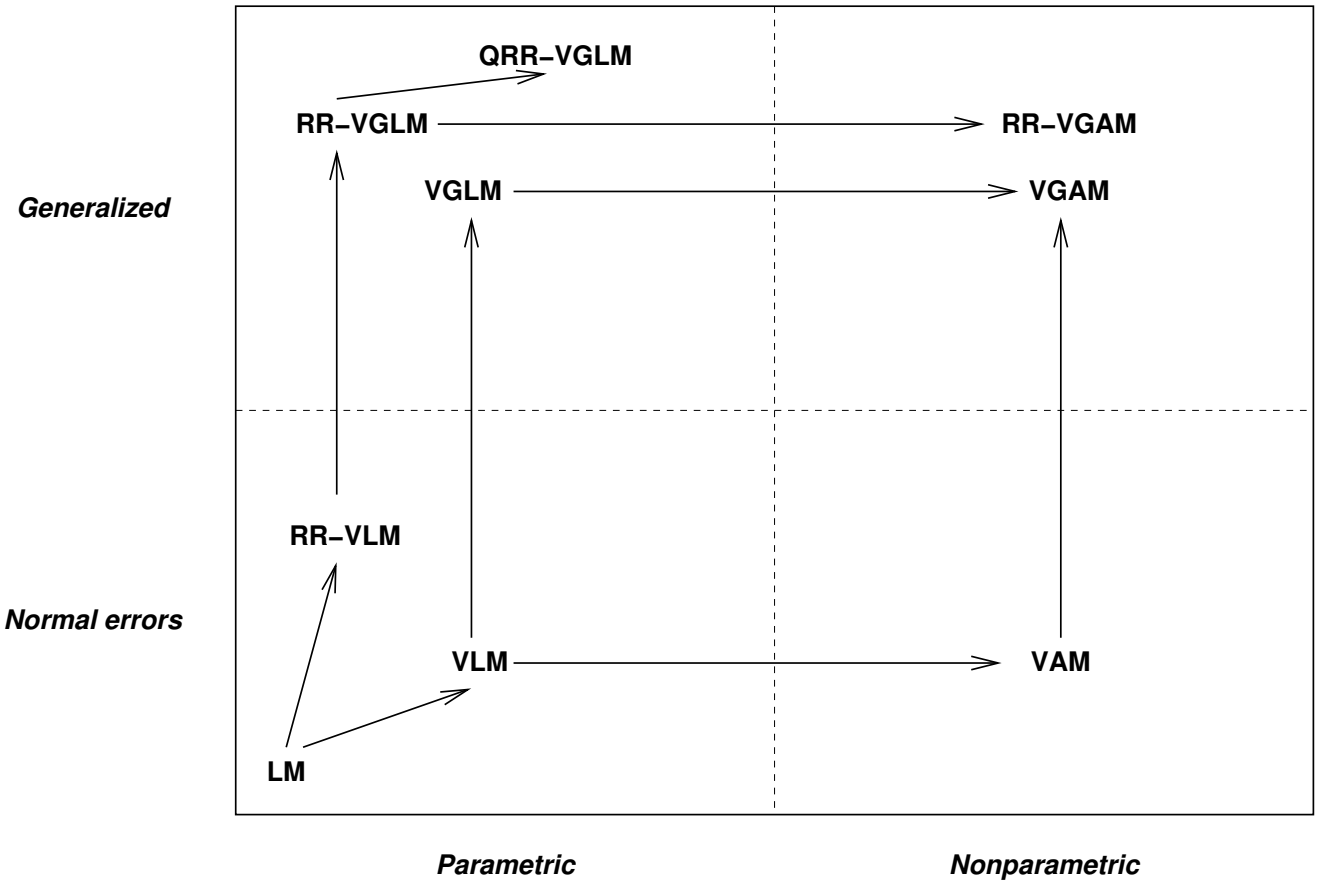


Figure: Flowchart for different classes of models. Legend: LM = linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, V = vector, G = generalized, A = additive, RR = reduced-rank.

Introduction to VGLMs and VGAMs VII

t

$\boldsymbol{\eta} = (\eta_1, \ldots, \eta_M)^T$	**Model**	**S function**	**Reference**
$\mathbf{B}_1^T \mathbf{x}_1 + \mathbf{B}_2^T \mathbf{x}_2 \ (= \mathbf{B}^T \mathbf{x})$	VGLM	`vglm()`	Yee & Hastie (2003)
$\mathbf{B}_1^T \mathbf{x}_1 + \sum_{k=p_1+1}^{p_1+p_2} \mathbf{H}_k \, \mathbf{f}_k^*(x_k)$	VGAM	`vgam()`	Yee & Wild (1996)
$\mathbf{B}_1^T \mathbf{x}_1 + \mathbf{A}\,\boldsymbol{\nu}$	RR-VGLM	`rrvglm()`	Yee & Hastie (2003)
$\mathbf{B}_1^T \mathbf{x}_1 + \mathbf{A}\,\boldsymbol{\nu} + \begin{pmatrix} \boldsymbol{\nu}^T \mathbf{D}_1 \boldsymbol{\nu} \\ \vdots \\ \boldsymbol{\nu}^T \mathbf{D}_M \boldsymbol{\nu} \end{pmatrix}$	QRR-VGLM	`cqo()`	Yee (2004)
$\mathbf{B}_1^T \mathbf{x}_1 + \sum_{r=1}^{R} \mathbf{f}_r(\nu_r)$	RR-VGAM	`cao()`	Yee (2006)

Table: VGAM & its framework. The latent variables $\boldsymbol{\nu} = \mathbf{C}^T \mathbf{x}_2$, $\mathbf{x}^T = (\mathbf{x}_1^T, \mathbf{x}_2^T)$. More abbreviations: C = constrained, O = ordination, Q = quadratic.

The VGAM Package I

See `VGAMrefcard.pdf` for a summary.

How does VGAM differ from other packages?

- its breadth,
- its similarity to `glm()` and `gam()`, e.g.,

```
> n = 20
> y = rchisq(n, df = exp(2))
> fit = vglm(y ~ 1, family = chisq)
> fitted(fit)
> summary(fit)
```

- its size,
- its room for future extension.

VGLMs I

Data $(\mathbf{x}_i, \mathbf{y}_i)$, $i = 1, \ldots, n$, from independent "individuals".

Definition Conditional distribution of $\mathbf{y}$ given $\mathbf{x}$ is

$$f(\mathbf{y}|\mathbf{x}; \boldsymbol{\beta}) = h(\mathbf{y}, \eta_1, \ldots, \eta_M, \boldsymbol{\phi}),$$

where for $j = 1, \ldots, M$,

$$\begin{aligned} \eta_j &= \eta_j(\mathbf{x}) = \boldsymbol{\beta}_j^T \mathbf{x}, \\ \boldsymbol{\beta}_j &= (\beta_{(j)1}, \ldots, \beta_{(j)p})^T, \\ \boldsymbol{\beta} &= (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_M^T)^T, \\ \boldsymbol{\phi} &= \text{a vector of scale factors.} \end{aligned} \tag{3}$$

Often $g_j(\theta_j) = \eta_j$ for parameters θ_j and link functions g_j.

Nb. $-\infty < \eta_j < \infty$.

VGLM Examples I

1. *Negative binomial distribution*. For $y = 0, 1, 2, \ldots,$

$$f(y; \mu, k) = \binom{y + k - 1}{y} \left(\frac{\mu}{\mu + k}\right)^y \left(\frac{k}{k + \mu}\right)^k, \quad \mu, \quad k > 0,$$

$$\begin{aligned} \eta_1 &= \log \mu, \\ \eta_2 &= \log k \quad \text{are good choices.} \end{aligned}$$

If $k > 1$ then could have $\eta_2 = \log \log k$. Use `negbinomial(lk = loglog)`.

*Later (Slide **??**):* will be able to fit

(i) $\mathrm{Var}(Y) = s_1 \, \mu,$
(ii) $\mathrm{Var}(Y) = \mu + \mu^{s_2},$

easily regardless whether s_1 and s_2 are known or unknown.
[cf. $\mathrm{Var}(Y) = \mu + \mu^2/k$]

VGLM Examples II

2. *Bivariate (logistic) odds-ratio model*

Data: (Y_1, Y_2) where $Y_j = 0$ or 1.

Examples

- $Y_1 = 1$ if left eye is blind, $Y_2 = 1$ if right eye is blind.
- $Y_j = 1/0$ if Species j is present/absent.
- $Y_1 = 1/0$ if person has/hasn't cancer,
  $Y_2 = 1/0$ if person has/hasn't diabetes.

$$
\begin{aligned}
p_j &= P(Y_j = 1), \quad \text{marginal probabilities,} \\
p_{rs} &= P(Y_1 = r, Y_2 = s), \quad r, s = 0, 1, \quad \text{joint probabilities,} \\
\psi &= p_{00}\, p_{11} / (p_{01}\, p_{10}) \quad \text{(Odds ratio).}
\end{aligned}
$$

VGLM Examples III

Model:

$$\begin{aligned} \text{logit}\, p_j(\mathbf{x}) &= \eta_j(\mathbf{x}), \qquad j = 1, 2\,, \\ \log\ \psi(\mathbf{x}) &= \eta_3(\mathbf{x}). \end{aligned}$$

Recover p_{rs}'s from p_1, p_2 and ψ.

Q: why not allow a probit or complementary log-log link?

```
> binom2.or("cloglog", exchangeable = TRUE, zero = NULL)
```

VGLM Examples IV

Exchangeable data $\Longrightarrow$ constrain $\eta_1 = \eta_2$, (e.g., ears and eyes), i.e.,

$$\begin{aligned} \texttt{cloglog}\, p_j(\mathbf{x}) &= \eta_1(\mathbf{x}), \qquad j = 1, 2\,, \\ \log\, \psi(\mathbf{x}) &= \eta_3(\mathbf{x}). \end{aligned}$$

Note:

$$\begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix} = \sum_{k=1}^{p} \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta^*_{(1)k} \\ \beta^*_{(2)k} \end{pmatrix} x_k = \sum_{k=1}^{p} \begin{pmatrix} \beta^*_{(1)k} \\ \beta^*_{(1)k} \\ \beta^*_{(2)k} \end{pmatrix} x_k. \tag{4}$$

General formula is

$$\boldsymbol{\eta}(\mathbf{x}) = \mathbf{B}^T \mathbf{x} = \sum_{k=1}^{p} \mathbf{H}_k \boldsymbol{\beta}^*_k \, x_k, \tag{5}$$

where $\boldsymbol{\beta}^*_k = \left(\beta^*_{(1)k}, \ldots, \beta^*_{(r_k)k}\right)^T$.

VGLM Examples V

3. *Nonproportional odds model* for a categorical response i.e., $Y \in \{1, 2, \ldots, M+1\}$

$$\text{logit}\, P(Y \leq j|\mathbf{x}) \quad = \quad \eta_j(\mathbf{x}), \qquad j = 1, \ldots, M.$$

Proportional odds model: constrain

$$\eta_j(\mathbf{x}) = \alpha_j + \eta(\mathbf{x})$$

(aka the *parallelism* assumption, which stops the probabilities from becoming negative or greater than 1). Have $\mathbf{H}_1 = \mathbf{I}_M$; $\mathbf{H}_k = \mathbf{1}_M$ for $k = 2, \ldots, p$.

VGLM Examples VI

Other links (for $0 < p < 1$):

probit	$\Phi^{-1}(p)$,
cloglog	$\log\{-\log(1-p)\}$,
cauchit	$\tan(\pi(p - \frac{1}{2}))$.

```
> vglm(y ~ x2, cumulative(link = probit))
```

VGLM algorithm I

Models with log-likelihood

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ell_i\{\eta_1(\mathbf{x}_i), \ldots, \eta_M(\mathbf{x}_i)\},$$

where $\eta_j = \boldsymbol{\beta}_j^T \mathbf{x}_i$. Then

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}_j} = \sum_{i=1}^{n} \frac{\partial \ell_i}{\partial \eta_j} \mathbf{x}_i$$

and

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}_j \, \partial \boldsymbol{\beta}_k^T} = \sum_{i=1}^{n} \frac{\partial^2 \ell_i}{\partial \eta_j \, \partial \eta_k} \mathbf{x}_i \, \mathbf{x}_i^T.$$

Newton-Raphson algorithm

$$\boldsymbol{\beta}^{(a+1)} = \boldsymbol{\beta}^{(a)} + \mathcal{J}\left(\boldsymbol{\beta}^{(a)}\right)^{-1} \mathbf{U}\left(\boldsymbol{\beta}^{(a)}\right)$$

VGLM algorithm II

written in *iteratively reweighted least squares (IRLS)* form is

$$\begin{aligned}\boldsymbol{\beta}^{(a+1)} &= \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}\mathbf{X}\,\boldsymbol{\beta}^{(a)} + \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}\mathbf{W}^{-1}\mathbf{u} \\ &= \left(\mathbf{X}_{\text{VLM}}^T\,\mathbf{W}^{(a)}\,\mathbf{X}_{\text{VLM}}\right)^{-1}\mathbf{X}_{\text{VLM}}^T\,\mathbf{W}^{(a)}\,\mathbf{z}^{(a)}\,.\end{aligned}$$

Let $\mathbf{z} = (\mathbf{z}_1^T, \ldots, \mathbf{z}_n^T)^T$ and $\mathbf{u} = (\mathbf{u}_1^T, \ldots, \mathbf{u}_n^T)^T$, where $\mathbf{u}_i$ has jth element

$$(\mathbf{u}_i)_j = \frac{\partial \ell_i}{\partial \eta_j},$$

and $\mathbf{z}_i = \boldsymbol{\eta}(\mathbf{x}_i) + \mathbf{W}_i^{-1}\mathbf{u}_i$ (*adjusted dependent vector* or *pseudo-response*).

Also, $\mathbf{W} = \text{Diag}(\mathbf{W}_1, \ldots, \mathbf{W}_n)$, $\quad (\mathbf{W}_i)_{jk} = -\dfrac{\partial^2 \ell_i}{\partial \eta_j\, \partial \eta_k}$,

$\mathbf{X}_{\text{VLM}} = (\mathbf{X}_1^T, \ldots, \mathbf{X}_n^T)^T$, $\quad \mathbf{X}_i = \text{Diag}(\mathbf{x}_i^T, \ldots, \mathbf{x}_i^T) = \mathbf{I}_M \otimes \mathbf{x}_i^T$.

VGLM algorithm III

$\boldsymbol{\beta}^{(a+1)}$ is the solution to

$$\mathbf{z}^{(a)} \;=\; \mathbf{X}_{\text{VLM}}\,\boldsymbol{\beta}^{(a+1)} + \boldsymbol{\varepsilon}^{(a)}, \qquad \text{Var}(\boldsymbol{\varepsilon}^{(a)}) \;=\; \phi\,\mathbf{W}^{(a)^{-1}}.$$

Fisher scoring:

$$(\mathbf{W}_i)_{jk} \;=\; -\,E\left[\frac{\partial^2 \ell_i}{\partial\eta_j\,\partial\eta_k}\right]$$

usually results in slower convergence but is preferable because the *working weight matrices* are positive-definite over a larger parameter space.

`wz` computed in `@weight` is usually

$$(\mathbf{W}_i)_{jk} \;=\; -\,E\left(\frac{\partial^2 \ell_i}{\partial\eta_j\,\partial\eta_k}\right), \quad \text{sometimes} \quad -\,\frac{\partial^2 \ell_i}{\partial\eta_j\,\partial\eta_k}.$$

VLMs† I

The *vector linear model (VLM)* is the central model behind VGLMs and VGAMs. Its crux is to minimize

$$\sum_{i=1}^{n} \left(\mathbf{z}_i - \sum_{k=1}^{p} \mathbf{H}_k \, \boldsymbol{\beta}_k^* \, x_{ik} \right)^T \mathbf{W}_i \left(\mathbf{z}_i - \sum_{k=1}^{p} \mathbf{H}_k \, \boldsymbol{\beta}_k^* \, x_{ik} \right),$$

where the $\mathbf{H}_k$ are known *constraint matrices* of full column rank.

With no constraints ($\mathbf{H}_k = \mathbf{I}_M$), this is equivalent to fitting

$$\mathbf{z}_i = \begin{pmatrix} \mathbf{x}_i^T \boldsymbol{\beta}_1 \\ \vdots \\ \mathbf{x}_i^T \boldsymbol{\beta}_M \end{pmatrix} + \boldsymbol{\varepsilon}_i, \quad \boldsymbol{\varepsilon}_i \sim (\mathbf{0}, \mathbf{W}_i^{-1}), \quad i = 1, \ldots, n.$$

VLMs† II

Cf. *Multivariate linear model (aka multivarate regression)*

$$\left(\mathbf{Y}_{(1)} \cdots \mathbf{Y}_{(M)}\right) \;=\; \mathbf{XB} + \mathbf{U}, \qquad \mathbf{u}_i \;\sim\; (\mathbf{0}, \mathbf{\Sigma}) \tag{6}$$

where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)^T$.

The VGAM package for R I

The central functions of VGAM

`vglm()`	Vector generalized linear models.
`vgam()`	Vector generalized additive models.
`rrvglm()`	Reduced-rank vector generalized linear models.
`cqo()`	Constrained quadratic (Gaussian) ordination (QRR-VGLM).
`cao()`	Constrained additive ordination (RR-VGAM).

Others:

`vlm()`	Vector linear models.
`grc()`	Goodman's RC(r) model.
`uqo()`	Unconstrained quadratic ordination (QU-VGLM).

The VGAM package for R II

Table: VGAM generic functions applied to a model called `fit`.

Function	Value
`coef(fit)`	$\widehat{\boldsymbol{\beta}}$
`coef(fit, matrix = TRUE)`	$\widehat{\mathbf{B}}$
`constraints(fit)`	$\mathbf{H}_k,\ k = 1, \ldots, p$
`deviance(fit)`	Deviance $D = \sum_{i=1}^{n} d_i$
`fitted(fit)`	$\widehat{\mu}_{ij}$ usually
`logLik(fit)`	Log-likelihood $\sum_{i=1}^{n} w_i \, \ell_i$
`model.matrix(fit, type = "lm")`	LM model matrix $(n \times p)$
`model.matrix(fit, type = "vlm")`	VLM model matrix $\mathbf{X}_{\text{VLM}}$
`predict(fit)`	$\widehat{\eta}_{ij}$

The VGAM package for R III

Table: VGAM generic functions applied to a model called `fit`.

Function	Value
`predict(fit, type = "response")`	$\widehat{\mu}_{ij}$ usually
`resid(fit, type = "response")`	$y_{ij} - \widehat{\mu}_{ij}$ usually
`resid(fit, type = "deviance")`	$\text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}$
`resid(fit, type = "pearson")`	$\mathbf{W}_i^{-\frac{1}{2}}\,\mathbf{u}_i$
`resid(fit, type = "working")`	$\mathbf{z}_i - \boldsymbol{\eta}_i = \mathbf{W}_i^{-1}\,\mathbf{u}_i$
`vcov(fit)`	$\widehat{\text{Var}}(\widehat{\boldsymbol{\beta}})$
`weights(fit, type = "prior")`	w_i (`weights` argument)
`weights(fit, type = "working")`	$\mathbf{W}_i$ (in matrix-band format)

The VGAM package for R IV

The VGAM package employs several feature to make the software more robust, e.g.,

- Parameter link functions, e.g.,
  - $\log \theta$ for $\theta > 0$,
  - $\text{logit}\ \theta$ for $0 < \theta < 1$,
  - $\log(\theta - 1)$ for $\theta > 1$.

  These improve the quadratic approximation to ℓ near the solution and avoid range-restriction problems.
- *Half-step sizing*.
- Good initial values, e.g., *self-starting* VGAM family functions.
- Numerical linear algebra based on orthogonal methods, e.g., QR method in **LINPACK**. Yet to do: use **LAPACK**.
- B-splines, not the Reinsch algorithm.

Working weights† I

Each $\mathbf{W}_i$ needs to be positive-definite.

1. *Expected information matrix (EIM)* is often positive-definite over a larger parameter space than the *observed information matrix (OIM)*. But the EIM may be intractable.
2. Under mild regularity conditions,

$$\text{Var}\left(\frac{\partial \ell_i}{\partial \boldsymbol{\theta}}\right) = -E\left(\frac{\partial^2 \ell_i}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}^T}\right).$$

Often the score vector is easy. Use random variates to compute the sample variance of the score vector. The `nsimEIM` argument implements this.

Working weights† II

For example, the negative binomial has

$$\frac{\partial^2 \ell_i}{\partial k^2} = \psi'(y_i + k) - \psi'(k) = -\sum_{r=0}^{y_i - 1} (k + r)^{-2},$$

where $\psi'(z)$ is the trigamma function (the digamma function $\psi(z) = \Gamma'(z)/\Gamma(z)$). Its expected value involves an infinite series.

The `weight` slot of `negbinomial()`:

Working weights† III

```
weight = eval(substitute(expression({
    wz = matrix(as.numeric(NA), n, M)  # wz is 'diagonal'
    run.varcov = matrix(0, n, NOS)
    ind1 = iam(NA, NA, M=M, both=TRUE, diag=TRUE)
    for(ii in 1:( .nsimEIM )) {
        ysim = rnbinom(n=n*NOS, mu=c(mu), size=c(kmat))
        dl.dk = digamma(ysim+kmat) - digamma(kmat) -
                (ysim+kmat)/(mu+kmat) + 1 + log(kmat/(kmat+mu))
        run.varcov = run.varcov + dl.dk^2
    }
    run.varcov = cbind(run.varcov / .nsimEIM)
# Can do even better if it is an intercept-only model
    wz[,2*(1:NOS)] = if(intercept.only)
        matrix(colMeans(run.varcov),
               n, ncol(run.varcov), byrow=TRUE) else run.varcov

    wz[,2*(1:NOS)] = wz[,2*(1:NOS)] * dk.deta^2
# The 1-1 element (known exactly):
    ed2l.dmu2 = 1/mu - 1/(mu+kmat)
```

Working weights† IV

```
        wz[,2*(1:NOS)-1] = dmu.deta^2 * ed2l.dmu2
        w * wz
    }), list( .cutoff = cutoff, .Maxiter = Maxiter, .nsimEIM = nsimEIM )))
```

Some computational and implementational details

- Is S4 object-orientated and very modular—simply have to write a VGAM "family function".
- `> args(vglm.control)`

```
function (checkwz = TRUE, criterion = names(.min.criterion.VGAM),
    epsilon = 1e-07, half.stepsizing = TRUE, maxit = 30, stepsize = 1,
    save.weight = FALSE, trace = FALSE, wzepsilon = .Machine$double.eps^0.75,
    xij = NULL, ...)
NULL
```

- Implements *"smart" prediction*, e.g., `bs(scale(x))`, `I(bs(x))`.

VGAMs I

These allow additive-model extensions to all η_j in a VGLM, i.e., from:

$$\eta_j(\mathbf{x}) \;=\; \boldsymbol{\beta}_j^T \mathbf{x} \;=\; \beta_{(j)1}\, x_1 + \cdots + \beta_{(j)p}\, x_p$$

to M *additive predictors*:

$$\eta_j(\mathbf{x}) \;=\; \beta_{(j)1} + f_{(j)2}(x_2) + \cdots + f_{(j)p}(x_p),$$

a sum of arbitary smooth functions. Equivalently,

$$\begin{aligned} \boldsymbol{\eta}(\mathbf{x}) &= \mathbf{f}_1(x_1) + \cdots + \mathbf{f}_p(x_p) \\ &= \mathbf{H}_1\, \mathbf{f}_1^*(x_1) + \cdots + \mathbf{H}_p\, \mathbf{f}_p^*(x_p) \end{aligned} \tag{7}$$

for some *constraint matrices* $\mathbf{H}_k$ (default: $\mathbf{H}_k = \mathbf{I}_M$).

- $\mathbf{H}_1, \ldots, \mathbf{H}_p$ are known and of full-column rank,

VGAMs II

- $\mathbf{f}_k^*$ is a vector containing a possibly reduced set of component functions,

Starred quantities in (7) are unknown and to be estimated.

The $\mathbf{f}_k^*$ are centered for identifiability.

Examples of constraints

1. **Exchangeable bivariate odds-ratio model** All

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{H}_k = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ or } \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad k = 2, \ldots, p.$$

2. **Proportional odds model**

$$\mathbf{H}_1 = \mathbf{I}_M, \qquad \mathbf{H}_2 = \cdots = \mathbf{H}_p = \mathbf{1}_M.$$

VGAM facilitates the implementation and use of constraints, e.g.,

```
> binom2.or(exchangeable = TRUE, zero = 2)
> cumulative(parallel = FALSE ~ x5 - 1)
```

Simple Example: Positive Poisson Distribution† I

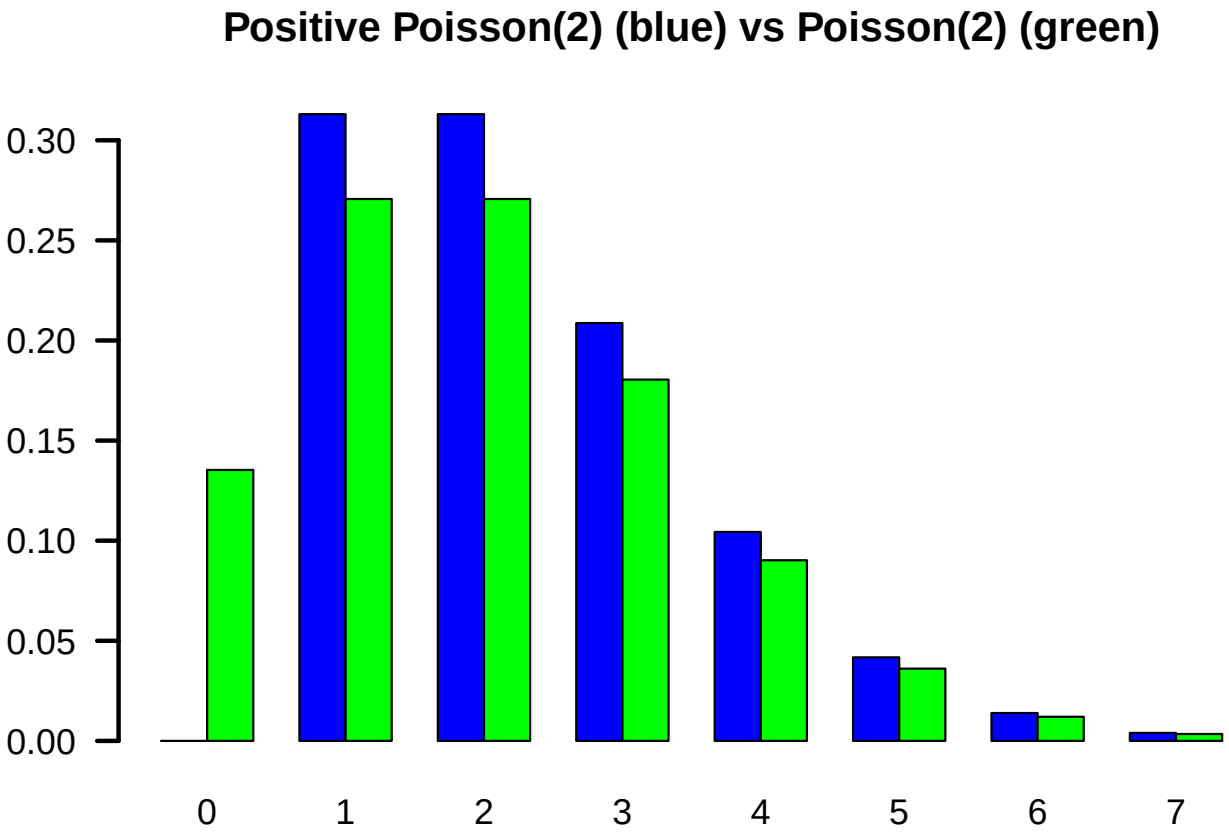
Its probability function is

$$f(y;\lambda) = \begin{cases} \dfrac{e^{-\lambda}\,\lambda^{y}}{y!\,(1-e^{-\lambda})} & y = 1, 2, \ldots \text{ and } \lambda > 0, \\ 0, & y = 0. \end{cases}$$

where λ is the *rate* parameter. Then

$$\begin{aligned} E(Y) &= \frac{\lambda}{1-e^{-\lambda}}, \\ \mathrm{Var}(Y) &= \frac{\lambda(1+\lambda)}{1-e^{-\lambda}} - \frac{\lambda^2}{(1-e^{-\lambda})^2}. \end{aligned}$$

Simple Example: Positive Poisson Distribution† II



Simple Example: Positive Poisson Distribution† III

The log-likelihood components are

$$\ell_i(\lambda; y_i) = -\lambda_i + y_i \log \lambda_i - \log(1 - e^{-\lambda_i}) - \log y_i! \,.$$

The score and Hessian functions are

$$\begin{aligned}
\frac{\partial \ell_i}{\partial \lambda_i} &= -1 + \frac{y_i}{\lambda_i} - \frac{1}{e^{\lambda_i} - 1}, \\
\frac{\partial^2 \ell_i}{\partial \lambda_i^2} &= -\frac{y_i}{\lambda_i^2} + \frac{e^{\lambda_i}}{(e^{\lambda_i} - 1)^2}, \\
\text{and } E\left(\frac{\partial^2 \ell_i}{\partial \lambda_i^2}\right) &= -\frac{e^{\lambda_i}}{(e^{\lambda_i} - 1)} \left(\frac{1}{\lambda_i} - \frac{1}{e^{\lambda_i} - 1}\right).
\end{aligned}$$

The default is a log link:

$$\eta = \log \lambda.$$

Simple Example: Positive Poisson Distribution† IV

Table: On a spring afternoon in Portland, Oregon, data on the sizes of different groups observed in public places were collected by Coleman and James (1961).

Group size	1	2	3	4	5	6
Frequency	1486	694	195	37	10	1

Now

```
> odata = data.frame(y = 1:6, w = c(1486, 694, 195,
+     37, 10, 1))
> fit = vglm(y ~ 1, pospoisson, data = odata, weights = w)
```

The output is

```
> print(summary(fit), presid = FALSE)
```

Simple Example: Positive Poisson Distribution† V

```
Call:
vglm(formula = y ~ 1, family = pospoisson, data = odata, weights = w)

Coefficients:
             Value Std. Error t value
(Intercept) -0.114     0.0268   -4.25

Number of linear predictors:  1

Name of linear predictor: log(lambda)

Dispersion Parameter for pospoisson family:   1

Log-likelihood: -2304.7 on 5 degrees of freedom

Number of Iterations: 6
```

Simple Example: Positive Poisson Distribution† VI

The maximum likelihood estimate $\widehat{\lambda} = e^{\widehat{\eta}}$, is

```
> Coef(fit)

lambda
0.8925
```

The estimate mean is

```
> fitted(fit)[1]

[1] 1.5118
```

That is, the mean group size is estimated to be 1.512 persons for the observed data and 0.89 persons for the expected Poisson distribution.

Simple Example: Positive Poisson Distribution† VII

The standard error of $\widehat{\lambda}$ is

```
> sqrt(vcov(fit, untransform = TRUE))
```

```
         lambda
lambda 0.023899
```

using the *delta method*.

The constraint matrix $\mathbf{H}_1$ is

```
> constraints(fit)
```

```
$`(Intercept)`
     [,1]
[1,]    1
```

Some Implemented Models I

The following are some distributions currently implemented by VGAM.

Positive Models

t

Distribution	Density function $f(y;\theta)$	VGAM family
Positive binomial	$\frac{1}{(1-p)^N}\binom{N}{Ny}p^{Ny}(1-p)^{N(1-y)}$	`posbinomial()`
Positive Poisson	$\frac{1}{1-e^{-\lambda}}\frac{e^{-\lambda}\lambda^y}{y!}$	`pospoisson()`
Positive negative binomial	$\frac{1}{1-(k/(k+\mu))^k}\binom{y+k-1}{y}\left(\frac{\mu}{\mu+k}\right)^y\left(\frac{k}{k+\mu}\right)^k$	`posnegbinomial()`
Positive normal	$\frac{1}{\sigma}\frac{\phi((y-\mu)/\sigma)}{[1-\Phi(-\mu/\sigma)]}$	`posnormal1()`

Table: Some positive distributions currently supported by the VGAM package.

Zero-inflated Models

t

Zero-inflated distribution	Probability function $f(y;\boldsymbol{\theta})$	VGAM family function
ZI negative binomial	$I(y=0)\phi + (1-\phi)\times$ $\binom{y+k-1}{y}\left(\frac{\mu}{\mu+k}\right)^{y}\left(\frac{k}{k+\mu}\right)^{k}$	`zinegbinomial()`
ZI binomial	$I(y=0)\phi + (1-\phi)\times$ $\binom{N}{Ny}p^{Ny}(1-p)^{N(1-y)}$	`zibinomial()`
ZI Poisson	$I(y=0)\phi + (1-\phi)\frac{e^{-\lambda}\lambda^{y}}{y!}$	`zipoisson()`

Table: Summary of zero-inflated discrete distributions currently supported by VGAM. "ZI" stands for "zero-inflated".

Zero-altered Models

t

Zero-altered distribution	Probability function $f(y;\boldsymbol{\theta})$	VGAM family
ZA negative binomial	$I(y=0)\,\phi + I(y>0)\,\frac{(1-\phi)}{1-(k/(k+\mu))^k} \times \binom{y+k-1}{y}\left(\frac{\mu}{\mu+k}\right)^y\left(\frac{k}{k+\mu}\right)^k$	`zanegbinomial()`
ZA Poisson	$I(y=0)\,\phi + I(y>0)\,(1-\phi)\,\frac{e^{-\lambda}\lambda^y}{(1-e^{-\lambda})y!}$	`zapoisson()`

Table: Summary of zero-altered discrete distributions currently supported by VGAM. "ZA" stands for "zero-altered".

Zero-inflated, Zero-Altered and Positive Models

t

Distribution	Random variates functions
Zero-altered negative binomial	`[dpqr]zanegbin()`
Zero-altered Poisson	`[dpqr]zapois()`
Zero-inflated binomial	`[dpqr]zibinom()`
Zero-inflated negative binomial	`[dpqr]zinegbin()`
Zero-inflated Poisson	`[dpqr]zipois()`
Positive binomial	`[dpqr]posbinom()`
Positive negative binomial	`[dpqr]posnegbinom()`
Positive normal	`[dpqr]posnorm()`
Positive Poisson	`[dpqr]pospois()`

Table: Some of VGAM functions for generating random variates etc. The prefix "d" means the density, "p" means the distribution function, "q" means the quantile function and "r" means random deviates. Note: most functions have a `[dpqr]foo()` for density, distribution function, quantile function and random generation.

GLMs

t

Distribution	Density/probability function $f(y)$	Range of y	VGAM family function
Gaussian	$(2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(y-\mu)^2/\sigma^2\right\}$	$(-\infty, \infty)$	`gaussianff()`
Binomial	$\binom{A}{Ay} p^{Ay}(1-p)^{A(1-y)}$	$0(1/A)1$	`binomialff()`
Poisson	$\frac{\exp\{-\lambda\}\lambda^y}{y!}$	$0(1)\infty$	`poissonff()`
Gamma	$\frac{(k/\mu)^k\, y^{k-1}\, \exp\{-ky/\mu\}}{\Gamma(k)}$	$(0, \infty)$	`gammaff()`
Inverse Gaussian	$\left(\frac{\lambda}{2\pi\, y^3}\right)^{\frac{1}{2}} \exp\left\{-\frac{\lambda}{2\,\mu^2}\frac{(y-\mu)^2}{y}\right\}$	$(0, \infty)$	`inverse.gaussianff()`

Table: Summary of GLMs supported by VGAM. The known prior weight is A.

Dispersion Models I

A *reproductive dispersion model* with positional parameter μ and dispersion parameter σ^2 is a family of distributions whose probability density functions are of the form

$$f(y;\mu,\sigma^2) \;=\; a(y;\sigma^2)\;\exp\left\{-\frac{1}{2\,\sigma^2}\,d(y;\mu)\right\}. \qquad (8)$$

where $a \geq 0$ is a suitable function, and d is a unit deviance.

Equation (8) is said to be of standard form. Jorgensen (1997) discusses many models within a dispersion model framework. The VGLM framework is too general to take advantage of the special structure of dispersion models, however, many of the models discussed in that book have been implemented in VGAM. See the table on Slide 49.

Dispersion Models II

t

Distribution	Density function $f(y;\boldsymbol{\theta})$	Range of y	VGAM family
Negative binomial	$\binom{y+k-1}{y}\left(\frac{\mu}{\mu+k}\right)^{y}\left(\frac{k}{k+\mu}\right)^{k}$	$\{0,1,\ldots\}$	`negbinomial`
Hyperbolic secant	$\frac{\exp\{\theta y+\log(\cos(\theta))\}}{2\cosh(\pi y/2)}$	$(-\infty,\infty)$	`hypersecant`
Hyperbolic secant	$\frac{\cos(\theta)}{\pi}\,u^{-\frac{1}{2}+\frac{\theta}{\pi}}\,(1-u)^{-\frac{1}{2}-\frac{\theta}{\pi}}$	$(0,1)$	`hypersecant.1`
Inverse binomial	$\frac{\lambda\ \Gamma(2y+\lambda)\ \{\rho(1-\rho)\}^{y}\ \rho^{\lambda}}{\Gamma(y+1)\,\Gamma(y+\lambda+1)}$	$\{0,1,\ldots\}$	`invbinomial`
Reciprocal inverse Gaussian	$\sqrt{\frac{\lambda}{2\pi y}}\,\exp\left\{-\frac{\lambda(y-\mu)^2}{2y}\right\}$	$(0,\infty)$	`rig`
Leipnik (transformed)	$\frac{\{y(1-y)\}^{-\frac{1}{2}}}{\text{Beta}(\frac{\lambda+1}{2},\frac{1}{2})}\left[1+\frac{(y-\mu)^2}{y(1-y)}\right]^{-\frac{\lambda}{2}}$	$(0,1)$	`leipnik`
Generalized Poisson	$\frac{\theta(\theta+y\lambda)^{y-1}}{y!}\,\exp(-y\lambda-\theta)$	$\{0,1,\ldots\}$	`genpoisson`
Simplex	$\frac{\exp\left\{-\frac{1}{2\sigma^2}\,\frac{(y-\mu)^2}{y(1-y)\,\mu^2\,(1-\mu)^2}\right\}}{\sqrt{2\pi\sigma^2\{y(1-y)\}^3}}$	$(0,1)$	`simplex`

Table: Dispersion models implemented in VGAM.

Other Distributions

t

Distribution	Density function $f(y;\boldsymbol{\theta})$	Range of y	Range of $\boldsymbol{\theta}$	VGAM family
Rice	$\frac{y}{\sigma^2}\ \exp\left\{\frac{-(y^2+v^2)}{2\,\sigma^2}\right\}\ I_0\left(\frac{y\,v}{\sigma^2}\right)$	$\mathbb{Z}$	$v>0,\ \sigma>0$	`riceff`
Skellam	$\left(\frac{\mu_1}{\mu_2}\right)^{y/2}\ e^{-\mu_1-\mu_2}\ I_y(2\sqrt{\mu_1\mu_2})$	$\mathbb{Z}$	$\mu_j>0$	`skellam`
Yule-Simon	$\rho\ B(y,\rho+1)$	$1(1)\infty$	$\rho>0$	`yulesimon`

Table: More discrete models implemented in VGAM.

More Distributions

t

Distribution	Density function $f(y;\boldsymbol{\theta})$	Range of y	Range of $\boldsymbol{\theta}$	VGAM family
Geometric	$(1-p)^y p$	$0(1)\infty$	$0 < p < 1$	`geometric`
Zeta	$\left[y^{p+1}\ \zeta(p+1)\right]^{-1}$	$1(1)\infty$	$0 < p < \infty$	`zetaff`
Zipf	$y^{-s}/\sum_{i=1}^{N} i^{-s},$	$1, 2, \ldots, N$	$s > 0$	`zipf`

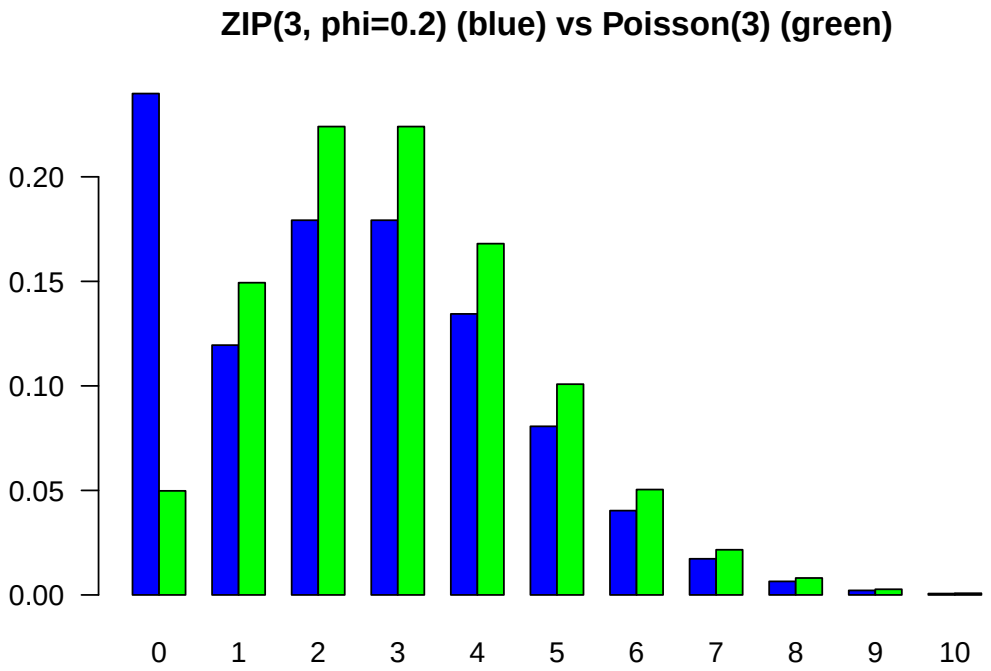
Table: More discrete univariate distributions currently supported by VGAM. Note: $H_{n,m} = \sum_{i=1}^{n} i^{-m}$ is known as the nth generalized harmonic number, B is the beta function.

Zero-inflated Poisson model I

Loosely,

$$P(Y = y; \phi, \lambda) \quad = \quad \phi\, P(Y = 0) + (1 - \phi)\, \text{Poisson}(\lambda).$$

where $0 < \phi < 1$, $\lambda > 0$.



Zero-inflated Poisson model II

Example 1: $Y =$ the number of insects on a leaf of a particular plant (some leaves have no insects because they are unsuitable for feeding).

Let the overall proportion of such leaves be ϕ.
Actually,

$$\begin{aligned} P(Y=0) &= \phi + (1-\phi)\, e^{-\lambda}, \\ P(Y=y) &= (1-\phi)\, \frac{e^{-\lambda}\, \lambda^{y}}{y!}, \qquad y = 1, 2, \ldots, \end{aligned}$$

where $0 < \phi < 1$ and $\lambda > 0$. Then a good idea is

$$\boldsymbol{\eta} = \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} = \begin{pmatrix} \text{logit}\, \phi \\ \log \lambda \end{pmatrix}.$$

Zero-inflated Poisson model III

If $\boldsymbol{\theta} = (\phi, \lambda)^T$ then the *expected information matrix (EIM)* is given by

$$\begin{pmatrix} \dfrac{1-e^{-\lambda}}{(1-\phi)(\phi+(1-\phi)\,e^{-\lambda})} & \dfrac{-e^{-\lambda}}{\phi+(1-\phi)\,e^{-\lambda}} \\ \dfrac{-e^{-\lambda}}{\phi+(1-\phi)\,e^{-\lambda}} & \dfrac{1-\phi}{\lambda} - \dfrac{\phi(1-\phi)\,e^{-\lambda}}{\phi+(1-\phi)\,e^{-\lambda}} \end{pmatrix}.$$

Here is some simulated data example.

```
> set.seed(1111)
> N = 2000
> zdata = data.frame(x2 = runif(n = N))
> zdata = transform(zdata, phi = logit(-1 + 1 * x2,
+     inverse = TRUE), lambda = loge(-2 + 2 * x2, inverse = TRUE))
> zdata = transform(zdata, y = rzipois(N, lambda, phi))
> with(zdata, table(y))
```

Zero-inflated Poisson model IV

```
y
   0    1    2    3    4
1608  308   66   17    1

> fit = vglm(y ~ x2, zipoisson, zdata, trace = TRUE)

VGLM    linear loop  1 :  loglikelihood = -1283.2
VGLM    linear loop  2 :  loglikelihood = -1860.3
Taking a modified step...
VGLM    linear loop  2 :  loglikelihood = -1243.5
VGLM    linear loop  3 :  loglikelihood = -1223.9
VGLM    linear loop  4 :  loglikelihood = -1196.7
VGLM    linear loop  5 :  loglikelihood = -1194.4
VGLM    linear loop  6 :  loglikelihood = -1194.3
VGLM    linear loop  7 :  loglikelihood = -1194.3
VGLM    linear loop  8 :  loglikelihood = -1194.3

> coef(fit, matrix = TRUE)
```

Zero-inflated Poisson model V

```
            logit(phi) log(lambda)
(Intercept)   -0.27175     -1.8193
x2            -0.28954      1.6095

> fit2 = vglm(y ~ x2, zipoisson(shrinkage.init = 0.95),
+     zdata, trace = TRUE)

VGLM    linear loop  1 :  loglikelihood = -1222.6
VGLM    linear loop  2 :  loglikelihood = -1208.0
VGLM    linear loop  3 :  loglikelihood = -1195.1
VGLM    linear loop  4 :  loglikelihood = -1194.3
VGLM    linear loop  5 :  loglikelihood = -1194.3
VGLM    linear loop  6 :  loglikelihood = -1194.3

> coef(fit2, matrix = TRUE)

            logit(phi) log(lambda)
(Intercept)   -0.26973     -1.8185
x2            -0.29238      1.6083
```

Robustness of negbinomial() I

Example 2: Robustness of negbinomial().

$$P(Y=y) = \binom{y+k-1}{y} \left(\frac{\mu}{\mu+k}\right)^y \left(\frac{k}{k+\mu}\right)^k, \quad y = 0, 1, \ldots, \quad (9)$$

where $k > 0$.

```
> set.seed(123)
> ndata = data.frame(x = runif(n <- 500))
> ndata = transform(ndata, y1 = rnbinom(n, mu = exp(3 +
+     x), size = exp(1)))
```

That is, $n = 500$, $X_i \sim \text{Unif}(0, 1)$, $\mu = \exp(3 + x)$, $k = e^1 \approx 2.72$ in (9). Then

```
> ndata$y1[1] <- 1e+08
> nbfit <- vglm(y1 ~ x, negbinomial, ndata, trace = TRUE)
```

Robustness of negbinomial() II

```
VGLM    linear loop  1 :  loglikelihood = -3357.5
VGLM    linear loop  2 :  loglikelihood = -3340.7
VGLM    linear loop  3 :  loglikelihood = -3323.2
VGLM    linear loop  4 :  loglikelihood = -3305.1
VGLM    linear loop  5 :  loglikelihood = -3286.7
VGLM    linear loop  6 :  loglikelihood = -3268.7
VGLM    linear loop  7 :  loglikelihood = -3253.1
VGLM    linear loop  8 :  loglikelihood = -3243.5
VGLM    linear loop  9 :  loglikelihood = -3241.1
VGLM    linear loop  10 :  loglikelihood = -3241.1
VGLM    linear loop  11 :  loglikelihood = -3241.1
```

Here the response might typically have a maximum of around 200, but one of the values is replaced by 10^8. Convergence is still achieved!

Yet to do ...

- Many probability distributions have been implemented. But there are many more to go...
- The ease by which estimation can be performed increases the need for goodness-of-fits tests.
- Call **LAPACK** instead of **LINPACK**.
- Add random effect to linear predictors, e.g.,

$$g_j(\theta_j) \;=\; \eta_j \;=\; \boldsymbol{\beta}_j^T \mathbf{x} + \boldsymbol{\gamma}_j^T \mathbf{z} \tag{10}$$

  where $\boldsymbol{\gamma}_j \sim N_g(\mathbf{0}, \boldsymbol{\Sigma})$, say.
  Obtain the classes of *VGLMMs* (and later *VGAMMs*).
- Add automatic smoothing parameter selection, e.g., by using a generalized cross validation (GCV) criterion. The associated ideas of penalized regression splines (P-splines) have become very popular recently. See Wood (2006).

Closing Comments I

1. VGLMs and VGAMs are a very large class of models; VGLMs are model-driven while VGAMs are data-driven.
2. The VGLMs/VGAMs framework naturally allows for estimation for many statistical distributions.
3. There is *a lot* of future work to develop the methodology fully.

Closing Comments II

Fini

Closing Comments III

Fini

The End