An overview of the VGAM package

© T. W. Yee

University of Auckland

12–13 Nov 2011 © Shanghai

t.yee@auckland.ac.nz
http://www.stat.auckland.ac.nz/~yee
Outline of this talk

1. Introduction to VGLMs and VGAMs
2. Vector generalized linear models
3. VGAMs
4. Some examples
   - Zero-inflated Poisson model
   - Loglinear models for binary responses†
5. More complicated constraints†
6. More on VGAMs†
7. RR-VGLMs
8. Constrained Quadratic Ordination (CQO)†
9. Concluding remarks
The **VGAM** package 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,

\[ g_j(\theta_j) = \eta_j = \beta_j^T \mathbf{x} = \beta_{(j)1} x_1 + \cdots + \beta_{(j)p} x_p \]  (1)

where \( g_j \) is a parameter link function \((-\infty < \eta_j < \infty)\).

VGAMs extend (1) to

\[ g_j(\theta_j) = \eta_j = f_{(j)1}(x_1) + \cdots + f_{(j)p}(x_p), \]  (2)

i.e., an additive model for each parameter. Estimated by smoothers, this is a data-driven approach.
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 \( \mu (= E(Y)) > 0 \) and \( k > 0 \).

The **MASS** implementation is restricted to a intercept-only estimate of \( k \), e.g., cannot fit \( \log k = \beta(2)_1 + \beta(2)_2 x_2 \). In contrast, **VGAM** can fit

\[
\log \mu = \eta_1 = \beta_1^T x, \\
\log k = \eta_2 = \beta_2^T x.
\]

\texttt{vglm(y \sim x2 + \cdots + xp, family = 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 \( \eta \) are handled,
(iii) \( \eta_j \) need not be a function of a mean \( \mu \): \( \eta_j = g_j(\theta_j) \) for any parameter \( \theta_j \).

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 \( \implies \) 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.
Figure: Flowchart for different classes of models. Legend: \( \text{LM} = \) linear model, \( \text{V} = \) vector, \( \text{G} = \) generalized, \( \text{A} = \) additive, \( \text{RR} = \) reduced-rank, \( \text{Q} = \) quadratic.
### Introduction to VGLMs and VGAMs VII

<table>
<thead>
<tr>
<th>$t$</th>
<th>$\eta$</th>
<th>Model</th>
<th>S function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1^T x_1 + B_2^T x_2$ ($= B^T x$)</td>
<td>VGLM</td>
<td>vglm()</td>
<td>Yee &amp; Hastie (2003)</td>
<td></td>
</tr>
<tr>
<td>$B_1^T x_1 + \sum_{k=p_1+1}^{p_1+p_2} H_k f_k^*(x_k)$</td>
<td>VGAM</td>
<td>vgam()</td>
<td>Yee &amp; Wild (1996)</td>
<td></td>
</tr>
<tr>
<td>$B_1^T x_1 + A \nu$</td>
<td>RR-VGLM</td>
<td>rrvglm()</td>
<td>Yee &amp; Hastie (2003)</td>
<td></td>
</tr>
<tr>
<td>$B_1^T x_1 + A \nu + \begin{pmatrix} \nu^T D_1 \nu \ \vdots \ \nu^T D_M \nu \end{pmatrix}$</td>
<td>QRR-VGLM</td>
<td>cqe()</td>
<td>Yee (2004)</td>
<td></td>
</tr>
<tr>
<td>$B_1^T x_1 + \sum_{r=1}^{R} f_r(\nu_r)$</td>
<td>RR-VGAM</td>
<td>cao()</td>
<td>Yee (2006)</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** A summary of VGAM and its framework. The latent variables $\nu = C^T x_2$, or $\nu = c^T x_2$ if rank $R = 1$. Here, $x^T = (x_1^T, x_2^T)$. Abbreviations: $A =$ additive, $C =$ constrained, $L =$ linear, $O =$ ordination, $Q =$ quadratic, $RR =$ reduced-rank, $VGLM =$ vector generalized linear model.
Vector generalized linear models

Data \((x_1, y_1), \ldots, (x_n, y_n)\) on \(n\) independent “individuals”.

Definition Conditional distribution of \(y\) given \(x\) is

\[
f(y|x; \beta) = h(y, \eta_1, \ldots, \eta_M, \phi),
\]

where for \(j = 1, \ldots, M\),

\[
\eta_j = \eta_j(x) = \beta_j^T x,
\]

\[
\beta_j = (\beta_{(j)1}, \ldots, \beta_{(j)p})^T,
\]

\[
\beta = (\beta_1^T, \ldots, \beta_M^T)^T,
\]

\[
\phi = \text{a vector of scale factors}.
\]

Often \(g_j(\theta_j) = \eta_j\) for parameters \(\theta_j\) and link functions \(g_j\).
VGLM examples I

GLMs $M = 1$

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

Table: Summary of GLMs supported by VGAM. The known prior weight is $A$. These are incompatible with glm().
VGLM examples II

2 Negative binomial distribution

For $y = 0, 1, 2, \ldots$,

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

Good choice:

$$\eta_1 = \log \mu,$$
$$\eta_2 = \log k.$$

vglm(y \sim x2 + x3 + \ldots, \text{family} = \text{negbinomial(}zero = \text{NULL}))$$
VGLM examples III

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.
VGLM examples IV

Table: The coalminers data set from UK collieries. Note: 
$B =$ Breathlessness, $W =$ Wheeze, 1 = presence, 0 = absence.

<table>
<thead>
<tr>
<th>Age Group</th>
<th>$(B = 1, W = 1)$</th>
<th>$(B = 1, W = 0)$</th>
<th>$(B = 0, W = 1)$</th>
<th>$(B = 0, W = 0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20–24</td>
<td>9</td>
<td>7</td>
<td>95</td>
<td>1841</td>
</tr>
<tr>
<td>25–29</td>
<td>23</td>
<td>9</td>
<td>105</td>
<td>1654</td>
</tr>
<tr>
<td>30–34</td>
<td>54</td>
<td>19</td>
<td>177</td>
<td>1863</td>
</tr>
<tr>
<td>35–39</td>
<td>121</td>
<td>48</td>
<td>257</td>
<td>2357</td>
</tr>
<tr>
<td>40–44</td>
<td>169</td>
<td>54</td>
<td>273</td>
<td>1778</td>
</tr>
<tr>
<td>45–49</td>
<td>269</td>
<td>88</td>
<td>324</td>
<td>1712</td>
</tr>
<tr>
<td>50–54</td>
<td>404</td>
<td>117</td>
<td>245</td>
<td>1324</td>
</tr>
<tr>
<td>55–59</td>
<td>406</td>
<td>152</td>
<td>225</td>
<td>967</td>
</tr>
<tr>
<td>60–64</td>
<td>372</td>
<td>106</td>
<td>132</td>
<td>526</td>
</tr>
</tbody>
</table>

$p_j = P(Y_j = 1), \quad$ marginal probabilities

$p_{rs} = P(Y_1 = r, Y_2 = s), \quad r, s = 0, 1, \quad$ joint probabilities

$\psi = \frac{p_{00} p_{11}}{p_{01} p_{10}} \quad$ (Odds ratio).
VGLM examples V

Model:

\[ \text{logit } p_j(x) = \eta_j(x), \quad j = 1, 2, \]
\[ \log \psi(x) = \eta_3(x). \]

Recover \( p_{rs} \)'s from \( p_1, p_2 \) and \( \psi \).

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

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

\[ \text{cloglog } p_j(x) = \eta_1(x), \quad j = 1, 2, \]
\[ \log \psi(x) = \eta_3(x). \]
**VGLM examples VI**

**Note:**

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

\[(4)\]

> vglm(..., family = binom2.or("cloglog", exchangeable = TRUE))
Models for a categorical response i.e., $Y \in \{1, 2, \ldots, M + 1\}$.

$Y$ may be unordered (nominal) or ordered (ordinal).

**Table:** Period of exposure (years) and severity of pneumoconiosis amongst a group of coalminers.

<table>
<thead>
<tr>
<th>Exposure Time</th>
<th>Normal</th>
<th>Mild</th>
<th>Severe</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.8</td>
<td>98</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15.0</td>
<td>51</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>21.5</td>
<td>34</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>27.5</td>
<td>35</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>33.5</td>
<td>32</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>39.5</td>
<td>23</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>46.0</td>
<td>12</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>51.5</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>
VGLM examples VIII

(i) Multinomial logit model (nominal $Y$)

$$P(Y = j | x) = \frac{\exp\{\eta_j(x)\}}{\sum_{t=1}^{M+1} \exp\{\eta_t(x)\}}, \quad j = 1, \ldots, M + 1.$$ 

For identifiability: $\eta_{M+1} \equiv 0$.

Equivalently,

$$\log \left( \frac{P(Y = j | x)}{P(Y = M + 1 | x)} \right) = \eta_j(x), \quad j = 1, \ldots, M + 1.$$ 

\[ \text{vglm(ymatrix } \sim x2 + x3 + \ldots, \text{ family = multinomial)} \]
VGLM examples IX

(ii) Nonproportional odds model (ordinal $Y$)

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

Proportional odds model: constrain

$$\eta_j(x) = \alpha_j + \eta(x)$$

(aka the parallelism assumption, which stops the probabilities from becoming negative or greater than 1).

vglm(ymatrix $\sim$ x2 + \ldots, family = cumulative(parallel = TRUE))
VGLM examples X

(iii) Continuation ratio model (ordinal $Y$)

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

Good for sequential-type data, e.g., the effect of $\mathbf{x}$ on a couple who decide to have one child, two children, three children, . . .

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

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

vglm(..., family = cratio(link = probit))
VGLM examples XI

**Some probability link functions**

- Logit
- Probit
- Cloglog
- Cauchit

**First derivative**

**Some inverse probability link functions**

- Logit
- Probit
- Cloglog
- Cauchit
VGLM algorithm \( \dagger \) I

Models with log-likelihood

\[
\ell(\beta) = \sum_{i=1}^{n} \ell_i\{\eta_1(x_i), \ldots, \eta_M(x_i)\},
\]

where \( \eta_j = \beta_j^T x_i \). Then

\[
\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial \ell_i}{\partial \eta_j} x_i
\]

and

\[
\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k^T} = \sum_{i=1}^{n} \frac{\partial^2 \ell_i}{\partial \eta_j \partial \eta_k} x_i x_i^T.
\]

Newton-Raphson algorithm

\[
\beta^{(a+1)} = \beta^{(a)} + J(\beta^{(a)})^{-1} U(\beta^{(a)})
\]
VGLM algorithm\(^*\) II

written in iteratively reweighted least squares (IRLS) form is

\[
\beta^{(a+1)} = (X^T W X)^{-1} X^T W X \beta^{(a)} + (X^T W X)^{-1} X^T W W^{-1} u
\]

\[
= \left( X_{VLM}^T W^{(a)} X_{VLM} \right)^{-1} X_{VLM}^T W^{(a)} z^{(a)}.
\]

Let \( z = (z_1^T, \ldots, z_n^T)^T \) and \( u = (u_1^T, \ldots, u_n^T)^T \), where \( u_i \) has \( j \)th element

\[
(u_i)_j = \frac{\partial \ell_i}{\partial \eta_j},
\]

and \( z_i = \eta(x_i) + W_i^{-1} u_i \) (adapted dependent vector or pseudo-response).

Also, \( W = \text{Diag}(W_1, \ldots, W_n) \), \( (W_i)_{jk} = -\frac{\partial^2 \ell_i}{\partial \eta_j \partial \eta_k} \),

\( X_{VLM} = (X_1^T, \ldots, X_n^T)^T \), \( X_i = \text{Diag}(x_1^T, \ldots, x_i^T) = I_M \otimes x_i^T \).
VGLM algorithm

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

\[
z^{(a)} = X_{VLM} \beta^{(a+1)} + \varepsilon^{(a)}, \quad \text{Var}(\varepsilon^{(a)}) = \phi W^{(a)}^{-1}.
\]

**Fisher scoring:**

\[
(W_i)_{jk} = -E \left[ \frac{\partial^2 \ell_i}{\partial \eta_j \partial \eta_k} \right]
\]

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

Some Notes

1. $wz$ computed in @weight is usually

$$ (W_i)_{jk} = - E \left( \frac{\partial^2 l_i}{\partial \eta_j \partial \eta_k} \right), \quad \text{sometimes} \quad - \frac{\partial^2 l_i}{\partial \eta_j \partial \eta_k}. $$

2. The following formulae are useful.

$$ \frac{\partial l}{\partial \eta_j} = \frac{\partial l}{\partial \theta_j} \frac{\partial \theta_j}{\partial \eta_j}, $$

$$ \frac{\partial^2 l}{\partial \eta_j^2} = \frac{\partial l}{\partial \theta_j} \frac{\partial^2 \theta_j}{\partial \eta_j^2} + \left( \frac{\partial \theta_j}{\partial \eta_j} \right)^2 \frac{\partial^2 l}{\partial \theta_j^2}, $$

$$ \frac{\partial^2 l}{\partial \eta_j \partial \eta_k} = \left\{ \frac{\partial^2 l}{\partial \theta_j \partial \theta_k} - \frac{\partial l}{\partial \theta_k} \frac{\partial \theta_k}{\partial \eta_k} \frac{\partial^2 \eta_k}{\partial \theta_j \partial \theta_k} \right\} \frac{\partial \theta_j}{\partial \eta_j} \frac{\partial \theta_k}{\partial \eta_k}, \quad j \neq k, $$
The vector linear model (VLM) is the central model behind VGLMs and VGAMs. Its crux is to minimize

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

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

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

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

\[
(Y_1 \cdots Y_M) = XB + U, \quad u_i \sim (0, \Sigma)
\]  

where \(U = (u_1, \ldots, u_n)^T\).
The VGAM package for R I

Written in S, its central core are the functions `vglm()`, `vgam()` and `rrvglm()`.

Generic functions include `coef()`, `fitted()`, `plot()`, `predict()`, `print()`, `resid()`, `summary()`. Others are `lvplot()`, `Coef()`, `df.residual()`, `logLik()`, `vcov()`.

Plus lots and lots of VGAM family functions.

Modular construction, flexible, easy to use and useful.

Runs S4 (Chambers, 1998) in R.

Can install the VGAM package in R by typing

```
> install.packages("VGAM")
```
The VGAM package for R II

The central functions of VGAM

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

Others:

- vlm() Vector linear models.
- grc() Goodman’s RC(r) model.
- rcam() Row-column association models (not complete).
The VGAM package for R III

Package: VGAM
Version: 0.8-4
Date: 2011-11-03
Title: Vector Generalized Linear and Additive Models
Author: Thomas W. Yee <t.yee@auckland.ac.nz>
Maintainer: Thomas Yee <t.yee@auckland.ac.nz>
Depends: R (>= 2.11.1), splines, methods, stats, stats4
Description: Vector generalized linear and additive models, and associated models (Reduced-Rank VGLMs, Quadratic RR-VGLMs, Reduced-Rank VGAMs). This package fits many models and distribution by maximum likelihood estimation (MLE) or penalized MLE. Also fits constrained ordination models in ecology.
License: GPL-2
URL: http://www.stat.auckland.ac.nz/~yee/VGAM

See the INDEX file.
The VGAM package for R IV

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>coef(fit)</td>
<td>( \hat{\beta}^* )</td>
</tr>
<tr>
<td>coef(fit, matrix = TRUE)</td>
<td>( \hat{B} )</td>
</tr>
<tr>
<td>constraints(fit, type = &quot;lm&quot;)</td>
<td>( H_k, \ k = 1, \ldots, p )</td>
</tr>
<tr>
<td>deviance(fit)</td>
<td>Deviance ( D = \sum_{i=1}^{n} d_i )</td>
</tr>
<tr>
<td>fitted(fit)</td>
<td>( \hat{\mu}_{ij} ) usually</td>
</tr>
<tr>
<td>logLik(fit)</td>
<td>Log-likelihood ( \sum_{i=1}^{n} w_i \ell_i )</td>
</tr>
<tr>
<td>model.matrix(fit, type = &quot;lm&quot;)</td>
<td>LM model matrix ((n \times p))</td>
</tr>
<tr>
<td>model.matrix(fit, type = &quot;vlm&quot;)</td>
<td>VLM model matrix (X_{VLM}^{VL} )</td>
</tr>
<tr>
<td>predict(fit)</td>
<td>( \hat{\eta}_{ij} )</td>
</tr>
</tbody>
</table>
The VGAM package for R V

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>predict(fit, type = &quot;response&quot;)</code></td>
<td>$\hat{\mu}_{ij}$ usually</td>
</tr>
<tr>
<td><code>resid(fit, type = &quot;response&quot;)</code></td>
<td>$y_{ij} - \hat{\mu}_{ij}$</td>
</tr>
<tr>
<td><code>resid(fit, type = &quot;deviance&quot;)</code></td>
<td>$\text{sign}(y_i - \hat{\mu}_i) \sqrt{d_i}$</td>
</tr>
<tr>
<td><code>resid(fit, type = &quot;pearson&quot;)</code></td>
<td>$W_i^{-\frac{1}{2}} u_i$</td>
</tr>
<tr>
<td><code>resid(fit, type = &quot;working&quot;)</code></td>
<td>$z_i - \eta_i = W_i^{-1} u_i$</td>
</tr>
<tr>
<td><code>vcov(fit)</code></td>
<td>$\text{Var}(\hat{\beta})$</td>
</tr>
<tr>
<td><code>weights(fit, type = &quot;prior&quot;)</code></td>
<td>$w_i$ (weights argument)</td>
</tr>
<tr>
<td><code>weights(fit, type = &quot;working&quot;)</code></td>
<td>$w_i W_i$ (in matrix-band format)</td>
</tr>
</tbody>
</table>
The VGAM package for R VI

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

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

- **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.
Some computational and implementational details†

- Is S4 object-orientated and very modular—simply have to write a VGAM “family function”.
- VGAM minimizes $\hat{\text{deviance}}$, or maximizes $\hat{\text{loglikelihood}}$ or iterates till change in $\hat{\beta}^{(a+1)}$ is sufficiently small. User may specify some other objective function, e.g., $\hat{\text{rss}}$.

**Convergence criteria:** for a scalar quantity $A$

$$\frac{|A^{(a+1)} - A^{(a)}|}{\text{epsilon} + |A^{(a)}|} < \text{epsilon}; \quad (6)$$

for the regression coefficients the maximum over $j = 1, \ldots, p$ of

$$\frac{|\hat{\beta}_j^{(a+1)} - \hat{\beta}_j^{(a)}|}{\text{epsilon} + |\hat{\beta}_j^{(a)}|} < \text{epsilon}.$$

for all $j = 1, \ldots, p$. 
Some computational and implementational details

> args(vglm)

function (formula, family, 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 = FALSE, 
smart = TRUE, ...)

NULL

> 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. [Safe prediction not as good as smart prediction, e.g., bs(scale(x)), I(bs(x)), poly(scale(x), 2)].
VGAMs I

VGAMs allow additive-model extensions to all $\eta_j$ in a VGLM, i.e., from

$$\eta_j(x) = \beta_{(j)1} x_1 + \cdots + \beta_{(j)p} x_p = \beta_j^T x$$

to $M$ additive predictors:

$$\eta_j(x) = f_{(j)1}(x_1) + \cdots + f_{(j)p}(x_p),$$

a sum of arbitrary smooth functions. Equivalently,

$$\eta(x) = f_1(x_1) + \cdots + f_p(x_p)$$

$$= H_1 f_1^*(x_1) + \cdots + H_p f_p^*(x_p) \quad (7)$$

for some constraint matrices $H_k$ (default: $H_k = I_M$).

- $H_1, \ldots, H_p$ 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 logistic model**
   
   All
   
   \[ H_k = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}. \]

2. **The proportional odds model**
   
   \[ H_1 = I_M, \quad H_2 = \cdots = H_p = 1_M. \]

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

\[ \texttt{binom2.or(exchangeable = TRUE)} \]

\[ \texttt{cumulative(parallel = FALSE ~ x5 - 1)} \]
Constraints I

General formula:

\[ \eta(x) = a + \sum_{k=1}^{p} H_k f^*_k(x_k). \]  

Some types of constraints can be more easily handled by arguments such as exchangeable, parallel, zero.

Examples

1. \( > \text{vglm}(y \sim x, \text{cumulative(parallel = FALSE)}) \)
   
   stops the parallelism constraint from being applied to any of the explanatory variables (default). The result is a non-proportional odds model.

2. \( > \text{vglm}(y \sim x, \text{multinomial(zero=2)}) \)
   
   will force \( \eta_2 \) to be an intercept-only, i.e., \( \eta_2 = \beta_{(2)1} \).
Both

Both

\>
\texttt{vgam(y~x2+s(x3,3)+s(x4)+x5, binom2.or(exch= TRUE ~ s(x3,3)+x5, zero=FALSE))}
\>
\texttt{vgam(y~x2+s(x3,3)+s(x4)+x5, binom2.or(exch= FALSE ~ x2+s(x4)-1, zero=FALSE))}

makes the effect of \(s(X_3)\) and \(X_5\) to be the same for both marginal probabilities. Explicitly, the model they fit is

\[
\text{logit } p_1 = \beta^{*\text{(1)}}_1 + \beta^{*\text{(1)}}_2 x_2 + f^{*\text{(1)}}_3(x_3) + f^{*\text{(1)}}_4(x_4) + \beta^{*\text{(1)}}_5 x_5,
\]

\[
\text{logit } p_2 = \beta^{*\text{(1)}}_1 + \beta^{*\text{(2)}}_2 x_2 + f^{*\text{(1)}}_3(x_3) + f^{*\text{(2)}}_4(x_4) + \beta^{*\text{(1)}}_5 x_5,
\]

\[
\log \psi(x) = \beta^{*\text{(2)}}_1 + \beta^{*\text{(3)}}_2 x_2 + f^{*\text{(2)}}_3(x_3) + f^{*\text{(3)}}_4(x_4) + \beta^{*\text{(2)}}_5 x_5.
\]
Constraints III

> vgam(y ~ x2 + s(x3), binom2.or(zero=3))

constrains $\eta_3 = \beta_{(3)1}$. So the odds ratio is simply a point estimate and not a function of the covariates. In general, zero may be assigned a vector of integers in the range 1 to $M$. Negative values allowed for multiple responses. Note that the exchangeability constraint applies to the intercepts, whereas the parallelism constraint doesn’t.

Set zero = NULL if the constraints argument is used.
More examples I

(\eta_1 = \eta_2; \text{ e.g., exchangeability in the bivariate logit model})

\begin{align*}
\eta_1 &= \alpha_1^* + f_{(1)2}(x_2) + f_{(1)3}(x_3) \\
\eta_2 &= \alpha_1^* + f_{(1)2}(x_2) + f_{(1)3}(x_3) \\
\eta_3 &= \alpha_2^* + f_{(2)2}(x_2) + f_{(2)3}(x_3)
\end{align*}

Then

\[
H_1 = H_2 = H_3 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.
\]

If the \( f_{(j)k}^* \) are linear then

\begin{verbatim}
> cmat = matrix(c(1,1,0, 0,0,1), 3, 2)
> clist = list("(Intercept)"=cmat, x2=cmat, x3=cmat)
> vglm(y ~ 1 + x2 + x3, constraints = clist, ...)
\end{verbatim}
More examples II

(\eta_j(\mathbf{x}) = \alpha_j + \eta(\mathbf{x}); \text{ e.g., parallelism in the proportional odds model})

\begin{align*}
\eta_1 &= \alpha_1^* + f_{(1)2}^*(x_2) + f_{(1)3}^*(x_3) \\
\eta_2 &= \alpha_2^* + f_{(1)2}^*(x_2) + f_{(1)3}^*(x_3) \\
\eta_3 &= \alpha_3^* + f_{(1)2}^*(x_2) + f_{(1)3}^*(x_3)
\end{align*}

Then

\[ H_1 = I_3, \quad H_2 = H_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \]

If the \( f_{(j)k} \) are linear then

\begin{verbatim}
> cm = matrix(1, 3, 1)
> clist = list("(Intercept)"=diag(3), x2=cm, x3=cm)
> vglm(y ~ x2 + x3, constraints=clist, ...)
\end{verbatim}
More examples III

**VGAM** allows the input of the constraint matrices $H_1, \ldots, H_p$ through the `constraints` argument. (The default value of `NULL` signifies no constraints at all).

`constraints` must be assigned a list containing (constraint) matrices or functions that create them. Each of these must be named with the variable name, or term in the case of `s()`, `bs()` etc.

If any explanatory variable is a factor then only the name of the factor (not any of its levels) needs to appear in `constraints`.

As it can be seen, `constraints` is powerful and flexible, although a little cumbersome. Users are encouraged to use `zero/parallel/exchangeable` where possible.

The constraint matrices of a **VGAM** object can be seen by typing, e.g., `constraints(fit)`. 
Estimation† I

Estimate the $f_k$ by *penalized likelihood*. Let

$$f_k = (f_{(1)k}(x_{1k}), \ldots, f_{(M)k}(x_{1k}), \ldots, f_{(1)k}(x_{nk}), \ldots, f_{(M)k}(x_{nk}))^T \quad \text{and}$$

$$\eta = (\eta_1^T, \ldots, \eta_n^T)^T = \sum_{k=1}^p f_k, \quad \text{where}$$

$$\eta_i^T = (\eta_1(x_i), \ldots, \eta_M(x_i)).$$

The penalized log-likelihood can be written as

$$j(f_1, \ldots, f_p) = \ell(\eta; y) - \frac{1}{2} \sum_{k=1}^p \sum_{j=1}^M \lambda_{(j)k} \int_{a_k}^{b_k} \left\{ f_{(j)k}''(x_k) \right\}^2 \, dx_k$$

$$= \ell(\eta; y) - \frac{1}{2} \sum_{k=1}^p f_k^T K_k f_k \quad (9)$$
where \( K_k \) is the penalty matrix corresponding to smoothing with respect to the \( k \)th covariate and depends upon \( \lambda_k \).

Maximizing \( j(f_1, \ldots, f_p) \) with respect to \( f_1, \ldots, f_p \) using Newton-Raphson gives

\[
\begin{pmatrix}
W + K_1 & W & \cdots & W \\
W & W + K_2 & \cdots & \vdots \\
\vdots & \vdots & \ddots & W \\
W & \cdots & W & W + K_p
\end{pmatrix}
\begin{pmatrix}
f_1^1 - f_1^0 \\
f_2^1 - f_2^0 \\
\vdots \\
f_p^1 - f_p^0
\end{pmatrix}
= \begin{pmatrix}
u - K_1 f_1^0 \\
u - K_2 f_2^0 \\
\vdots \\
u - K_p f_p^0
\end{pmatrix}.
\]

See Hastie and Tibshirani (1990, sec. 6.5.2) and Green and Silverman (1994).

With \( p = 1 \) covariates, this leads to vector smoothing with vector splines.

**Result:** A VGAM using a vector spline maximizes the penalized likelihood (9).
Vector smoothing†

One has a *vector response* $y_i \ (M \times 1)$ at each scalar $x_i$:

$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \sim (0, \Sigma_i)$$

(10)

independently. Here,

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_M(x) \end{pmatrix}$$

is a vector of $M$ arbitrary smooth functions and $i = 1, \ldots, n$. 
Applications of vector smoothing†

Actually, the vector smoothing problem has few applications in practice because $\Sigma_i$ are unknown. Its primary use is in fitting the VGAM class. There are two good ways of vector smoothing:

- **Vector splines** Originally proposed by Fessler (1991), his solution was based on the Reinsch algorithm. Unfortunately, this may sometimes be numerically unstable. Better solution: use B-splines.

- **Local regression for vector responses** Joint work with Alan Welsh has resulted in a quasi-likelihood estimator, and a derivation of its asymptotic properties (for the simplest case of $M = 2$ and local linear fitting).
Vector splines†

Vector splines minimize

\[
\sum_{i=1}^{n} \{y_i - f(x_i)\}^T \Sigma_i^{-1} \{y_i - f(x_i)\} + \sum_{j=1}^{M} \lambda_j \int_{a}^{b} \{f_j''(x)\}^2 \, dx,
\]

over a Sobolev space of order 2. Here, \(a < x_1 < \cdots < x_n < b\) for some \(a\) and \(b\), and \(\lambda \geq 0\).

Write

\[
\begin{align*}
\mathbf{y} &= (y_1^T, \ldots, y_n^T)^T \\
\Sigma &= \text{diag}(\Sigma_1, \ldots, \Sigma_n) \\
f &= (f_1(x_1), \ldots, f_M(x_1), \ldots, f_1(x_n), \ldots, f_M(x_n))^T.
\end{align*}
\]
Some notes† I

• Special case of $M = 1$ simplifies to a cubic smoothing spline.
• Fits into the penalty function framework of Green and Silverman (1994).
• Penalized least squares

Minimize:

$$
(y - f)^T \Sigma^{-1}(y - f) + f^T K f.
$$

Solution:

$$
\hat{f} = A(\lambda) y
$$

where $A(\lambda) = (I_{nM} + \Sigma K)^{-1}$ is the influence or smoother matrix.
Some notes† II

- Degrees of freedom of smooth:

\[ df = \text{trace}(A), \]
\[ df^{\text{var}} = \text{trace}(AA^T), \]
\[ df^{\text{err}} = nM - \text{trace}(2A - A^TA) \]
\[ df_{(m)} = df \text{ of } f_m = \text{sum of diagonal elements corresponding to } f_m \]

- Standard Errors

\[ \text{Cov}(\hat{f}) = A(\lambda)\Sigma A(\lambda)^T \text{ is used to form pointwise SE bands } \]
\[ \text{Cov}(\hat{f}) = A(\lambda)\Sigma \text{ is a Bayesian alternative. Cheaper. } \]
VGAM algorithm: vector backfitting

Vector backfitting is the backfitting algorithm using vector smoothing.

\[ E(Y|X) = f_t(x_t) + \sum_{k=1, k \neq t}^p f_k(x_k) \]

So

\[ f_t(x_t) = E \left( Y - \sum_{k=1, k \neq t}^p f_k(x_k) \bigg| X_t \right). \]

Modified vector backfitting possible—and is implemented. It decomposes

\[ \eta(x) = X \beta + \sum_{k=1}^p r_k(x_k) \]

i.e., into a linear and nonlinear components.
Some examples I

The following are some simple VGAM examples.
Some examples II

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

Table: Dispersion models implemented in VGAM (Jørgensen, 1997).
Gamma Distributions

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Density function $f(y; \theta)$</th>
<th>Range of $y$</th>
<th>Mean</th>
<th>VGAM family</th>
</tr>
</thead>
<tbody>
<tr>
<td>gamma (2-parameter)</td>
<td>$\frac{\lambda (\lambda y/\mu)^{\lambda - 1} \exp(-\lambda y/\mu)}{\mu \Gamma(\lambda)}$</td>
<td>$(0, \infty)$</td>
<td>$\mu$</td>
<td>gamma2</td>
</tr>
<tr>
<td>gamma (generalized)</td>
<td>$\frac{\mu \Gamma(\lambda)}{d b - d k \Gamma(k)} \frac{y^{d k - 1} \exp(-(y/b)^d)}{\exp {k y - \exp(y)}}$</td>
<td>$(0, \infty)$</td>
<td>$b k$</td>
<td>gengamma</td>
</tr>
<tr>
<td>log gamma (standard)</td>
<td>$\frac{\Gamma(k)}{\exp \left{k y - \exp(y)\right}}$</td>
<td>$(-\infty, \infty)$</td>
<td>$\psi(k)$</td>
<td>lgammaaff</td>
</tr>
<tr>
<td>log gamma (3-parameter)</td>
<td>$\frac{b \Gamma(k)}{\exp \left{k(y-a)/b - \exp \left(y-a \over b\right)\right}}$</td>
<td>$(-\infty, \infty)$</td>
<td>$a + b \psi(k)$</td>
<td>lgamma3ff</td>
</tr>
<tr>
<td>McCullagh (1989)</td>
<td>$\frac{\Gamma(k)}{\exp \left{k(y-a)/b - \exp \left(y-a \over b\right)\right}} \frac{(y^2 - 1)^{\nu - 1/2}}{{1 - y^2}^{\nu - 1/2}}$</td>
<td>$(-1, 1)$</td>
<td>$\frac{\nu \theta}{1+\nu}$</td>
<td>mccullagh89</td>
</tr>
<tr>
<td>Prentice (1974)</td>
<td>$\frac{B(\nu + \frac{1}{2}, \frac{1}{2}) (1 - 2\theta y + \theta^2)^\nu}{</td>
<td>q</td>
<td>\exp(w/q^2 - e^w)} \frac{b \Gamma(1/q^2)}{\Gamma(1/q^2)^\nu}$</td>
<td>$(-\infty, \infty)$</td>
</tr>
</tbody>
</table>

Table: Some gamma-type distributions currently supported by VGAM.
Some examples

## Size Distributions

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Density function $f(y; \theta)$</th>
<th>Mean (subject to range restrictions)</th>
<th>VGAM family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beta II</td>
<td>$y^{p-1}$</td>
<td>$\frac{b \Gamma(p+1) \Gamma(q-1)}{\Gamma(p) \Gamma(q)}$</td>
<td>betaII</td>
</tr>
<tr>
<td>Dagum</td>
<td>$b^p B(p, q) \left{ 1 + \frac{y}{b} \right}^{p+q}$</td>
<td>$\frac{b \Gamma(p + a^{-1}) \Gamma(1 - a^{-1})}{\Gamma(p)}$</td>
<td>dagum</td>
</tr>
<tr>
<td>Fisk</td>
<td>$b^a \left{ 1 + \left( \frac{y}{b} \right)^a \right}^{p+1}$</td>
<td>$\frac{b \Gamma(1 + a^{-1}) \Gamma(1 - a^{-1})}{\Gamma(p)}$</td>
<td>fisk</td>
</tr>
<tr>
<td>Generalized beta II</td>
<td>$b^{ap} B(p, q) \left{ 1 + \left( \frac{y}{b} \right)^a \right}^{p+q}$</td>
<td>$\frac{b \Gamma(p + a^{-1}) \Gamma(q - a^{-1})}{\Gamma(p) \Gamma(q)}$</td>
<td>genbetaII</td>
</tr>
<tr>
<td>Inverse Lomax</td>
<td>$b^p \left{ 1 + \frac{y}{b} \right}^{p+1}$</td>
<td>NA</td>
<td>invlomax</td>
</tr>
<tr>
<td>Inverse paralogistic</td>
<td>$b^a \left{ 1 + \left( \frac{y}{b} \right)^a \right}^{a+1}$</td>
<td>$\frac{b \Gamma(a + a^{-1}) \Gamma(1 - a^{-1})}{\Gamma(a)}$</td>
<td>invparalogistic</td>
</tr>
<tr>
<td>Lomax</td>
<td>$b \left{ 1 + \left( \frac{y}{b} \right)^a \right}^{1+q}$</td>
<td>$\frac{b}{\Gamma(a)}$</td>
<td>lomax</td>
</tr>
<tr>
<td>Paralogistic</td>
<td>$b^a \left{ 1 + \left( \frac{y}{b} \right)^a \right}^{1+a}$</td>
<td>$\frac{b \Gamma(1 + a^{-1}) \Gamma(1 - a^{-1})}{\Gamma(a)}$</td>
<td>paralogistic</td>
</tr>
<tr>
<td>Singh-Maddala</td>
<td>$b^a \left{ 1 + \left( \frac{y}{b} \right)^a \right}^{1+q}$</td>
<td>$\frac{b \Gamma(1 + a^{-1}) \Gamma(q - a^{-1})}{\Gamma(q)}$</td>
<td>sinmad</td>
</tr>
</tbody>
</table>

**Table:** Kleiber and Kotz (2003) models implemented in **VGAM**.
# Bivariate Distributions

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Cumulative distribution function $F(y_1, y_2; \theta)$</th>
<th>VGAM family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ali-Mikhail-Haq</td>
<td>$\frac{y_1 y_2}{1 - \alpha(1 - y_1)(1 - y_2)}$</td>
<td>amh</td>
</tr>
<tr>
<td>Farlie-Gumbel-Morgenstern</td>
<td>$y_1 y_2 \left[ 1 + \alpha(1 - y_1)(1 - y_2) \right]$</td>
<td>fgm</td>
</tr>
<tr>
<td>Frank</td>
<td>$\log_{\alpha} \left[ 1 + \frac{(\alpha y_1 - 1)(\alpha y_2 - 1)}{\alpha - 1} \right]$</td>
<td>frank</td>
</tr>
<tr>
<td>Gamma hyperbola</td>
<td>$f(y) = \exp \left{ -e^{-\theta} y_1 / \theta - \theta y_2 \right}$</td>
<td>gammahyp</td>
</tr>
<tr>
<td>Gumbel’s Type I</td>
<td>$e^{-y_1 - y_2 + \alpha y_1 y_2} + 1 - e^{-y_1} - e^{-y_2}$</td>
<td>gumbelIbiv</td>
</tr>
<tr>
<td>McKay’s bivariate gamma</td>
<td>$f(y) = \frac{y_1^{p-1} (y_2 - y_1)^{q-1} e^{-y_2/a}}{a^{p+q} \Gamma(p) \Gamma(q)}$</td>
<td>bivgamma.mckay</td>
</tr>
<tr>
<td>Morgenstern</td>
<td>$e^{-y_1 - y_2} \left( 1 + \alpha [1 - e^{-y_1}] [1 - e^{-y_2}] \right) + 1 - e^{-y_1} - e^{-y_2}$</td>
<td>morgenstern</td>
</tr>
<tr>
<td>Plackett</td>
<td>$f(y) = \frac{\psi \left[ 1 + (\psi - 1)(y_1 + y_2 - 2y_1 y_2) \right]}{\left[ (1 + (\psi - 1)(y_1 + y_2))^2 - 4\psi(\psi - 1)y_1 y_2 \right]^{3/2}}$</td>
<td>plackett</td>
</tr>
</tbody>
</table>

**Table:** Some bivariate distributions currently supported by **VGAM**.
### Zero-inflated, Zero-Altered and Positive Models

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Random variates functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-altered negative binomial</td>
<td>[dpqr]zegbin()</td>
</tr>
<tr>
<td>Zero-altered Poisson</td>
<td>[dpqr]zpoin()</td>
</tr>
<tr>
<td>Zero-inflated binomial</td>
<td>[dpqr]zingbin()</td>
</tr>
<tr>
<td>Zero-inflated geometric</td>
<td>[dpqr]zigeom()</td>
</tr>
<tr>
<td>Zero-inflated negative binomial</td>
<td>[dpqr]zingbin()</td>
</tr>
<tr>
<td>Zero-inflated Poisson</td>
<td>[dpqr]zpoin()</td>
</tr>
<tr>
<td>Positive binomial</td>
<td>[dpqr]posbinom()</td>
</tr>
<tr>
<td>Positive negative binomial</td>
<td>[dpqr]posnegbinom()</td>
</tr>
<tr>
<td>Positive normal</td>
<td>[dpqr]posnorm()</td>
</tr>
<tr>
<td>Positive Poisson</td>
<td>[dpqr]pospois()</td>
</tr>
</tbody>
</table>

**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.
Zero-inflated Poisson model

Loosely,

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

where \( 0 < \phi < 1 \) and \( \lambda > 0 \).
Zero-inflated Poisson model II

**Example:** \( Y = \) 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 \( \phi \).

Actually,

\[
P(Y = 0) = \phi + (1 - \phi) e^{-\lambda},
\]
\[
P(Y = y) = (1 - \phi) \frac{e^{-\lambda} \lambda^y}{y!}, \quad y = 1, 2, \ldots,
\]

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

\[
\eta = \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} = \begin{pmatrix} \text{logit} \phi \\ \log \lambda \end{pmatrix}.
\]
Zero-inflated Poisson model III

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

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

Here is some simulated data example.
Zero-inflated Poisson model IV

```r
> 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))

y
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
868 229 218 190 122 116 91 65 45 20 19 5 7 2 1
  15 16
  1 1

> fit = vglm(y ~ x2, zipoisson, zdata, trace=TRUE)
```
Zero-inflated Poisson model V

VGLM  linear loop  1 :  loglikelihood = -3402.505
VGLM  linear loop  2 :  loglikelihood = -3338.938
VGLM  linear loop  3 :  loglikelihood = -3338.638
VGLM  linear loop  4 :  loglikelihood = -3338.638

> # fit = vglm(y ~ x2, zipoisson, zdata, crit = "coef", trace=TRUE)
> coef(fit, matrix=TRUE)  # These should agree with the above values

        logit(\phi)  log(\lambda)  
(Intercept) -0.9494578   2.034661  
x2 0.7684691  -2.122872

> # fit2 = vglm(y ~ x2, zipoisson(shrinkage.init = 0.95), zdata, trace=TRUE)
> # coef(fit2, matrix=TRUE)  # These should agree with the above values
Loglinear models for binary responses

For bivariate binary responses $Y_1$ and $Y_2,$

$$\log P(Y_1 = y_1, Y_2 = y_2 | x) = u_0(x) + u_1(x) y_1 + u_2(x) y_2 + u_{12}(x) y_1 y_2$$

(11)

where $y_j = 0$ or $1,$

$$\begin{pmatrix}
  u_1(x) \\
  u_2(x) \\
  u_{12}(x)
\end{pmatrix}
= \eta(x)
= \begin{pmatrix}
  \eta_1(x) \\
  \eta_2(x) \\
  \eta_3(x)
\end{pmatrix}.$$
Loglinear models for binary responses

In general, suppose the data are \((y_{i1}, \ldots, y_{iS}, x_i), i = 1, \ldots, n\), where each \(y_{ij}\) is a binary response. Then only allow for pairwise associations:

\[
\log P(Y_1 = y_1, \ldots, Y_S = y_S | x) = u_0(x) + \sum_{j=1}^{S} u_j(x) y_j + \sum_{j < k} u_{jk}(x) y_j y_k.
\] (12)

The normalizing parameter \(u_0\) satisfies

\[
e^{-u_0} = 1 + \sum_{j=1}^{S} e^{u_j} + \sum_{j<k} e^{u_j+u_k+u_{jk}} + \sum_{j<k<\ell} e^{u_j+u_k+u_\ell+u_{jk}+u_{j\ell}+u_{k\ell}} + \cdots + \exp \left( \sum_{j=1}^{S} u_j + \sum_{j<k} u_{jk} \right).
\]
Loglinear models for binary responses† III

One has

$$\eta = (\eta_1, \ldots, \eta_M)^T = (u_1, \ldots, u_S, u_{12}, \ldots, u_{S-1,S})^T$$

where $M = S(S + 1)/2$. (An identity link for each of the $u$'s is chosen because the parameter space is unconstrained.)

With IRLS, Newton-Raphson $\equiv$ Fisher scoring.

```r
> # data(hunua)
> fits = vgam(cbind(dacdac, metrob) ~ s(altitude, df=c(4,4,1.5)),
              loglinb2, hunua)
> par(mfrow=c(2,2), las=1, mar=c(5,5,1,1))
> plot(fits, se=TRUE, cex=0.9, lcol="blue")
```
Loglinear models for binary responses

Some examples

Loglinear models for binary responses

IV

altitudes(altitude, df = c(4, 4, 1.5)):1

altitudes(altitude, df = c(4, 4, 1.5)):2

altitudes(altitude, df = c(4, 4, 1.5)):3
More complicated constraints†

The $x_{ij}$ facility allows for quite a lot of flexibility among all the regression coefficients, e.g., mixed logit models and nested logit models in discrete choice models. Further examples follow soon.

The crucial equation is (13):

$$\eta_i = \sum_{k=1}^{p} \text{diag}(x_{ik1}, \ldots, x_{ikM}) H_k \beta_k^* \quad \left(= \sum_{k=1}^{p} X_{(ik)}^* H_k \beta_k^*, \text{ say.}\right)$$

Each component of the list $x_{ij}$ is a formula having $M$ terms (ignoring the intercept) which specifies the successive diagonal elements of the matrix $X_{(ik)}^*$. Thus each row of the constraint matrix may be multiplied by a different vector of values. The constraint matrices themselves are not affected by the $x_{ij}$ argument.
More complicated constraints

Example 1

We assume that the data frame \texttt{myframe} has variables \( x_1 \) and \( x_2 \).

\textbf{Q1:} How can we fit

\[
\eta_1 = \beta_{(1)1} x_1 + \beta_{(1)2} x_2 \quad (14)
\]
\[
\eta_2 = \beta_{(2)1} x_1 + \beta_{(2)2} x_2 \quad (15)
\]

subject to \( \beta_{(1)1} + \beta_{(2)2} = \beta_{(2)1} + \beta_{(1)2} \)?

\textbf{A1:}

\[
\eta_2 = \beta_{(2)1} x_1 + \left( \beta_{(2)1} + \beta_{(1)2} - \beta_{(1)1} \right) x_2 \\
= \beta_{(2)1} (x_1 + x_2) + \beta_{(1)2} x_2 - \beta_{(1)1} x_2.
\]
More complicated constraints

\[ \eta = \text{diag}(x_1, x_2) \begin{pmatrix} 1 \\ -1 \end{pmatrix} \beta_{(1)1} + x_2 \begin{pmatrix} 1 \\ -1 \end{pmatrix} \beta_{(1)2} + x_3 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \beta_3 \]

\[ = \text{diag}(x_1, x_2) \begin{pmatrix} 1 \\ -1 \end{pmatrix} \beta^*_{(1)1} + (x_2 \mathbf{1}_2) \begin{pmatrix} 1 \\ -1 \end{pmatrix} \beta^*_{(1)2} + (x_3 \mathbf{1}_2) \begin{pmatrix} 0 \\ 1 \end{pmatrix} \beta^*_{(3)1} \]

where \( x_3 = x_1 + x_2 \) and \( \beta_3 = \beta_{(2)1} \), say. That is, \( \beta_{(1)1} = \beta^*_{(1)1} \), \( \beta_{(1)2} = \beta^*_{(1)2} \), \( \beta^*_{(3)1} = \beta_3 \).

myframe = transform(myframe, X1 = x1, x3 = x1 + x2)
Hlist = list(X1 = rbind(1, -1),
             x2 = rbind(1, 1), x3 = rbind(0, 1))
fit = vglm(y ~ -1 + X1 + x2 + x3,
           VGAMfamilyFunction, myframe,
           constraints = Hlist,
           xij = list(X1 ~ -1 + x1 + x2),
           form2 = ~ -1 + x1 + x2 + x3 + X1)
More complicated constraints† IV

Example 2

Q2: How can we fit

\[ \eta_1 = \beta_{(1)1} x_1 + \beta_{(1)2} x_2 \]  \hspace{1cm} (16)
\[ \eta_2 = \beta_{(2)1} x_1 + \beta_{(2)2} x_2 \]  \hspace{1cm} (17)
\[ \eta_3 = \beta_{(3)1} x_1 + \beta_{(3)2} x_2 \]  \hspace{1cm} (18)

subject to \( \beta_{(1)1} + \beta_{(2)1} + \beta_{(3)1} = \beta_{(1)2} + \beta_{(2)2} + \beta_{(3)2} \)?

A2: Now

\[ \eta_3 = \beta_{(3)1} x_1 + \left( \beta_{(1)1} + \beta_{(2)1} + \beta_{(3)1} - \beta_{(1)2} - \beta_{(2)2} \right) x_2. \]
More complicated constraints

Usually (trivial constraints)

\[ \eta = x_1 I_3 \begin{pmatrix} \beta_{(1)1} \\ \beta_{(2)1} \\ \beta_{(3)1} \end{pmatrix} + x_2 I_3 \begin{pmatrix} \beta_{(1)2} \\ \beta_{(2)2} \\ \beta_{(3)2} \end{pmatrix} \]

but here,

\[ \eta_3 = \beta_{(3)1} (x_1 + x_2) + \beta_{(1)1} x_2 + \beta_{(2)1} x_2 + \beta_{(1)2} (-x_2) + \beta_{(2)2} (-x_2). \]

So

\[ \eta = I_3 \begin{pmatrix} x_1 & 0 & 0 \\ 0 & x_1 & 0 \\ x_2 & x_2 & x_1 + x_2 \end{pmatrix} \begin{pmatrix} \beta_{(1)1} \\ \beta_{(2)1} \\ \beta_{(3)1} \end{pmatrix} + \begin{pmatrix} x_2 & 0 \\ 0 & x_2 \\ -x_2 & -x_2 \end{pmatrix} \begin{pmatrix} \beta_{(1)2} \\ \beta_{(2)2} \end{pmatrix}. \]
More complicated constraints† VI

The second term is easy:

\[ x_2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} \beta_{(1)2} \\ \beta_{(2)2} \end{pmatrix}. \]

The first term is problematic if dealt with wholly. Instead, we need to break it up by columns:

\[
\begin{pmatrix} x_1 \\ 0 \\ x_2 \end{pmatrix} \beta_{(1)1} + \begin{pmatrix} 0 \\ x_1 \\ x_2 \end{pmatrix} \beta_{(2)1} + \begin{pmatrix} 0 \\ 0 \\ x_1 + x_2 \end{pmatrix} \beta_{(3)1} =
\begin{pmatrix} x_1 & 0 & 0 \\ 0 & a_1 & 0 \\ 0 & 0 & x_2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \beta_{(1)1} + \begin{pmatrix} a_2 & 0 & 0 \\ 0 & x_1 & 0 \\ 0 & 0 & x_2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \beta_{(2)1} + \]
More complicated constraints† VII

\[
\begin{pmatrix}
    a_3 & 0 & 0 \\
    0 & a_4 & 0 \\
    0 & 0 & x_3
\end{pmatrix}
\begin{pmatrix}
    0 \\
    0 \\
    1
\end{pmatrix}
\beta_{(3)1}
\]

where \( x_3 = x_1 + x_2 \) and \( a_j \) has any value. Thus \( p = 4 \).
More complicated constraints† VIII

Try something like

cmat = matrix(c(1, 0, -1, 0, 1, -1), 3, 2)
myframe = transform(myframe, X1 = x1, X2 = x1, X3 = x1,
                    x3 = x1 + x2,
                    a1 = 0 * x1, a2 = 0 * x1,
                    a3 = 0 * x1, a4 = 0 * x1)

fit = vglm(y ~ X1 + X2 + X3 + x2 - 1,
           VGAMfamilyFunction, myframe,
           constraints = list(X1 = rbind(1, 0, 1),
                              X2 = rbind(0, 1, 1),
                              X3 = rbind(0, 0, 1),
                              x2 = cmat),
           form2 = ~ X1 + X2 + X3 - 1 +
                   x1 + x2 + x3 +
                   a1 + a2 + a3 + a4,
           xij = list(X1 ~ x1 + a1 + x2 - 1,
                      X2 ~ a2 + x1 + x2 - 1,
                      X3 ~ a3 + a4 + x3 - 1))
More on VGAMs† I

Let’s do two simultaneous logistic regressions: We will fit two species’ presence/absence versus $X_2 = \text{altitude}$. Data is from 392 sites from the Hunua forest.

- **agaaus** is *Agathis australis*, better known as “Kauri”.
- **kniexc** is *Knightia excelsa*, or “Rewarewa”.

```r
> # nrow(hunua)
> fit2 = vgam(cbind(agaaus, kniexc) ~ s(altitude, df = c(2, 3)),
            binomialff(mv = TRUE), hunua)
> round(coef(fit2, mat = TRUE), dig=6) # Not really interpretable

logit(E[agaaus]) logit(E[kniexc])
(Intercept)         -1.308694       -0.073857
s(altitude, df = c(2, 3)) 0.000122        0.002744

> #coef(fit2, mat = TRUE)  # Not really interpretable
> plot(fit2, se = TRUE, overlay = TRUE, lcol = c("blue", "orange"),
      scol = c("blue", "orange"))
```

The output to `coef()` is not really interpretable; they are the coefficients to the linear part of the fit.
Figure: Two nonparametric logistic regressions fitted as a VGAM. Blue is Kauri, orange is Rewarewa.
Now notice the O-splines here

```r
> fit2@Bspline

$s(altitude, df = c(2, 3))$
An object of class "vsmooth.spline.fit"
Slot "Bcoefficients":

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.545752005 -1.32505661</td>
</tr>
<tr>
<td>2</td>
<td>0.543526883 -1.31945612</td>
</tr>
<tr>
<td>3</td>
<td>0.532401279 -1.29145370</td>
</tr>
<tr>
<td>4</td>
<td>0.514600728 -1.24665158</td>
</tr>
<tr>
<td>5</td>
<td>0.483461950 -1.16830000</td>
</tr>
<tr>
<td>6</td>
<td>0.450130888 -1.08448848</td>
</tr>
<tr>
<td>7</td>
<td>0.401395611 -0.96221952</td>
</tr>
<tr>
<td>8</td>
<td>0.357345763 -0.85219250</td>
</tr>
<tr>
<td>9</td>
<td>0.302986231 -0.71737760</td>
</tr>
<tr>
<td>10</td>
<td>0.260222187 -0.61218805</td>
</tr>
<tr>
<td>11</td>
<td>0.197658830 -0.45944000</td>
</tr>
<tr>
<td>12</td>
<td>0.147647842 -0.33736200</td>
</tr>
<tr>
<td>13</td>
<td>0.094891851 -0.20811500</td>
</tr>
<tr>
<td>14</td>
<td>0.067812019 -0.14250131</td>
</tr>
<tr>
<td>15</td>
<td>0.033036408 -0.05701255</td>
</tr>
<tr>
<td>16</td>
<td>0.003421857  0.03312613</td>
</tr>
<tr>
<td>17</td>
<td>0.041013992  0.12713371</td>
</tr>
<tr>
<td>18</td>
<td>0.075331053  0.21431114</td>
</tr>
<tr>
<td>19</td>
<td>0.100125600  0.27843259</td>
</tr>
<tr>
<td>20</td>
<td>0.133206057  0.36579467</td>
</tr>
<tr>
<td>21</td>
<td>0.156166632  0.42873344</td>
</tr>
<tr>
<td>22</td>
<td>0.185364953  0.51403679</td>
</tr>
</tbody>
</table>
```
More on VGAMs† IV

\[
\begin{array}{cccc}
[23,] & 0.202522534 & 0.57150973 \\
[24,] & 0.213323031 & 0.62056162 \\
[25,] & 0.215241541 & 0.63970159 \\
[26,] & 0.213283985 & 0.65107630 \\
[27,] & 0.206914270 & 0.64577245 \\
[28,] & 0.190703158 & 0.61480542 \\
[29,] & 0.171845576 & 0.57357295 \\
[30,] & 0.147476587 & 0.52270725 \\
[31,] & 0.119320416 & 0.46710009 \\
[32,] & 0.084921160 & 0.40655642 \\
[33,] & 0.024948684 & 0.31190939 \\
[34,] & -0.044699535 & 0.21367149 \\
[35,] & -0.149124456 & 0.06944959 \\
[36,] & -0.234927976 & -0.05584417 \\
[37,] & -0.326203264 & -0.19816957 \\
[38,] & -0.405866064 & -0.32832379 \\
[39,] & -0.524187668 & -0.53272684 \\
[40,] & -0.668164989 & -0.79889328 \\
[41,] & -1.014853850 & -1.49298616 \\
[42,] & -1.458548841 & -2.62195642 \\
[43,] & -1.840213406 & -3.62007860 \\
[44,] & -1.967465889 & -3.95187756 \\
\end{array}
\]

Slot "knots":

\[
\begin{array}{cccc}
[1] & 0.000000000 & 0.000000000 & 0.000000000 & 0.000000000 \\
[5] & 0.001515152 & 0.007575758 & 0.012121212 & 0.022727273 \\
[9] & 0.030303030 & 0.045454545 & 0.053030303 & 0.068181818 \\
[13] & 0.075757576 & 0.098484848 & 0.106060606 & 0.118181818 \\
[17] & 0.121212121 & 0.136363636 & 0.151515152 & 0.159090909 \\
\end{array}
\]
More on VGAMs† V

<table>
<thead>
<tr>
<th></th>
<th>0.174242424</th>
<th>0.181818182</th>
<th>0.204545455</th>
<th>0.212121212</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>0.242424242</td>
<td>0.257575758</td>
<td>0.280303030</td>
<td>0.287878788</td>
</tr>
<tr>
<td>25</td>
<td>0.318181818</td>
<td>0.333333333</td>
<td>0.363636364</td>
<td>0.378787879</td>
</tr>
<tr>
<td>29</td>
<td>0.393939394</td>
<td>0.424242424</td>
<td>0.439393939</td>
<td>0.484848485</td>
</tr>
<tr>
<td>33</td>
<td>0.515151515</td>
<td>0.560606061</td>
<td>0.575757576</td>
<td>0.606060606</td>
</tr>
<tr>
<td>37</td>
<td>0.636363636</td>
<td>0.681818182</td>
<td>0.727272727</td>
<td>0.909090909</td>
</tr>
<tr>
<td>41</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
</tr>
</tbody>
</table>

Slot "xmin":
```r
s(altitude, df = c(2, 3))
```

Slot "xmax":
```r
c(altitude, df = c(2, 3), 660)
```
More on VGAMs† VI

> ooo = with(hunua, order(altitude))
> with(hunua, matplot(altitude[ooo], fitted(fit2)[ooo,], ylim = c(0, .8),
> xlab = "Altitude (m)", ylab = "Probability of presence", las = 1,
> col = c("blue", "orange"),
> main = "Two plant species' response curves", type = "l", lwd = 2))
> with(hunua, rug(altitude))

Figure: Two nonparametric logistic regressions fitted as a VGAM.
Working weights† I

Each $W_i$ 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,

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

Often the score vector is easy. Use random variates to compute the sample variance of the score vector. The \texttt{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

\[
I_{k,k} = k^4 \left( \sum_{j=0}^{\infty} (k + j)^{-2} P(Y_i \geq j) - \frac{\mu_i/k}{k + \mu_i} \right). \tag{19}
\]

The weight slot of `negbinomial()`:
Working weights† III

\[
\text{weight} = \text{eval(substitute(expression({}
\text{wz} = \text{matrix(as.numeric(NA), n, M)} \quad \# \text{wz is 'diagonal'}
\text{run.varcov} = \text{matrix(0, n, NOS)}
\text{ind1} = \text{iam(NA, NA, M = M, both = TRUE, diag = TRUE)}
\text{for(ii in 1:(.nsimEIM))} \{ 
\text{ysim} = \text{rnbinom(n = n*NOS, mu = c(mu), size = c(kmat))}
\text{dl.dk} = \text{digamma(ysim+kmat) - digamma(kmat)} -
\quad (ysim+kmat)/(mu+kmat) + 1 + \log(kmat/(kmat+mu))
\text{run.varcov} = \text{run.varcov} + dl.dk^2
\}
\text{run.varcov} = \text{cbind(run.varcov} / .nsimEIM)
\quad \# \text{Can do even better if it is an intercept-only model}
\text{wz[,2*(1:NOS)]} = \text{if(intercept.only)}
\quad \text{matrix(colMeans(run.varcov),}
\quad \text{n, ncol(run.varcov), byrow = TRUE)} \quad \text{else run.varcov}
\text{wz[,2*(1:NOS)]} = wz[,2*(1:NOS)] * dk.deta^2
\quad \# \text{The 1-1 element (known exactly)}:
\text{ed2l.dmu2} = 1/mu - 1/(mu+kmat)
\text{wz[,2*(1:NOS)-1]} = dmu.deta^2 * ed2l.dmu2
\text{w} \times \text{wz}
}}, \text{list(.cutoff = cutoff, .Maxiter = Maxiter, .nsimEIM = nsimEIM ))}
\]
Recall $\eta = B^T x$ for VGLMs.

**Motivation:** if $M$ and $p$ are large then $B$ is “too big” ($Mp$ elements).

**Idea:** approximate part of $B$ by a lower rank matrix, i.e, the product of two ‘thin’ matrices $A C^T$.

Great for dimension reduction! Can do biplots etc.

**A simple and important result**

Solving by an alternating algorithm shows that RR-VGLMs are VGLMs where the constraint matrices are unknown and to be estimated.
RR-VGLMs II

Some special cases

1. A reduced-rank multinomial logit model is a stereotype model (Anderson, 1984).

2. A reduced-rank negative binomial distribution has variance function

\[ \text{Var}(Y) = \mu + \delta_1 \mu^{\delta_2} \]

for parameters \( \delta_1 \) and \( \delta_2 \).

```
rrvglm(y ~ x2 + \cdots + xp, negbinomial(zero = NULL))
```


```
rrvglm(y ~ x2 + \cdots + xp, zipoisson(zero = NULL))
```
**Example: Hunting spider data**

Data consists of abundances (numbers trapped over a 60 week period) of \( S = 12 \) species of hunting spiders in a Dutch dune area. Have \( n = 28 \) and \( p_2 = 6 \).

**Table:** Hunting spiders data. Log transformed environmental variables.

<table>
<thead>
<tr>
<th>R variable</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WaterCon</td>
<td>Water Content</td>
<td>Percentage of soil dry mass</td>
</tr>
<tr>
<td>BareSand</td>
<td>Bare Sand</td>
<td>Percentage cover of bare sand</td>
</tr>
<tr>
<td>FallTwig</td>
<td>Fallen Twigs</td>
<td>Percentage cover of fallen leaves and twigs</td>
</tr>
<tr>
<td>CoveMoss</td>
<td>Cover Moss</td>
<td>Percentage cover of the moss layer</td>
</tr>
<tr>
<td>CoveHerb</td>
<td>Cover Herbs</td>
<td>Percentage cover of the herb layer</td>
</tr>
<tr>
<td>ReflLux</td>
<td>Light Refl</td>
<td>Reflection of the soil surface with cloudless sky</td>
</tr>
</tbody>
</table>
**Table:** Hunting spiders data. Species’ abbreviations.

<table>
<thead>
<tr>
<th>R variable</th>
<th>Species name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alopacce</td>
<td>Alopecosa accentuata</td>
</tr>
<tr>
<td>Alopcune</td>
<td>Alopecosa cuneata</td>
</tr>
<tr>
<td>Alopfabr</td>
<td>Alopecosa fabrilis</td>
</tr>
<tr>
<td>Arctlute</td>
<td>Arctosa lutetiana</td>
</tr>
<tr>
<td>Arctperi</td>
<td>Arctosa perita</td>
</tr>
<tr>
<td>Auloalbi</td>
<td>Aulonia albimana</td>
</tr>
<tr>
<td>Pardlugu</td>
<td>Pardosa lugubris</td>
</tr>
<tr>
<td>Pardmont</td>
<td>Pardosa monticola</td>
</tr>
<tr>
<td>Pardnigr</td>
<td>Pardosa nigriceps</td>
</tr>
<tr>
<td>Pardpull</td>
<td>Pardosa pullata</td>
</tr>
<tr>
<td>Trocterr</td>
<td>Trochosa terricola</td>
</tr>
<tr>
<td>Zoraspin</td>
<td>Zora spinimana</td>
</tr>
</tbody>
</table>
Constrained Quadratic Ordination (CQO)† III

VGAM has the data in a data frame called hspider. A rank-1 Poisson CQO with equal tolerances can be obtained from

```r
> data(hspider)
> set.seed(111)
> hspider[,1:6] = scale(hspider[,1:6]) # Standardize environ vars
> p1et = cqo(cbind(Alopacce, Alopucne, Alopfabr, Arctlute,
> Arctperi, Auloalbi, Pardlugu, Pardmont,
> Pardnigr, Pardpull, Trocterr, Zoraspin) ~
> WaterCon + BareSand + FallTwig +
> CoveMoss + CoveHerb + ReflLux,
> family = quasipoissonff, data = hspider,
> Crow1pos =FALSE,
> # EqualTolerances=FALSE, ITolerances=FALSE, trace=FALSE)
> persp(p1et,
> main="Hunting spider data",
> col=1:ncol(p1et@y), llwd=2, las=1, llty=1)
```
Hunting spider data

Figure: Rank-1 Poisson QRR-VGLM with equal tolerances.
Constrained Quadratic Ordination (CQO)† V

> summary(p1et)

Call:
cqo(formula = cbind(Alopacce, Alopcline, Alopfabr, Arctlute, Arctperi,
Auloalbi, Pardlugu, Pardmont, Pardnigr, Pardpull, Trocterr,
Zoraspin) ~ WaterCon + BareSand + FallTwig + CoveMoss + CoveHerb +
ReflLux, family = quasipoissonff, data = hspider, Crow1pos = FALSE,
trace = FALSE)

C matrix (constrained/canonical coefficients)

   lv
WaterCon -0.3572911
BareSand  0.5534843
FallTwig -0.9204249
CoveMoss  0.3170025
CoveHerb -0.3055749
ReflLux   0.7022278

B1 and A matrices
Constrained Quadratic Ordination (CQO)† VI

<table>
<thead>
<tr>
<th></th>
<th>(Intercept)</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>log(E[Alopecce])</td>
<td>1.0946607</td>
<td>1.9879733</td>
</tr>
<tr>
<td>log(E[Alopcune])</td>
<td>2.8465611</td>
<td>-0.4417111</td>
</tr>
<tr>
<td>log(E[Alopfabr])</td>
<td>-2.2453933</td>
<td>3.1040488</td>
</tr>
<tr>
<td>log(E[Arctlute])</td>
<td>0.9216726</td>
<td>-0.7216428</td>
</tr>
<tr>
<td>log(E[Arctperi])</td>
<td>-8.6858178</td>
<td>4.8349970</td>
</tr>
<tr>
<td>log(E[Auloalbi])</td>
<td>2.5729096</td>
<td>-0.6633768</td>
</tr>
<tr>
<td>log(E[Pardlugu])</td>
<td>-0.6767298</td>
<td>-2.5515725</td>
</tr>
<tr>
<td>log(E[Pardmont])</td>
<td>3.5830119</td>
<td>0.8916557</td>
</tr>
<tr>
<td>log(E[Pardnigr])</td>
<td>3.7606464</td>
<td>-0.5870566</td>
</tr>
<tr>
<td>log(E[Pardpull])</td>
<td>4.2098964</td>
<td>-0.4076585</td>
</tr>
<tr>
<td>log(E[Trocterr])</td>
<td>4.4439428</td>
<td>-0.8412503</td>
</tr>
<tr>
<td>log(E[Zoraspin])</td>
<td>2.7697050</td>
<td>-0.8593639</td>
</tr>
</tbody>
</table>

Optima and maxima

<table>
<thead>
<tr>
<th></th>
<th>Optimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alopecce</td>
<td>1.9879733</td>
<td>21.556547</td>
</tr>
<tr>
<td>Alopcune</td>
<td>-0.4417111</td>
<td>18.993855</td>
</tr>
<tr>
<td>Alopfabr</td>
<td>3.1040488</td>
<td>13.094160</td>
</tr>
</tbody>
</table>
### Constrained Quadratic Ordination (CQO)$

<table>
<thead>
<tr>
<th>Species</th>
<th>Quadratic Score</th>
<th>Term Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arctlute</td>
<td>-0.7216428</td>
<td>3.261074</td>
</tr>
<tr>
<td>Arctperi</td>
<td>4.8349970</td>
<td>20.141453</td>
</tr>
<tr>
<td>Auloalbi</td>
<td>-0.6633768</td>
<td>16.329021</td>
</tr>
<tr>
<td>Pardlugu</td>
<td>-2.5515725</td>
<td>13.177771</td>
</tr>
<tr>
<td>Pardmont</td>
<td>0.8916557</td>
<td>53.545775</td>
</tr>
<tr>
<td>Pardnigr</td>
<td>-0.5870566</td>
<td>51.058094</td>
</tr>
<tr>
<td>Pardpull</td>
<td>-0.4076585</td>
<td>73.184902</td>
</tr>
<tr>
<td>Trocterr</td>
<td>-0.8412503</td>
<td>121.242640</td>
</tr>
<tr>
<td>Zoraspin</td>
<td>-0.8593639</td>
<td>23.079810</td>
</tr>
</tbody>
</table>

### Tolerance

<table>
<thead>
<tr>
<th>Species</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alopacce</td>
<td>1</td>
</tr>
<tr>
<td>Alopcune</td>
<td>1</td>
</tr>
<tr>
<td>Alopfabr</td>
<td>1</td>
</tr>
<tr>
<td>Arctlute</td>
<td>1</td>
</tr>
<tr>
<td>Arctperi</td>
<td>1</td>
</tr>
<tr>
<td>Auloalbi</td>
<td>1</td>
</tr>
<tr>
<td>Pardlugu</td>
<td>1</td>
</tr>
</tbody>
</table>
Constrained Quadratic Ordination (CQO)† VIII

Pardmont 1
Pardnigr 1
Pardpull 1
Trocterr 1
Zoraspin 1

Standard deviation of the latent variables (site scores)

\[ \text{lv} \]

2.371051

Dispersion parameters:

\begin{align*}
\text{Alopacce} & & \text{Alopcune} & & \text{Alopfabr} & & \text{Arctlute} \\
3.486247e+00 & & 7.725849e+00 & & 4.306666e+00 & & 1.953142e+00 \\
\text{Arctperi} & & \text{Auloalbi} & & \text{Pardlugu} & & \text{Pardmont} \\
8.560377e-01 & & 4.770231e+00 & & 1.356310e+05 & & 1.569950e+01 \\
\text{Pardnigr} & & \text{Pardpull} & & \text{Trocterr} & & \text{Zoraspin} \\
1.235828e+01 & & 6.590131e+00 & & 4.100787e+01 & & 3.089348e+00
\end{align*}
Concluding remarks

1. VGLMs and VGAMs fit a very large class of models. VGLMs are model-driven while VGAMs are data-driven.

2. The framework is purposely general. More general $\implies$ more useful.

3. VGAM is freely available on CRAN or at the author’s web page.
Tēnā koutou katoa
谢谢！
Xiè xie nǐ
That’s all folks!