VGAM Reference Card

by T. W. Yee, t.yee@auckland.ac.nz, 2011-11-04
See http://www.stat.auckland.ac.nz/~yee/VGAM for the VGAM package and some other documentation. This document is current for version VGAM-0.8-4.

The theory[1] underlying the software can be found in Yee and Wild (1996), Yee and Mackenzie (2002), Yee and Hastie (2003), Yee (2004a), Yee (2004b), Yee (2006), Yee and Stephenson (2007), Yee (2008), Yee (2011a), Yee and Dirnböck (2009), Yee (2010a), Yee (2010b), Yee (2011b). These references contain additional examples etc. Please cite the appropriate references if you use the software!

DISCLAIMER: VGAM is still in the development stage, meaning that new features are still being added and bugs are still being found on a regular basis. Function and argument names may change at any time, as well as default values of arguments; see the NEWS file. The VGAM package is available on a use-at-your-own-risk basis: the Author assumes no liability for loss or damage of any kind resulting from its use.

Modelling functions

The following functions form the heart of the VGAM package and use `formula` and `family` arguments, e.g., **`vglm(y ~ 1, family = maxwell)`**.

`vglm()` Vector generalized linear models.
`vgam()` Vector generalized additive models.
`rrvglm()` Reduced-rank vector generalized linear models (same as constrained linear ordination, or CLO).
`cqo()` Constrained quadratic (Gaussian) ordination (QRR-VGLM).
`cao()` Constrained additive ordination (RR-VGAM).
`rcam(table or matrix)` Row-column association models (developmental stage only).

The following functions do *not* use the **`formula, family =`** arguments.

`grc(table or matrix)` Goodman's RC model.
`vsmooth.spline(x, y, w, ...)` Vector smoothing spline.

Useful controlling options for the modelling functions

`trace = TRUE` Print a running log of the estimation.
`criterion = "coef"` Criterion on which to test convergence.
`maxit = 40` Maximum number of iterations allowed.
`coefstart`, **`etastart`**, **`mustart`** Starting values for β, η_i ($n \times M$ matrix), μ_i respectively. For example, **`vglm(y ~ x2, maxwell, mydata, etastart = predict(simplerModel))`**.

The following arguments are standard to formula-based modelling functions.

`data =` Data frame with the formula variables.
`subset =` Vector of logicals.
`na.action =` What to do with missing values. `"na.fail"` causes an error, `"na.omit"` deletes rows. Can be assigned a user-defined function.

Extractor functions

`class()` The object's class.
`coef()` Regression coefficients (the β_k^* in (1) but enumerated in a different order).
`Coef()` Regression coefficients, especially if the formula comprises of intercept only, i.e., **`~ 1`**.
`constraints()` Constraint matrices, $\mathbf{H}_k$.
`deviance()` deviance, D.
`df.residual()` Residual degrees of freedom
`fitted()` Fitted values, usually $\widehat{\mu}_i$.
`logLik()` log-likelihood, ℓ.
`model.matrix()` The big model matrix $\mathbf{X}$. This is never smaller than the **`lm`**-type model matrix.
`predict()` Prediction of μ or η from x.
`print()` Print.
`resid()` Residuals (working, Pearson, deviance, response).
`summary()` Summary.
`terms()` Terms.
`vcov()` Variance-covariance matrix, $\widehat{\mathrm{Var}}(\widehat{\beta})$.
`weights()` Prior weights w_i, and working weights $(w_i \mathbf{W}_i)$ in matrix-band format.

Plotting functions

Not all of the following apply to a given fitted model.
`biplot()` Biplot for RR-VGLMs.
`deplot()` Density plot, e.g., for quantile regression.
`lvplot()` Latent variable plot (ordination diagram).
`persp()` Perspective (3-D; sometimes 2-D) plot.
`plot()` General plotting function.
`guplot()` Gumbel plot, e.g., for extreme values regression.
`meplot()` Mean excess plot, e.g., for extreme values regression.
`qtplot()` Quantile plot, e.g., for quantile regression.
`rlplot()` Return level plot, e.g., for extreme values regression.
`trplot()` Trajectory plot, e.g., for constrained ordination.

Link functions

Most parameters θ_j are transformed into a linear/additive predictor $\eta_j = \beta_j^T x$ or $\eta_j = \sum_{k=1}^{p} f_{(j)k}(x_k)$. All logarithms are to base e unless specified otherwise.
Usage: use full link function name (in quotes is ok), e.g., **`vglm(y ~ 1, family = maxwell(link = "identity"))`**.

`loge()` Log, $\log(\theta)$, $\theta > 0$.
`logoff()` Log with an offset, $\log(\theta + A)$, $\theta + A > 0$.
`nlogoff()` Negative log with an offset, $\log(-\theta + A)$, $-\theta + A > 0$.
`loglog()` Log-log, $\log\log(\theta)$, $\theta > 1$.
`cloglog` Complementary log-log, $\log(-\log(1-\theta))$, $0 < \theta < 1$.
`golf()` Gamma-ordinal link function, $0 < \theta < 1$.
`polf()` Poisson-ordinal link function, $0 < \theta < 1$.
`nbolf()` Negative binomial-ordinal link function, $0 < \theta < 1$.
`logc()` Complementary log, $\log(1-\theta)$, $\theta < 1$.
`logit()` Logit, $\log \frac{\theta}{1-\theta}$, $0 < \theta < 1$.
`elogit()` Extended logit, $\log \frac{\theta - A}{B - \theta}$, $A < \theta < B$.
`fsqrt()` Folded square root link function, $A < \theta < B$.
`probit()` Probit, $\Phi^{-1}(\theta)$, $0 < \theta < 1$.

[1] A small resume is given below.

cauchit() Cauchit, $\tan(\pi(\theta - \frac{1}{2}))$, $0 < \theta < 1$.
identity() Identity, θ.
nidentity() Negative-identity, $-\theta$.
inverse() Inverse, $1/\theta$, $\theta \neq 0$.
fisherz() Fisher's Z-transformation, $\frac{1}{2}\log\frac{1+\theta}{1-\theta}$, $-1 < \theta < 1$.
rhobit() Twice Fisher's Z-transformation, $\log\frac{1+\theta}{1-\theta}$.

Utility functions

theta2eta() $\eta_j = g(\theta_j)$.
eta2theta() $\theta_j = g^{-1}(\eta_j)$.
m2adefault() Conversion from weight matrices (matrix-band) format to array format.

Common arguments in family functions

`zero` allows for $\eta_j = \beta_{(j)1}$, i.e., intercept-only. Can be assigned a vector with values from the set $\{1, 2, \ldots, M\}$. Negative values allowed for multiple responses. A `NULL` means *all* the η_j are modelled as functions of the covariates.

`exchangeable` For some s and t, $\eta_s - \eta_t = 0$ or some constant. That is, η_s and η_t differ at most by an intercept.

`parallel` If `TRUE`, for all k, $\beta_{(s)k} = \beta_{(t)k}$ for some s and t; i.e., the slope for X_k of some of the linear/additive predictors are parallel or differ by a constant.

`parallel = FALSE ~ X1 + X2 - 1` means all terms, except for X_1 and X_2, have parallel slopes.

Next is a listing of VGAM family functions. These are assigned to the **family =** argument to the modelling functions, e.g., **vglm(y ~ x2 + x3, family = multinomial)**.

Generalized linear models

binomialff() Binomial. For multivariate responses use **binomialff(mv = TRUE)**.
quasibinomialff() Binomial with dispersion parameter to be estimated.
gaussianff() Gaussian or normal.
Gamma() (same as **gammaff()**) Gamma.
inverse.gaussianff() Inverse Gaussian.
poissonff() Poisson.
quasipoissonff() Poisson with dispersion parameter to be estimated.
quasiff() Quasi- family (not working yet).
dexpbinomial() Double exponential binomial distribution.

Nonlinear regression models

micmen() Michaelis-Menten model, $\mu_i = \theta_1 x_i/(\theta_2 + x_i)$.

Categorical data

In the following, g is a link function, η_j the linear/additive predictors, and $Y \in \{1, \ldots, M+1\}$ is a categorical response.
acat() Adjacent categories model, $\eta_j = g(P[Y = j+1]/P[Y = j])$.
cumulative() Cumulative categories model, $\eta_j = g(P[Y \leq j])$. Includes the proportional odds model. Derivatives available through **margeff()**.
propodds() Proportional odds model $\eta_j = \text{logit}(P[Y > j])$.
cratio() Continuation ratio model, $\eta_j = g(P[Y > j | Y \geq j])$.
sratio() Stopping ratio model $\eta_j = g(P[Y = j | Y \geq j])$.
multinomial() Multinomial logit model, $\eta_j = \log(P[Y = j]/P[Y = M+1])$. Derivatives available through **margeff()**.
brat() Bradley Terry model (without ties).
bratt() Bradley Terry model (with ties).
ordpoisson() Ordinal Poisson model.
The argument `reverse` reverses the direction of many of the above probabilities.
See documentation on the `xij` argument at the package's website: this handles covariates that have different values for differing linear/additive predictors η_j. This will handle consumer choice or discrete choice models.

Bivariate responses

binom2.or(dr) Bivariate logistic/probit/...odds ratio model for two binary responses.
bilogistic4(dpr) 4-parameter bivariate logistic distribution.
binom2.rho(dr) Bivariate probit model for two binary responses. Based on a standard N_2 with correlation parameter ρ.
loglinb2()/loglinb3() Loglinear model for two/three binary responses.
amh(dpr) Ali-Mikhail-Haq's bivariate distribution.
binormal(d) Bivariate normal distribution, $N_2(\mu_1, \mu_2, \sigma_{11}, \sigma_{22}, \rho)$.
fgm(dpr) Farlie-Gumbel-Morgenstern's bivariate distribution
freund61() Freund's (1961) bivariate extension of the exponential distribution.
frank(dpr) Frank's bivariate distribution.
gammahyp() 1-parameter gamma hyperbola bivariate distribution.
bivgamma.mckay() McKay's bivariate gamma distribution.
morgenstern() Morgenstern's bivariate distribution
gumbelIbiv() Gumbel's Type I bivariate distribution
plackett(dpr) Plackett's bivariate distribution.

Extreme value data

gev() 3-parameter generalized extreme value distribution.
egev() 3-parameter generalized extreme value distribution (univariate response).
gpd() 2-parameter generalized Pareto distribution.
egumbel() 2-parameter Gumbel distribution (univariate response).
gumbel() 2-parameter Gumbel distribution for multivariate responses.
recnormal1() Records: univariate normal data.
recexp1() Records: univariate exponential data.
See also **guplot()**, **meplot()**, **rlplot()**.

Quantile and expectile regression

(I) LMS methods
lms.bcn() Box-Cox transformation to normality.
lms.bcg() Box-Cox transformation to the gamma distribution.
lms.yjn() Yeo-Johnson transformation to normality.

Special methods functions for these models are:
qtplot() Quantile plot.
deplot() Density plot.
cdf() Cumulative distribution function.

(II) Asymmetric Laplace distribution (ALD) methods
alaplace1() 1-parameter ALD.
alaplace2() 2-parameter ALD.
alaplace3() 3-parameter ALD.

(III) Asymmetric Maximum Likelihood (AML) estimation methods

amlnormal() Asymmetric least squares (expectile) regression (Efron, 1991).
amlbinomial() Logistic (expectile) regression.
amlexponential() Exponential (expectile) regression.
amlpoisson() Poisson (expectile) regression (Efron, 1992).

Genetical data

AA.Aa.aa() AA-Aa-aa blood group system.
AB.Ab.aB.ab2() AB-Ab-aB-ab2 blood group system.
AB.Ab.aB.ab() AB-Ab-aB-ab blood group system.
ABO() ABO blood group system.
G1G2G3() G1G2G3 blood group system.
MNSs() MNSs blood group system.

Positive and zero-inflated or zero-altered distributions

Here, the probability that the response is zero is zero.
posbinomial(dpqr) Positive binomial distribution.
posnormal1(dpqr) Positive normal distribution (univariate).
posnegbinomial(dpqr) Positive negative binomial distribution.
pospoisson(dpqr) Positive Poisson distribution.
zanegbinomial(dpqr) Zero-altered negative binomial distribution.
zapoisson(dpqr) Zero-altered Poisson distribution.
yip88() Zero-inflated Poisson distribution.
zibinomial(dpqr) Zero-inflated binomial distribution.
zigeometric(dpqr) Zero-inflated geometric distribution.
zinegbinomial(dpqr) Zero-inflated negative binomial distribution.
zipoissonff(dpqr), **zipoisson(dpqr)** Zero-inflated Poisson distribution.
zipebcom() Exchangeable zero-inflated Poisson distribution and **binom2.or()**.

Univariate distributions

Additionally, some of these distributions have random variates generation, density functions, cumulative distribution functions and quantile functions associated with them. For example, **rsinmad()**, **dlomax()**, **pdagum()**, **qfisk()**. Those distribution with **d**, **p**, **q**, **r** functions have characters inside their arguments, e.g., **sinmad(r)**.

alaplace[123](dpqr) Asymmetric Laplace distribution.
[dpqr]benf() 0-parameter Benford distribution.
benini() 1-parameter Benini distribution.
betabinomial(dpr) Beta-binomial distribution.
betabinomial.ab(dpr) Beta-binomial distribution.
beta.ab() 2-parameter beta distribution.
betaff() 2-parameter beta distribution.
[dpr]betageometric() Beta-geometric distribution.
betaprime() 2-parameter beta-prime distribution.
betaII() 3-parameter beta II distribution.
bisa(dpqr) Birnbaum-Saunders distribution.
borel.tanner(dr) Borel-Tanner distribution.
cardioid(dpqr) Cardioid distribution.
cauchy() 2-parameter Cauchy distribution.
cauchy1() 1-parameter Cauchy distribution.
cexpon() Censored exponential distribution.
chisq() Chi-squared distribution.
cgumbel() Censored Gumbel distribution.
cennormal1() Censored univariate normal distribution.
cenrayleigh() Censored Rayleigh distribution.
dagum(dpqr) 3-parameter Dagum distribution.
dcennormal1() Double censored 2-parameter normal distribution.
dexpbinomial() Double exponential binomial distribution.
erlang() Erlang distribution.
expexp() Exponentiated exponential distribution.
expexp1() Exponentiated exponential distribution.
explogarithmic() Exponential logarithmic distribution.
exponential() Exponential distribution.
fff() F-distribution.
fisk(dpqr) 2-parameter Fisk distribution.
fnormal1(dpqr) Folded normal distribution (univariate and generalized).
frechet[2](dpqr) 2-parameter Fréchet distribution.
gamma1() 1-parameter gamma distribution.
gamma2(), **gamma2.ab()** 2-parameter gamma distribution.
genbetaII() 4-parameter generalized beta II distribution.
gengamma(dpqr) Generalized gamma distribution.
genpoisson() Generalized Poisson distribution.
geometric() Geometric distribution.
gumbel() 2-parameter Gumbel distribution.
hypersecant(), **hypersecant.1()** Hyperbolic secant distribution.
hzeta(dpqr) Haight's Zeta Function.
inv.gaussianff() 2-parameter inverse Gaussian distribution.
invlomax(dpqr) 2-parameter inverse Lomax distribution.
invparalogistic(dpqr) 2-parameter inverse paralogistic distribution.
koenker(dpqr) Koenker's distribution.
kumar(dpqr) Kumaraswamy distribution.
laplace(dpqr) Laplace distribution.
leipnik() Leipnik distribution.
levy() Levy distribution.
lgammaff() Log-gamma distribution.
lino(dpqr) 3-parameter generalized beta distribution (Libby and Novick, 1982).
logff() Logarithmic distribution.
logistic1() 1-parameter logistic distribution.
logistic2() 2-parameter logistic distribution.
lognormal3() 3-parameter lognormal distribution.
lognormal() 2-parameter lognormal distribution.
lomax(dpqr) Lomax distribution.
mbinomial() Matched case-control study binomial distribution.
maxwell(dpqr) Maxwell distribution.
mccullagh89() McCullagh (1989) distribution.
nakagami(dpqr) Nakagami distribution.
negbinomial() Negative binomial distribution with parameters μ and k.
normal1() 2-parameter univariate normal distribution.
paralogistic(dpqr) 2-parameter paralogistic distribution.
pareto1(dpqr) 1-parameter Pareto distribution (Pareto(I)).
paretoIV(dpqr) Pareto(IV) distribution.
paretoIII(dpqr) Pareto(III) distribution.
paretoII(dpqr) Pareto(II) distribution.
poissonp() Distances to a fixed point, in a Poisson plane or volume.
[dpr]polono() Poisson-lognormal distribution.
polya() Pólya (negative binomial) distribution with parameters p and k.

rayleigh(dpqr) Rayleigh distribution.
riceff(dr) Rice distribution.
rig() Reciprocal inverse Gaussian distribution.
seq2binomial() 2-stage sequential binomial distribution.
simplex(dr) 2-parameter simplex distribution.
sinmad(dpqr) 3-parameter Singh-Maddala distribution.
skellam(dr) Skellam distribution.
skewnormal1(dr) 1-parameter skew-normal distribution.
studentt[23]() Student t distribution.
tikuv(dpqr) Short-tailed symmetric distribution of Tiku and Vaughan (1999).
tobit() Tobit model.
triangle(dpqr) Triangle distribution.
vonmises() von Mises distribution.
wald() Standard Wald distribution.
weibull() 2-parameter Weibull distribution.
yulesimon(dpr) Yule-Simon distribution.
zetaff() Zeta distribution.
zipf(dp) Zipf distribution.

Finite mixture models

mix2normal1() Two univariate normals.
mix2poisson() Two Poisson distributions.

Miscellaneous models and distributions

DeLury() De Lury's model for fish depletion analysis.
rrar() Reduced-rank autoregressive model for multiple time series.
dirichlet() Dirichlet distribution.
dirmultinomial(), dirmul.old() Dirichlet-multinomial distribution.
huber(dpqr) Huber's robust regression method.
huggins91(dpqr) Huggins (1991) capture-recapture model (not working properly).

Miscellaneous mathematical functions

erf() Error function.
erfc() Complementary error function.
lerch() Lerch's $\Phi(x,s,v)$ function.
zeta() Riemann's $\zeta(x)$ zeta function.

Quadratic and additive ordination

cqo() Canonical quadratic (Gaussian) ordination (QRR-VGLM).
cao() Constrained additive ordination (RR-VGAM). Not fully finished yet.
The fast algorithm currently works for families **poissonff()**, **gaussianff()**, **negbinomial()**, **gamma2()**, and **binomialff()** (`logit` and `cloglog` links available).

Special methods functions for these models are:

Coef() $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}_1$, $\widehat{\mathbf{C}}$, $\widehat{\mathbf{D}}$, $\widehat{u}_s$, $\widehat{\mathbf{T}}_s$, $\widehat{\mathrm{v}}_i$, etc.
ccoef() Canonical coefficients $\widehat{\mathbf{C}}$
lv() Latent variables $\widehat{\mathrm{v}}_i = \widehat{\mathbf{C}}^T x_{2i}$ (site scores)
Max() Maxima $E[Y_s|\widehat{u}_s] = g^{-1}(\widehat{\alpha}_s)$
Opt() Optima $\widehat{u}_s$ (species scores)
Tol() Tolerances $\widehat{\mathbf{T}}_s$
lvplot() Latent variable plot (ordination diagram; for rank $R = 1$ or 2)
persp() Perspective (3-D; sometimes 2-D) plot
calibrate() Calibration: estimate ν from y
trplot() Trajectory plot (for $R = 1$ only)

Miscellaneous

methods(class = class(fit)) lists all the methods to handle objects of class `fit`.
slotNames(fit) lists the slots of the object `fit`, but it is best to use extractor functions where possible.

Some formulae

For most VGLMs the log-likelihood $\ell = \sum_{i=1}^n w_i \ell_i(\eta_1, \ldots, \eta_M)$ is maximized, where $\eta_j = \beta_j^T x$. The prior weights w_i are inputted using **vglm(..., weights =...)**. For VGLMs

$$\eta(x) \;=\; \mathbf{H}_1 \beta_1^* x_1 + \cdots + \mathbf{H}_p \beta_p^* x_p \;=\; \mathbf{B}^T x \qquad (1)$$

where $\mathbf{H}_1, \ldots, \mathbf{H}_p$ are *known* full-column rank *constraint matrices*, and β_k^* is a vector containing a possibly reduced set of unknown regression coefficients. With no constraints at all, $\mathbf{H}_k = \mathbf{I}_M$ for all k. Usually $x_1 = 1$ (intercept term). In general,

$$\mathbf{B}^T \;=\; \left(\mathbf{H}_1 \beta_1^* \;\cdots\; \mathbf{H}_p \beta_p^*\right). \qquad (2)$$

Then **coef(fit, matrix = TRUE)** is the estimate of $\mathbf{B}$, and **constraints(fit)** are the $\mathbf{H}_k$. The $\mathbf{H}_k$ can be inputted with **vglm(..., constraints = list("(Intercept)" = ..., x2 = ...))**, or with arguments such as `parallel`, `exchangeable` and `zero` in the VGAM family function itself.

For VGAMs, (1) extends to

$$\eta(x) \;=\; \mathbf{H}_1 \beta_1^* x_1 + \mathbf{H}_2 f_2^*(x_2) + \cdots + \mathbf{H}_p f_p^*(x_p) \qquad (3)$$

where $f_k^*(x_k) = (f_{(1)k}(x_k), \ldots, f_{(r_k)k}(x_k))^T$ is a r_k-vector of smooth functions of x_k (estimated by a vector smoothing spline). With no constraints, $\eta_j = \sum_{k=1}^p f_{(j)k}(x_k)$.

For RR-VGLMs,

$$\eta(x) \;=\; \mathbf{B}_1^T x_1 + \mathbf{A}\nu \qquad (4)$$

where $x = (x_1^T, x_2^T)^T$, $\nu = \mathbf{C}^T x_2$ is a vector of latent variables, $\mathbf{A}$ is $M \times R$ and $\mathbf{C}$ is $p_2 \times R$. Here, $\mathbf{A}$ and $\mathbf{C}$ are estimated, and $\mathbf{B} = (\mathbf{B}_1^T \; \mathbf{B}_2^T)^T$ with $\mathbf{B}_2 = \mathbf{C}\mathbf{A}^T$, a reduced-rank approximation of a subset of $\mathbf{B}$ (cf. (1)). The *rank* R is often 1 or 2, maybe 3 ....

For QRR-VGLMs,

$$\begin{aligned} \eta(x) \;&=\; \mathbf{B}_1^T x_1 + \mathbf{A}\nu + \sum_{j=1}^M (\nu^T \mathbf{D}_j \nu) e_j \qquad (5) \\ &=\; \mathbf{B}_1^T x_1 + \mathbf{A}\nu + \begin{pmatrix} \nu^T \mathbf{D}_1 \nu \\ \vdots \\ \nu^T \mathbf{D}_M \nu \end{pmatrix}, \end{aligned}$$

where e_i is a vector of zeros but with a one in the ith position, and $\mathbf{D}_j$ are $R \times R$ symmetric matrices. Then $\mathbf{T}_j = -\frac{1}{2}\mathbf{D}_j^{-1}$ are *tolerance matrices*.

References

Yee, T. W., 2004a. A new technique for maximum-likelihood canonical Gaussian ordination. Ecol. Monogr. 74 (4), 685–701.

Yee, T. W., 2004b. Quantile regression via vector generalized additive models. Stat. Med. 23 (14), 2295–2315.

Yee, T. W., 2006. Constrained additive ordination. Ecology 87 (1), 203–213.

Yee, T. W., 2008. The VGAM package. R News 8 (2), 28–39. URL `http://CRAN.R-project.org/doc/Rnews/`

Yee, T. W., 2010a. The VGAM package for categorical data analysis. Journal of Statistical Software 32 (10), 1–34. URL http://www.jstatsoft.org/v32/i10/

Yee, T. W., 2010b. VGLMs and VGAMs: an overview for applications in fisheries research. Fisheries Research 101 (1–2), 116–126.

Yee, T. W., 2011a. Quantile regression for counts and proportions. In preparation .

Yee, T. W., 2011b. Some issues raised by an analysis of the 2008 World Fly Fishing Championships data. In preparation .

Yee, T. W., Dirnböck, T., 2009. Models for analysing species' presence/absence data at two time points. Journal of Theoretical Biology 259 (4), 684–694.

Yee, T. W., Hastie, T. J., 2003. Reduced-rank vector generalized linear models. Statistical Modelling 3 (1), 15–41.

Yee, T. W., Mackenzie, M., 2002. Vector generalized additive models in plant ecology. Ecol. Model. 157 (2–3), 141–156.

Yee, T. W., Stephenson, A. G., 2007. Vector generalized linear and additive extreme value models. Extremes 10 (1–2), 1–19.

Yee, T. W., Wild, C. J., 1996. Vector generalized additive models. J. Roy. Statist. Soc. Ser. B 58 (3), 481–493.