

Introduction to Generalized Nonlinear Models in *R*

Heather Turner and David Firth

Department of Statistics
University of Warwick, UK

useR! 2009, Rennes, 2009-07-07

Copyright © Heather Turner and David Firth, 2009

Introduction to Generalized Nonlinear Models in R

Preface

Generalized linear models (logit/probit regression, log-linear models, etc.) are now part of the standard empirical toolkit.

Sometimes the assumption of a *linear* predictor is unduly restrictive.

This short course shows how *generalized nonlinear models* may be viewed as a unified class, and how to work with such models using the R package `gnm`.

The course will give an overview of the functionality of the `gnm` package; further details and examples can be found in the vignette distributed with the package:

<http://www.cran.r-project.org/package=gnm>

Introduction to Generalized Nonlinear Models in R

└ Outlines

Plan

Part I: Overview of Generalized Nonlinear Models in R

Part II: Further Examples

Part I: Overview of Generalized Nonlinear Models in R

Linear and generalized linear models

Generalized nonlinear models

Structured interactions

Introduction to the `gnm` package

Part II: Further Examples

Introduction

Stereotype model for ordinal response

UNIDIFF (log-multiplicative) models for strength of association

Biplot models for two-way data

Lee-Carter models for mortality trends

More in the package...

Part I

Overview of Generalized Nonlinear Models in R

Linear models:

e.g.,

$$E(y_i) = \beta_0 + \beta_1 x_i + \beta_2 z_i$$

$$E(y_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

$$E(y_i) = \beta_0 + \gamma_1 \delta_1 x_i + \exp(\theta_2) z_i$$

In general:

$$E(y_i) = \eta_i(\beta) = \text{linear function of unknown parameters}$$

Also assumes variance essentially constant:

$$\text{var}(y_i) = \phi a_i$$

with a_i known (often $a_i \equiv 1$).

Generalized linear models

Problems with linear models in many applications:

- ▶ range of y is restricted (e.g., y is a count, or is binary, or is a duration)
- ▶ effects are not additive
- ▶ variance depends on mean (e.g., large mean \Rightarrow large variance)

Generalized linear models specify a non-linear *link function* and *variance function* to allow for such things, while maintaining the simple interpretation of linear models.

Generalized linear model:

$$g[E(y_i)] = \eta_i = \text{linear function of unknown parameters}$$

$$\text{var}(y_i) = \phi a_i V(\mu_i)$$

with the functions g (link function) and V (variance function) known.

Examples:

- ▶ binary logistic regressions
- ▶ rate models for event counts
- ▶ log-linear models for contingency tables (including multinomial logit models)
- ▶ multiplicative models for durations and other positive measurements
- ▶ hazard models for event history data

etc., etc.

e.g., binary logistic regression:

$$y_i = \begin{cases} 1 & \text{event happens} \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_i = E(y_i) = \text{probability that event happens}$$

$$\text{var}(y_i) = \mu_i(1 - \mu_i)$$

Variance is completely determined by mean.

Common link functions are logit, probit, and (complementary) log-log, all of which transform constrained μ into unconstrained η .

e.g., multiplicative (i.e., log-linear) rate model for event counts.

'Exposure' for observation i is a fixed, known quantity t_i .

Rate model:

$$E(y_i) = t_i \exp(\beta_0) \exp(\beta_1 x_i) \exp(\beta_2 z_i)$$

i.e.,

$$\log E(y_i) = \log t_i + \beta_0 + \beta_1 x_i + \beta_2 z_i$$

— effects are rate multipliers.

Variance is typically taken as the Poisson-like function $V(\mu) = \mu$ (variance is equal to, or is proportional to, the mean).

Generalized linear: $\eta = g(\mu)$ is a linear function of the unknown parameters. Variance depends on mean through $V(\mu)$.

Generalized *nonlinear*: still have g and V , but now relax the linearity assumption.

Many important aspects remain unchanged:

- ▶ fitting by maximum likelihood or quasi-likelihood
- ▶ analysis of deviance to assess significance of effects
- ▶ diagnostics based on residuals, etc.

But technically more difficult [essentially because $\partial\eta/\partial\beta = X$ becomes $\partial\eta/\partial\beta = X(\beta)$].

Some practical consequences of the technical difficulties:

- ▶ automatic detection and elimination of redundant parameters is very difficult — it's no longer just a matter of linear algebra
- ▶ automatic generation of good starting values for ML fitting algorithms is hard
- ▶ great care is needed in cases where the likelihood has more than one maximum (which cannot happen in the linear case).

Some motivation: structured interactions

GNNs are not exclusively about structured interactions, but many applications are of this kind.

A classic example is log-linear models for structurally-square contingency tables (e.g., pair studies, before-after studies, etc.).

Pairs are classified twice, into row and column of a table of counts.

The independence model is

$$\log E(y_{rc}) = \theta + \beta_r + \gamma_c$$

or with `glm`

```
> glm(y ~ row + col, family = poisson)
```

Some standard (generalized linear) models for departure from independence are

- ▶ quasi-independence,
`y ~ row + col + Diag(row, col)`
- ▶ quasi-symmetry,
`y ~ row + col + Symm(row, col)`
- ▶ symmetry,
`y ~ Symm(row, col)`

Functions `Diag` and `Symm` are provided by the `gnm` package along with the function `Topo` for fully-specified 'topological' association structures, see `?Topo`.

Row-column association

The uniform association model (for ordered categories) has

$$\log E(y_{rc}) = \beta_r + \gamma_c + \delta u_r v_c$$

with the u_r and v_c defined as fixed, equally-spaced scores for the rows and columns.

A natural generalization is to allow the *data* to determine the scores (Goodman, 1979). This can be done either heterogeneously,

$$\log E(y_{rc}) = \beta_r + \gamma_c + \phi_r \psi_c$$

or (in the case of a structurally square table) homogeneously,

$$\log E(y_{rc}) = \beta_r + \gamma_c + \phi_r \phi_c$$

These are generalized non-linear models.

Introduction to the `gnm` package

The `gnm` package aims to provide a unified computing framework for specifying, fitting and criticizing generalized nonlinear models in R.

The central function is `gnm`, which is designed with the same interface as `glm`.

Since generalized linear models are included as a special case, the `gnm` function can be used in place of `glm`, and will give equivalent results.

For the special case $g(\mu) = \mu$ and $V(\mu) = 1$, the `gnm` fit is equivalent to an `nls` fit.

Limitations

`gnm` does not allow for correlated responses or dependence of the variance on covariates. For Gaussian data, this can be handled by `gnls`.

An important limitation of `gnm` (and indeed of the standard `glm`) is to models in which the mean-predictor function is completely determined by available explanatory variables. Latent variables (random effects) are not handled.

Nonlinear model terms

Nonlinear model terms are specified in model formulae using functions of class `"nonlin"`.

These functions specify the term structure, possibly also labels and starting values.

There are a number of `"nonlin"` functions provided by `gnm`. Some of these specify basic mathematical functions of predictors, e.g. a term of the form

$$(\alpha + \beta x)\gamma_{jk}$$

where j and k index levels of factors A and B , is specified as `Mult(1 + x, A:B)`.

Other basic `"nonlin"` functions include `Exp` and `Inv`.

Specialized `"nonlin"` functions

There are two specialized `"nonlin"` functions provided by `gnm`

`MultHomog`: for homogeneous row and column scores, as in

$$\alpha_r + \beta_c + \phi_r \phi_c$$

specified as `MultHomog(row, col)`

`Dref`: 'diagonal reference' dependence on a square classification,

$$w_1 \gamma_r + w_2 \gamma_c$$

(Sobel, 1981, 1985) specified as `Dref(row, col)`

Any (differentiable) nonlinear term can be specified by nesting existing `"nonlin"` functions or writing a custom `"nonlin"` function.

Example: Occupational Status Data

The `occupationalStatus` dataset is a contingency table classified by the occupational status of fathers (`origin`) and their sons (`destination`).

Following Goodman (1979) we fit a homogeneous row-column association model, separating out the diagonal effects

```
> RChomog <- gnm(Freq ~ origin + destination + Diag(origin, destination)
+               MultHomog(origin, destination), ofInterest = "MultHomog",
+               family = poisson, data = occupationalStatus)
```

We use the `ofInterest` argument to specify that only the parameters of the multiplicative interaction should be shown in model summaries.

Call:

```
gnm(formula = Freq ~ origin + destination + Diag(origin, destination) +
     MultHomog(origin, destination), ofInterest = "MultHomog",
     family = poisson, data = occupationalStatus)
```

Deviance Residuals:

| Min | 1Q | Median | 3Q | Max |
|---------|---------|--------|--------|--------|
| -1.6588 | -0.4297 | 0.0000 | 0.3862 | 1.7208 |

Coefficients of interest:

| | Estimate | Std. Error | z value | Pr(> z) |
|---------------------------------|----------|------------|---------|----------|
| MultHomog(origin, destination)1 | -1.5691 | NA | NA | NA |
| MultHomog(origin, destination)2 | -1.3508 | NA | NA | NA |
| MultHomog(origin, destination)3 | -0.7527 | NA | NA | NA |
| MultHomog(origin, destination)4 | -0.1688 | NA | NA | NA |
| MultHomog(origin, destination)5 | -0.1516 | NA | NA | NA |
| MultHomog(origin, destination)6 | 0.3601 | NA | NA | NA |
| MultHomog(origin, destination)7 | 0.7763 | NA | NA | NA |
| MultHomog(origin, destination)8 | 1.0199 | NA | NA | NA |

Std. Error is NA where coefficient has been constrained or is unidentified

Residual deviance: 32.561 on 34 degrees of freedom

AIC: 414.9

Over-parameterization

The `gnm` function makes no attempt to remove redundant parameters from nonlinear terms. This is deliberate.

As a consequence, fitted models are typically represented in a way that is *over-parameterized*: not all of the parameters are 'estimable' (i.e., 'identifiable', 'interpretable').

In the previous example, for instance:

$$\alpha_r + \beta_c + \phi_r \phi_c = -k^2 + (\alpha_r - k\phi_r) + (\beta_c - k\phi_c) + (\phi_r + k)(\phi_c + k)$$

The `gnm` package provides various tools (`checkEstimable`, `getContrasts`, `se`) for checking the estimability of parameter combinations, and for obtaining valid standard errors for estimable combinations.

```
> getContrasts(RChomog, pickCoef(RChomog, "MultHomog"))
```

| | estimate | SE | quasiSE | quasiVar |
|---------------------------------|----------|--------|---------|----------|
| MultHomog(origin, destination)1 | 0.0000 | 0.0000 | 0.15725 | 0.024729 |
| MultHomog(origin, destination)2 | 0.2183 | 0.2346 | 0.11901 | 0.014164 |
| MultHomog(origin, destination)3 | 0.8165 | 0.1667 | 0.06112 | 0.003736 |
| MultHomog(origin, destination)4 | 1.4003 | 0.1603 | 0.05184 | 0.002687 |
| MultHomog(origin, destination)5 | 1.4175 | 0.1720 | 0.07979 | 0.006367 |
| MultHomog(origin, destination)6 | 1.9293 | 0.1574 | 0.03598 | 0.001295 |
| MultHomog(origin, destination)7 | 2.3454 | 0.1726 | 0.07960 | 0.006336 |
| MultHomog(origin, destination)8 | 2.5890 | 0.1887 | 0.10953 | 0.011998 |

(The “quasi standard errors” allow the calculation of an approximate standard error for *any* contrast; they are independent of the choice of reference category. For the theory see Firth and de Menezes (2004). More in later examples.)

Example: Barley Heights Data

The row-column association models are special cases of the generalized additive main effects and interaction (GAMMI) model:

$$g(\mu_{rc}) = \alpha_r + \beta_c + \sum_{k=1}^K \sigma_k \phi_{kr} \psi_{kc},$$

Such models are often used in the analysis of crop yields. For example, the **BarleyHeights** dataset contains the average heights of 15 genotypes of barley over 9 years. We can model the year-genotype interaction using an AMMI-1 model:

```
> data(barleyHeights)
> barleyModel <- gnm(height ~ year + genotype + Mult(year, genotype),
+                   ofInterest = "[.]year", data = barleyHeights)
```

```
Call:
gnm(formula = height ~ year + genotype + Mult(year, genotype),
     ofInterest = "[.]year", data = barleyHeights)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.90972 -1.43403 -0.07098  1.70322  6.34119
```

```
Coefficients of interest:
              Estimate Std. Error t value Pr(>|t|)
Mult(., genotype).year1974  -5.225         NA      NA      NA
Mult(., genotype).year1975 -11.839         NA      NA      NA
Mult(., genotype).year1976 -10.121         NA      NA      NA
Mult(., genotype).year1977   4.621         NA      NA      NA
Mult(., genotype).year1978  12.292         NA      NA      NA
Mult(., genotype).year1979  -3.270         NA      NA      NA
Mult(., genotype).year1980   7.665         NA      NA      NA
Mult(., genotype).year1981   4.046         NA      NA      NA
Mult(., genotype).year1982   1.826         NA      NA      NA
```

Std. Error is NA where coefficient has been constrained or is unidentified

```
Residual deviance: 649.49 on 91 degrees of freedom
AIC: 685.19
```

AMMI and SVD

An AMMI-2 model may be appropriate here. Due to the identity link the additional coefficients can be found directly from SVD of the residuals from the previous model. We use the `residSVD` function to do this

```
> mult2 <- residSVD(barleyModel, year, genotype, d = 1)
```

then set the `start` argument to `gnm` in a call to fit the AMMI-2 model, specified using the `instances` function:

```
> barleyModel2 <- gnm(height ~ year + genotype +
+                   instances(Mult(year, genotype), 2),
+                   start = c(coef(barleyModel), mult2),
+                   data = barleyHeights)
```

No iterations are performed.

Comparing Models

The two AMMI models can be compared using `anova`

```
> anova(barleyModel, barleyModel2, test = "F")
```

Analysis of Deviance Table

Model 1: `height ~ year + genotype + Mult(year, genotype)`

Model 2: `height ~ year + genotype + Mult(year, genotype, inst = 1) + Mult(year, genotype, inst = 2)`

| | Resid. Df | Resid. Dev | Df | Deviance | F | Pr(>F) |
|---|-----------|------------|----|----------|--------|-----------|
| 1 | 91 | 649.49 | | | | |
| 2 | 72 | 448.93 | 19 | 200.55 | 1.6929 | 0.05761 . |

Showing only marginal significance for the extra component.

Estimable Contrasts for GAMMI Models

Now since, for example,

$$\begin{aligned}\alpha_r + \beta_r + \phi_r \psi_c &= \alpha_r + (\beta_r - \psi_c) + (\phi_r + 1) \psi_c \\ &= \alpha_r + \beta_r + (2\phi_r)(\psi_c/2)\end{aligned}$$

we need to constrain both the location and scale. To obtain the parameterization in which σ_k is the singular value for component k , the row and column scores must be constrained so that

$$\begin{aligned}\sum_r \phi_r &= \sum_c \psi_c = 0 \\ \text{and } \sum_r \phi_r^2 &= \sum_c \psi_c^2 = 1\end{aligned}$$

These constraints are met by the scaled contrasts

$$\frac{\phi_r - \bar{\phi}}{\sqrt{\sum_r (\phi_r - \bar{\phi})^2}}, \quad \frac{\psi_c - \bar{\psi}}{\sqrt{\sum_c (\psi_c - \bar{\psi})^2}}$$

Which can be obtained as follows:

```
> phi <- getContrasts(barleyModel, pickCoef(barleyModel, "[.]y"),
+                   ref = "mean", scaleWeights = "unit")
> psi <- getContrasts(barleyModel, pickCoef(barleyModel, "[.]g"),
+                   ref = "mean", scaleWeights = "unit")
```

```
> phi$qvframe[c(1, 5, 9),]
```

| | Estimate | Std. Error |
|----------------------------|-------------|------------|
| Mult(., genotype).year1974 | -0.22662084 | 0.05621029 |
| Mult(., genotype).year1978 | 0.53320299 | 0.04775769 |
| Mult(., genotype).year1982 | 0.07922729 | 0.05770322 |

```
> psi$qvframe[c(1, 5, 9),]
```

| | Estimate | Std. Error |
|--------------------------|------------|------------|
| Mult(year, .).genotype1 | -0.0791433 | 0.05913864 |
| Mult(year, .).genotype13 | 0.1683167 | 0.05843057 |
| Mult(year, .).genotype3 | -0.1697188 | 0.05841527 |

```
> svd(termPredictors(barleyModel)[, "Mult(year, genotype)"])$d
```

```
43.49599
```

A note of caution on interpretation

Care is needed in interpreting apparent multiplicative effects.

For example, in political science much use has been made of generalized logit and probit models in which the standard binary-response assumption (in terms of probit)

$\text{pr}(y_i = 1) = \Phi(x_i' \beta / \sigma)$ is replaced by a model which allows non-constant variance in the underlying latent regression:

$$\text{pr}(y_i = 1) = \Phi[x_i' \beta / \exp(z_i' \gamma)]$$

This clearly results in a multiplicative model for the mean: in R, the above would be specified as

```
> gnm(y ~ -1 + Mult(x, Exp(z)), family = binomial(link="probit"))
```

It is therefore impossible to distinguish, with binary data, between two distinct generative mechanisms: underlying variance depends on z ; or effect of x is modulated by z .

Part II

Further Examples

In this part we will consider some particular applications of `gnm` in some detail, showing some further features of the package as we go along.

The models we shall demonstrate are as follows

- ▶ the stereotype model (Anderson, 1984), for ordered categorical response
- ▶ UNIDIFF models (Xie, 1992; Erikson and Goldthorpe, 1992) for 3-way contingency tables
- ▶ biplot models (Gabriel, 1998) for projecting data onto two linearly independent dimensions
- ▶ the Lee-Carter model (Lee and Carter, 1992) for mortality data

Stereotype Models

The stereotype model (Anderson, 1984) is suitable for ordered categorical data. It is a special case of the multinomial logistic model:

$$pr(y_i = c | \mathbf{x}_i) = \frac{\exp(\beta_{0c} + \beta_c^T \mathbf{x}_i)}{\sum_r \exp(\beta_{0r} + \beta_r^T \mathbf{x}_i)}$$

in which only the *scale* of the relationship with the covariates changes between categories:

$$pr(y_i = c | \mathbf{x}_i) = \frac{\exp(\beta_{0c} + \gamma_c \beta_c^T \mathbf{x}_i)}{\sum_r \exp(\beta_{0r} + \gamma_r \beta_r^T \mathbf{x}_i)}$$

Further Examples

└ Stereotype model for ordinal response

The stereotype model can be fitted using `gnm` by re-expressing the categorical data as counts and fitting the log-linear model

$$\log \mu_{ic} = \alpha_i + \beta_{0c} + \gamma_c \sum_r \beta_r x_{ir}$$

The function `expandCategorical` converts categorical data into the desired format.

Further Examples

└ Stereotype model for ordinal response

Example: Back Pain Data

The `backPain` data set is taken from Anderson's paper. For 101 patients it records 3 prognostic variables recorded at baseline and their level of back pain after 3 weeks.

```
> backPain[1,]
  x1 x2 x3 pain
1  1  1  1 same
```

```
> backPainLong <- expandCategorical(backPain, "pain")
> head(backPainLong)
```

| | x1 | x2 | x3 | pain | count | id |
|-----|----|----|----|----------------------|-------|----|
| 1 | 1 | 1 | 1 | worse | 0 | 1 |
| 1.1 | 1 | 1 | 1 | same | 1 | 1 |
| 1.2 | 1 | 1 | 1 | slight.improvement | 0 | 1 |
| 1.3 | 1 | 1 | 1 | moderate.improvement | 0 | 1 |
| 1.4 | 1 | 1 | 1 | marked.improvement | 0 | 1 |
| 1.5 | 1 | 1 | 1 | complete.relief | 0 | 1 |

Further Examples

└ Stereotype model for ordinal response

The stereotype model can then be fitted as follows

```
> stereotype <- gnm(count ~ pain + Mult(pain, x1 + x2 + x3),
+   eliminate = id, family = poisson, data = backPainLong)
```

The `eliminate` argument of `gnm` is used to specify that the `id` parameters replace the intercept in the model. This way `gnm` will use a method exploiting the structure of these parameters in order to improve the computational efficiency of their estimation, and the parameters will be excluded from summaries of the model object.

Further Examples

└─Stereotype model for ordinal response

We can compare the stereotype model to the multinomial logistic model:

```
> logistic <- gnm(count ~ pain + pain:(x1 + x2 + x3),
+   eliminate = id, family = poisson, data = backPainLong)
> anova(stereotype, logistic)
```

```
anova(stereotype, logistic, test = "Chisq")
```

Analysis of Deviance Table

```
Model 1: count ~ pain + Mult(pain, x1 + x2 + x3)
Model 2: count ~ pain + pain:x1 + pain:x2 + pain:x3
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1         493     303.100
2         485     299.015   8    4.085    0.849
```

Further Examples

└─Stereotype model for ordinal response

In order to make the category-specific multipliers identifiable, we must constrain both the location and scale.

One way to do this is to set the last multiplier to one and fix the coefficient of the first covariate to one. We can do this using the **constrain** and **constrainTo** arguments of `gnm`:

```
> stereotype <- gnm(count ~ pain + Mult(pain, offset(x1) + x2 + x3),
+   eliminate = id, family = poisson, data = backPainLong,
+   constrain = "[.]paincomplete.relief", constrainTo = 1)
> ofInterest(stereotype) <- pickCoef(stereotype, "Mult")
```

Further Examples

└─Stereotype model for ordinal response

```
> parameters(stereotype)
```

Coefficients of interest:

```
  Mult(., x2 + x3 + offset(x1)).painworse
      6.3718456
  Mult(., x2 + x3 + offset(x1)).painsame
      2.6621173
  Mult(., x2 + x3 + offset(x1)).painslight.improvement
      2.8621571
  Mult(., x2 + x3 + offset(x1)).painmoderate.improvement
      3.7389111
  Mult(., x2 + x3 + offset(x1)).painmarked.improvement
      1.7602579
  Mult(., x2 + x3 + offset(x1)).paincomplete.relief
      1.0000000
  Mult(pain, . + x3 + offset(x1)).x2
      0.5735620
  Mult(pain, x2 + . + offset(x1)).x3
      0.5050045
```

Further Examples

└ UNIDIFF (log-multiplicative) models for strength of association

UNIDIFF-type models

High-order interactions in their 'raw' form can involve large numbers of parameters. Often more economical/interpretable summaries are possible.

The classic 'UNIDIFF' model relates to a 3-way table of counts y_{rct} , viewed as a set of T two-way tables $y_{rc1}, y_{rc2}, \dots, y_{rct}$.

Interest is in the row-column association, and variation between tables t in the strength of that association.

The UNIDIFF model postulates a common pattern of (log) odds ratios, modulated by a constant that is specific to each table:

$$\log(\mu_{rct}) = \alpha_{rt} + \beta_{ct} + e^{\gamma_t} \delta_{rc}$$

Further Examples

└ UNIDIFF (log-multiplicative) models for strength of association

Example: Social Mobility Data

The **yaish** dataset is a study of social mobility by Yaish (1998, 2004). It is a 3-way contingency table classified by:

orig father's social class (7 levels)

dest son's social class (7 levels)

educ son's education level (5 levels)

Interest is in the affect of `educ` on the interaction between `orig` and `dest`

Further Examples

└ UNIDIFF (log-multiplicative) models for strength of association

UNIDIFF Model

The UNIDIFF model

$$\log(\mu_{rct}) = \alpha_{rt} + \beta_{ct} + e^{\gamma_t} \delta_{rc}$$

can be specified using `Mult` and `Exp`

```
> unidiff <- gnm(Freq ~ educ*orig + educ*dest +
+               Mult(Exp(educ), orig:dest),
+               ofInterest = "[.]educ", family = poisson,
+               data = yaish, subset = (dest != 7))
```

Primary interest is in $\gamma_1, \dots, \gamma_5$, which measure the relative strength of interaction in the 5 tables. Only the differences $\gamma_t - \gamma_s$, and other such *contrasts*, are identifiable.

Further Examples

└ UNIDIFF (log-multiplicative) models for strength of association

```
> unidiffContrasts <- getContrasts(unidiff, ofInterest(unidiff))
> summary(unidiffContrasts, digits = 2)
```

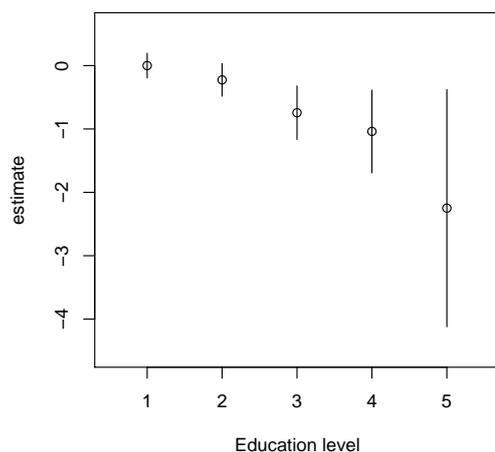
```
Model call: gnm(formula = Freq ~ educ * orig + educ * dest +
  Mult(Exp(educ), orig:dest), ofInterest = "[.]educ",
  family = poisson, data = yaish, subset = (dest != 7))
              estimate SE quasiSE quasiVar
Mult(Exp(.), orig:dest).educ1    0.00 0.00  0.098  0.0095
Mult(Exp(.), orig:dest).educ2   -0.23 0.16  0.129  0.0166
Mult(Exp(.), orig:dest).educ3   -0.74 0.23  0.212  0.0449
Mult(Exp(.), orig:dest).educ4   -1.04 0.34  0.326  0.1063
Mult(Exp(.), orig:dest).educ5   -2.25 0.95  0.936  0.8754
Worst relative errors in SEs of simple contrasts (%): -0.9 1.4
Worst relative errors over *all* contrasts (%): -3.6 2.1
```

Further Examples

└ UNIDIFF (log-multiplicative) models for strength of association

Contrasts Plot

```
plot(unidiffContrasts, xlab = "Education Level", levelNames = 1:5)
```

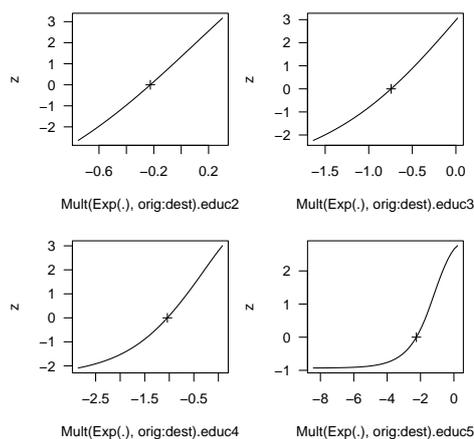


Further Examples

└ UNIDIFF (log-multiplicative) models for strength of association

Profiling

```
> unidiff2 <- update(unidiff, constrain = "[.]educ1")
> prof <- profile(unidiff2, ofInterest(unidiff2), trace = TRUE)
> plot(prof)
```



Further Examples

└ UNIDIFF (log-multiplicative) models for strength of association

Profile Confidence Intervals

```
> conf <- confint(prof)
> print(conf, digits = 2)
```

```

                2.5 % 97.5 %
Mult(Exp(.), orig:dest).educ1    NA    NA
Mult(Exp(.), orig:dest).educ2  -0.6    0.1
Mult(Exp(.), orig:dest).educ3  -1.5   -0.2
Mult(Exp(.), orig:dest).educ4  -2.6   -0.3
Mult(Exp(.), orig:dest).educ5  -Inf   -0.7
```

Further Examples

└ Biplot models for two-way data

Biplots

Biplots are graphical displays of data arrays which represent the objects that index all dimensions of the array on the same plot.

So for a two-way table, a biplot represents both the rows and columns at the same time.

The biplot is constructed from a rank-2 representation of the data. Here we consider the generalized bilinear model

$$g(\mu_{ij}) = \alpha_{1i}\beta_{1j} + \alpha_{2i}\beta_{2j}$$

Further Examples

└ Biplot models for two-way data

Example: Leaf Blotch Data

The **barley** data set gives the percentage of leaf area affected by leaf blotch for 10 varieties of barley grown at nine sites.

As suggested by Wedderburn (1974) we model these data using a logit link and a variance proportional to the square of that of the binomial, i.e. $V(\mu) = \mu^2(1 - \mu)^2$

```
> set.seed(138)
> biplotModel <- gnm(y ~ -1 + instances(Mult(site, variety), 2),
+                   family = wedderburn, data = barley)
```

The effect of site i can be represented by the point

$$(\alpha_{1i}, \alpha_{2i})$$

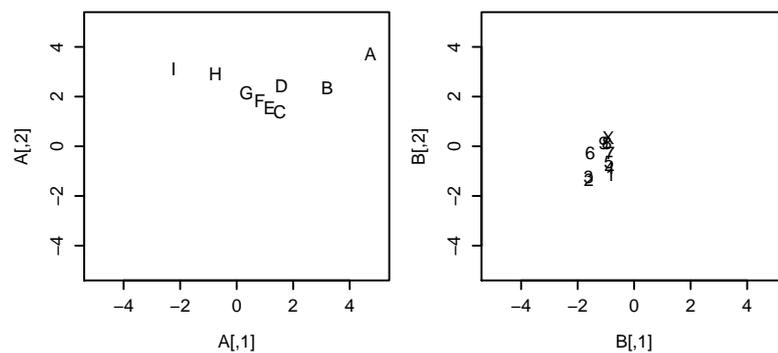
in the space spanned by the linearly independent basis vectors

$$a_1 = (\alpha_{11}, \alpha_{12}, \dots, \alpha_{19})^T$$

$$a_2 = (\alpha_{21}, \alpha_{22}, \dots, \alpha_{29})^T$$

Thus we can represent the sites and varieties separately as follows

```
> coefs <- matrix(coef(biplotModel), nc = 2)
> A <- coefs[1:9,]
> B <- coefs[-c(1:9),]
> par(mfrow = c(1, 2))
> plot(A, pch = levels(barley$site),
+      xlim = c(-4, 4), ylim = c(-4, 4))
> plot(B, pch = levels(barley$variety),
+      xlim = c(-4, 4), ylim = c(-4, 4))
```



Further Examples

└─ Biplot models for two-way data

Interpreting such plots can be misleading however, because the basis vectors are not orthogonal.

Furthermore, to provide insight into the model as a whole, we would like to create a combined plot, therefore require basis vectors with the same scale.

We can obtain basis vectors with the desired properties via SVD of the predictors: the model stays the same, but the parametrization changes.

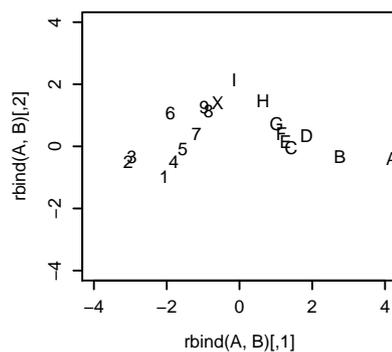
Further Examples

└─ Biplot models for two-way data

```
> barleySVD <- svd(matrix(biplotModel$predictors, 10, 9), 2, 2)
> A <- sweep(barleySVD$v, 2, sqrt(barleySVD$d), "*")
> B <- sweep(barleySVD$u, 2, sqrt(barleySVD$d), "*")
> # all.equal(c(t(A) %*% t(B)), unname(biplotModel$predictors))
> plot(rbind(A, B),
+      pch = c(levels(barley$site), levels(barley$variety)),
+      xlim = c(-5, 5), ylim = c(-5, 5))
```

Further Examples

└─ Biplot models for two-way data



Further Examples

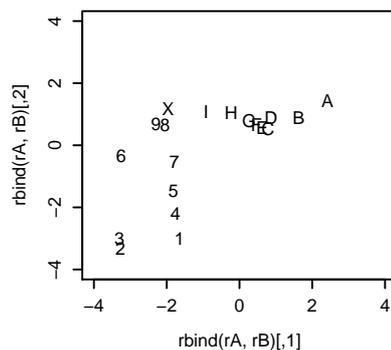
└ Biplot models for two-way data

The product AB^T is unaffected by rotation or reciprocal scaling, so we apply a rotation to align the points with the axes and scale so that the scores for the sites are about +1 on the vertical scale

```
> a <- pi/5
> rotation <- matrix(c(cos(a), sin(a), -sin(a), cos(a)),
+                   2, 2, byrow = TRUE)
> rA <- (2 * A/3) %%% rotation
> rB <- (3 * B/2) %%% rotation
> plot(rbind(rA, rB),
+      pch = c(levels(barley$site), levels(barley$variety)),
+      xlim = c(-5, 5), ylim = c(-5, 5))
```

Further Examples

└ Biplot models for two-way data



Further Examples

└ Biplot models for two-way data

In the original biplot, the co-ordinates for the sites and varieties were given by the rows of A and B respectively, i.e

$$\alpha_i^T = \sqrt{d}(u_{1i}, u_{2i})$$

$$\beta_j^T = \sqrt{d}(v_{1j}, v_{2j})$$

The rotated and scaled biplot suggests the simpler model

$$\alpha_i^T = (\gamma_i, 1)$$

$$\beta_j^T = (\delta_0 + \delta_1 I(j \in \{2, 3, 6\}), \tau_j)$$

which Gabriel describes as a double additive model.

We fit the double additive model as follows

```
> variety.binary <- barley$variety %in% c(2, 3, 6)
> doubleAdditive <- update(biplotModel,
+                           . ~ -1 + variety + Mult(site, variety.binary))
```

Comparing the models, we can confirm that the double additive model is adequate

[Analysis of Deviance Table](#)

```
Model 1: y ~ 0 + variety + Mult(site, variety.binary)
Model 2: y ~ 0 + Mult(site, variety, inst = 1) +
          Mult(site, variety, inst = 2)
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         71      50.966
2         56      41.024 15    9.942    0.803
```

Lee-Carter model for mortality trends

For the study and projection of age-specific population mortality rates, Lee and Carter (1992) proposed a model that has been the basis of many subsequent analyses.

Suppose that death count D_{ay} for individuals of age a in year y has mean μ_{ay} and 'quasi-Poisson' variance $\phi\mu_{ay}$.

Lee-Carter model:

$$\log(\mu_{ay}/e_{ay}) = \alpha_a + \beta_a\gamma_y,$$

where e_{ay} is the 'exposure' (number of lives at risk).

A generalized nonlinear model with a single multiplicative term.

Example: Mortality in Canada

Data from the [Human Mortality Database](#) on male deaths in Canada between 1921 and 2003.

Fit the Lee-Carter model:

```
> LCmodel <- gnm(mDeaths ~ Age + Mult(Exp(Age), Year),
+               offset = log(mExposure), family = "quasipoisson",
+               data = Canada)
```

Here we use the fact that the model only really makes sense if all of the β_a parameters, which represent the 'sensitivity' of age group a to a change in the level of general mortality, have the same sign: we impose this constraint by using `Exp(Age)` instead of just `Age` in the multiplicative term.

Residual deviance is 32422.68 on 6400 degrees of freedom — substantial overdispersion relative to Poisson.

Further Examples

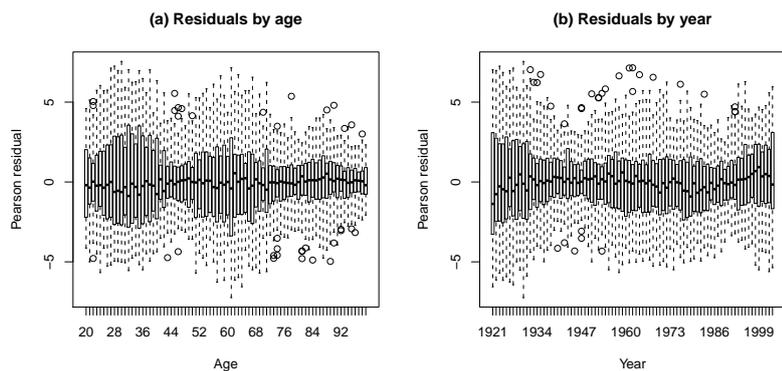
└ Lee-Carter models for mortality trends

Where is the overdispersion? Look at residuals:

```
> with(Canada, {
+
+   res <- residuals(LCmodel, type = "pearson")
+
+   plot(Age, res, xlab="Age", ylab="Pearson residual",
+        main = "(a) Residuals by age")
+
+   plot(Year, res, xlab="Year", ylab="Pearson residual",
+        main = "(b) Residuals by year")
+
+ })
```

Further Examples

└ Lee-Carter models for mortality trends



Further Examples

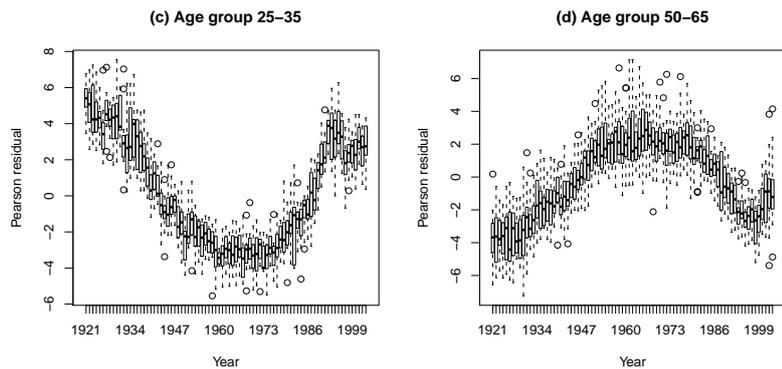
└ Lee-Carter models for mortality trends

Look more closely at the two age groups that exhibit most of the over-dispersion:

```
> plot(Year[(age>24) & (age<36)], res[(age>24) & (age<36)],
+      xlab = "Year", ylab = "Pearson residual",
+      main = "(c) Age group 25-35")
> plot(Year[(age>49) & (age<66)], res[(age>49) & (age<66)],
+      xlab = "Year", ylab = "Pearson residual",
+      main = "(d) Age group 50-65")
```

Further Examples

└ Lee-Carter models for mortality trends



(for more detail on this example see the `gnm` vignette)

Further Examples

└ More in the package...

Other examples in the `gnm` package

In the `gnm` *vignette*, other examples include:

- ▶ more general $RC(m)$ association models;
- ▶ diagonal reference models in social mobility studies;
- ▶ compound exponential decay curves.

Further examples in `gnm` package *help files* include:

- ▶ “double unidiff” models for a 4-dimensional association structure in political sociology (`cautres`);
- ▶ Rasch-type scaling of legislative roll calls (`House2001`).

Further Examples

└ More in the package...

To conclude...

- ▶ Current `gnm` handles many much-used generalized nonlinear model types.
- ▶ Formula interface encourages experimentation, and uninhibited modelling.
- ▶ Key to generality of implementation is the use of *over-parameterized* model representation; tools are provided for handling this correctly.
- ▶ More generality still? *Random effects* in nonlinear terms. MLE is then difficult; approximations (PQL) or composite (pseudo) likelihoods? On-going.

References

- Anderson, J. A. (1984). Regression and ordered categorical variables. *J. R. Statist. Soc. B* 46(1), 1–30.
- Erikson, R. and J. H. Goldthorpe (1992). *The Constant Flux*. Oxford: Clarendon Press.
- Firth, D. and R. X. de Menezes (2004). Quasi-variances. *Biometrika* 91, 65–80.
- Gabriel, K. R. (1998). Generalised bilinear regression. *Biometrika* 85, 689–700.
- Goodman, L. A. (1979). Simple models for the analysis of association in cross-classifications having ordered categories. *J. Amer. Statist. Assoc.* 74, 537–552.
- Lee, R. D. and L. Carter (1992). Modelling and forecasting the time series of US mortality. *Journal of the American Statistical Association* 87, 659–671.

References

- Sobel, M. E. (1981). Diagonal mobility models: A substantively motivated class of designs for the analysis of mobility effects. *Amer. Soc. Rev.* 46, 893–906.
- Sobel, M. E. (1985). Social mobility and fertility revisited: Some new models for the analysis of the mobility effects hypothesis. *Amer. Soc. Rev.* 50, 699–712.
- Wedderburn, R. W. M. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika* 61, 439–447.
- Xie, Y. (1992). The log-multiplicative layer effect model for comparing mobility tables. *American Sociological Review* 57, 380–395.
- Yaish, M. (1998). *Opportunities, Little Change. Class Mobility in Israeli Society, 1974–1991*. Ph. D. thesis, Nuffield College, University of Oxford.
- Yaish, M. (2004). *Class Mobility Trends in Israeli Society, 1974–1991*. Lewiston: Edwin Mellen Press.