4.8 Constraints on the Component Functions

Constraining the component functions is very important for some models because often one wishes to force some of the component functions to be equal or zero. For example, if one fitted eye data to a bivariate logistic model, then one might wish to constrain $\eta_1 = \eta_2$ (eyes are exchangeable). Another example is the proportional odds model where there is a parallelism (or proportional odds) assumption $\beta_{(1)k} = \cdots = \beta_{(M)k},\ k = 1, \ldots, p$.

Casual users of VGAM will be able to get by using the `parallel`, `exchangeable` and `zero` arguments provided in those family functions which anticipate its use. This is discussed in detail in the following section. Users wanting to constrain component functions in more elaborate ways need to understand the underlying details given in Section 4.8.4. It should be noted that the arguments `parallel`, `exchangeable` and `zero` merely provide a convenient short-cut for the method described in Section 4.8.4.

4.8.1 The `nointercept` and `zero` Arguments

Many VGAM family functions have one or more of the arguments `zero`, `nointercept`, `parallel` and `exchangeable`. These provide a convenient way of enforcing a common type of constraint on that particular model.

The default values of the arguments `parallel` and `exchangeable` may or may not apply to the intercepts; this depends on that particular family function. For example, setting `cumulative(parallel=T)` never affects the intercepts in a cumulative logit model, whereas setting `binom2.or(exchangeable=T)` does apply to the intercepts in a bivariate logistic model.

The first type of constraint supported by VGAM are `zero` and `nointercept`. They are roughly opposites of each other. The former allows any number of linear predictors to be modelled with an intercept term only. For example,

```
vglm(y ~ x1, binom2.or(exchangeable=T, zero=3))
```

fits the model

$$\begin{aligned} \text{logit}\, P(Y_j = 1) &= \beta_{(1)0} + \beta_{(1)1}\, x_1, \qquad j = 1, 2\,, \\ \log\ \psi &= \beta_{(2)0}\ . \end{aligned}$$

Thus the estimated odds ratio is simply $\hat{\psi} = \exp\{\hat{\beta}_{(2)0}\}$. The argument `zero` is often applied to models where one of the parameters is often treated as a scalar unknown rather than being modelled with respect to the covariates. Parameters such as odds ratios, correlation coefficients or powers are common examples. In general, `zero` can be assigned a vector whose (integer) values lie between 1 and M inclusive. Note that a null model corresponds to `zero=1:M`, though this won't work in practice.

4.8.2 The `parallel`, `exchangeable`, etc. Arguments

Input for the `parallel` and `exchangeable` arguments are different from `nointercept` and `zero`. They may be assigned a single logical value or a formula. They are best illustrated by way of examples.

1. `parallel=T` means the parallelism assumption is applied to all terms (or equivalently, to all columns of the model matrix), except possibly the intercept.

2. `parallel = T ~ x2 + x5` means the parallelism assumption is only applied to X_2, X_5, and the intercept.

3. `parallel = ~ x2 + x5` means the same as `parallel = T ~ x2 + x5` because the 'response' is `TRUE` by default.

4. `exchangeable = F ~ x2` applies the exchangeability constraint to all terms except for X_2 and the intercept.

5. `exchangeable = F ~ x2 - 1` applies the exchangeability constraint to all terms (including the intercept) except for X_2.

6. `exchangeable = T ~ s(x2, df=2) - 1` applies the exchangeability constraint only to the smooth term `s(x2, df = 2)`. Note that one doesn't have to worry about white spaces etc. when typing in the terms to match (but you **do** have to worry about white spaces when using the `constraints` argument; see Section 4.8.4); nevertheless the term must be typed in exactly. Typing `exchangeable = T ~ s(x2) - 1` will have no effect.

It can be seen that special care must be taken regarding the intercept term when a constraints argument is assigned a formula. The user in such a case must specify explicitly whether or not the constraint applies to the intercept term. Recall that, by default, an intercept term is included, and it can be dropped by adding a `-1` to the formula.

A final note is that the formula must be simple: `.` and `-x1` etc. are not handled.

4.8.3 The `s()` Term and Constraints

The specification of `df`/`spar` in `s()` under `zero`, `parallel` and `exchangeable` is simple: the successive values of `df`/`spar` correspond to successive unique component functions of that variable. For example,

```
vgam(y ~ s(x, df=c(4,1)), binom2.or(exchangeable=T))
```

fits the model

$$\begin{aligned} \text{logit}\, P(Y_j = 1) &= \beta_{(1)0} + f_{(1)1}(x), \qquad j = 1, 2\,, \\ \log\ \psi(\boldsymbol{x}) &= \beta_{(2)0} + \beta_{(2)1}\, x, \end{aligned}$$

where $f_{(1)1}$ has 4 degrees of freedom (1 degree of freedom denotes a linear fit). If the length of `df`/`spar` is less than the number of unique component functions of that variable then the values are recycled.

4.8.4 More Complicated Constraints

VGAM supports a wider range of constraints than offered by `zero`/`parallel`/`exchangeable` through the `constraints` argument. Constraints have the form

$$\begin{aligned} \boldsymbol{\eta} &= \boldsymbol{h} + \boldsymbol{\beta}_0 + \boldsymbol{f}_1 + \cdots + \boldsymbol{f}_p \\ &= \boldsymbol{h} + \mathbf{H}_0\, \boldsymbol{\beta}_0^* + \mathbf{H}_1\, \boldsymbol{f}_1^* + \cdots + \mathbf{H}_p\, \boldsymbol{f}_p^* \end{aligned} \tag{4.7}$$

where $\boldsymbol{h}$ is a vector of known offsets, $\mathbf{H}_0, \mathbf{H}_1, \ldots, \mathbf{H}_p$ are known full-column rank *constraint matrices*, $\boldsymbol{f}_k^*$ is a vector containing a possibly reduced set of component functions and $\boldsymbol{\beta}_0^*$ is a vector of unknown parameters.

Note that starred quantities in (4.7) are unknown and to be estimated. With no constraints at all, $\mathbf{H}_0 = \mathbf{H}_1 = \cdots = \mathbf{H}_p = \mathbf{I}_M$ and $\boldsymbol{\beta}_0^* = \boldsymbol{\beta}_0 = (\beta_{(1)0}, \cdots, \beta_{(M)0})^T$. Like the $\boldsymbol{f}_k$, the $\boldsymbol{f}_k^*$ are centered. The default value of the $M \times 1$ offset vector $\boldsymbol{h}$ is $\mathbf{0}$, i.e., no offset at all. The following are some examples of (4.7), none of which have offsets.

1. ($\eta_1 = \eta_2$; e.g., exchangeability in the bivariate logit model)

$$\begin{aligned} \eta_1 &= \alpha_1 + f_{(1)1}(x_1) + f_{(1)2}(x_2) \\ \eta_2 &= \alpha_1 + f_{(1)1}(x_1) + f_{(1)2}(x_2) \\ \eta_3 &= \alpha_2 + f_{(2)1}(x_1) + f_{(2)2}(x_2) \end{aligned}$$

   Then

$$\mathbf{H}_0 = \mathbf{H}_1 = \mathbf{H}_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

2.

$$\begin{aligned} \eta_1 &= \alpha_1 + f_{(1)1}(x_1) + f_{(1)2}(x_2) \\ \eta_2 &= \alpha_2 + f_{(1)1}(x_1) + f_{(2)2}(x_2) \\ \eta_3 &= \alpha_3 + f_{(2)1}(x_1) + f_{(3)2}(x_2) \end{aligned}$$

   Then

$$\mathbf{H}_0 = \mathbf{I}_3, \quad \mathbf{H}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{H}_2 = \mathbf{I}_3 \, .$$

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

$$\begin{aligned} \eta_1 &= \alpha_1 + f_{(1)1}(x_1) + f_{(1)2}(x_2) \\ \eta_2 &= \alpha_2 + f_{(1)1}(x_1) + f_{(1)2}(x_2) \\ \eta_3 &= \alpha_3 + f_{(1)1}(x_1) + f_{(1)2}(x_2) \end{aligned}$$

   Then

$$\mathbf{H}_0 = \mathbf{I}_3, \quad \mathbf{H}_1 = \mathbf{H}_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \, .$$

4.

$$\begin{aligned} \eta_1 &= \alpha_1 + f_{(1)1}(x_1) + f_{(1)2}(x_2) \\ \eta_2 &= \alpha_2 + f_{(1)1}(x_1) \\ \eta_3 &= \alpha_3 + f_{(2)1}(x_1) + f_{(2)2}(x_2) \end{aligned}$$

   Then

$$\mathbf{H}_0 = \mathbf{I}_3, \quad \mathbf{H}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{H}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \, .$$

5. For $\eta_1 = \eta_2 = \cdots = \eta_M$ use $\mathbf{H}_0 = \mathbf{H}_1 = \cdots = \mathbf{H}_p = \mathbf{1}_M$. □

VGAM allows the input of the constraint matrices $\mathbf{H}_0, \mathbf{H}_1, \ldots, \mathbf{H}_p$ through the `constraints=` argument. (The default value of `NULL` signifies no constraints at all). Like `contrasts=`, `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()`. For example, the constraint matrices for Example 3 could be set up by

```
clist = list(x1=matrix(1,3,1), x2=matrix(1,3,1))
```

and then invoked with something like `vgam(..., constraints=clist, ...)`. Similarly, for Example 1,

```
cm = matrix(c(1,1,0,0,0,1), 3, 2)
clist = list("(Intercept)"=cm, x1=cm, x2=cm)
vgam(..., constraints=clist, ...)
```

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

When using the `constraints` argument with functions such as `s()`, `bs()` etc., one must be very careful to get any white spaces right. For example,

```
clist = list("(Intercept)"=diag(3),
             "poly(x1,3)"=matrix(1,3,1),
             "s(x2,df=3)"=matrix(1,3,1))
vgam(y ~ poly(x1,3) + s(x,df=3), ..., constraints=clist)
```

will not work, but

```
clist = list("(Intercept)"=diag(3),
             "poly(x1, 3)"=matrix(1,3,1),
             "s(x2, df = 3)"=matrix(1,3,1))
vgam(y ~ poly(x1,3) + s(x,df=3), ..., constraints=clist)
```

will. If you are unsure, type something like

```
as.character(y ~ poly(x1,3) + s(x,df=3))
```

to see how to get the white spaces correct. It is necessary to get it exactly right because the names of the `constraints` list are matched with the character representation of the terms in the formula.

The specification of `df`/`spar` in `s()` under constraints is simple: the successive values of `df`/`spar` correspond to successive columns of the constraint matrix of that variable. If the length of `df`/`spar` is less than the number of columns of the constraint matrix then values are recycled. Here are some further examples:

1. Suppose in Example 1 above that we wish to fit $f_{(1)1}$ with 4 degrees of freedom, $f_{(2)1}$ with 3 degrees of freedom, $f_{(1)2}$ with 5 degrees of freedom and $f_{(2)2}$ with 4 degrees of freedom. Then `~ s(x1,df=c(4,3)) + s(x2,df=c(5,4))`.

2. Suppose in Example 2 above that we wish to fit all terms linear except for $f_{(2)2}$ which has 4 degrees of freedom. Then `~ s(x1,df=1) + s(x2,df=c(1,4,1))`, or equivalently, `~ x1 + s(x2,c(1,4))`.

3. Suppose in Example 3 above that we wish to fit $f_{(1)1}$ and $f_{(1)2}$ with 4 degrees of freedom. Then `~ s(x1) + s(x2)`.