Fitting Paired Comparison Models in R

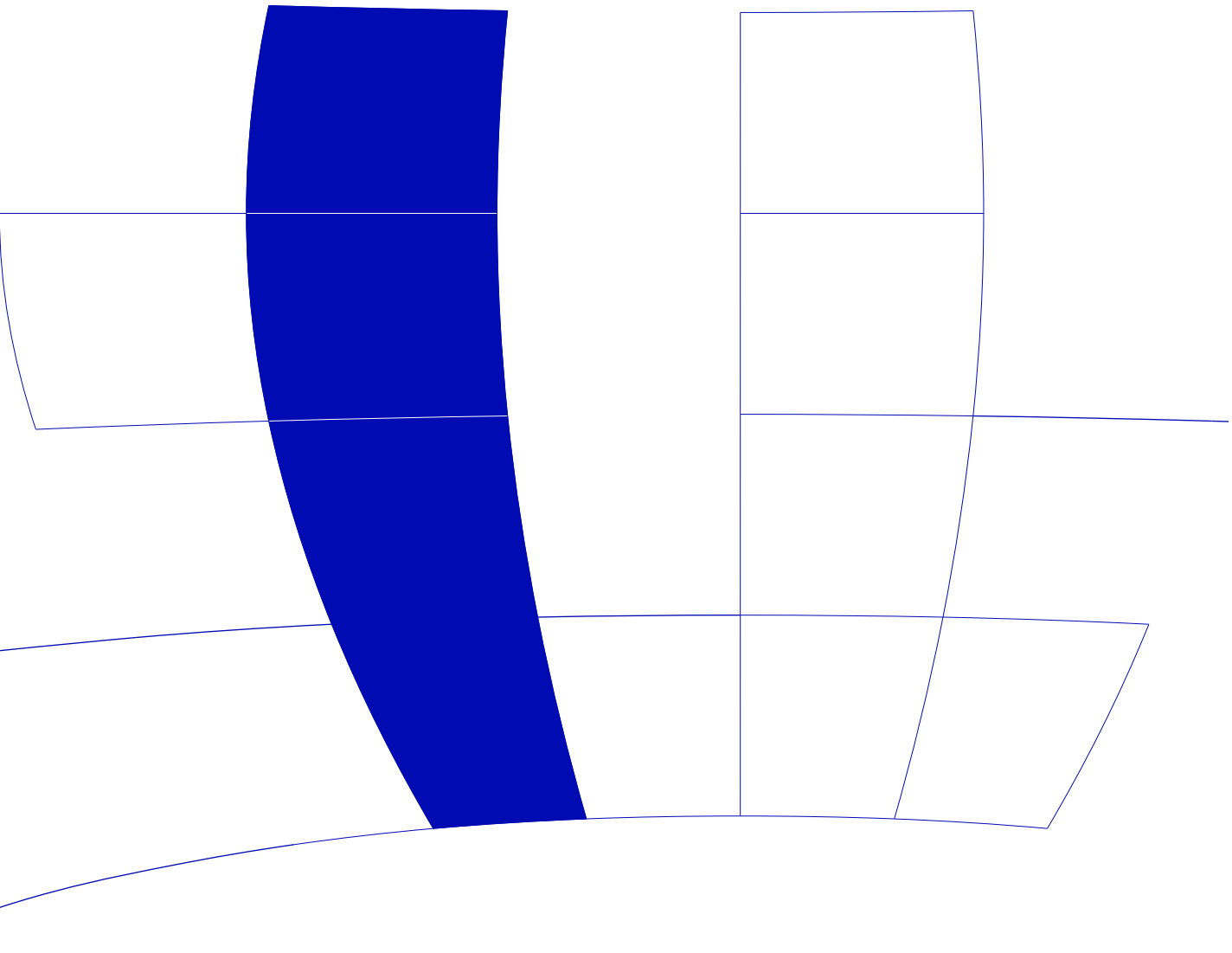
Reinhold Hatzinger, Brian J. Francis

Department of Statistics and Mathematics
Wirtschaftsuniversität Wien

Research Report Series

Report 3
May 2004

http://statistik.wu-wien.ac.at/



Fitting Paired Comparison Models in R

Reinhold Hatzinger Brian J. Francis

May, 2004

Abstract

Paired comparison models in loglinear form are generalised linear models and can be fitted using the IWLS algorithm. Unfortunately, the design matrices can become very large and thus a method is needed to reduce computational load (relating to both space and time). This paper discusses an algorithm for fitting loglinear paired comparison models in the presence of many nuisance parameters which is based on partition rules for symmetric matrices and takes advantage of the special structure of the design matrix in Poisson loglinear models. The algorithm is implemented as an R function. Some simple examples illustrate its use for fitting both paired comparison models and (multinomial) logit models.

1 Some definitions for loglinear paired comparison models

The basic paired comparison model (or Bradley-Terry model) can be written in loglinear formulation as

$$\ln E(y_{jj'h}) = \theta_{jj'} + a_h(\lambda_j - \lambda_{j'}) \,, \tag{1}$$

where $y_{jj'h}$ is the observed number of decisions for category h in the comparison between objects j and j', and $E(y_{jj'h})$ is the expected number. The λs describe the location of the objects on a preference scale for all objects and the θs are normalising constants. The weights a_h depend on the response category. If object j is preferred then $h = 0$ and $a_h = 1$, otherwise $h = 1$ and $a_h = -1$.

This model may be extended in several ways. If we allow for an ordinal comparison of objects in the sense of adjacent categories, i.e., object j may be preferred to object j' on a graded or Likert type of scale then model (1) becomes

$$\ln E(y_{jj'h}) = \theta_{jj'} + \gamma_h + a_h(\lambda_j - \lambda_{j'}) \,, \tag{2}$$

where the weights a_h are now defined as $a_h = H - 2h$, $h = 0, \ldots, H$ with $H + 1$ as the total number of response categories. The parameters γ_h represent a bias towards preferring category h. They are usually not of specific interest except in the case of an odd number of possible response categories. Then we can model a 'neutral' response category ($h = H/2$ is integer valued) indicating no preference for either object. This case corresponds to an undecided response, the corresponding γ reflects proneness to indecisiveness in the sample.

Another extension covered in this paper concerns the introduction of subject specific covariates to model the effect of some variables on the general preference scaling of all objects. This is easily achieved by replacing the λs with linear predictor such as $\lambda_j \rightarrow \lambda_j + \lambda_{j\ell}$, where $\lambda_{j\ell}$ reflects the interaction between object j and a certain combination of covariates, cf. Dittrich, Hatzinger and Katzenbeisser (1998) and Dittrich, Hatzinger and Katzenbeisser (2004).

2 Parameter estimation - the IWLS algorithm

Models (1) and (2) are generalised linear models or more specifically, Poisson loglinear models with canonical link function $g(\mu) = \ln \mu$. The parameters can be estimated using the IWLS (iterative weighted least squares) algorithm which we discuss for the specific case of a Poisson loglinear model.

The aim is to find solutions for

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{z} \ , \tag{3}$$

where $\hat{\boldsymbol{\beta}}$ are the estimates for the parameters of interest, $\boldsymbol{X}$ is a matrix of covariates, $\boldsymbol{W}$ is a diagonal matrix of weights, and $\boldsymbol{z}$ is a vector of new random variables Z_i with elements

$$z_i = \eta_i + (y_i - \mu_i)/\mu_i$$

and expectations $E(Z_i) = \eta_i = \sum_{p=1}^{P} x_{ip}\beta_p$. η_i is called the linear predictor for subject i, $E(y_i) = \mu_i$, and $\ln \mu_i = \eta$. Thus if the z_i were known, $\beta_1, \ldots, \beta_P$ could be estimated using weighted least squares, with weights reciprocal to $\mathrm{Var}(Z_i) = \mu_i$.

Unfortunately $z_1, \ldots, z_n$ are not known since neither η_i nor μ_i are known. But an iterative procedure might be as follows:

1. The starting point is the choice of suitable values for the expectation $\hat{\mu}_i^{(0)} = y_i$ and the linear predictor $\hat{\eta}_i^{(0)} = \ln \hat{\mu}_i^{(0)}$. The link function $\ln \mu$ need special consideration, as it is implied that all $y_i > 0$. If not, then the adjustment $\hat{\mu}_i^{(0)} = \max\{y_i, \varepsilon\}$, with small positive ε can be made.

2. Given $\hat{\mu}_i^{(t)}$ and $\hat{\eta}_i^{(t)}$, the adjusted dependent variable is

   $$\hat{z}_i^{(t)} = \hat{\eta}_i^{(t)} + (y_i - \hat{\mu}_i^{(t)})/\hat{\mu}_i^{(t)}$$

   with iterative weight

   $$\hat{w}_i^{(t)} = (\hat{\mu}_i^{(t)})^{-1}, \quad i = 1, \ldots, n.$$

3. In step $t+1$ the estimates for $\boldsymbol{\beta}^{(t+1)}$ are calculated using weighted least squares

   $$\hat{\boldsymbol{\beta}}^{(t+1)} = (\boldsymbol{X}^T \hat{\boldsymbol{W}}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^T \hat{\boldsymbol{W}}^{(t)} \hat{\boldsymbol{z}}^{(t)},$$

   with $\hat{\boldsymbol{W}}^{(t)} = diag\{\hat{w}_i^{(t)}\}$. This leads to $\hat{\eta}_i^{(t+1)} = \boldsymbol{X}\hat{\boldsymbol{\beta}}^{(t+1)}$ and $\hat{\mu}_i^{(t+1)} = \exp(\hat{\mu}_i^{(t+1)})$.

4. Steps 2) and 3) are repeated until a suitable criterion of convergence is fulfilled. Usually the algorithm is stopped when either the βs or the likelihood ratio statistics (or deviances) of two consecutive steps do not differ too much. The deviance for Poisson models is

   $$D(y; \hat{\mu}) = 2 \sum_{i=1}^{n} [y_i \ln (y_i/\hat{\mu}_i) - (y_i - \hat{\mu}_i)] \tag{4}$$

3 The design matrix in loglinear paired comparison models

3.1 The design matrix for models without subject covariates

We first consider the simple case where there are no subject covariates. In general, the design matrix for paired comparison models $\tilde{\boldsymbol{X}}$ has two parts conceptually, one describing the objects, denoted as $\boldsymbol{X}$. This will be called the *model part* of the design matrix. The second, $\boldsymbol{X}_*$, comprises terms which can be regarded as nuisance and which will be called the *nuisance part*. We thus may write

$$\tilde{\boldsymbol{X}} = (\boldsymbol{X}|\boldsymbol{X}_*) \; . \tag{5}$$

The model part $\boldsymbol{X}$ is for model (1)(the simplest possible model) given as

$$\boldsymbol{X} = \boldsymbol{B} \otimes \boldsymbol{a} \, ,$$

The $(\binom{J}{2} \times J)$ matrix $\boldsymbol{B}$ is the paired comparison design matrix

$$\boldsymbol{B} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix} .$$

Each column of this matrix corresponds to one of the J objects, and each row to one of the $\binom{J}{2}$ paired comparisons with order $(O_1, O_2), (O_1, O_3), \dots, (O_{J-1}, O_J)$. $\boldsymbol{a}$ is a column vector with elements $\boldsymbol{a}^T = (1, -1)$ in the case of model (1) representing the scores for the different response categories, i.e. the weights for the λ-parameters in model (1).

The second part of $\tilde{\boldsymbol{X}}$, i.e., $\boldsymbol{X}_*$, consists of dummy variables representing the comparison between each pair of objects. These are generated by a factor having $\binom{J}{2}$ levels, where J is the number of objects. Each row appears as many times as the number of response categories which is two in this case.

Accordingly, the nuisance part $\boldsymbol{X}_*$ of the design matrix for model (1) may be written as

$$\boldsymbol{X}_* = \boldsymbol{I}_{\binom{J}{2}} \otimes \boldsymbol{1}_2 \, ,$$

Example 1 . Consider the case of of three objects O_1, O_2, O_3 and two response categories. The model part $\boldsymbol{X}$ and the nuisance part $\boldsymbol{X}_*$ of the design matrix $\tilde{\boldsymbol{X}}$ would be

$$\boldsymbol{X} = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{X}_* = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} .$$

The three columns of $\boldsymbol{X}$ indicate the three objects, the three columns of $\boldsymbol{X}_*$ indicate the three different comparisons (in the order (O_1, O_2), (O_1, O_3), and (O_2, O_3)), and represent the nuisance parameters θ, generated by a factor defined as having three levels.

It can be seen that $\tilde{\boldsymbol{X}} = (\boldsymbol{X}|\boldsymbol{X}_*)$ does not contain an intercept term. The reason for this will become clear later after a discussion of the computational aspects involved in the nuisance part.

But the model is still overparameterised. To avoid this the parameter for one object (usually the last) is set to zero.

When we proceed to model (2) only minor changes are needed in the specification of $\boldsymbol{X}$ and $\boldsymbol{X}_*$. Instead of two we now have $H+1$ response categories. The nuisance part becomes

$$\boldsymbol{X}_* = \boldsymbol{I}_{\binom{J}{2}} \otimes \mathbf{1}_{H+1}\,. \tag{6}$$

Accordingly, the model part is

$$\boldsymbol{X} = \left(\boldsymbol{B} \otimes \boldsymbol{a}, \mathbf{1}_{\binom{J}{2}} \otimes \boldsymbol{I}_{H+1}\right)\,, \tag{7}$$

where $\boldsymbol{a}$ is now a $((H+1)\times 1)$ column vector $\boldsymbol{a}^T = (H, H-2, H-4, \ldots, -H)$ and the second term specifies the covariates for the parameters γ_h which describe the proneness to the categories h. As already mentioned, in most instances only the γ for the 'undecided' category is of interest, if at all. Then $\boldsymbol{I}_{H+1}$ is reduced to its $(\frac{H}{2}+1)$th column.

3.2 The design matrix for models with subject covariates

In the more general case, we have additional covariates describing subject characteristics. We assume them to be categorical but the generalisation to continous covariates is straightforward, cf. Aitkin and Francis (1992). Let K be the cumulative product of all numbers of levels for all factors, i.e., the number of cells in the contingency table when crossclassifying the subjects according to some categorical explanatory variables. Then the model part X is simply

$$\boldsymbol{X} = \mathbf{1}_K \otimes \left(\boldsymbol{B} \otimes \boldsymbol{a}, \mathbf{1}_{\binom{J}{2}} \otimes \boldsymbol{I}_{H+1}\right)\,, \tag{8}$$

in case of the null model, cf. (7). When fitting certain subject covariates to the data additional columns have to be added to (8). A discussion of this follows later.

Let us consider the nuisance part. Since paired comparison models are multinomial logit models their formulation as loglinear models requires some special considerations. In contrast to loglinear models we distinguish between explanatory factors and a response factor in binomial or multinomial logit models. The loglinear formulation of these explanation–response structured models is achieved by conditioning on the joint marginal distribution of all explanatory factors. This means, we have to include the highest possible interaction terms of all explanatory factors into the model.

Example 2. When we include a two level factor (say SEX) into example 1, then $\boldsymbol{X}_*$ and $\boldsymbol{X}_*$ become

$$\boldsymbol{X} = \begin{pmatrix}
1 & -1 & 0\\ -1 & 1 & 0\\ 1 & 0 & -1\\ -1 & 0 & 1\\ 0 & -1 & -1\\ 0 & 1 & 1\\
1 & -1 & 0\\ -1 & 1 & 0\\ 1 & 0 & -1\\ -1 & 0 & 1\\ 0 & -1 & -1\\ 0 & 1 & 1
\end{pmatrix} \quad \text{and} \quad
\boldsymbol{X}_* = \begin{pmatrix}
1&0&0&1&0&1&0&0&0&0&0\\
1&0&0&1&0&1&0&0&0&0&0\\
0&1&0&1&0&0&1&0&0&0&0\\
0&1&0&1&0&0&1&0&0&0&0\\
0&0&1&1&0&0&0&1&0&0&0\\
0&0&1&1&0&0&0&1&0&0&0\\
1&0&0&0&1&0&0&0&1&0&0\\
1&0&0&0&1&0&0&0&1&0&0\\
0&1&0&0&1&0&0&0&0&1&0\\
0&1&0&0&1&0&0&0&0&1&0\\
0&0&1&0&1&0&0&0&0&0&1\\
0&0&1&0&1&0&0&0&0&0&1
\end{pmatrix}$$

(9)

Again the three columns of $\boldsymbol{X}$ indicate the three objects and the first three column of $\boldsymbol{X}_*$ indicate the three comparisons, both now duplicated for the two levels of the added subject covariate. The next two columns represent these two levels and the remaining six columns represent the interactions. Introducing further covariates leads to analogous expansions. We can immediately see that $\boldsymbol{X}_*$ is overparameterised and a simplifiaction is needed.

Let `TH` denote the factor representing the comparisons and `SEX` the two-level factor introduced into the model. Using Wilkinson-Rogers notation, matrix (9) can be denoted as `TH * SEX = TH + SEX + TH.SEX`. An alternative parameterisation provides the simplification required , i.e., `TH * SEX = TH(1).SEX(1) + TH(2).SEX(1) + ... + TH(3).SEX(2)`. Instead of representing the model by main effects plus additional interaction terms we just use the interaction terms (this model formulation specifies one parameter for each cell of the 3×2 contingency table). Thus, $\boldsymbol{X}_*$ becomes

$$\boldsymbol{X}_* = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

However, we usually are not interested in the marginal distribution of these variables and therefore they can be considered as nuisance and their respective covariates are columns of the nuisance part $\boldsymbol{X}_*$.

So far, we only considered the case of categorical subject covariates. If continuous (and categorical) subject covariates have to be included in the model, $\boldsymbol{X}_*$ consists of the interactions between a factor representing the comparisons and a factor representing the n subjects, i.e., a factor having n levels. In this case, $\boldsymbol{X}_*$ is always given as

$$\boldsymbol{X}_* = \boldsymbol{I}_n \otimes (\boldsymbol{I}_{\binom{J}{2}} \otimes \mathbf{1}_{H+1}) \,. \tag{11}$$

irrespective of the number of (categorical and/or continuous) subject covariates. Unfortunately, $\boldsymbol{X}_*$ can become very large in any case and thus requires special computational attention. For example, for continuous covariates with 200 subjects and 5 objects, we would need to include a nuisance factor with 2000 levels.

GLIM's current implementation (Francis, Green & Payne, 1993) provides a nice solution to the problem via a so-called elimination procedure (implemented as the command `$eliminate` *`model-formula-for-nuisance-terms`*) by which a great deal of computational burdens can be avoided. The basic idea is to use a coding for the interaction terms of the parameters as given by $\boldsymbol{X}_*$ in (10) leading to a submatrix of the weighted SSP matrix $\boldsymbol{X}^T\hat{\boldsymbol{W}}\boldsymbol{X}$ which is diagonal and easy to invert. We then can take advantage of using rules for inverting partitioned matrices, as presented in the following.

4 Inverting partitioned matrices

The crucial part of algorithms for calculating estimates in (generalised) linear models is the inversion of the (weighted) SSP matrix $\boldsymbol{A} = \tilde{\boldsymbol{X}}^T \hat{\boldsymbol{W}} \tilde{\boldsymbol{X}}$. Due to the special symmetric structure of some problems it is feasible to make use of partition rules where only properly defined submatrices have to be inverted. The basic idea is to avoid inversion of the full SSP matrix $\boldsymbol{A}$, and to use a numerical technique to obtain the inverse of only the part of $\boldsymbol{A}$ in which we are interested (cf., Healy, 2000).

Let $\boldsymbol{A}$ and $\boldsymbol{A}^{-1}$ (assumed to exist) be partitioned as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix}, \qquad \text{and} \qquad \boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{A}^{11} & \boldsymbol{A}^{12} \\ \boldsymbol{A}^{21} & \boldsymbol{A}^{22} \end{pmatrix},$$

From $AA^{-1} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$ we obtain

$$A_{21}A^{11} + A_{22}A^{21} = 0 \qquad \rightarrow \qquad A^{21} = -(A_{22}^{-1}A_{21}A^{11}) \tag{12}$$

and

$$A_{11}A^{11} + A_{12}A^{21} = I \qquad \text{inserting (12)} \rightarrow \qquad A^{11} = (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} \tag{13}$$

From $A^{-1}A = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$, we obtain

$$A^{11}A_{12} + A^{12}A_{22} = 0 \qquad \rightarrow \qquad A^{12} = -(A^{11}A_{12}A_{22}^{-1}) \tag{14}$$

and

$$A^{21}A_{12} + A^{22}A_{22} = I \qquad \text{inserting (14)} \rightarrow \qquad A^{22} = (A_{22}^{-1} - A^{21}A_{12}A_{22})^{-1} \tag{15}$$

To invert the whole matrix $\boldsymbol{A}$ we need the inverses of $\boldsymbol{A}_{22}$ and $(\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})$. The necessary steps are

- $\boldsymbol{A}_{22}^{-1}$
- $(\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})^{-1}$ yields $\boldsymbol{A}^{11}$, cf. (13)
- from these both we get $\boldsymbol{A}^{21} = -(\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}^{11})$, cf. (12)
- and finally $\boldsymbol{A}^{22} = \boldsymbol{A}_{22}^{-1} - \boldsymbol{A}^{21}\boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}$, cf. (15)

5 Partitioning the paired comparison model SSP matrix

According to the previous two sections the weighted SSP matrix $\boldsymbol{A} = \tilde{\boldsymbol{X}}^T \hat{\boldsymbol{W}} \tilde{\boldsymbol{X}}$ for paired comparisons may be partitioned as follows (for ease of notation we use $\boldsymbol{W}$ instead of $\hat{\boldsymbol{W}}$).

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}_* \\ \boldsymbol{X}_*^T\boldsymbol{W}\boldsymbol{X} & \boldsymbol{X}_*^T\boldsymbol{W}\boldsymbol{X}_* \end{pmatrix}$$

In this problem we are interested in $\boldsymbol{A}_{11}$ as it corresponds to the model part of the paired comparison design matrix. We will see that $\boldsymbol{A}_{22}$ is of diagonal form and thus the inverse of $\boldsymbol{A}_{22}$ can easily be calculated. Submatrix $\boldsymbol{A}_{22}$ (for example 2) is

$$\begin{aligned}\boldsymbol{A}_{22} &= \boldsymbol{X}_*^{\boldsymbol{T}} \boldsymbol{W} \boldsymbol{X}_* \\ &= \begin{pmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} w_{011} & 0 & 0 & 0 & \dots & 0 \\ 0 & w_{111} & 0 & 0 & & 0 \\ 0 & 0 & w_{021} & 0 & & 0 \\ 0 & 0 & 0 & w_{121} & & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & w_{132} \end{pmatrix} \cdot \boldsymbol{X}_* \\ &= \begin{pmatrix} w_{+11} & 0 & 0 & 0 & \dots & 0 \\ 0 & w_{+21} & 0 & 0 & & 0 \\ 0 & 0 & w_{+12} & 0 & & 0 \\ 0 & 0 & 0 & w_{+22} & & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & w_{+32} \end{pmatrix}\end{aligned} \tag{16}$$

The $w_{hk\ell}$, are weights corresponding to the cells of the complete contingency table, where $h = 0, \dots, 1$ denote the response categories, $k = 1, \dots, 3$ denote the comparisons and $\ell = 1, 2$ the levels of the subject covariate `SEX`. Submatrix $\boldsymbol{A}_{22}$ is diagonal with elements corresponding to the sums of $w_{hk\ell}$ over all response categories.

The transpose of $\boldsymbol{X}_*$ has a nice property when multiplying from left with a properly dimensioned matrix or vector, $\boldsymbol{Q}$ say. The resulting matrix or vector $\boldsymbol{Q}^* = \boldsymbol{X}_*^T \boldsymbol{Q}$ consists of elements which result from the sum of n consecutive elements of $\boldsymbol{Q}$ per column, where n is the number of consecutive ones in a row of ${\boldsymbol{X}_*}^T$. To illustrate this property, let $\boldsymbol{X}_*^T$ and $\boldsymbol{Q}$ be

$$\boldsymbol{X}_*^{\boldsymbol{T}} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{Q} = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \\ q_{31} & q_{32} \\ q_{41} & q_{42} \end{pmatrix}$$

The resulting $\boldsymbol{Q}^*$ is

$$\boldsymbol{Q}^* = \begin{pmatrix} q_{11} + q_{21} & q_{12} + q_{22} \\ q_{31} + q_{41} & q_{32} + q_{42} \end{pmatrix}$$

This property can be used to compute all quantities where left multiplication of $\boldsymbol{X}_*^{\boldsymbol{T}}$ is involved without having to set up the possibly huge $\boldsymbol{X}_*$. We will call this the *summation property* of the nuisance part.

In fact, we can take advantage of this property when looking at $\boldsymbol{A}_{21}$. Since $\boldsymbol{A}_{21} = \boldsymbol{X}_*^{\boldsymbol{T}} \boldsymbol{W} \boldsymbol{X}$ we get

$$\boldsymbol{A}_{21} = \begin{pmatrix} w_{+11}(x_{11} + x_{21}) & w_{+11}(x_{12} + x_{22}) & \dots \\ w_{+21}(x_{31} + x_{41}) & w_{+21}(x_{32} + x_{42}) & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

where $\boldsymbol{X}_*^{\boldsymbol{T}}$ and $\boldsymbol{W}$ are as above.

Conversely, $\boldsymbol{X}_*$ has what we will call *duplication property* of the nuisance part. When multiplying from left with a suitably dimensioned vector or matrix $\tilde{\boldsymbol{Q}}$ the in the resulting matrix or vector $\tilde{\boldsymbol{Q}}^* = \boldsymbol{X}_*\tilde{\boldsymbol{Q}}$ each row is of the original $\tilde{\boldsymbol{Q}}$ is repeated n times, when n is the number of consecutive ones in a column of $\boldsymbol{X}_*$. This corresponds to

$$\tilde{\boldsymbol{Q}}^* = \tilde{\boldsymbol{Q}} \otimes \mathbf{1}_n \,. \tag{17}$$

The other two submatrices will not be considered here in more detail. Since $\boldsymbol{A}$ is symmetric, $\boldsymbol{A}_{12} = \boldsymbol{A}_{21}^T$ and possible simplifications for $\boldsymbol{A}_{11}$ have not yet been investigated in this project.

6 Estimating the model parameters and standard errors

Instead of using equation (3)for estimating the model parameters we use a different approach (cf. Healy, 1986). If in a linear regression model the design matrix $\boldsymbol{X}$ is augmented by adding the response vector $\boldsymbol{y}$ as a column (the first, say) then the least squares estimates can be easily obtained from $\left[(\boldsymbol{y}|\boldsymbol{X})^T(\boldsymbol{y}|\boldsymbol{X})\right]^{-1}$. The same approach may be applied to the IWLS algorithm. The vector $\hat{\boldsymbol{\beta}}$ can be extracted from the first columns of the inverse of the augmented SSP matrix

$$\boldsymbol{A}^{-1} = \left[(\boldsymbol{z}|\tilde{\boldsymbol{X}})^T\hat{\boldsymbol{W}}(\boldsymbol{z}|\tilde{\boldsymbol{X}})\right]^{-1} \tag{18}$$

by

$$\hat{\beta}_j = -\frac{1}{a_{11}}\, a_{j+1,1} \,, \tag{19}$$

where $j = 1, \ldots, J$, and J is the number of all parameters, for the model part and the nuisance part, respectively, and $\boldsymbol{z}$ is the adjusted dependent variate of section 2. Thus, we only have to explicitely compute the two submatrices $\boldsymbol{A}^{11}$ and $\boldsymbol{A}^{21}$. We neither need (the possibly huge) $\boldsymbol{A}^{22}$ nor $\boldsymbol{A}^{12}$, since $\boldsymbol{A}^{12} = (\boldsymbol{A}^{21})^T$.

The estimated variances $\hat{\sigma}^2_{\hat{\beta}}$ of the estimates $\hat{\beta}$ are contained in the diagonal of the inverse of the original SSP matrix. After convergence we can again apply rules for inverting partitioned matrices to easily calculate $\left[\boldsymbol{X}^T\hat{\boldsymbol{W}}\boldsymbol{X}\right]^{-1}$ from $\boldsymbol{B}^{-1} = \left[(\boldsymbol{z}|\tilde{\boldsymbol{X}})^T\hat{\boldsymbol{W}}(\boldsymbol{z}|\tilde{\boldsymbol{X}})\right]^{-1}$. The necessary calculations are particularly simple since we do not need the standard errors for the nuisance parameters. If, according to section 4, we partition the whole augmented SSP matrix $\boldsymbol{B}$ (from which we already know the inverse) such that $\boldsymbol{B}_{22}$ corresponds to $\boldsymbol{z}^T w_{11}\boldsymbol{z}$, $\boldsymbol{B}_{11} = \boldsymbol{A}_{11}$ corresponds to the model part $\boldsymbol{X}^T\hat{\boldsymbol{W}}\boldsymbol{X}$, and $\boldsymbol{B}_{12}$ corresponds to the weighted crossproducts of $\boldsymbol{z}$ and $\boldsymbol{X}$, then we need the diagonal elements of $\boldsymbol{B}_{11}^{-1}$, i.e.,

$$\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_{\hat{\beta}} = \mathrm{diag}(\boldsymbol{B}^{11} - \boldsymbol{B}^{12}(\boldsymbol{B}^{22})^{-1}\boldsymbol{B}^{21}) \tag{20}$$

or

$$\hat{\sigma}^2_{\hat{\beta}_j} = b_{jj} + \sum_j b^2_{j+1,1}\frac{1}{b_{11}} \,,$$

where b_{ij} are elements of $\boldsymbol{B}^{-1}$ and j is the index of the model parameters. Note, that $(\boldsymbol{B}^{22})^{-1} = \frac{1}{b_{11}}$ and (20) follows from (15) and (14).

7 The R code for fitting paired comparison models

This section describes the function `fitpc` for fitting the paired comparison model. The function takes a number of arguments - all are necessary and none can be defaulted. These are:

`p`	Number of model parameters (number of columns of $\boldsymbol{X}$)
`q`	Number of nuisance parameters (number of columns of $\boldsymbol{X}_*$)
`ncat`	Number of response categories ($H + 1$)
`Xmodel`	Design matrix $\boldsymbol{X}$ for the model part. The number of rows is the number of comparisons times `ncat` (times the cumulative product of the number of levels for all subject factors or times the number of subjects in case of categorical or continuos subject covariates, respectively), the number of columns is `p`, i.e., the number of objects - 1 plus additional covariates (usually category parameters and interaction terms between objects and subject variables).
`y`	Vector of responses (same length as number of rows of `Xmodel`).

We describe the internal workings of the function `fitpc`. Illustrations of the use of the function can be found in Appendix A.

First we initialise the IWLS loop. The `p` (model) plus `q` (eliminated) elements of the parameter vector `b` and the deviance `dev` are set to zero. The fitted values $\hat{y}_i$ (coded as `fv` and the linear predictor $\ln(\hat{y}_i) = \eta_i = \sum_{p=1}^{P} x_{ip}\beta_p$ (coded as `eta`) are initialised using the observation y_i and a small constant is added if $y_i = 0$.

```
> b <- rep(0, p + q)
> dev <- 0
> fv <- y
> fv[y == 0] <- 1e-10
> eta <- log(fv)
```

Finally we define some quantities for assessing convergence.

```
> conv.dev.eps <- 1e-04
> maxiter <- 25
> iter <- 0
```

At this point, the main loop starts. All statements that follow are within this loop.

```
> repeat {
+     statements
+ }
```

The estimates `b.old` and deviance `dev.old` of the previous iteration are updated with the new values `b` and `dev`.

```
> b.old <- b
> dev.old <- dev
```

Part 2 of the iteration cycle calculates the adjusted dependent variate which in the case of Poisson models is $z_i = \eta_i + (y_i - \mu_i)/\mu_i$.

```
> z <- eta + (y - fv)/fv
```

The critical part of the algorithm is to find numerical solutions for the normal equations involving heavy calculations on (very) large though sparse matrices. The next piece of code implements the ideas as described in sections 4 to 6.

First of all we need the inverse $\boldsymbol{A}_{22}^{-1} = \left(\boldsymbol{X}_*^{\boldsymbol{T}} \boldsymbol{W} \boldsymbol{X}_*\right)^{-1}$. The $(n \times n)$ weight matrix $\boldsymbol{W}$ is diagonal with elements $w_{ii} = \hat{y}_i$ (in the Poisson case). The summation property of $\boldsymbol{X}_*^T$ allows to use the function `colSums()` to obtain the diagonal elements of $\boldsymbol{A}_{22}$ without multiplications (as illustrated in (16) in section 5). In fact, we redefine a vector into a matrix such that each column contains the `ncat` elements for which we need the sums. Since $\boldsymbol{A}_{22}^{-1}$ is diagonal we store the elements into a vector (i.e., `a22inv` [1]).

```
> a22inv <- 1/colSums(matrix(fv, nr = ncat))
```

For $\boldsymbol{A}^{11}$ we additionally need two other submatrices of the (enlarged) $\boldsymbol{A}$ of section 6, i.e., $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_{21}$. First we compute $\boldsymbol{A}_{21} = \boldsymbol{X}_*^T \boldsymbol{W}(\boldsymbol{y}|\boldsymbol{X})$. The basic $\boldsymbol{X}$ is the paired comparison design matrix (7) and contains only object covariates as columns. For models with subject covariates we have $\boldsymbol{X}$ as in (8) with additional columns for the subject effects. $\boldsymbol{X}$ is coded as `Xmodel`. To obtain the $\boldsymbol{W}(\boldsymbol{z}|\boldsymbol{X})$ part of $\boldsymbol{A}_{21}$ we add the adjusted dependent variate `z` as the first column to $\boldsymbol{X}$ and multiply with $\boldsymbol{W}$ (we do not need the diagonal matrix $\boldsymbol{W}$ because pointwise multiplication with a column vector containing the diagonal elments yields the same result). Finally we again take advantage of the summation property of the nuisance part as above. Each row of $\boldsymbol{W}(\boldsymbol{z}|\boldsymbol{X})$ is passed to the function `colSums()` in turn. The result is redefined to form a (properly dimensioned) matrix (i.e., `A21`).

```
> WX <- fv * cbind(z, Xmodel)
> A21 <- matrix(colSums(matrix(WX[, 1:(p + 1)], nr = ncat)), nc = p +
+     1)
```

The part corresponding to the model is straightforward, $\boldsymbol{A}_{11} = (\boldsymbol{y}|\boldsymbol{X})^T \boldsymbol{W}(\boldsymbol{y}|\boldsymbol{X})$.

```
> A11 <- crossprod(cbind(z, Xmodel), WX)
```

We now have all ingredients for $\boldsymbol{A}^{11}$. However, for $\boldsymbol{A}^{11} = (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}$ we need $A_{22}^{-1}A_{21}$. But this is also required in $\boldsymbol{A}^{21} = -(\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}^{11})$ (and in $\boldsymbol{A}^{22} = \boldsymbol{A}_{22}^{-1} - \boldsymbol{A}^{21}\boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1} = \boldsymbol{A}_{22}^{-1} - \boldsymbol{A}^{21}(A_{22}^{-1}A_{21})^T$, if we would like to compute all submatrices of $\boldsymbol{A}^{-1}$). Therefore we first calculate this quantity multiplied by -1 and then appropriately adjust the other operations.

```
> A22inv.A21 <- -a22inv * A21
> A.11 <- solve(A11 + crossprod(A21, A22inv.A21))
> A.21 <- A22inv.A21 %*% A.11
```

We can now again tidy up some workspace.

[1] Please note, that we use the following coding. Vectors and scalars are lowercase, matrices start with an uppercase character. The submatrices of $\boldsymbol{A}^{-1}$ are denoted with a dot as opposed to that of $\boldsymbol{A}$, e.g., the name for $\boldsymbol{A}^{21}$ is `A.21` wheras the name for $\boldsymbol{A}_{21}$ is `A21`.

```
> rm(A11, A21, a22inv, A22inv.A21, WX, z)
```

To obtain the updated estimates for $\boldsymbol{\beta}$ (coded as `b`) we can now simply extract them from the first columns of $\boldsymbol{A}^{11}$ and $\boldsymbol{A}^{21}$ according to (19).

```
> pseudo.rss <- -1/A.11[1, 1]
> b.model <- pseudo.rss * A.11[1, 2:(p + 1)]
> b.elim <- pseudo.rss * A.21[, 1]
> b <- c(b.model, b.elim)
```

The standard errors of the parameter estimates (`se`) for the model part are obtained in accordance to (20).

```
> se <- sqrt(diag(A.11[2:(p + 1), 2:(p + 1)]) + pseudo.rss * A.11[2:(p +
+     1), 1] * A.11[2:(p + 1), 1])
```

To save workspace we remove unrequired larger structures.

```
> rm(A.11, A.21)
```

Finally we have to calculate the updated $\hat{y}_i$s and deviance. Computing the linear predictor for the model part (`eta.model`) is straightforward, the linear predictor for the nuisance part $\hat{\boldsymbol{\eta}}_* = \boldsymbol{X}_*\hat{\boldsymbol{\beta}}_*$ (coded as `eta.elim`) is obtained taking advantage of the duplication property of the nuisance part of the design matrix as described in (17).

```
> eta.model <- crossprod(t(Xmodel), b.model)
> eta.elim <- rep(b.elim, rep(ncat, q))
> eta <- as.vector(eta.model + eta.elim)
> fv <- exp(eta)
```

For the deviance (`dev`) we only sum up the increments where $y_i > 0$.

```
> dev <- sum(2 * (y * log(ifelse(y == 0, 1, y/fv)) - (y - fv)))
```

To complete an iteration cycle we check for convergence or stop the iterations if the number of cycles exceeds the predefined value of `maxiter`. Currently, we define the algorithm to be converged (in analogy to GLIM) if the relative difference of the deviances of the previous and current cycle falls below a predefined value (`conv.dev.eps`).

```
> iter <- iter + 1
> GLIM.dev.diff <- (dev - dev.old)/dev
> GLIM.converged.dev <- abs(GLIM.dev.diff) < conv.dev.eps
> if (iter > maxiter || GLIM.converged.dev) break
```

This ends the iteration loop. After convergence the function prints the parameter estimates, standard errors, the scaled deviance and the degrees of freedom of the model.

```
> cat("------------------------------\n\n")
> for (i in 1:p) {
+     writeLines(paste("   ", sprintf("%10.5f", b[i]), "   ",
+         sprintf("%10.5f", se[i])))
+ }
> cat("\n")
> writeLines(paste("Deviance:    ", sprintf("%10.5f", dev)))
> writeLines(paste("Residual df =", sprintf("%10.0f", length(y) - p - q + 1)))
> cat("------------------------------\n\n")
```

8 Concluding remarks

This technical report is primarily concerned with efficient fitting of the Bradley Terry paired comparison model. However, the algorithmic method can be used for many application where a large number of nuisance parameters need to be fitted as part of the linear model. If these nuisance parameters can be represented by a factor with a large number of levels, then the algorithm presented here can be used. Examples of such problems are numerous. The Poisson representation of the multinomial logit model is a relatively well-known example but other examples include conditional logistic regression , the Rasch model, and the piecewise exponential proportional hazards model. All of these can be fitted as Poisson log-linear models using generalised linear modelling software, and this algorithmic method speeds up computiation substantially.

References

Aitkin, M. & Francis, B. (1992). Fitting the multinomial logit model with continuous covariates in GLIM. *Computational Statistics and Data Analysis*, *14*, 89–97.

Dittrich, R., Hatzinger, R. & Katzenbeisser, W. (1998). Modelling the effect of subject-specific covariates in paired comparison studies with an application to university rankings. *Applied Statistics*, *47*, 511–525.

Dittrich, R., Hatzinger, R. & Katzenbeisser, W. (2004). A log-linear approach for modelling ordinal paired comparison data on motives to start a PhD programme. *accepted for Statistical Modelling*.

Francis, B., Green, M. & Payne, C. (1993). *The GLIM system. Release 4 manual.* Oxford University Press.

Healy, M. J. R. (2000). *Matrices for statistics.* Oxford University Press.

A The R function and some simple examples

The complete function as described in Section 7 is:

```
fitpc<-function(p,q,ncat,Xmodel,y)
{
b <- rep(0, p + q)
dev <- 0
fv <- y
fv[y == 0] <- 1e-10
eta <- log(fv)
conv.dev.eps <- 1e-04
maxiter <- 25
iter <- 0
repeat {
      b.old <- b
      dev.old <- dev
      z <- eta + (y - fv)/fv
      a22inv <- 1/colSums(matrix(fv, nr = ncat))
      WX <- fv * cbind(z, Xmodel)
      A21 <- matrix(colSums(matrix(WX[, 1:(p + 1)], nr = ncat)), nc = p + 1)
      A11 <- crossprod(cbind(z, Xmodel), WX)
      A22inv.A21 <- -a22inv * A21
      A.11 <- solve(A11 + crossprod(A21, A22inv.A21))
      A.21 <- A22inv.A21 %*% A.11
      rm(A11, A21, a22inv, A22inv.A21, WX, z)
      pseudo.rss <- -1/A.11[1, 1]
      b.model <- pseudo.rss * A.11[1, 2:(p + 1)]
      b.elim <- pseudo.rss * A.21[, 1]
      b <- c(b.model, b.elim)
      se <- sqrt(diag(A.11[2:(p + 1), 2:(p + 1)]) + pseudo.rss *
            A.11[2:(p + 1), 1] * A.11[2:(p + 1), 1])
      rm(A.11, A.21)
      eta.model <- crossprod(t(Xmodel), b.model)
      eta.elim <- rep(b.elim, rep(ncat, q))
      eta <- as.vector(eta.model + eta.elim)
      fv <- exp(eta)
      dev <- sum(2 * (y * log(ifelse(y == 0, 1, y/fv)) - (y - fv)))
      iter <- iter + 1
      GLIM.dev.diff <- (dev - dev.old)/dev
      GLIM.converged.dev <- abs(GLIM.dev.diff) < conv.dev.eps
      if (iter > maxiter || GLIM.converged.dev) break
}
cat("-----------------------------\n\n      estimate           s.e.\n")
for (i in 1:p) {
    writeLines(paste("   ", sprintf("%10.5f", b[i]), "   ",
        sprintf("%10.5f", se[i])))  }
writeLines(paste("\nDeviance:    ", sprintf("%10.5f", dev)))
writeLines(paste("Residual df =", sprintf("%10.0f", length(y) - p - q + 1)))
cat("-----------------------------\n\n")
}
```

To illustrate the usage of the function we provide three small examples: i) a simple paired comparison model, ii) a paired comparison model with one categorical subject covariate, and iii) a loglinear formulation of a simple logit problem.

A.1 Example: a simple paired comparison model

The first example (fictitious data) describes an experiment, where 3 objects O_j have been compared in a paired comparison experiment. Aim of the analysis is to find the location of the objects on a preference scale. The response frequencies for comparison (O_1, O_2) are 10 in preference for O_1, 20 for O_2. The respective numbers for comparisons (O_1, O_3) are 30 and 40, and finally for (O_2, O_3) 50 and 60.

The following table shows the setup of the data structure:

y	o1	o2	nuisance		
10	1	-1	1	0	0
60	-1	1	1	0	0
30	1	0	0	1	0
40	-1	0	0	1	0
20	0	1	0	0	1
50	0	-1	0	0	1

The R code to fit the model is

```
> y  <- c(10,60,30,40,20,50)
> o1 <- c(1,-1,1,-1,0,0)
> o2 <- c(-1,1,0,0,1,-1)
> p <-2
> q <-3
> ncat <-2
> Xmodel <- cbind(o1,o2)
>
> fitpc(p,q,ncat,Xmodel,y)
```

The output provided by `fitpc()` is

```
-----------------------------

      estimate           s.e.
      -0.51012        0.10749
      -0.10174        0.10115

Deviance:       26.75900
Residual df =          2
-----------------------------
```

The first line of the estimates correponds to O_1 and the second to O_2 (O_3 is set to zero to avoid overparameterisation).

A.2 Example: a simple paired comparison model with one categorical subject covariate

The second example is analogous to the first one but a categorical subject covariate with two categories (say `sex`) is included. The setup of the data structure is

y	o1	o2	o1:sex2	o2:sex2	nuisance					
25	1	-1	0	0	1	0	0	0	0	0
5	-1	1	0	0	1	0	0	0	0	0
5	1	0	0	0	0	1	0	0	0	0
25	-1	0	0	0	0	1	0	0	0	0
15	0	1	0	0	0	0	1	0	0	0
15	0	-1	0	0	0	0	1	0	0	0
35	1	-1	1	-1	0	0	0	1	0	0
5	-1	1	-1	1	0	0	0	1	0	0
25	1	0	1	0	0	0	0	0	1	0
15	-1	0	-1	0	0	0	0	0	1	0
5	0	1	0	1	0	0	0	0	0	1
35	0	-1	0	-1	0	0	0	0	0	1

The R code to fit the model is

```
y  <- c(25,5,5,25,15,15,35,5,25,15,5,35)
o1 <- c(1,-1,1,-1,0,0)
o1 <- c(o1,o1)
o2 <- c(-1,1,0,0,1,-1)
o2 <- c(o2,o2)
sex2<- c(0,0,0,0,0,0,1,1,1,1,1,1)
Xmodel<-cbind(o1,o2)
Xmodel<-cbind(Xmodel, Xmodel * sex2)
p<-4
q<-6
ncat<-2

fitpc(p,q,ncat,Xmodel,y)
```

The output would be

```
-----------------------------

      estimate           s.e.
      -0.23410        0.15521
      -0.46821        0.16118
       0.44201        0.21319
      -0.40888        0.24186

Deviance:       20.45180

Residual df =          3
-----------------------------
```

The lines for the estimates correspond to the terms `o1`, `o2`, `o1:sex2`, and `o2:sex2`.

A.3 Example: a loglinear representation of a logit model

The example is from Francis et al. (1993), pages 451ff. The data are from a study of infant respiratory disease, namely the number of children developing/not developing bronchitis or pneumonia (`ill`) in their first year of life by sex (`sex`) and type of feeding (`food`). In a loglinear formulation of this logit problem the data structure for a main effects model (`sex+food`)is

y	ill	ill:sex2	ill:food2	ill:food3	nuisance (sex:food)					
77	1	0	0	0	1	0	0	0	0	0
381	0	0	0	0	1	0	0	0	0	0
19	1	0	1	0	0	1	0	0	0	0
128	0	0	0	0	0	1	0	0	0	0
47	1	0	0	1	0	0	1	0	0	0
447	0	0	0	0	0	0	1	0	0	0
48	1	1	0	0	0	0	0	1	0	0
336	0	0	0	0	0	0	0	1	0	0
16	1	1	1	0	0	0	0	0	1	0
111	0	0	0	0	0	0	0	0	1	0
31	1	1	0	1	0	0	0	0	0	1
433	0	0	0	0	0	0	0	0	0	1

The R code to fit the main effects model (`sex+food`) is

```
> y <- c(77,381,19,128,47,447,48,336,16,111,31,433)
> ill  <- c(1,0,1,0,1,0,1,0,1,0,1,0)
> sex2 <- c(0,0,0,0,0,0,1,1,1,1,1,1)
> food2<- c(0,0,1,1,0,0,0,0,1,1,0,0)
> food3<- c(0,0,0,0,1,1,0,0,0,0,1,1)
>
> Xmodel<-cbind(ill,ill*sex2,ill*food2,ill*food3)
> p<-ncol(Xmodel)
> q<-6
> ncat<-2
>
> fitpc(p,q,ncat,Xmodel,y)
```

Please note that in the loglinear formulation of logit models `ncat` is the number of categories of the response variable, i.e., `ill` in this example. The output from `fitpc()` is

```
-----------------------------

      estimate           s.e.
      -1.61270        0.11241
      -0.31255        0.14104
      -0.17254        0.20558
      -0.66929        0.15301

Deviance:        0.72192
Residual df =          3
-----------------------------
```

Again the estimates are given in the same order as the columns of `Xmodel`, i.e., `ill`, `ill:sex2`,`ill:food2`, and `ill:food3`.