Psychology Science Quarterly, Volume 51, 2009 (2), pp. 216 - 242

Fitting loglinear Bradley-Terry models (LLBT) for paired comparisons using the R package prefmod

REGINA DITTRICH & REINHOLD HATZINGER[1]

Abstract

This paper aims at introducing the R package prefmod (Hatzinger, 2009) which allows the user to fit various models to paired comparison data. These models give estimated overall rankings of objects or items where each subject (respondent/judge) makes one or more comparisons between pairs of objects (items). The focus is on the loglinear Bradley-Terry (LLBT) model, the loglinear formulation of the Bradley-Terry(-Luce) model, both assuming independence between comparisons. Five types of data structures are covered: (i) simple paired comparisons, (ii) paired comparisons including an undecided category, (iii) categorical subject covariates (for estimating different overall rankings for different subject groups), (iv) object covariates for reparameterizing objects, and (v) order (position) effects. Additionally, the discussion briefly addresses other response formats such as ratings and (partial) rankings.

Key words: Bradley-Terry(-Luce) model; loglinear Bradley-Terry model; prefmod; paired comparison; preference scale

[1] Correspondence should be addressed to Reinhold Hatzinger, PhD, Department of Statistics and Mathematics, WU (Vienna University of Economics and Business), Augasse 2-6, 1090 Vienna, Austria;
email: reinhold.hatzinger@wu-wien.ac.at

1. The basic Bradley-Terry model

The paired comparison method is a very old psychometric technique that has been used by generations of researchers in various fields. It is a well-developed method of ordering attributes or characteristics of a given set of items. With this method items are presented in pairs to respondents/judges who for each pair select the item that best satisfies the specified judgment criterion. The result of these paired choices is a rank-ordering of the items on an interval scale.

One of the most prominent and well-known models that covers such situations is due to Bradley and Terry (1952). The basic Bradley-Terry (BT) model defines the probability that object j (O_j) is preferred to object k (O_k) in a given comparison (jk) as

$$p(O_j \succ O_k \mid \pi_j, \pi_k) = \frac{\pi_j}{\pi_j + \pi_k} \qquad \textit{for all } j \neq k, \tag{1}$$

where π_j and π_k are non-negative parameters describing the location of the objects on the preference scale. The number of all possible pairs for a set of J objects is given by $\binom{J}{2} = J(J-1)/2$. For instance, given $J = 4$ objects labeled 1,2,3,4 there are 6 pairs of items, i.e., (12),(13),(14),(23),(24),(34). In each pair of items either the first or the second item can be preferred. For example, if item 1 is preferred in the comparison (12) this will be denoted by (12)1 and if item 2 is preferred by (12)2. For the moment, we consider 2 possible responses in each comparison.

In psychometric contexts, the Bradley-Terry model is often called BTL model due to the relation to Luce's (1959) choice axiom. One consequence of this is the assumption of independence from irrelevant alternatives, i.e., the probability of selecting one item over another from a pool (a choice set) of many items is not affected by the presence or absence of other items in the pool. A second consequence of Luce's axiom is that a (latent) response strength for item O_j, i.e., π_j, associated with a certain response to O_j, implies that the response probabilities are proportional to response strengths. This representation is closely related to logit analysis used in statistics. The Bradley-Terry model can be seenc as a special case (restricted to two-element choice sets) of Luce's model though formulated earlier. The relation to Item Response Theory Models (IRT) and logit regression is easily seen when we use a slightly different presentation of the Bradley-Terry model,

$$p(O_j \succ O_k \mid \beta_j, \beta_k) = p_{(jk)} = \frac{\exp(\beta_j - \beta_k)}{1 + \exp(\beta_j - \beta_k)}. \tag{2}$$

If we consider some of the objects j to represent subjects and the other objects k to represent items, then the comparison (jk) is that of a person against an item as specified in the Rasch model. In fact, the Rasch model can be seen as an incomplete paired comparison model since comparison within the set of items and within the set of subjects do not occur. There is also an obvious relation to the linear logistic test model (LLTM), when objects are reparameterized according to certain characteristics (for a detailled discussion of these common properties see, e.g., Fischer & Tanzer, 1994).

Identifiability of (2) and also of (1) requires a constraint such as $\beta_J = 0$ or $\pi_J = 1$. The model describes $\binom{J}{2}$ such probabilites with $J-1$ parameters, thus the degrees of freedom are $\binom{J}{2}-(J-1)$. To fit a logit model we assume that each pair of random variables $Y_{(jk)j}$ (the number of preferences for object j) and $Y_{(jk)k}$ (the number of preferences for object k) is an independent binomial variate with parameters $p_{(jk)}$ and $n_{(jk)} = Y_{(jk)j} + Y_{(jk)k}$ with realisations (observed counts) $y_{(jk)j}$ and $y_{(jk)k}$. For each $\ln(p_{(jk)j}/p_{(jk)k}) = \beta_j - \beta_k$ we have to set up J explanatory variables corresponding to the coefficients of the β s, i.e., the variable for β_j is 1 and the variable for β_k is -1, all other variables are 0 for that comparison. This gives a $\binom{J}{2}-(J-1)$ design matrix where one of the J columns has to be left out for identifiability reasons. Standard software for logistic regression can be used to compute the parameter estimates (see, e.g., Tutz, 1989, Agresti, 1990, p.371).

1.1 The loglinear Bradley-Terry model (LLBT) for two response categories

Alternatively, the BT-model can be fitted as a loglinear model (see, e.g., Sinclair, 1982; Agresti, 1990; Dittrich, Hatzinger & Katzenbeisser, 1998). Fienberg and Larntz (1976) give a detailled description of the original logistic formulation of the BT-model (based on the binomial distribution) and its loglinear representation (based on the Poisson distribution) and discuss various advantages of the loglinear form (see also Appendix). Let again $n_{(jk)}$ be the number of comparisons between object j and object k and let $Y_{(jk)j}$ be the number of preferences for object j and $Y_{(jk)k}$ the number of preferences for object k. The outcome of a paired comparison experiment can also be regarded as a $\binom{J}{2}\times J$ incomplete two-dimensional *objectpair* $\times$ *decision* contingency table. For instance, for three objects the corresponding contingency table is given as in Table 1.

Table 1:
Contingency table for *objectpair* $\times$ *decision*

comparison	*decision* *for object 1*	*for object 2*	*for object 3*	*total number of comparisons*
(12)	$y_{(12)1}$	$y_{(12)2}$	---	$n_{(12)}$
(13)	$y_{(13)1}$	---	$y_{(13)3}$	$n_{(13)}$
(23)	---	$y_{(23)2}$	$y_{(23)3}$	$n_{(23)}$

The distribution of the random variables $Y_{(jk)j}$ and $Y_{(jk)k}$ is now assumed to be Poisson. Conditional on fixed $n_{(jk)} = Y_{(jk)j} + Y_{(jk)k}$, the $(Y_{(jk)j}, Y_{(jk)k})$'s follow a binomial (or more generally a multinomial) distribution. The expected number of preferences of object j to object k is denoted by $m_{(jk)j}$ and given by $n_{(jk)}\,p_{(jk)j}$.

Using the respecification for the $p_{(jk)j}$'s suggested by Sinclair (1982) and standard notation for loglinear models for contingency tables, the basic form of the loglinear Bradley-Terry model (LLBT) is given by the equations

$$\ln m_{(jk)j} = \mu_{(jk)} + \lambda_j^O - \lambda_k^O$$
$$\ln m_{(jk)k} = \mu_{(jk)} - \lambda_j^O + \lambda_k^O$$

where the μ's are nuisance parameters and may be interpreted as interaction parameters representing the objects involved in the respective comparisons, fixing therefore the corresponding $n_{(jk)}$ marginal distributions. The λ^O's represent object-parameters and are related to the π's in (1) by $\ln\pi = 2\lambda^O$.

This model is restricted to two possible outcomes for each comparison (jk): either object j is preferred and object k not preferred or object k is preferred and object j is not preferred.

Table 2 shows the design structure where the elements (columns) are the *counts*, a factor for μ, and variates O_1, O_2, and O_3 for λ_1^O, λ_2^O, and λ_3^O.

Table 2:
Design structure for a simple LLBT

comparison	*decision*	*counts*	μ	λ_1^O	λ_2^O	λ_3^O
(12)	O_1	$y_{(12)1}$	1	1	-1	0
(12)	O_2	$y_{(12)2}$	1	-1	1	0
(13)	O_1	$y_{(13)1}$	2	1	0	-1
(13)	O_3	$y_{(13)3}$	2	-1	0	1
(23)	O_2	$y_{(23)2}$	3	0	1	-1
(23)	O_3	$y_{(23)3}$	3	0	-1	1

Example 1 Two response categories (no undecided category)

First, the necessary packages have to be loaded:

```
> library(prefmod)
> library(gnm)
```

The prefmod was written by Hatzinger (2009) and gnm by Turner and Firth (2009) and available help on these packages may be obtained using

```
> help("prefmod")
> help("gnm")
```

Data preparation:

This is an artificial example for 4 items. Therefore we have $\binom{4}{2} = 4*3/2 = 6$ comparisons. The data are given at the individual level, one line for each of the 100 respondents; we have 2 response categories in each comparison (ij).

The name of the data is (dat4) and the data for a few respondents are given below:

```
     comp1  comp2  comp3  comp4  comp5  comp6
1        1      1      1      1      1      1
2       -1      1      1      1      1     -1
3        1      1     -1      1      1     -1
         .                                  .
         .                                  .
100     -1      1      1      1     -1      1
```

The data are response vectors for the subjects for 4 items and thus 6 comparisons. The order for the comparisons has to be as follows:

(12)(13)(23)(14)(24)(34)

for the comparisons comp1, comp2, ... comp6.

For each of the two possible responses in a given comparison, e.g., comp1 is the column for comparison (12), the coding is

1 if first object (O_1) is preferred
-1 if second object (O_2) is preferred

(This is just one possible coding for the responses, the alternative would be 0/1 with 0 as preferred).

Data input:

To get the data we use the command:

```
> data(dat4)
```

Here the data are part of the package prefmod therefore we can use the command data(). Otherwise the data have to be supplied in a dataframe, obtained, e.g., by read.table("filename", header = TRUE).

Design matrix:

The command to set up the design matrix is:

```
> des1 <- llbt.design(dat4, nitems = 4)
```

The design matrix is stored in des1 and created by using the function llbt.design, assigning the data file by dat4 and the number of items nitems = 4.

The vectors in the design matrix are the y's which are the counts, one factor mu with 6 levels for the comparisons, and four variates for the objects (o1,o2,o3,o4) indicating whether a specific object is preferred (+1) or not preferred (-1) in a given comparison $(1,2,\ldots,6)$

y	mu	o1	o2	o3	o4
64	1	1	-1	0	0
36	1	-1	1	0	0
74	2	1	0	-1	0
26	2	-1	0	1	0
68	3	0	1	-1	0
32	3	0	-1	1	0
73	4	1	0	0	-1
27	4	-1	0	0	1
71	5	0	1	0	-1
29	5	0	-1	0	1
37	6	0	0	1	-1
63	6	0	0	-1	1

Model fit and output:

The command to fit the basic BT-models is:

```
> res1 <- gnm(y   o1 + o2 + o3 + o4, eliminate = mu, data = des1,
+       family = poisson)
```

This command is fitting the basic model with 4 object parameters using the function gnm. The model formula is y ~ o1+o2+o3+o4, where the y's are the counts and o1,o2,o3,o4 are the object variates. The nuisance parameters for the comparisons mu are 'eliminated' by the option eliminate = mu (without going into detail – they are fitted but not shown in the output). The input for the fit routine is the design matrix des1 generated by llbt.design and the model is a loglinear model defined by the option family = poisson.

To show the result we specify:

```
> summary(res1)

Call:

gnm(formula = y ~ o1 + o2 + o3 + o4, eliminate = mu, family = poisson, data =
des1)

Deviance Residuals:
       1         2         3         4         5         6         7
 0.52431  -0.65951  -0.45764   0.83026  -0.28396   0.42950  -0.01128
       8         9        10        11        12
 0.01858   0.79273  -1.12249  -0.99333   0.83205
```

```
Coefficients of interest:
      Estimate  Std. Error   z value   Pr(>|z|)
o1     0.49976     0.07753     6.446   1.15e-10
o2     0.29918     0.07449     4.017   5.91e-05
o3    -0.13327     0.07484    -1.781      0.075
o4          NA          NA        NA         NA

(Dispersion parameter for poisson family taken to be 1)

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

Residual deviance: 5.4416 on 3 degrees of freedom
AIC: 91.504
Number of iterations: 1
```

Interpretation:

How to interpret the output? First we are interested to know if the model is fitting well. We perform a goodness-of-fit test, i.e., a likelihood ratio test of the current model against the saturated model (a model reproducing the data perfectly). The corresponding teststatistic is the residual deviance 5.442 which has an asymptotic χ^2 -distribution with df = 3. We get the probability using the command `1-pchisq(5.4416,3)` which is 0.142 in this case. So the fit of the model is OK.

The object parameters $\lambda_1^O, \lambda_2^O, \lambda_3^O, \lambda_4^O$ are given by the shown estimates `o1, o2, o3, o4` where the estimate for o4, the last object in the list, is NA as one parameter has been set to zero (constraint)[2].

In **R** we have access to the parameter estimates which are stored in `coefficients` as part of the object named `res1`, the output of the `gnm` function. The elements 1 to 6 are the mu's, the nuisance parameters for all 6 comparisons. The next elements are the 4 object parameter estimates and can be accessed by `res1$coefficients[7:10]`, or, more easily using the extractor function `coef()`, i.e.,

```
> coef(res1)

Coefficients of interest:
       o1          o2          o3          o4
0.4997604   0.2991802  -0.1332744          NA
```

The magnitude of the estimates is telling us that `o1` is the most preferred object, second is `o2`, third is `o4` with value zero and the least preferred object is `o3` with a negative estimate.

---

[2] The `R` function `gnm` produces parameter estimates and standard errors for the λ_i^O which are contrasts, taking the last object by default as the reference object. To choose a different object as reference, simply refit the model, but specify the object identifiers in a different order. The last object in the model formula will be the reference object and set to zero.

To calculate the worth parameters π one has to take into account that the BT-model is invariant under change of scale, and identifiability (estimability) is obtained by the requirement that $\sum_i \pi_i = 1$. The relationship between the λ's and the worth parameters π is then given by

$$\pi_j = \frac{\exp\{2\lambda_j^O\}}{\Sigma_i \exp\{2\lambda_i^O\}}, j = 1,2,\ldots,J.$$

We get the worth parameters for all objects by:

```
> res1$coefficients[10] <- 0
> worth <- round(exp(2 * res1$coefficients[7:10])/(sum(exp(2 *
+    res1$coefficients[7:10]))), digits = 4)
> names(worth) <- paste("pi", 1:4, sep = "")
> print(worth)

    pi1    pi2    pi3    pi4
 0.4311 0.2887 0.1215 0.1587
```

The sum of the worth parameters is 1.

Plot:

For plotting the worth parameters we can use the function plotworth which needs a matrix containing the worth parameter as input. In this case it is a matrix with one column as there is only one group. This can be done by the following command:

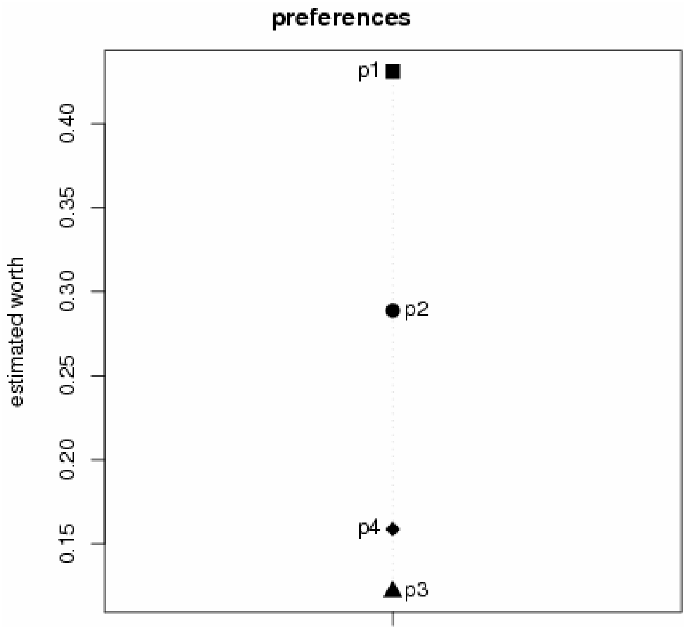
```
> worth <- as.matrix(worth)
```

The following commands specify the column label to be empty and the names for the rows (which are the objects):

```
> colnames(worth) <- ""
> rownames(worth) <- c("p1", "p2", "p3", "p4")
```

To produce a simple plot we use plotworth and specify a few options (more details are given in the help files which we get by the command ?plotworth). With the option pcol = "black" the colours of the symbols are defined to be black (by default, when no pcol option is defined, the colours for the objects are defined from the rainbow palette). With the option main = "preferences" a title is added to the plot and with ylab = "estimated worth" the y-axis is labeled.

```
> plotworth(worth, pcol = "black", main = "preferences", ylab = "estimated worth")
```



2. Some extensions to the LLBT

2.1 The LLBT for three response categories – including an undecided category

However, there are often ties, when no decision can be made. This extensions can easily be incorporated into the LLBT when using the respecification suggested by Davidson and Beaver (1977).

$$\begin{aligned} \ln m_{(jk)j} &= \mu_{(jk)} + \lambda_j^O - \lambda_k^O \\ \ln m_{(jk)0} &= \mu_{(jk)} \qquad\qquad + \gamma \\ \ln m_{(jk)k} &= \mu_{(jk)} - \lambda_j^O + \lambda_k^O \end{aligned} \tag{3}$$

where $m_{(jk)j}$ denotes the expected number of preferences for object j, $m_{(jk)k}$ denotes the expected number of preferences for object k, $m_{(jk)0}$ is the expected number of no decisions in the comparison of (jk) and γ is the undecided effect.

The following table gives the design structure where the elements (columns) are the *counts* y, a factor for μ, a dummy for γ and variates O_1, O_2, and O_3 for λ_1^O, λ_2^O, and λ_3^O.

The model formula is extended by introducing an undecided response category. There are three possible outcomes for each comparison (jk), when either object j or object k is preferred or *no* decision is made. We get one extra parameter γ for the case of *no* decision.

The function `llbt.design` builds up the necessary design matrix for the factor μ, the variates, i.e. for the object parameters λ^O and the parameter γ for undecided category.

Table 3:
Design structure for three responses

comparison	*decision*	*counts*	μ	γ	λ_1^O	λ_2^O	λ_3^O
(12)	O_1	$y_{(12)1}$	1	0	1	-1	0
(12)	*no*	$y_{(12)0}$	1	1	0	0	0
(12)	O_2	$y_{(12)2}$	1	0	-1	1	0
(13)	O_1	$y_{(13)1}$	2	0	1	0	-1
(13)	*no*	$y_{(13)0}$	2	1	0	0	0
(13)	O_3	$y_{(13)3}$	2	0	-1	0	1
(23)	O_2	$y_{(23)2}$	3	0	0	1	-1
(23)	*no*	$y_{(23)0}$	3	1	0	0	0
(23)	O_3	$y_{(23)3}$	3	0	0	-1	1

Example 2 The basic LLBT and an undecided category (ties)

A survey of 303 students was carried out in Vienna to examine the student's preferences of universities (Dittrich et al., 1998). The universities are located in London (LO), Paris (PA), Milano (MI), Barcelona (BA), St. Gallen (SG) and Stockholm (ST) and denote the 6 objects in this example. There are $15 = \binom{6}{2}$ different pairwise comparisons of two universities for each student where three responses, including 'no preference' are possible within each comparison.

Data preparation:

In this example, the data are given at the individual level (one line for each respondent) for all 303 subjects (students); we have 3 response categories because there is an undecided category, when no decision was made. The first few lines of the data for Example 2 are given in Table 4.

Table 4:
First six lines of the data cemspc

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
0	0	NA	2	2	2	0	0	1	0	0	0	1	0	1
0	0	NA	0	2	2	0	2	2	2	0	2	2	0	2
1	0	NA	0	0	2	0	0	1	0	0	0	1	0	1
0	0	NA	0	2	0	0	0	0	0	0	0	0	0	2
0	0	NA	2	2	2	2	2	2	0	0	0	0	0	2
2	2	NA	0	0	0	2	2	2	2	0	0	0	0	2

The data for the comparisons (`V1` to `V15`) are response vectors for the subjects for all comparisons. The order for the comparisons is the same as in Example 1 but for 6 items now:

(12)(13)(23)(14)(24)(34)(15)(25)(35)(45)(16)(26)(36)(46)(56)

where 1 is London, 2 is Paris, 3 is Milano, 4 is Barcelona, 5 is St. Gallen, and 6 is Stockholm. The coding for the responses in the comparisons (e.g., for `V2`) is as follows:

0 if first object (LO) is preferred
1 if no decision was made
2 if second object (MI) is preferred
Missing values are coded by `NA`

This is another possible coding for the responses (see help file for the function llbt.design, i.e., `help("llbt.design")`, section Input Data.

Data input:
The CEHS data are already included in the package and therefore they can be read in by `data(cemspc)` once the prefmod package is loaded by using the command `library(prefmod)`.

```
> data(cemspc)
```

Design matrix:
To setup the design matrix for this LLBT the following `R` commands have to be specified:

```
> des2 <- llbt.design(cemspc, nitems = 6)
```

To change the object names we first look at the current names of the data frame `des2`:

```
> names(des2)

[1] "y" "mu" "g0" "g1" "g2" "o1" "o2" "o3" "o4" "o5" "o6"
```

The variates for the objects are in the 6th to the 11th column of the data frame des2 and to change the names the following command can be used

```
> names(des2)[6:11] <- c("LO", "PA", "MI", "SG", "BA", "ST")
```

Model fit and output:

To fit the model the following R commands have to be specified again using the gnm-function.

```
> res2 <- gnm(y   LO + PA + MI + SG + BA + ST + g1, eliminate = mu,
+ data = des2, family = poisson)
```

Applying the function summary(res2) prints the following estimates for the parameters λ_i^O for the objects and the parameter γ (g1 in the output).

```
Call:
gnm(formula = y   LO + PA + MI + SG + BA + ST + g1, eliminate = mu, family =
poisson, data = des2)

Deviance Residuals:
      Min          1Q      Median          3Q         Max
-4.87907    -1.30140     0.01653     1.22732     4.70251

Coefficients of interest:
      Estimate   Std. Error    z value     Pr(>|z|)
LO     0.79062      0.04053     19.506      < 2e-16
PA     0.39743      0.03784     10.503      < 2e-16
MI     0.10450      0.03727      2.804      0.00505
SG     0.18196      0.03677      4.949     7.46e-07
BA     0.08047      0.03681      2.186      0.02883
ST          NA           NA         NA           NA
g1    -1.32619      0.04845    -27.370      < 2e-16

(Dispersion parameter for poisson family taken to be 1)

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

Residual deviance: 140.48 on 24 degrees of freedom AIC: 460.12

Number of iterations: 2
```

Interpretation:

The negative value of g1, the parameter related to 'no preference', indicates a strong tendency in favour of a decision. A strong preference for London can be seen, with Paris in second place, and so on. A goodness-of-fit test can be performed by evaluating the deviance, e.g., using the commands

```
> dev2 <- round(res2$deviance, digits = 4)
> df <- res2$df.residual
> prob2 <- 1 - pchisq(dev2, df2)
> print(prob2)

[1] 0
```

The model does not fit well as can be seen from the result: `Residual deviance:` 140.4829 with df = 24 showing a very small p-value (`prob2`). So the fit of the model is not OK. One reason might be that different groups of students have different preference orderings. In Example (3) it will be shown how to include subject-specific covariates into the model.

2.2 The LLBT with categorical subject covariates

In this section it will be explained how to incorporate subject covariates into the LLBT and how to fit these models using R.

Subject covariates allow the data modeller to move away from the assumption that all subjects have the same ordering, and instead to allow the object (item) parameters to vary according to some characteristics of the subjects (respondent/judge).

The starting point is again the LLBT for paired comparison experiments including an undecided category as defined in (3).

To illustrate the approach for incorporating one subject covariate (such as gender), assume that the judges are classified according to *one categorical covariate* S with levels l, $l = 1,2,\ldots L$. Let $m_{(jk)j|l}$ be the expected number of preferences for object j (when compared to object k) for subjects in covariate class l. The loglinear representation of this extended Bradley-Terry model is given by the following equations:

$$\begin{aligned}
\ln m_{(jk)j|l} &= \mu_{(jk)l} + \lambda_j^O - \lambda_k^O + \lambda_l^S + \lambda_{jl}^{OS} - \lambda_{kl}^{OS} \\
\ln m_{(jk)k|l} &= \mu_{(jk)l} - \lambda_j^O + \lambda_k^O + \lambda_l^S - \lambda_{jl}^{OS} + \lambda_{kl}^{OS} \\
\ln m_{(jk)0|l} &= \mu_{(jk)l} \quad + \lambda_l^S \quad + \gamma
\end{aligned}$$

There are different ways to parameterise the model. One possibility is to define a reference group, where the λ_j^O's represent the ordering for that group. The orderings for the other groups are obtained by adding λ_{jl}^{OS}'s specific to group l to the λ_j^O's for the reference (baseline) group.

For example, the preference value for groups 1 and 2 are:

$$\begin{aligned}
(\textit{reference group})\ \textit{Group}\ 1 &: \textit{preference for object}\ j : \lambda_j^O \\
\textit{Group}\ 2 &: \textit{preference for object}\ j : \lambda_j^O + \lambda_{j2}^{OS}
\end{aligned}$$

Accordingly, the specification for groups 1 (reference group) is

$$\ln m_{(jk)j|1} = \mu_{(jk)1} + \lambda_j^O - \lambda_k^O$$
$$\ln m_{(jk)k|1} = \mu_{(jk)1} - \lambda_j^O + \lambda_k^O$$
$$\ln m_{(jk)0|1} = \mu_{(jk)1} \quad + \gamma$$

and for group 2

$$\ln m_{(jk)j|2} = \mu_{(jk)2} + \lambda_j^O - \lambda_k^O + \lambda_2^S + \lambda_{j2}^{OS} - \lambda_{k2}^{OS}$$
$$\ln m_{(jk)k|2} = \mu_{(jk)2} - \lambda_j^O + \lambda_k^O + \lambda_2^S - \lambda_{j2}^{OS} + \lambda_{k2}^{OS}$$
$$\ln m_{(jk)0|2} = \mu_{(jk)2} \quad + \lambda_2^S \quad + \gamma$$

The corresponding design structure for 3 objects and a subject covariate with 2 levels consists of a factor for μ, dummies for λ^S and γ, and variates O_1, O_2, O_3, O_1S, O_2S, O_3S for λ_1^O, λ_2^O, λ_3^O, λ_1^{OS}, λ_2^{OS}, and λ_3^{OS}. The y's are again the *counts*.

The set of (nuisance) parameters λ_l^S represent the main effect of the subject covariate measured on the l-th level (in the above table we use a dummy coding, i.e., the first level is coded with 0, the reference category, and the second level with 1). λ_{jl}^{OS} and λ_{kl}^{OS} are the (useful) subject-object interaction parameters describing the effect of the subject covariate (observed on category l) on the preference for object j and k, respectively. The parameters λ^O describe the effect of the subject covariate observed on category 1 (i.e. the reference category). γ is the undecided parameter. Please note that in the context of e.g. analysis of variance the object parameters are the dependent variables.

Effectively, a separate contingency table is constructed for each level of the categorical covariate. The dimension of the complete table is

no of subject groups (levels of covariate S) $\times$ *no of comparisons* $\times$ *no of preferences.*

The LLBT can again be fitted as a Generalised Linear Model using Poisson error and log-link. The extension to multiple categorical subject covariates is straightforward.

Table 5:
Design structure for three responses and one subject covariate

comparison	*decision*	*counts*	μ	λ^S	γ	λ_1^O	λ_2^O	λ_3^O	λ_{12}^{OS}	λ_{22}^{OS}	λ_{32}^{OS}
(12)	O_1	$y_{(12)1\|1}$	1	0	0	1	-1	0	0	0	0
(12)	*no*	$y_{(12)0\|1}$	1	0	1	0	0	0	0	0	0
(12)	O_2	$y_{(12)2\|1}$	1	0	0	-1	1	0	0	0	0
⋮	⋮	⋮	⋮					⋮			
(12)	O_1	$y_{(12)1\|2}$	4	1	0	1	-1	0	1	-1	0
(12)	*no*	$y_{(12)0\|2}$	4	1	1	0	0	0	0	0	0
(12)	O_2	$y_{(12)2\|2}$	4	1	0	-1	1	0	-1	1	0
⋮	⋮	⋮	⋮					⋮			

Example 3 Categorical subject covariate

The following example demonstrates how to fit the LLBT using categorical subject information. For demonstration purposes we use again Example 2, the survey carried out at the Vienna University of Economics to examine the student's preferences of universities for a study year abroad (Dittrich et al., 1998). The objects in this example are universities located in London (LO), Paris (PA), Milano (MI), Barcelona (BA), St. Gallen (SG) and Stockholm (ST). There are 15 pairwise comparisons of universities for each student where within each comparison three responses including an undecided category were possible.

To illustrate the effect of subject covariates, gender (`SEX`: 1 = female, 2 = male) and a language factor (knowledge of English, `ENG`: 1 = good, 2 = poor), is used to define groups of students.

Data preparation:

For fitting this model we need a data file containing the results of all comparisons (responses) made by the subjects in the defined order (see Example 2) and further columns for the subject covariates. The subject covariates have to be coded numerically with consecutive integers starting with 1. (When using the function `llbt.design()` to set up the design structure, the subject covariates must be in the rightmost columns of the data, i.e., after the columns for the comparisons.)

Data file for Example 3 (`cemspc`):

```
    V1  V2  V3  V4  V5  V6  V7  V8  V9  V10 V11 V12 V13 V14 V15 ENG SEX
1   0   0   NA  2   2   2   0   0   1   0   0   0   1   0   1   1   2
2   0   0   NA  0   2   2   0   2   2   2   0   2   2   0   2   1   1
   .                               .
   .                               .
303 1   2   2   1   1   1   2   2   1   2   1   0   0   0   2   1   1
```

Remember, the order for the comparisons has to be

(12)(13)(23)(14)(24)(34)(14)(24)(34)(15)(25)(35)(45)(16)(26)(36)(46)(56)

The names of the comparisons are `V1, V2, V3, ..., V15`, where 1 is London, 2 is Paris, 3 is Milano, 4 is Barcelona, 5 is St. Gallen, and 6 is Stockholm.

The responses should be coded by 0 if the first object was preferred, 1 if no decision was made (undecided) and by 2 if the second object was preferred. Missing values should be coded by `NA`.

Data input and design matrix:

To set up the necessary input for the model fit we need the following specifications:

```
> data(cemspc)
> des3 <- llbt.design(cemspc, nitems = 6, cov.sel = "ENG")
> des3$ENG <- factor(des3$ENG)
> names(des3)[6:11] <- c("LO", "PA", "MI", "SG", "BA", "ST")
```

In the function llbt.design we need to specify the data file cemspc, the number of items nitems = 6 and which subject covariate should be included when building up the design matrix, here the knowledge of English by using the option cov.sel="ENG". As the subject covariate ENG is categorial it has to be defined as a factor (3rd line of the specifications before).

Model fit and output:

To fit the LLBT we need the following specifications:

```
> res3 <- gnm(y   LO + PA + MI + SG + BA + ST + g1, eliminate = mu:ENG,
+       data = des3, family = poisson)
> summary(res3)
```

In a first step, we fitted the basic BT-model y ~ O + PA + MI + SG + BA + ST + g1 (object parameters and undecided effect) but using the extended design matrix, which is doubled for the factor ENG, the fit routine is gnm().

The eliminated term[3] has changed from mu to mu:ENG and has accordingly been specified by eliminate = mu:ENG.

The parameter estimates are the same as in the model fit without subject covariate (Example 2) but the Residual deviance and the df's have changed, because the contingency table has been extended (doubled) due to the 2 level subject covariate ENG.

A model containing *object-subject interaction terms* can now easily be fitted. If the effects of the knowledge of English on the preference ordering of the universities is of interest one would fit the following model:

```
> res31 <- gnm(y   LO + PA + MI + SG + BA + ST + g1 + (LO +
+       PA + MI + SG + BA + ST):ENG, eliminate = mu:ENG, data = des3,
+       family = poisson)
```

The model terms are:

LO+PA+MI+SG+BA+ST	objects -- reference group
(LO+PA+MI+SG+BA+ST):ENG	objects -- comparison group they are obtained by the interactions (:) between objects and ENG
g1	undecided effect
mu:ENG	interactions between mu and ENG to be eliminated with no relation to the objects

The output of the model fit is obtained by:

```
> summary(res31)
```

[3] The : operator indicates interaction between two variables in R. For technical reason (relation between the Poisson and the multinomial distribution) we need the interaction terms between the comparison factor mu and the subject covariate ENG to define the objects being dependent variables in the model. For details see Hatzinger & Francis (2004).

```
Call:
gnm(formula = y   LO + PA + MI + SG + BA + ST + g1 + (LO + PA + MI + SG +
BA + ST):ENG, eliminate = mu:ENG, family = poisson, data = des3)

Deviance Residuals:
      Min         1Q      Median          3QMax
-5.03429   -0.78976   -0.04852     0.669734.24992

Coefficients of interest:
              Estimate    Std. Error      z value      Pr(>|z|)
LO            0.802571    0.047502        16.896       < 2e-16
PA            0.434114    0.044420         9.773       < 2e-16
MI            0.103194    0.043568         2.369       0.01786
SG            0.132598    0.043031         3.081       0.00206
BA            0.100728    0.043059         2.339       0.01932
ST                  NA          NA            NA            NA
g1           -1.323793    0.048468       -27.313       < 2e-16
ENG2:LO      -0.038506    0.090536        -0.425       0.67061
ENG2:PA      -0.134203    0.085057        -1.578       0.11461
ENG2:MI       0.007432    0.084375         0.088       0.92981
ENG2:SG       0.184651    0.083350         2.215       0.02673
ENG2:BA      -0.075317    0.083290        -0.904       0.36585
ENG2:ST             NA          NA            NA            NA

(Dispersion parameter for poisson family taken to be 1)

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

Residual deviance: 162.90 on 49 degrees of freedom AIC: 727.54

Number of iterations: 3
```

Interpretation:

Now the estimates for LO, PA, MI, SG, BA, and ST are the parameter estimates for the reference group, i.e., for students with a good knowledge in English (ENG = 1).

The ranking for the universities for students with a poor knowledge of English (ENG = 2) can be obtained by adding the corresponding interaction terms λ_{il}^{OS} to the object terms λ_i^O. For instance, the parameter estimate for SG for the group ENG = 2 is 0.3173 = (0.1326 + 0.1847) compared to SG = 0.1326 (for the reference group ENG = 1). By the way, SG:ENG is the only significant term, moving St. Gallen up in the preference scale. This might be due to the fact that poor English knowledge prevents students from choosing a non German-language university.

Other possible models:

The interaction between the term g1 ('no preference') and a subject covariate can be examined by adding g1:ENG to the model formula

```
> res32 <- gnm(y   LO + PA + MI + SG + BA + ST + g1 + g1:ENG +
+       (LO + PA + MI + SG + BA + ST):ENG, eliminate = mu:ENG,
+       data = des3, family = poisson)
```

In fact, the term `(LO+PA+MI+SG+BA+ST):ENG` is equivalent to specifying `LO:ENG+PA:ENG+...+ST:ENG`. Some of these terms might be removed from the model formula if only certain universities are to be evaluated.

Additional subject covariates can also be introduced. For examples see Dittrich et al. (1998) who fit a full range of covariates to this datset.

3. Advanced feature and model specification

3.1 The LLBT with Object covariates

In this section it will be explained how to incorporate object covariates into the LLBT and how to fit these models using R.

As before, the starting point is the LLBT for paired comparison experiments including an undecided category as defined in (3).

The extension we consider here is to take into account the effects of object covariates on the preferences of the judges. A common idea is to reparameterize the object parameters as a linear combination of P covariates $X_1,\ldots,X_P$, which represent P properties of the objects. In order to incorporate object covariates, let us replace the object-related parameters λ_j^O by the linear reparameterisation

$$\lambda_j^O = \sum_{p=1}^{P} x_{jp}\beta_p^X ,$$

where the x_{jp}'s denote the covariates describing the p th property of the object j and the β^X's are unknown regression parameters.

The LLBT including the effects of one object covariate is:

$$\begin{aligned}
\ln m_{(jk)j} &= \mu_{(jk)} + \lambda_j^O - \lambda_k^O \\
&= \mu_{(jk)} + \beta_1^X x_{j1} - \beta_1^X x_{k1} \\
&= \mu_{(jk)} + \beta_1^X (x_{j1} - x_{k1})
\end{aligned}$$

and the other equations are defined analogously.

The LLBT can again be fitted as a Generalized Linear Model using Poisson error and log-link; the design matrix consists of column vectors with suitable entries under μ, γ and of a further P column vector containing the values $x_{jp} - x_{kp}$ of the object covariates.

Example 4 Categorical object covariate

For demonstration purposes we start with Example 2, the survey carried out at the Vienna University of Economics to examine the student's preferences of universities for a study year abroad (Dittrich et al., 1998).

We are interested if universities with a common attribute can be regarded as a group having the same rank. In this case one would not consider each university separately but rather look at the attributes of the universities and their contribution to the estimates. In our example the universities can be grouped by various attributes. To show how object covariates can be incorporated one two level object covariate called LOC is used. LOC means that the universities are either located in Latin countries or in other European countries. The universities LO, SG, ST would form a contrast to the group PA, MI, BA. The values for LOC are given in the following table:

Objects	LO	PA	MI	SG	BA	ST
LOC	0	1	1	0	1	0

In general, to fit the model for object covariates we have to set up the 'basic model' for all objects first. Therefore the specifications are the same as in Example 2. The design matrix has been stored in des2.

Design matrix:

Now we have to set up a new object variate, e.g. called LAT, where each object is multiplied by its value of the object covariate LOC. This is achieved by performing a matrix multiplication of the matrix formed by the design columns for the objects (OBJ) in des2 with the new object covariate vector LOC (if there are more than one object covariate, LOC is a matrix with P columns).

```
> OBJ <- as.matrix(des2[, c("LO", "PA", "MI", "SG", "BA", "ST")])
> LOC <- c(0, 1, 1, 0, 1, 0)
> LAT <- OBJ  %*% LOC
```

LAT thus takes the value +1 when a latin university is preferred to a non-latin, -1 when a non-latin is preferred, and 0 otherwise.

The model has now to be refitted because we want to replace all objects LO, PA, MI, SG, BA, ST by an object variate.

The undecided term g1 should be included in the model specification as there is an undecided category.

Model fit and output:

We include the object variate LAT into the model (the objects are now replaced by the variate LAT).

```
> res4 <- gnm(y  LAT + g1, eliminate = mu, data = des2, family = poisson)
> summary(res4)
```

```
Call:
gnm(formula = y ~ LAT + g1, eliminate = mu, family = poisson,
   data = des2)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-10.3948   -2.6886   -0.5405    2.1238    8.8488

Coefficients of interest:
      Estimate  Std. Error     z value    Pr(>|z|)
LAT  -0.11201     0.02041      -5.488    4.06e-08
g1   -1.40052     0.04804     -29.156     < 2e-16

(Dispersion parameter for poisson family taken to be 1)

Residual deviance: 692.1 on 28 degrees of freedom
AIC: 1003.7

Number of iterations: 3
```

The six object parameters are substituted by just one parameter describing the effect of the variate LAT. The negative parameter LAT means that the set of universities located in a Latin country (LOC = 1) are preferred less than the universities of other European countries (LOC = 0).

However, it has to be checked if the reduction from a model including all universities (stored in res2 calculated in Example 2) to a model with just one parameter (in res4) is permissible. A formal procedure is to compare the deviances of the two models. In R the function anova() can be used

```
> anova(res2, res4)

Analysis of Deviance Table

Model 1: y   LO + PA + MI + SG + BA + ST + g1
Model 2: y   LAT + g1
    Resid. Df    Resid. Dev      Df    Deviance
1          24        140.48
2          28        692.10      -4     -551.62
```

From a modelling point of view the incorporation of this object covariate alone does not make sense in this data set because the deviance difference between the two models is 551.62 on 4 degrees of freedom, a very large value. Thus the objects can not be substituted by LAT. However, in many cases it is possible to find a parsimonious model by including object covariates.

3.2 Position effect

In case there is a position (or order) effect then comparison (jk) is not equal to comparison (kj), with other words it matters if object j or object k is presented first.

This extension can also be incorporated into the LLBT when using the following respecification

$$
\begin{aligned}
\ln m_{(jk)j\cdot j} &= \mu_{(jk)j} + \lambda_j^O - \lambda_k^O + \delta \\
\ln m_{(jk)k\cdot j} &= \mu_{(jk)j} - \lambda_j^O + \lambda_k^O \\
\ln m_{(jk)j\cdot k} &= \mu_{(jk)k} + \lambda_j^O - \lambda_k^O + \delta \\
\ln m_{(jk)k\cdot k} &= \mu_{(jk)k} - \lambda_j^O + \lambda_k^O
\end{aligned}
$$

where δ represents the position effect and $m_{(jk)j\cdot j}$ denotes the expected number of preferences for object j in the comparison of (jk) given $(\cdot)$ that j is presented first.

Sometimes a short notation is used when it is clear that the order of the objects in the description of the comparison is meaningful i.e. it makes a difference to say (12) or (21): $m_{(21)1\cdot 2}$ becomes $m_{(21)1}$

The corresponding design matrix is given in the following table where the elements are $\mu, \delta, \lambda_1^O, \lambda_2^O, \lambda_3^O$ where μ is a factor and the y's are the *counts*.

Table 6:

Design structure for two responses and a position effect

comparison	*decision*	*counts*	μ	δ	λ_1^O	λ_2^O	λ_3^O
(12)	O_1	$y_{(12)1}$	1	1	1	-1	0
(12)	O_2	$y_{(12)2}$	1	0	-1	1	0
⋮	⋮	⋮	⋮			⋮	
(21)	O_2	$y_{(21)2}$	4	1	-1	1	0
(21)	O_1	$y_{(21)1}$	4	0	1	-1	0
⋮	⋮	⋮	⋮			⋮	

The model formula is extended by a *position effect*. We get an extra parameter δ whenever the object presented first had been chosen. The number of lines in the design matrix are dublicated and so are the number of μ's.

The R-package builds up the necessary design matrix for the covariates and factors and provide estimates for the object parameters λ^O and if needed the parameter γ for the undecided category and the position parameter δ. Again the nuisance parameters μ are not of interest and are fitted using the eliminate option in gnm.

Example 5 *Position effect*

We use the data on results of the 1987 season for professional baseball teams in the Eastern Division of the American League published and analysed by Agresti (1990, pp 371-373) to illustrate a possible position effect.

The objects are the 7 teams, e.g. object 1 is Milwaukee, object 2 is Detroit,...., object 7 is Baltimore, where each team played each other 13 times. One game can be seen as one paired comparison between two objects. A game cannot end in a draw and there is therefore no need for an undecided catogory. We thus have 2 response categories. Taking into account a position effect (home field advantage) there are two possible comparisons between any two teams: for example, the comparison (12) in this example means that Milwaukee played at home and Detroit played away and another comparison (21) where Detroit played at home and Milwaukee played away. The number of wins and losses in the 42 different comparisons are given in Table 10.8 in Agresti (1990, pp. 373) and are already given in aggregated form.

Data preparation:

The data input for the macro has to be as follows: in a given comparison (jk) the number of preferences of object j presented first is denoted by $y_{(jk)j}$ and number of preferences for object k presented second is $y_{(jk)k}$. (Please keep in mind that here the order of the *objects* in the comparison is meaningful.)

The order for the *comparisons* has to be as follows: (jk) for all $j = 1,....,k-1$ and $k = 2,....,J$ where k is the object on the second position and J is the number of objects. So the results should have the following order:

$y_{(12)1}\ \ y_{(12)2}\ \ y_{(13)1}\ \ y_{(13)3}\ \ y_{(23)2}\ \ y_{(23)3}\ \cdots\ y_{(17)1}\ \ y_{(17)7}\ \ y_{(27)2}\ \ y_{(27)7}\ \cdots$

Since there is a position effect, we have another set of comparisons (kj) and results $y_{(kj)k}$, $y_{(kj)j}$, where object k $(k = 2,....,J)$ is on the first position, and object j is on the second position $j = 1,....,k-1$.

$y_{(21)2}\ \ y_{(21)1}\ \ y_{(31)3}\ \ y_{(31)1}\ \ y_{(32)3}\ \ y_{(32)2}\ \cdots\ y_{(71)7}\ \ y_{(71)1}\ \ y_{(72)7}\ \ y_{(72)2}\ \cdots$

Data input:

Accordingly, the input data file could be of the following form:

```
43                                          !results of (12)
42   4   2                                  !results of (13) and (23)
43   4   3   2   4                          !(14)(24)(34)
61   6   0   4   3   4   3                  !(15)(25)(35)(45)
42   6   1   4   2   4   2   5   2          !(16)(26)(36)(46)(56)
60   4   3   6   0   6   1   6   0   2   4  !(17)(27)(37)(47)(57)(67)

33                                          !(21)
52   3   4                                  !(31)(32)
33   1   5   5   2                          !(41)(42)(43)
15   5   2   3   3   2   4                  !(51)(52)(53)(54)
 5   2   3   3   4   3   3   4   2   4      !(61)(62)(63)(64)(65)
52   5   1   6   1   4   2   6   1   4   3  !(71)(72)(73)(74)(75)(76)
```

If the data are given in this form, the command to read them would be

```
data <- scan (" ", comment.char="!")
```

This is an example for aggregated data with a position effect but no undecided category.

Design matrix:

The corresponding commands to set up the data structure and to read the R-data baseball are as follows:

```
> d1 <- c(rep(0, 21), 1)
> d2 <- c(1, rep(0, 20), 2)
> d <- data.frame(rbind(d1, d2))
> names(d) <- c(paste("v", 1:21, sep = ""), "cov")
> des5 <- llbt.design(d, nitems = 7, cov.sel = "cov")
> des5$mm <- gl(42, 2)
> des5$pos <- c(des5$g0[1:42]), des5$g1[1:42])
> data(baseball)
> des5$yy <- baseball
```

By the function gl(42,2) a factor will be generated and mm is now the column for the 42 different matches (mm replaces mu), yy are the counts – number of won matches and pos is a variate indicating if the winning team played at home; in this case pos = 1. (The columns y,mu g0,g1 and cov are only auxiliary variables.)

The corresponding design structure stored in des5 is as follows (only the first and the last 5 lines are displayed):

```
> names(des5)[5:11] <- c("MIL", "DET", "TOR", "NY", "BOS",
+     "CLE", "BAL")
> head(des5)
```

	y	mu	g0	g1	MIL	DET	TOR	NY	BOS	CLE	BAL	cov	mm	pos	yy
1	1	1	1	0	1	-1	0	0	0	0	0	1	1	1	4
2	0	1	0	1	-1	1	0	0	0	0	0	1	1	0	3
3	1	2	1	0	1	0	-1	0	0	0	0	1	2	1	4
4	0	2	0	1	-1	0	1	0	0	0	0	1	2	0	2
5	1	3	1	0	0	1	-1	0	0	0	0	1	3	1	4
6	0	3	0	1	0	-1	1	0	0	0	0	1	3	0	2

```
> tail(des5)

    y  mu  g0  g1 MIL DET TOR  NY BOS CLE BAL cov mm pos yy
79  1  19   1   0   0   0   0   1   0   0  -1   2 40   0  4
80  0  19   0   1   0   0   0  -1   0   0   1   2 40   1  2
81  1  20   1   0   0   0   0   0   1   0  -1   2 41   0  6
82  0  20   0   1   0   0   0   0  -1   0   1   2 41   1  1
83  1  21   1   0   0   0   0   0   0   1  -1   2 42   0  4
84  0  21   0   1   0   0   0   0   0  -1   1   2 42   1  3
```

Model fit and output:
To fit the basic model including a position effect δ we specify:

```
> res5 <- gnm(yy ~ MIL + DET + TOR + NY + BOS + CLE + BAL + pos,
+       eliminate = mm, data = des5, family = poisson)
> summary(res5)

Call:
gnm(formula = yy ~ MIL + DET + TOR + NY + BOS + CLE + BAL + pos,
eliminate = mm, family = poisson, data = des5)

Deviance Residuals:
      Min          1Q       Median          3Q         Max
-2.039547   -0.320198    0.001841    0.341847    1.594799

Coefficients of interest:
     Estimate  Std. Error  z value   Pr(>|z|)
MIL    0.8098      0.1737    4.662   3.13e-06
DET    0.7377      0.1723    4.282   1.85e-05
TOR    0.6636      0.1702    3.900   9.64e-05
NY     0.6407      0.1702    3.764   0.000167
BOS    0.5719      0.1689    3.386   0.000710
CLE    0.3523      0.1675    2.104   0.035417
BAL        NA          NA       NA         NA
pos    0.3023      0.1309    2.308   0.020981

(Dispersion parameter for poisson family taken to be 1)

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

Residual deviance: 38.643 on 35 degrees of freedom
AIC: 377.87

Number of iterations: 3
```

The residual deviance 38.64 with `residual df = 35` compares to the result of Agresti (p. 373: $G^2 = 38.64$ and $df = 35$). `mm` denotes a factor for the μ's with as many factor levels as comparisons (here `21*2 = 42`).

Interpretation:

The object parameters `(MIL,DET,TOR,NY,BOS,CLE,BAL)` are the estimated $\lambda_i^O, i = 1,...,7$ when the teams play away and the parameter `pos` denotes the position effect δ, the general home field advantage in this example. The odds for a team to win a game are increased by exp(0.3023) = 1.35 if it plays at home. There is a (general) advantage for the teams when playing at home.

We get the worth parameters for all teams when playing away by:

```
> res5$coefficients[49] <- 0
> nominator <- exp(2 * res5$coefficients[43:49])
> denominator <- sum(exp(2 * res5$coefficients[43:49]))
> round(nominator/denominator, digits = 3)

  MIL    DET    TOR     NY    BOS    CLE    BAL
0.220  0.190  0.164  0.157  0.137  0.088  0.044
```

Let us look at the odds ratio for TOR (Toronto) to win against NY (New York) in case both teams play away (e.g. in a third city):

```
> exp(2 * res5$coefficients[45] - 2 * res5$coefficients[46])

     TOR
1.046834
```

We get the same result when using the πs

```
> round(0.164226/0.156879, digits = 5)

[1] 1.04683
```

As both teams have a similar strength (λ for TOR is 0.664 and λ for NY is 0.641), the odds ratio is close to one (1.0468) and therefore both teams have a similar chance to win against each other when not playing at home.

In case TOR plays at home and NY away the odds ratio for TOR to win is given by:

```
> exp(2 * res5$coefficients[45] - 2 * res5$coefficients[46] +
+      res5$coefficient[50])

     TOR
1.416276
```

The odds for Toronto to win against New York is now 1.416 times higher if Toronto plays at home and New York plays away.

4. Discussion

This paper is dealing with a variety of paired comparison (PC) models focusing on true paired comparison data. Some extensions to the basic BT model are presented, as the introduction of an undecided response category and subject covariates. Furthermore advanced features are presented to model object specific covariates and position effects. It is shown how these models can be fitted using the R package prefmod. The common assumption in all models described in this paper is the independence of the decisions in the paired comparisons.

However, this class of models is not restricted to real PC responses, as paired comparisons have much in common with ranking tasks but also with ratings (often called Likert items). If we know the order of the objects is (3,1,2), object 3 is best, object 1 is second and object 2 is the last one, we also know that in the comparison (13) object 3 is preferred to object 1 and so on. For Likert answer formats similar transformations are possible.

In these cases we can derive PC-data from the original data which can be either

- rankings
- partial rankings (optionally with ties)
- ratings (Likert items)

But real paired comparison data and derived paired comparison data differ in various ways and therefore need special treatment when applying PC models. With the prefmod package these response formats can also be modelled. Moreover model terms can be introduced taking into account that the independence assumption might not be appropriate.

Acknowledgements. The authors are particularly grateful for helpful comments by an anonymous reviewer. Special thanks to Brian Francis and Walter Katzenbeisser who contributed to all the work that has been the basis of this article. The R code and ouput was typeset using the Sweave-framework (by Fritz Leisch) which is part of the R program. Thanks Fritz.

References

Agresti, A. (1990). *Categorical Data Analysis*. J.Wiley, New York.

Bradley, R.A. & Terry, M.E. (1952). Rank Analysis of Incomplete Block Designs. I. The Method of Paired Comparisons. *Biometrika, 39*, 324-345.

Davidson, R.R. & Beaver, R.J. (1977). On extending the Bradley-Terry model to incorporate within-pair order effects. *Biometrics, 33*, 693-702.

Dittrich, R., Hatzinger, R., & Katzenbeisser, W. (1998). Modelling the effect of subject-specific covariates in paired comparison studies with an application to university rankings. *Journal of the Royal Statistical Society, Series C: Applied Statististics, 47*, 511-525.

Fienberg, S.E., & Larntz, K. (1976). Loglinear representation for paired and multiple comparison models. *Biometrika, 63*, 245-254.

Fischer, G.H., & Tanzer, N.(1994). Some LLTM and LBTL relationships. In: G.H. Fischer, & D. Laming (Eds.): *Contributions to mathematical psychology, psychometrics, and methodology*. New York: Springer-Verlag

Hatzinger, R. (2009). prefmod: Utilities to fit paired comparison models for preferences. *R package version 0.8-14* URL http://CRAN.R-project.org/package=prefmod.

Hatzinger, R., & Francis, B.J.(2004). Fitting Paired Comparison Models in R. Research Report Series/Department of Statistics and Mathematics, Nr. 3, May 2004 URL http://epub.wu-wien.ac.at

Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis*. New York: Wiley.

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Sinclair, C.D. (1982). GLIM for preference. In: Gilchrist, R. (Eds.): *GLIM 82. Proceedings of the International Conference on Generalised Linear Models*, Springer Lecture Notes in Statistics, 14.

Turner, H., & Firth, D. (2009). Generalized nonlinear models in R: An overview of the gnm package. *R package version 0.9-8* URL http://CRAN.R-project.org/package=gnm.

Tutz, G. (1989). Latent Trait-Modelle für ordinale Beobachtungen *[Latent trait-models for ordinal observations]*. Lehr- und Forschungstexte Psychologie, Springer-Verlag.

Appendix: The relation between the biomial and the Poisson distribution

Let y_1, y_2 be the number of subjects preferring alternative 1 or 2. A Poisson model treats these as independent random variables (Y_1,Y_2) with realisations (y_1,y_2) and parameters (μ_1,μ_2). The joint probability mass function is product of the two mass functions of the form

$$p(y_1,y_2)=\frac{\mu_1^{y_1}e^{-\mu_1}}{y_1!}\cdot\frac{\mu_2^{y_2}e^{-\mu_2}}{y_2!}$$

Then, the sum $Y_1 + Y_2$ is also Poisson distributed with parameter $\mu_1 + \mu_2 = \mu$. If we condition on $n = y_1 + y_2$, the Y_is are no longer Poisson distributed (because $y_1 + y_2$ cannot exceed n). The conditional probability is binomial

$$\begin{aligned} p(Y_1=y_1,Y_2=y_2\mid y_1+y_2=n) &= \frac{(\mu^{y_1}e^{-\mu_1}/y_1!)(\mu^{y_2}e^{-\mu_2}/y_2!)}{\mu^n e^{-\mu}/n!} \\ &= \frac{n!}{y_1!y_2!}\pi_1^{y_1}\pi_2^{y_2} \end{aligned}$$