Paired Comparison Preference Models

The prefmod Package: Day3

Repetition, News , position effect

Repeated measurements

Regina Dittrich & Reinhold Hatzinger

Department of Statistics and Mathematics, WU Vienna

Model fitting with `gnm()` using `llbt.design()`

▷ **step 1:** generate the design matrix with llbt.design()

♠*new option:* `cat.scovs = c("SEX")`

```
> load("cpc.Rdata")
> cities<-c("LO","PA","MI","SG","BA","ST")
> dd <- llbt.design(cpc, 6, objnames = cities, undec = T,
+       cat.scov = c("SEX") )

> head(dd)
    y mu g0 g1 g2 LO PA MI SG BA ST SEX
1  91  1  1  0  0  1 -1  0  0  0  0   1
2  10  1  0  1  0  0  0  0  0  0  0   1
3  51  1  0  0  1 -1  1  0  0  0  0   1
4 102  2  1  0  0  1  0 -1  0  0  0   1
5  12  2  0  1  0  0  0  0  0  0  0   1
6  38  2  0  0  1 -1  0  1  0  0  0   1
```

(♠ We **need not** to declare categorical subject covariates as factors as we did when using `cov.sel =`)

▷ **step 2:** fit a model using `gnm()`

```
> mds <- gnm(y ~ LO+PA+MI+SG+BA+ST + (LO+PA+MI+SG+BA+ST):SEX + g1,
+        elim = mu:SEX,
+        family = poisson,
+        data = dd)
```

▷ **step 3:** ♠ To plot the results we **can now** use `llbt.worth` and `plotworth()`

```
> estmds <- llbt.worth(mds)
> rownames(estmds) <- cities
> colnames(estmds) <- c("female", "male")
```

(♠ We **need not** to extract the coefficients and calculate a matrix)

Object Specific Covariates

(Dittrich, Hatzinger, Katzenbeisser, J. Royal Statistical Society, C, 1998)

To model the objects by a few characteristics

$$\lambda_j^o = \sum_{q=1}^{Q} \beta_q x_{jq}$$

x_{jq} covariate for characteristic q of object j

β_q effect of characteristic q

(cf. LLTM)

▷ subject and object specific covariates can be combined

Example: CEMS exchange programme

- We considered the following university attributes – object covariates `EC`, `MS`, `FS`, `LAT`:

The values are given as follows:

Objects	LO	PA	MI	SG	BA	ST
EC (specialised in economics)	1	0	1	0	0	0
MS (specialised in management science)	0	1	0	0	1	0
FS (specialised in finance)	0	0	0	1	0	1
LAT (Latin city)	0	1	1	0	1	0

`llbt.design()` ♠ new option: `objcovs =`

- To fit a model with object covariates (attributes) we only need the following steps:

▷ (1) – generate object covariate(s):

```
> LAT <-  c(0,1,1,0,1,0)
```

▷ (2) – make a `data.frame()` for object covariates

```
> LAT <- data.frame(LAT)
```

▷ (3) – make a `llbt.design()` using option: `objcovs =`

```
> des.neu <- llbt.design(cemspc,6, objnames=cities, undec=TRUE,
+              objcovs = LAT)
```

▷ (4) – fit model using `gnm()`

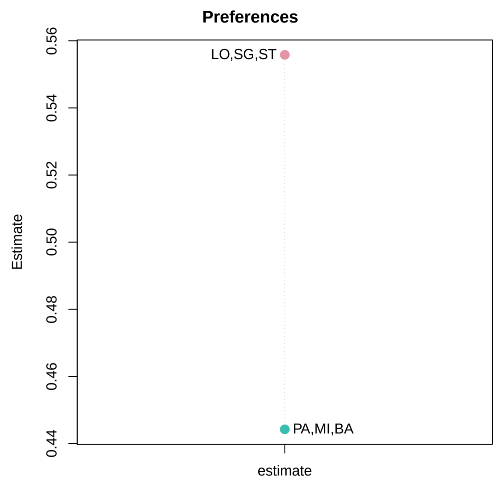
```
> md1  <- gnm(y ~ LAT + g1,
+               eliminate = mu, family = poisson, data = des.neu)
```


▷ (5) – ♠ We can apply `llbt.worth()` if we have used the new option `objcovs =`

```
> w_md1 <- llbt.worth(md1)
> w_md1
         estimate
LO,SG,ST  0.55577
PA,MI,BA  0.44423
attr(,"objtable")
  LAT x.1 x.2 x.3
1   0  LO  SG  ST
2   1  PA  MI  BA
```

▷ (6) – ♠ plot the worth

```
> plotworth(w_md1)
```

more object covariates

- To fit a model with more object covariates (attributes) proceed as follows:

▷ (1) – generate object covariates:

```
> LAT <-  c(0,1,1,0,1,0)
> EC  <- c(1,0,1,0,0,0)
> MS  <- c(0,1,0,0,1,0)
> FS  <- c(0,0,0,1,0,1)
```

▷ (2) – make a `data.frame()` for object covariates

```
> OBJ <- data.frame(LAT,EC,MS,FS)
> cities<-c("LO","PA","MI","SG","BA","ST")
```

▷ (3) – make a `llbt.design()` using option: `objcovs =`

```
> des.n1 <- llbt.design(cpc, 6, objcovs = OBJ,
+           objnames = cities)
```

▷ (4) – fit model using `gnm()`

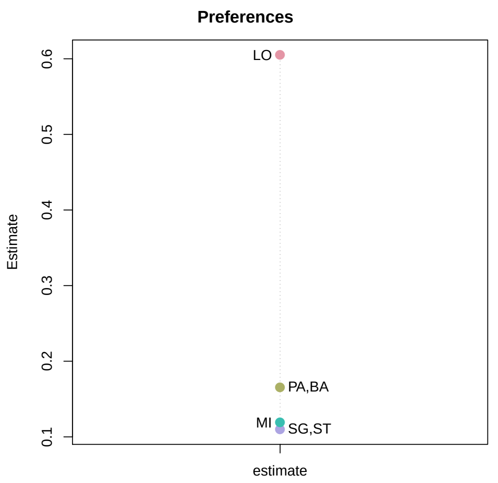
```
> ml3 <- gnm(y ~ LAT+EC+MS+FS+ EC:MS ,
+     eliminate = mu, family=poisson, data=des.n1)
```


▷ (1) – calculate the worth

```
> ww3 <- llbt.worth(ml3)
> ww3
       estimate
LO      0.60504
PA,BA   0.16563
MI      0.11946
SG,ST   0.10988
attr(,"objtable")
  LAT EC MS FS      x
1   0  1  0  0     LO
2   1  1  0  0     MI
3   1  0  1  0 PA, BA
4   0  0  0  1 SG, ST
```

▷ (2) – plot the worth

```
> plotworth(ww3)
```

- To fit a model with **object covariates** (attributes) and **categorical subject covariates** proceed as follows:

▷ (1) – generate object covariates:

```
> LAT <-  c(0,1,1,0,1,0)
> EC  <- c(1,0,1,0,0,0)
> MS  <- c(0,1,0,0,1,0)
> FS  <- c(0,0,0,1,0,1)
```

▷ (2) – make a `data.frame()` for object covariates

```
> OBJ <- data.frame(LAT,EC,MS,FS)
> cities<-c("LO","PA","MI","SG","BA","ST")
```

▷ (3) – make a `llbt.design()` using option: `objcovs =` and `cat.scovs =`

```
> des.n2 <- llbt.design(cpc,6, objnames = cities,
+           objcovs=OBJ,
+           cat.scovs = c("ENG","SEX"))
```

▷ (4) – fit model using `gnm()`

```
> m.n2 <- gnm(y ~ LAT *EC + MS + FS+ ENG + LAT:ENG +LO:SEX, ,
+     eliminate=ENG:SEX:mu, family=poisson, data=des.n2)
```

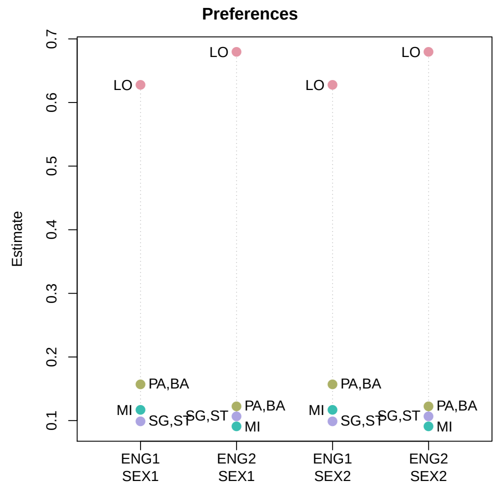
- Calculate worth and plot for
object covariates and **categorical subject covariates**:

▷ (1) – calculate the worth

```
> w.n2 <- llbt.worth(m.n2)
> w.n2
      ENG1:SEX1 ENG2:SEX1 ENG1:SEX2 ENG2:SEX2
LO     0.627880  0.679654  0.627608  0.679401
PA,BA  0.156699  0.122352  0.156814  0.122448
MI     0.116657  0.091087  0.116742  0.091159
SG,ST  0.098763  0.106907  0.098836  0.106992
attr(,"objtable")
  LAT EC MS FS LO      x
1   1  1  0  0  0     MI
2   1  0  1  0  0 PA, BA
3   0  0  0  1  0 SG, ST
4   0  1  0  0  1     LO
```

▷ (2) – plot the worth

```
> plotworth(w.n2)
```

Numerical Subject Covariates

(Francis, Dittrich, Hatzinger, Penn, J. Royal Statistical Society, C, 2002)

The basic LLBT-model has to be extended for each individual i

The equation for individual i, 1 comparison (jk), 1 response is:

$$\ln m_{(i,j \succ k)} = \mu_{i,(jk)} + \lambda_{i,j} - \lambda_{i,k}$$

We model the $\lambda_{i,j}$ through the relationship

$$\lambda_{i,j} = \lambda_j + \sum_{r=1}^{R} \beta_{jr} x_{r,i}$$

where $x_{r,i}$ corresponds to the rth covariate for individual i

For each object j, there is a separate set of β-parameters which describe the effect of the covariates on that item.

Example for 4 objects (ficticious)

▷ we simulate PC-data using `simPC()`

```
> dat <- simPC(4, 20, c(1,3,5,7), pr = T )
used worth parameters are:  0.0625 0.1875 0.3125 0.4375
```

1st number of objects = 4

2nd number of individuals = 20

3rd numbers to calc worth: c(1,3,5,7)

$1/16, 3/16, 5/16, 7/16$, where 16 is $(1+3+5+7)$

▷ we simulate a numerical subject covariate p for 20 individuals

```
> p<-rnorm(20)
```

▷ we make a data.frame with PC-data and subject covariate

```
> dat<-data.frame(dat,p)
```

Model fitting with gnm() using llbt.design()

▷ **step 1:** generate the design matrix with llbt.design()

♠*new option:* num.scovs =

```
> des <- llbt.design(dat, 4, num.scovs = "p")

> head(des)
  y mu g0 g1 o1 o2 o3 o4       p CASE
1 0  1  1  0  1 -1  0  0 0.54134    1
2 1  1  0  1 -1  1  0  0 0.54134    1
3 0  2  1  0  1  0 -1  0 0.54134    1
4 1  2  0  1 -1  0  1  0 0.54134    1
5 1  3  1  0  0  1 -1  0 0.54134    1
6 0  3  0  1  0 -1  1  0 0.54134    1
```


▷ **step 2:** fit models using gnm()

CASE is a subject covariate, therefore we use

eliminate = mu:CASE

```
> m2 <- gnm(y ~ o1+o2+o3+o4+(o1+o2+o3+o4):p,
+           family = poisson, data = des,
+           eliminate=mu:CASE)

> m2
Call:
gnm(formula = y ~ o1 + o2 + o3 + o4 + (o1 + o2 + o3 + o4):p,
    eliminate = mu:CASE, family = poisson, data = des)

Coefficients of interest:
    o1      o2      o3      o4    o1:p    o2:p    o3:p    o4:p
-1.087  -0.599  -0.572      NA   0.336   0.224   0.295      NA

Deviance:            130.48
Pearson chi-squared: 114.14
Residual df:         114
```


▷ **step 3:** calculate the worth (♠can not use llbt.worth())

extract coefficients ofInterest

```
> cc <- coef(m2)
```

replace all NA coefficients with zero

```
> cc <- ifelse(is.na(cc),0,cc)
```

extract coefficients

```
> a <- cc[1:4]
> b <- cc[5:8]
```

make a sequence for X coordinate (here person variable p) to be plotted

```
> s <- seq(min(p),max(p),0.01)
```


we write a function to calculate worth

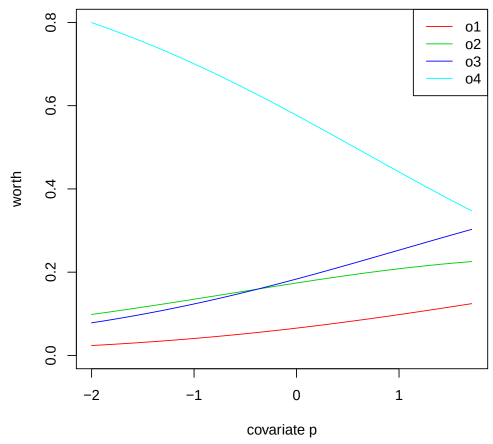
```
> ww <- function(x,a,b){exp(2*(a+b*x))/sum(exp(2*(a+b*x)))}
```

calculate worth matrix

```
> res <- sapply(s,ww,a,b)
> # lambdas for person 1
> res[,1]
      o1       o2       o3       o4
0.023740 0.098398 0.078250 0.799611
```

plot the worth

```
> plot(s,res[1,],type="l",ylim=c(0,max(res)),xlim=range(p),
+      col = 2,ylab = "worth",xlab = "covariate p")
> lines(s,res[2,],col=3)
> lines(s,res[3,],col=4)
> lines(s,res[4,],col=5)
> legend("topright",rownames(res),lty=1,col=2:5)
```



Position effect

- it makes a difference which object is presented first
  we differentiate between:
  (jk) if j is presented first and (kj) if k is presented first
  $m_{(j \succ k) \cdot j}$ expected preferences for j if presented first
  $m_{(j \succ k) \cdot k}$ expected preferences for j if **not** presented first

the LLBT model formulas for the comparison (jk) are now:
$\ln m_{(j \succ k) \cdot j} = \mu_{(jk)} + \lambda_j - \lambda_k + \delta$
$\ln m_{(k \succ j) \cdot j} = \mu_{(jk)} - \lambda_j + \lambda_k$

and the LLBT model formulas for the comparison (kj) are:
$\ln m_{(j \succ k) \cdot k} = \mu_{(kj)} + \lambda_j - \lambda_k$
$\ln m_{(k \succ j) \cdot k} = \mu_{(kj)} - \lambda_j + \lambda_k + \delta$

for 3 objects we have 6 different comparisons
▷ δ represents a general position effect

Example: Baseball

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)

- the objects are the 7 teams

  Milwaukee (MIL), Detroit (DET), Toronto (TOR), New York (NY), Boston (BOS), Cleveland (CLE) and Baltimore (BAL)

- each game is a paired comparison
- no draw – no undecided decision
- possible position effect (home field advantage)
- How many comparisons do we have?

  the number of wins and losses
  are given in the R - datafile "baseball"

```
> data(baseball)
```


- Data are given in aggregated form (already counts)

4 3 4 2 4 2 4 3 4 3 2 4 6 1 6 0 4 3 4 3 4 2 6 1 4 2 4 2 5 2 6
0 4 3 6 0 6 1 6 0 2 4

3 3 5 2 3 4 3 3 1 5 5 2 1 5 5 2 3 3 2 4 5 2 3 3 4 3 3 4 2 4 5
2 5 1 6 1 4 2 6 1 4 3

Example: Baseball

- Preparation

1. generate **two** dummy response pattern, one for each of two groups

21 responses (`v1` − `v21`) where both responses 0, 1 occur
and one covariate `cov` with levels (1, 2)

```
> 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")
> d
   v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 v16 v17 v18 v19 v20
d1  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
d2  1  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
   v21 cov
d1   0   1
d2   0   2
```


2. make design matrix using `llbt.design()`

```
> des5<-llbt.design(d, nitems=7,
+          objnames=c("MIL","DET","TOR","NY","BOS","CLE","BAL"),
+          cat.scov="cov")
> head(des5)
  y mu g0 g1 MIL DET TOR NY BOS CLE BAL cov
1 1  1  1  0   1  -1   0  0   0   0   0   1
2 0  1  0  1  -1   1   0  0   0   0   0   1
3 1  2  1  0   1   0  -1  0   0   0   0   1
4 0  2  0  1  -1   0   1  0   0   0   0   1
5 1  3  1  0   0   1  -1  0   0   0   0   1
6 0  3  0  1   0  -1   1  0   0   0   0   1
```

3. replace the old factor `mu` with a factor of length 42 each number repeated two times (number of different comparisons is now 21* 2 = 42)

```
> des5$mu <- gl(42,2)
> des5$mu
 [1] 1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9  10 10 11 11 12
[24] 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20 21 21 22 22 23 23
[47] 24 24 25 25 26 26 27 27 28 28 29 29 30 30 31 31 32 32 33 33 34 34 35
[70] 35 36 36 37 37 38 38 39 39 40 40 41 41 42 42
42 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 ... 42
```


4. construct δ for posititition effect – § **must have** the name `pos`

```
> pos <- c(rep(1:0, 21), rep(0:1,21) )
```

- for comparisons 1 − 21
the first team ($1 \succ 2$) is playing at home, pos = 1 and
the second team ($2 \succ 1$) is playing away, pos = 0

```
> pos[1:42]
 [1] 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
[36] 0 1 0 1 0 1 0
```

- for comparisons 22 − 42
the first team ($1 \succ 2$) is playing away, pos = 0 and
and the second team ($2 \succ 1$) is playing at home, pos = 1

```
> pos[43:length(pos)]
 [1] 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
[36] 1 0 1 0 1 0 1
```


5. read in the number of wins and losses given in `baseball.rda` which is a dataset in `prefmod` and assign it to y

```
> data(baseball)
> head(baseball)
[1] 4 3 4 2 4 2
> des5$y <- baseball
```

6. The design matrix `des5` now looks like

```
> head(des5)
  y mu g0 g1 MIL DET TOR NY BOS CLE BAL cov
1 4  1  1  0   1  -1   0  0   0   0   0   1
2 3  1  0  1  -1   1   0  0   0   0   0   1
3 4  2  1  0   1   0  -1  0   0   0   0   1
4 2  2  0  1  -1   0   1  0   0   0   0   1
5 4  3  1  0   0   1  -1  0   0   0   0   1
6 2  3  0  1   0  -1   1  0   0   0   0   1
```

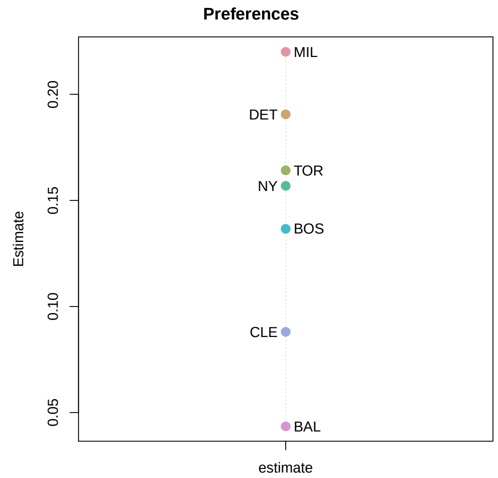
(The columns `g0,g1,cov` are only auxiliary variables)

7. fit the basic model including a position effect

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

8. Calculate the worth and plot

```
> w5<- llbt.worth(res5)
> plotworth(w5)
```

▷ The λs and the worth (given in `w5`) are the strength of the teams when **play away**:

```
> res5
Call:
gnm(formula = y ~ MIL + DET + TOR + NY + BOS + CLE + BAL + pos,
    eliminate = mu, family = poisson, data = des5)

Coefficients of interest:
  MIL    DET    TOR     NY    BOS    CLE    BAL    pos
0.810  0.738  0.664  0.641  0.572  0.352     NA  0.302

Deviance:            38.643
Pearson chi-squared: 34.963
Residual df:         35
```

9. How can we interpret `pos` which is the "home effect" ?

$$\exp(0.3023) = 1.35$$

the estimated ODDS for all teams to win when plying at home are 1.35 higher compared to playing away.

Comparing two teams

▷ Compare TOR and NY when playing against each other;

- if NY is playing at home and Toronto is playing away the odds for NY (team 4) to win against TOR (team 3) are:

```
> home <- coef(res5)[8]
> team <- coef(res5)[1:7]
> oddsNY <- exp((2 * team[4] + home) - 2 * team[3])
> oddsNY
   NY
1.2924
```

▷ odds for New York to win against Toronto is now 1.292 times higher if New York plays at home and Toronto plays away.

Remarks

1. it is assumed that the decisions are independent!
   (may be not reasonable)

2. missing values (NA) can occur in the comparisons
   just reduce the number of respondents N_{ij}
   but no missing values are allowed in the subject covariates

3. the number of rows of the design matrix is:

   number of comparisons $\times$
   number of possible decisions (response categories) $\times$
   number of subject groups

Response-format	Model		Designmatrix	Estimation	Notes
real PCs	LLBT	Data	`llbt.design()`	`glm(),gnm()`	1,2,(3),4, (5)
		Data	`llbt.design()`	`llbt.fit()`	1,3,4,5
		Data	⟶	`llbtPC.fit()`	1,3,5
	Pattern	Data	`patt.design()`	`glm(),gnm()`	2,4,(5),6
		Data	⟶	`pattPC.fit()`	1,3,(5),6
Rankings	Pattern	Data	`patt.design()`	`glm(), gnm()`	2,4,(5)
		Data	⟶	`pattR.fit()`	1,3,5
Ratings (Likert)	Pattern	Data	`patt.design()`	`glm(),gnm()`	2,4,(5)
		Data	⟶	`pattL.fit()`	1,3,5,6

(1) NAs
(2) R standard Output
(3) larger number of comparisons (objects)
(4) object specific covariates
(5) continuous subject covariates
(6) dependencies

The Basic Bradley-Terry Model (BT)

In the comparison (jk) the probability that object j is preferred to object k is defined as:

$$p_{(jk)(+)} = p(j \succ k) = \frac{\pi_j}{\pi_j + \pi_k} = c_{(jk)} \frac{\sqrt{\pi_j}}{\sqrt{\pi_k}}$$

and

$$p_{(jk)(-)} = p(k \succ j) = c_{(jk)} \frac{\sqrt{\pi_k}}{\sqrt{\pi_j}}$$

the π's are the location of the objects
$c_{(jk)}$ is constant for a given compariso

$(+)$ indicates that the first object is preferred and
$(-)$ indicates that the second object is preferred

The Basic Loglinear BT Model (LLBT)

the model can be formulated as a log-linear model following the usual Multinomial / Poisson - equivalence.

our basic paired comparison model for one comparison (jk)
is given by two equations

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

and the expected values are

$$m_{(jk)(+)} = N_{(jk)} p_{(jk)(+)} \quad \text{and} \quad m_{(jk)(-)} = N_{(jk)} p_{(jk)(-)}$$

this model formulation is feasible for further extensions

♠ Repeated observations at two time points

In the comparison (jk) the probability that $(j \succ k)$ at both timepoints is defined as:

$$p_{(jk)(++)} = c_{(jk)} \frac{\sqrt{\pi_{j1}}}{\sqrt{\pi_{k1}}} \cdot \frac{\sqrt{\pi_{j2}}}{\sqrt{\pi_{k2}}}$$

The other probabilities are:

$$p_{(jk)(+-)} = c_{(jk)} \frac{\sqrt{\pi_{j1}}}{\sqrt{\pi_{k1}}} \cdot \frac{\sqrt{\pi_{k2}}}{\sqrt{\pi_{j2}}} \qquad p_{(jk)(-+)} = c_{(jk)} \frac{\sqrt{\pi_{k1}}}{\sqrt{\pi_{j1}}} \cdot \frac{\sqrt{\pi_{j2}}}{\sqrt{\pi_{k2}}}$$

$$p_{(jk)(--)} = c_{(jk)} \frac{\sqrt{\pi_{k1}}}{\sqrt{\pi_{j1}}} \cdot \frac{\sqrt{\pi_{k2}}}{\sqrt{\pi_{j2}}}$$

where $c_{(jk)}$ is different constant for a given comparison

The corresponding loglinear model has now four equations for each comparison (jk)

$$\begin{aligned} \ln m_{(jk)(++)} &= \mu_{(jk)} + \lambda^O_{j1} - \lambda^O_{k1} + \lambda^O_{j2} - \lambda^O_{k2} \qquad (1)\\ \ln m_{(jk)(-+)} &= \mu_{(jk)} - \lambda^O_{j1} + \lambda^O_{k1} + \lambda^O_{j2} - \lambda^O_{k2}\\ \ln m_{(jk)(+-)} &= \mu_{(jk)} + \lambda^O_{j1} - \lambda^O_{k1} - \lambda^O_{j2} + \lambda^O_{k2}\\ \ln m_{(jk)(--)} &= \mu_{(jk)} - \lambda^O_{j1} + \lambda^O_{k1} - \lambda^O_{j2} + \lambda^O_{k2} \end{aligned}$$

terms and relations

- relation between π and λ:

$$\lambda_{jt} = \ln \sqrt{\pi_{jt}} \qquad \text{for all times } t = 1, \ldots, T$$

$$\pi_{jt} = \exp 2\lambda_{jt}$$

- identifiability of πs is obtained by the restriction $\pi_{Jt} = 1$ via $\lambda_{Jt} = 0$
- the worth parameters are calculated by

$$\pi_{jt} = \frac{\exp(2\lambda_{jt})}{\sum_j \exp(2\lambda_{jt})} \qquad t = 1, \ldots, T$$

where $\sum_j \pi_{jt} = 1$ for all times $t = 1, \ldots, T$

Within-comparison Dependencies

one important feature of a multivariate-LLBT is:

- we can introduce *within-comparison* dependencies
  – association between
  responses to (jk) at time t_1 and responses to (jk) at time t_2
- for 2 times there are $\binom{J}{2}$ *within-comparison* dependencies
- for T times there are $\binom{T}{2} \times \binom{J}{2}$ such dependencies

these dependence terms are denoted by: $\zeta_{(ij)}$

▷ repeated – dependencies between 2 or more timepoints
for all pairs of comparisons
▷ multivariate – dependencies between 2 dimensions (e.g. α_1, α_2)

Within-comparison Dependencies

we look at one comparisons (jk)

			time 2 $(1 \succ 3)$ +	time 2 $(3 \succ 1)$ −
time 1	$(1 \succ 2)$	+	m_{++}	m_{+-}
	$(2 \succ 1)$	−	m_{-+}	m_{--}

$$OR_{(jk)} = \frac{m_{++}m_{--}}{m_{+-}m_{-+}}$$

nominator are "coherent" decisions
denominator are "incoherent" decisions

Interpretation in terms of the parameters $\zeta_{(ij)}$

$$\ln OR_{(jk)} = 4\zeta_{(ij)} \qquad OR_{(jk)} = \exp(4\zeta_{(ij)})$$

Extending (1), the four equations for comparison (jk) become

$$\begin{aligned}
\ln m_{(jk)(++)} &= \mu_{(jk)} + \lambda^O_{j1} - \lambda^O_{k1} + \lambda^O_{j2} - \lambda^O_{k2} + \zeta_{(jk)} \\
\ln m_{(jk)(-+)} &= \mu_{(jk)} - \lambda^O_{j1} + \lambda^O_{k1} + \lambda^O_{j2} - \lambda^O_{k2} - \zeta_{(jk)} \\
\ln m_{(jk)(+-)} &= \mu_{(jk)} + \lambda^O_{j1} - \lambda^O_{k1} - \lambda^O_{j2} + \lambda^O_{k2} - \zeta_{(jk)} \\
\ln m_{(jk)(--)} &= \mu_{(jk)} - \lambda^O_{j1} + \lambda^O_{k1} - \lambda^O_{j2} + \lambda^O_{k2} + \zeta_{(jk)}
\end{aligned}$$

The sign of $\zeta_{(ij)}$ depends on the response pattern and can be regarded as the interaction of the responses at t_1 and t_2.

multivariate-LLBT Design Structure

- for 2 timepoints and 3 objects

			time 1			time 2			dependencies		
PC	*counts*	μ	λ_{11}	λ_{21}	λ_{31}	λ_{12}	λ_{22}	λ_{32}	$\zeta_{(12)}$	$\zeta_{(13)}$	$\zeta_{(23)}$
(12)	$n_{(12)++}$	1	1	-1	0	1	-1	0	1	0	0
(12)	$n_{(12)-+}$	1	-1	1	0	1	-1	0	-1	0	0
(12)	$n_{(12)+-}$	1	1	-1	0	-1	1	0	-1	0	0
(12)	$n_{(12)--}$	1	-1	1	0	-1	1	0	1	0	0
(13)	⋮	2	⋮	⋮	⋮						
(23)	$n_{(23)++}$	3	0	1	-1	0	1	-1	0	0	1
(23)	$n_{(23)-+}$	3	0	-1	1	0	1	-1	0	0	-1
(23)	$n_{(23)+-}$	3	0	1	-1	0	-1	1	0	0	-1
(23)	$n_{(23)--}$	3	0	-1	1	0	-1	1	0	0	1

the design matrix **X** with: μ which is a factor (dummies for μ_1, μ_2, μ_3), variates for the objects O_{11}, O_{21}, O_{31} at time 1, for the objects O_{12}, O_{22}, O_{32} at time 2 and `within-comparison` dependencies ζ

Inglehart Index (ficticious PC data)

theory states: personal values shifted after the Second World War from a materialist (M) to a post-materialist (P) orientation (Inglehart, 1977).

the 4 values are:

1	O	Maintain order in nation	order	M
2	S	Give people more to say s	say	P
3	P	Fight rising prices	prices	M
4	F	Protect freedom of speech	freedom	P

▷ the 4 values compared pairwise
people were asked which value should have higher priority for the country

▷ This investigation was done at two timepoints time 1 and time 2

- aim of the study:
  - preference order for the 4 values at each time
  - was there a change of values in time?

(Francis, Dittrich, Hatzinger, Penn, J. Royal Statistical Society, C, 2002)

Coding

all possible comparisons for time 1 and time 2 are:

`v1.1 ,v2.1, v3.1, v4.1, v5.1, v6.1, v1.2, v2.2, v3.2, v4.2, v5.2, v6.2`

v1.1	v2.1	v3.1	v4.1	v5.1	v6.1
(12)1	(13)1	(23)1	(14)1	(24)1	(34)1
(OS)1	(OP)1	(SP)1	(OF)1	(SF)1	(PF)1
-1	1	1	1	1	1

v1.2	v2.2	v3.2	v4.2	v5.2	v6.2
((12)2	(13)2	(23)2	(14)2	(24)2	(34)2
(OS)2	(OP)2	(SP)2	(OF)2	(SF)2	(PF)2
-1	-1	1	1	1	1

We get the data (`ingle.dat`):

```
> data <- read.table("D:/talk_seminar10/tag3/inglehart/ingle.dat",
+     header = TRUE)
```


To generate a design matrix for 2 times
we use a new ♠ function `llbtrep()`

▷ get function `llbtrep()`

```
> library(prefmod)
> source("D:/talk_seminar10/tag3/inglehart/llbtrep.R")
```

▷ use `llbtrep()` :

```
> des <- llbtrep(data,  4,  mpoints = 2)
```

the options in `llbtrep()` are:

1st data object: `data`

2nd number of items: `4`

3rd number of times: mpoints = 2

▷ give names to the columns of the design matrix

```
> tail(des)
     y mu X1 X2 X3 X4 X5 X6 X7 X8 X1.1 X2.1 X3.1 X4.1 X5.1 X6.1
19  59  5  0  1  0 -1  0 -1  0  1    0    0    0    0   -1    0
20  15  5  0 -1  0  1  0 -1  0  1    0    0    0    0    1    0
21 152  6  0  0  1 -1  0  0  1 -1    0    0    0    0    0    1
22  85  6  0  0 -1  1  0  0  1 -1    0    0    0    0    0   -1
23  37  6  0  0  1 -1  0  0 -1  1    0    0    0    0    0   -1
24  26  6  0  0 -1  1  0  0 -1  1    0    0    0    0    0    1
> objnam <- paste(c("O","S","P","F"),  rep(1:2,each=4),  sep="")
> objnam
[1] "O1" "S1" "P1" "F1" "O2" "S2" "P2" "F2"
> #
> ianam<-paste("I", 1:6, sep="")
> ianam
[1] "I1" "I2" "I3" "I4" "I5" "I6"
> #
> names(des)[3:16]<-c(objnam,ianam)
> names(des)
 [1] "y"  "mu" "O1" "S1" "P1" "F1" "O2" "S2" "P2" "F2" "I1" "I2" "I3" "I4"
[15] "I5" "I6"
```

▷ fit basic model using gnm()

```
> m0 <- gnm(y~ O1+S1+P1+F1+O2+S2+P2+F2,
+           elim = mu, family = poisson, data = des)
> m0
Call:

gnm(formula = y ~ O1 + S1 + P1 + F1 + O2 + S2 + P2 + F2, eliminate = mu,
    family = poisson, data = des)

Coefficients of interest:
   O1     S1     P1     F1     O2     S2     P2     F2
0.446  0.697  0.352     NA  0.332  0.522  0.684     NA

Deviance:             10.696
Pearson chi-squared:  10.837
Residual df:          12
```


▷ fit a model including interaction terms I1+I2+I3+I4+I5+I6

- with update() we can add new terms to old model m0

```
> mia<-update(m0,~.
+                     +I1+I2+I3+I4+I5+I6)
> mia
Call:
gnm(formula = y ~ O1 + S1 + P1 + F1 + O2 + S2 + P2 + F2 + I1 +
    I2 + I3 + I4 + I5 + I6 - 1, eliminate = mu, family = poisson,
    data = des)

Coefficients of interest:
      O1        S1        P1        F1        O2        S2        P2
 0.44553   0.67816   0.33373        NA   0.31455   0.49016   0.67290
      F2        I1        I2        I3        I4        I5        I6
      NA   0.08084   0.09167   0.02663   0.04016   0.07164  -0.00232

Deviance:             5.0483
Pearson chi-squared:  5.1015
Residual df:          6
```


- compare models with and without interactions

$I1 = \zeta_{(OS)}, I2 = \zeta_{(OP)}, I3 = \zeta_{(SP)},$
$I4 = \zeta_{(OF)}, I5 = \zeta_{(SF)}, I6 = \zeta_{(FP)}$

```
> anova(m0,mia)
Analysis of Deviance Table

Model 1: y ~ O1 + S1 + P1 + F1 + O2 + S2 + P2 + F2 - 1
Model 2: y ~ O1 + S1 + P1 + F1 + O2 + S2 + P2 + F2 + I1 + I2 + I3 + I4 +
    I5 + I6 - 1
  Resid. Df Resid. Dev Df Deviance
1        12      10.70
2         6       5.05  6     5.65
```

interaction terms are within-comparison dependencies
▷ interaction terms not needed,
decisons at time 1 are independent of decision at time 2

▷ calculate the worth (♠can not use llbt.worth())
extract coefficients for time 1 (set F1 to zero) and calculate worth

```
> e1<-coef(m0)[1:4]
> #
> e1[4]<-0
> #
> w1<-exp(2*e1)/sum(exp(2*e1))
```

do the same for time 2

```
> e2<-coef(m0)[5:8]
> #
> e2[4]<-0
> #
> w2<-exp(2*e2)/sum(exp(2*e2))
```

▷ combine the worth in matrix and give names

```
> wm<-cbind(w1,w2)
> rownames(wm)<-c("order","say","price","freedom")
> colnames(wm)<-c("time 1","time 2")
```

▷ plot the worth

```
> plotworth(wm)
```