Paired Comparison Preference Models

The prefmod Package: Day2

Some Examples

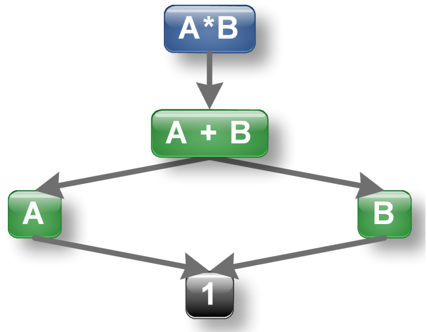
Regina Dittrich & Reinhold Hatzinger

Department of Statistics and Mathematics, WU Vienna

Two subject covariates: A and B

- various models are possible

Hierarchical models – Modeltree for 2 categorical variables



Two subject covariates: SEX and WORK

- various models are possible:

model with different preference scales for SEX *WORK interaction → `formel = ~ SEX*WORK`

internal

→ `LO + PA + MI + SG + BA + ST + LO + PA + MI + SG + BA + ST):(SEX + WORK + SEX:WORK)`

```
> msw <- llbtPC.fit(cpc, nitems = 6, undec = TRUE, formel = ~SEX *
+     WORK, elim = ~SEX * WORK, obj.names = cities)
```


model with different preference scales for `SEX` + `WORK`

```
> mworksex <- llbtPC.fit(cpc, nitems = 6, undec = TRUE, formel = ~SEX +
+     WORK, elim = ~SEX * WORK, obj.names = cities)
```

model with different preference scales for `WORK`

```
> mwork <- llbtPC.fit(cpc, nitems = 6, undec = TRUE, formel = ~WORK,
+     elim = ~SEX * WORK, obj.names = cities)
```

model with different preference scales for `SEX`

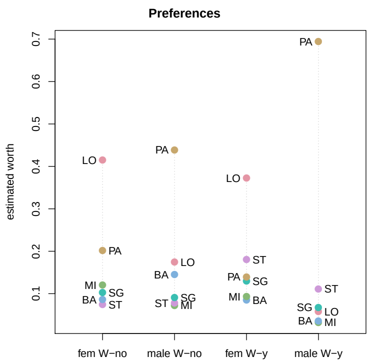
```
> msex <- llbtPC.fit(cpc, nitems = 6, undec = TRUE, formel = ~SEX,
+     elim = ~SEX * WORK, obj.names = cities)
```

null model (without subject covariates) – minimal model

```
> m0 <- llbtPC.fit(cpc, nitems = 6, undec = TRUE, formel = ~1,
+     elim = ~SEX * WORK, obj.names = cities)
```

plot for model `SEX*WORK` (stored in **msw**)

```
> wsw <- llbt.worth(msw)
> colnames(wsw) <- c("fem W-no", "male W-no", "fem W-y", "male W-y")
> plotworth(wsw, ylab = "estimated worth")
```



Interpretation of parameters ofInterest

- $\lambda_j^{O_j}$ parameter estimate for O_j for the reference group (all subjects covariates on level $= 0$ in dummy coding)
- $\lambda_{js}^{O_j S}$ change of $\lambda_j^{O_j}$ for group s

model: SEX * WORK for object PA model in **msw**

SEX1	WORK1	λ^{PA}			
SEX2	WORK1	λ^{PA}	$+\lambda^{PA:SEX2}$		
SEX1	WORK2	λ^{PA}		$+\lambda^{PA:WORK2}$	
SEX2	WORK2	λ^{PA}	$+\lambda^{PA:SEX2}$	$+\lambda^{PA:WORK2}$	$+\lambda^{PA:SEX2:WORK2}$

Interpretation of parameters ofInterest

```
> oi <- ofInterest(msw)
> cmsw <- coef(msw)[oi]
> cmsw
          LO           PA           MI           SG           BA
    0.856631     0.496625     0.237881     0.159190     0.070510
          ST            u      LO:SEX2     LO:WORK2      PA:SEX2
          NA    -1.316320    -0.126799    -0.169345    -0.236921
    PA:WORK2      MI:SEX2     MI:WORK2      SG:SEX2     SG:WORK2
    0.694191    -0.298964     0.673216     0.053633    -0.244417
     BA:SEX2     BA:WORK2      ST:SEX2     ST:WORK2 LO:SEX2:WORK2
   -0.011523     0.243747           NA           NA     0.566718
PA:SEX2:WORK2 MI:SEX2:WORK2 SG:SEX2:WORK2 BA:SEX2:WORK2 ST:SEX2:WORK2
   -0.120981    -0.429520     0.324849     0.325920           NA
```

model: SEX * WORK for object PA model in **msw**

		λ^{PA}	$+\lambda^{PA:SEX2}$	$+\lambda^{PA:WORK2}$	$+\lambda^{PA:SEX2:WORK2}$
SEX1	WORK1	0.497			
SEX2	WORK1	0.497	-0.237		
SEX1	WORK2	0.497		0.694	
SEX2	WORK2	0.497	-0.237	0.694	-0.299

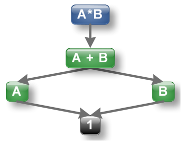
General model definitions:

`llbtPC.fit(): formel, elim`

formel	elim	
A $*$ B	A $*$ B	(o1+o2+o3) : (A+ B + A:B)
A + B	A $*$ B	(o1+o2+o3) : (A+ B)
A	A $*$ B	(o1+o2+o3) : A
B	A $*$ B	(o1+o2+o3) : B
1	A $*$ B	(o1+o2+o3)

(o1+o2+o3) . . . objects
A, B . . . are subject covariates

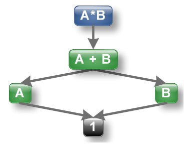
Modeltree for 2 categorical variables A and B



- step 1
  compare models: m2 = A * B with m1 = A + B
  use deviance change `anova(m2,m1)`
  if deviance change is small (large p-value) →
  term A : B not needed ⇒ proceed
  otherwise stop

- step 2
  compare models: m1 = A + B with m0 = 1
  use deviance change `anova(m1,m0)`
  if deviance change is small (large p-value) →
  A + B can be removed
  if deviance change is big (small p-value) →
  at least one main effect A, B is relevant

Modeltree for 2 categorical variables A and B



- steps 3

(1) Is A needed? compare models: m1 = A + B with m1a = A
`anova(m1,m1a)` if deviance change is big (small p-value) → A is relevant

(2) Is B needed? compare models: m1 = A + B with m1b = B
`anova(m1,m1b)` if deviance change is big (small p-value) → B is relevant

(3) check each single main effect by taking it out

(3a) compare m1a = A with m0 = 1 `anova(m1a,m0)`
(3b) compare m1b = B with m0 = 1 `anova(m1b,m0)`

▷ if there is a contradiction between (1), (2) and (3) → A,B probably correlated

Let us continue with CEMS– example: We already calculated the following models:

model with 2-way interation	msw = SEX * WORK
model with all main effects	mworksex = SEX + WORK
null model	m0 = 1

- step 1 and 2


```
> anova(msw, mworksex, m0)
Analysis of Deviance Table

Model 1: y ~ LO + PA + MI + SG + BA + ST + LO:SEX + LO:SEX:WORK + LO:WORK +
    PA:SEX + PA:SEX:WORK + PA:WORK + MI:SEX + MI:SEX:WORK + MI:WORK +
    SG:SEX + SG:SEX:WORK + SG:WORK + BA:SEX + BA:SEX:WORK + BA:WORK +
    ST:SEX + ST:SEX:WORK + ST:WORK + u - 1
Model 2: y ~ LO + PA + MI + SG + BA + ST + LO:SEX + LO:WORK + PA:SEX +
    PA:WORK + MI:SEX + MI:WORK + SG:SEX + SG:WORK + BA:SEX +
    BA:WORK + ST:SEX + ST:WORK + u - 1
Model 3: y ~ LO + PA + MI + SG + BA + ST + u - 1
  Resid. Df Resid. Dev  Df Deviance
1        99        224
2       104        232  -5     -8.4
3       114        292 -10    -60.0
> 1 - pchisq(8.4, 5)
[1] 0.13553
```

▷ we can remove interaction SEX:WORK

- steps 3

tree – left hand side (leave out WORK)

```
> anova(mworksex, msex, m0)
Analysis of Deviance Table

Model 1: y ~ LO + PA + MI + SG + BA + ST + LO:SEX + LO:WORK + PA:SEX +
    PA:WORK + MI:SEX + MI:WORK + SG:SEX + SG:WORK + BA:SEX +
    BA:WORK + ST:SEX + ST:WORK + u - 1
Model 2: y ~ LO + PA + MI + SG + BA + ST + LO:SEX + PA:SEX + MI:SEX +
    SG:SEX + BA:SEX + ST:SEX + u - 1
Model 3: y ~ LO + PA + MI + SG + BA + ST + u - 1
  Resid. Df Resid. Dev Df Deviance
1       104        232
2       109        252 -5    -19.2
3       114        292 -5    -40.8
> 1 - pchisq(19.2, 5)
[1] 0.0017640
```

▷ we can not remove WORK

- steps 3

tree – right hand side (leave out SEX)

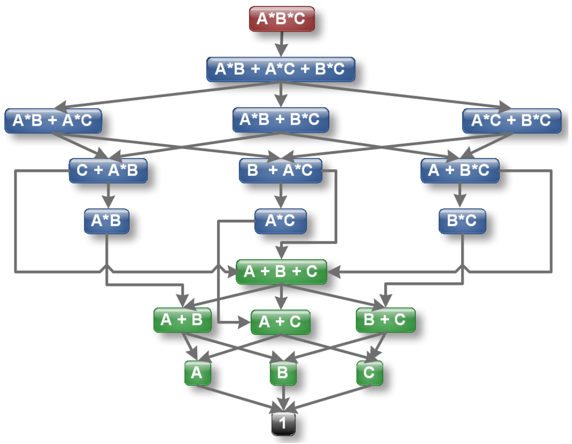
```
> anova(mworksex, mwork, m0)
Analysis of Deviance Table

Model 1: y ~ LO + PA + MI + SG + BA + ST + LO:SEX + LO:WORK + PA:SEX +
    PA:WORK + MI:SEX + MI:WORK + SG:SEX + SG:WORK + BA:SEX +
    BA:WORK + ST:SEX + ST:WORK + u - 1
Model 2: y ~ LO + PA + MI + SG + BA + ST + LO:WORK + PA:WORK + MI:WORK +
    SG:WORK + BA:WORK + ST:WORK + u - 1
Model 3: y ~ LO + PA + MI + SG + BA + ST + u - 1
  Resid. Df Resid. Dev Df Deviance
1       104        232
2       109        274 -5    -41.7
3       114        292 -5    -18.4
```

▷ we can not remove SEX

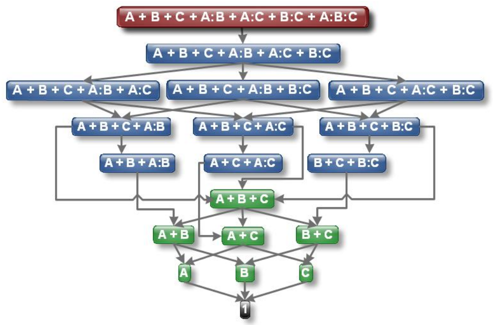
▷ final model is: SEX + WORK

→ LO + PA + MI + SG + BA + ST +
LO + PA + MI + SG + BA + ST):(SEX + WORK)

Modeltree for 3 categorical variables A, B and C
short " * " notation

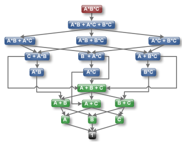


Modeltree for 3 categorical variables A, B and C
extended notation (A * B = A + B + A:B)



▷ Rule: no term can be taken out as long this term is included in a higher order term
e.g. A can not be taken out if A:B is still in model
e.g. A:B can not be taken out if A:B:C is still in model

Complicated with complex models - need strategy



▷ e.g. Christens (1997), Loglinear Models, Springer:

model with 3-way interation	m3 = A * B * C
model with all 2-way interations	m2 = A*B + A*C+ B*C
model with all main effects	m1 = A + B+ C
null model	m0 = 1

- compare deviances of m3 with m2, m1 m0 until significant
- in the group above should be the appropriate model (final model)

`anova(m3,m2)`– if not significant – we do not need 3-way interaction
`anova(m3,m1)`– if significant ▷ at least one 2-way interaction is needed

Compare model fitting with `llbtPC.fit()`, `gnm()`

- Remember: fit model with `llbtPC.fit()` for `SEX*WORK`

```
> msw <- llbtPC.fit(cpc, nitems = 6, undec = TRUE, formel = ~SEX *
+     WORK, elim = ~SEX * WORK, obj.names = cities)
> msw
```

▷ now we can easily generate worth by using

```
> wsw <- llbt.worth(msw)
```

▷ and plot results by

```
> plotworth(wsw)
```

`gnm()` generalized nonlinear models (Turner, Firth)

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

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

(a) `llbt.design()` without subject covariates

```
> d1 <- llbt.design(cpc, 6, objnames = cities)
```

```
> head(d1)
    y mu g0 g1 g2 LO PA MI SG BA ST
1 186  1  1  0  0  1 -1  0  0  0  0
2  26  1  0  1  0  0  0  0  0  0  0
3  91  1  0  0  1 -1  1  0  0  0  0
4 221  2  1  0  0  1  0 -1  0  0  0
5  26  2  0  1  0  0  0  0  0  0  0
6  56  2  0  0  1 -1  0  1  0  0  0
```


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

(b) `llbt.design()` select subject covariates

```
> d3<-llbt.design(cpc,6,objnames=cities,
+ cov.sel=c("SEX", "WORK"))
> head(d3)
    y mu g0 g1 g2 LO PA MI SG BA ST SEX WORK
1  90  1  1  0  0  1 -1  0  0  0  0   1    1
2  10  1  0  1  0  0  0  0  0  0  0   1    1
3  46  1  0  0  1 -1  1  0  0  0  0   1    1
4 101  2  1  0  0  1  0 -1  0  0  0   1    1
5  11  2  0  1  0  0  0  0  0  0  0   1    1
6  34  2  0  0  1 -1  0  1  0  0  0   1    1
```

▷ IMPORTANT: categorical covariates must be declared as `factor()`

```
> d3$SEX <- factor(d3$SEX)
> d3$WORK <- factor(d3$WORK)
```

▷ **step 2:** fit a model using `gnm()` (design matrix in `d3`)

- Main effect model **SEX**

```
> mds<-gnm(y ~ LO+PA+MI+SG+BA+ST + (LO+PA+MI+SG+BA+ST):SEX + g1,
+ elim=mu:SEX:WORK,
+ family=poisson,
+ data=d3)

> mds
Call:
gnm(formula = y ~ LO + PA + MI + SG + BA + ST + (LO + PA + MI +
    SG + BA + ST):SEX + g1, eliminate = mu:SEX:WORK, family = poisson,
    data = d3)

Coefficients of interest:
      LO        PA        MI        SG        BA        ST        g1
 0.84441   0.51770   0.26048   0.15052   0.07880        NA  -1.32040
 LO:SEX2   PA:SEX2   MI:SEX2   SG:SEX2   BA:SEX2   ST:SEX2
-0.09947  -0.23418  -0.31177   0.06476   0.00429        NA

Deviance:            251.55
Pearson chi-squared: 242.26
Residual df:         109
```


▷ general model definitions in `gnm()`

```
> gnm(y ~ LO+PA+MI+SG+BA+ST + (LO+PA+MI+SG+BA+ST):SEX + g1,
+ elim=mu:SEX:WORK,
+ family=poisson,
+ data=d3)
```

formula = $y \sim$	eliminate =
(o1+o2+o3)+(o1+o2+o3) : (A * B)	mu : A : B
(o1+o2+o3)+(o1+o2+o3) : (A + B)	mu : A : B
(o1+o2+o3)+(o1+o2+o3) : A	mu : A : B
(o1+o2+o3)+(o1+o2+o3) : B	mu : A : B
(o1+o2+o3)	mu : A : B

y ... is the dependent variable (counts $n_{(o1 \succ o2)}$)
(o1+o2+o3) ... objects
A, B ... are subject covariates
$g1$ ... is the undecided parameter
family ... defines the distribution, here poisson
data are generated by `llbt.design()`

- To plot the results we **can not** use `llbt.worth`

▷ **step 3:** from model **mds** we extract the parameters of interest and calculate the parameters for both groups

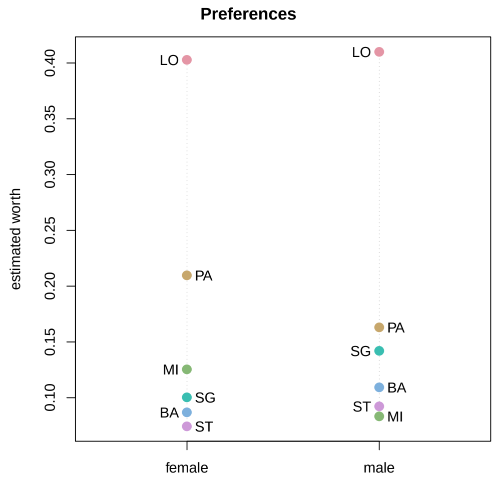
```
> c3 <- coef(mds)
> c3
Coefficients of interest:
        LO         PA         MI         SG         BA         ST
 0.8444125  0.5177037  0.2604805  0.1505150  0.0788050         NA
        g1    LO:SEX2    PA:SEX2    MI:SEX2    SG:SEX2    BA:SEX2
-1.3203958 -0.0994717 -0.2341781 -0.3117667  0.0647604  0.0042943
   ST:SEX2
        NA
> s1 <- c(c3[1:5], 0)
> s2 <- s1 + c(c3[8:12], 0)
> s1
      LO       PA       MI       SG       BA
0.844413 0.517704 0.260481 0.150515 0.078805 0.000000
> s2
       LO        PA        MI        SG        BA
 0.744941  0.283526 -0.051286  0.215275  0.083099  0.000000
```


- make matrix with estimates for both groups and calculate worth

```
> est <- cbind(s1, s2)
> est
         s1        s2
LO 0.844413  0.744941
PA 0.517704  0.283526
MI 0.260481 -0.051286
SG 0.150515  0.215275
BA 0.078805  0.083099
   0.000000  0.000000
```

- calculate worth

```
> worthm <- apply(est, 2, function(x) exp(2 * x)/sum(exp(2 *
+     x)))
> rownames(worthm) <- cities
> colnames(worthm) <- c("female", "male")

> plotworth(worthm, ylab = "estimated worth")
```



Object Specific Covariates

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 are interested if universities with a common attribute can be regarded as a group having the same rank
- consider the attribute **LAT** (with two levels): universities are either located south or north
- the universities **LO**, **SG**, **ST** located north: values of LAT are 0
- the universities **PA**, **MI**, **BA** located south: values of LAT are 1

The values for **LAT** are given as follows:

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

Example: CEMS

$$\begin{pmatrix} LO & PA & MI & SG & BA & ST \\ 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} LAT \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} mLAT \\ -1 \\ 1 \\ -1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ -1 \end{pmatrix}$$

$$mLAT = \begin{cases} 1 & \text{if any south uni } (PA, MI, BA) \succ \text{ to any north uni} \\ -1 & \text{if any north uni } (LO, SG, ST) \succ \text{ to any south uni} \\ 0 & \text{if north uni's compared with each other} \\ 0 & \text{if south uni's compared with each other} \end{cases}$$

Function: llbt.design()

- To fit a model with object covariates (attributes) llbtPC.fit() **can not** be used

▷ USE: llbt.design()
it is more flexible e.g. to modify design matrix

- first – generate the design matrix for all objects into data frame by

```
> des0 <- llbt.design(cpc, 6, objnames = cities)
> head(des0)
    y mu g0 g1 g2 LO PA MI SG BA ST
1 186  1  1  0  0  1 -1  0  0  0  0
2  26  1  0  1  0  0  0  0  0  0  0
3  91  1  0  0  1 -1  1  0  0  0  0
4 221  2  1  0  0  1  0 -1  0  0  0
5  26  2  0  1  0  0  0  0  0  0  0
6  56  2  0  0  1 -1  0  1  0  0  0
```


- fit model for all objects using gnm()

```
> md6 <- gnm(y ~ LO + PA + MI + SG + BA + ST + g1, elim = mu,
+     family = poisson, data = des0)

> md6
Call:
gnm(formula = y ~ LO + PA + MI + SG + BA + ST + g1, eliminate = mu,
    family = poisson, data = des0)

Coefficients of interest:
     LO       PA       MI       SG       BA       ST       g1
 0.7906   0.3974   0.1045   0.1820   0.0805       NA  -1.3262

Deviance:            140.48
Pearson chi-squared: 142.7
Residual df:         24
```


object specific covariate

▷ first – **generate object covariate:**
reparameterizing the objects (cf. LLTM)

```
> LAT <- c(0, 1, 1, 0, 1, 0)
> objects <- as.matrix(des0[6:11])
> mLAT <- objects %*% LAT
> head(mLAT)
     [,1]
[1,]   -1
[2,]    0
[3,]    1
[4,]   -1
[5,]    0
[6,]    1
```


- fit model for mLAT (instead of objects) using using gnm()

```
> md1 <- gnm(y ~ mLAT + g1, elim = mu, family = poisson, data = des0)
> md1
Call:
gnm(formula = y ~ mLAT + g1, eliminate = mu, family = poisson,
    data = des0)

Coefficients of interest:
  mLAT      g1
-0.112  -1.401

Deviance:            692.1
Pearson chi-squared: 676.03
Residual df:         28

> anova(md6, md1)
Analysis of Deviance Table

Model 1: y ~ LO + PA + MI + SG + BA + ST + g1 - 1
Model 2: y ~ mLAT + g1 - 1
  Resid. Df Resid. Dev Df Deviance
1        24        140
2        28        692 -4     -552
```

fit model with subject covariates using gnm

▷ generate the design matrix, but include SEX and WORK

```
> des <- llbt.design(cpc, 6, objnames = cities, cov.sel = c("SEX",
+     "WORK"))
```

▷ IMPORTANT: categorical covariates must be declared as `factor()`

```
> des$SEX <- factor(des$SEX)
> des$WORK <- factor(des$WORK)
```

- fit model using gnm()

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


object specific covariate:
reparameterizing the objects

```
> LAT <- c(0, 1, 1, 0, 1, 0)
> objects <- as.matrix(des[6:11])
> mLAT <- objects %*% LAT
```

▷ fitting a specific model:
different preference scales for SEX
but Latin cities (mLAT) combined with WORK

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


Example: CEMS exchange programme

- We also considered the following university attributes `EC, MS, FS`:

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


```
> EC <- c(1, 0, 1, 0, 0, 0)
> MS <- c(0, 1, 0, 0, 1, 0)
> FS <- c(0, 0, 0, 1, 0, 1)
> att <- cbind(EC, MS, FS)
> att
     EC MS FS
[1,]  1  0  0
[2,]  0  1  0
[3,]  1  0  0
[4,]  0  0  1
[5,]  0  1  0
[6,]  0  0  1
> head(des)
    y mu g0 g1 g2 LO PA MI SG BA ST SEX WORK
1  90  1  1  0  0  1 -1  0  0  0  0   1    1
2  10  1  0  1  0  0  0  0  0  0  0   1    1
3  46  1  0  0  1 -1  1  0  0  0  0   1    1
4 101  2  1  0  0  1  0 -1  0  0  0   1    1
5  11  2  0  1  0  0  0  0  0  0  0   1    1
6  34  2  0  0  1 -1  0  1  0  0  0   1    1
```

Example: CEMS

$$\begin{pmatrix} LO & PA & MI & SG & BA & ST \\ 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} EC & MS & FS \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} EC & MS & FS \\ 1 & -1 & 0 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & -1 & 1 \end{pmatrix}$$


```
> des_obj <- as.matrix(des[, cities])
> des_obj <- des_obj %*% att
> head(des_obj)
     EC MS FS
[1,]  1 -1  0
[2,]  0  0  0
[3,] -1  1  0
[4,]  0  0  0
[5,]  0  0  0
[6,]  0  0  0
> des2 <- data.frame(des, des_obj)
> head(des2)
    y mu g0 g1 g2 LO PA MI SG BA ST SEX WORK EC MS FS
1  90  1  1  0  0  1 -1  0  0  0  0   1    1  1 -1  0
2  10  1  0  1  0  0  0  0  0  0  0   1    1  0  0  0
3  46  1  0  0  1 -1  1  0  0  0  0   1    1 -1  1  0
4 101  2  1  0  0  1  0 -1  0  0  0   1    1  0  0  0
5  11  2  0  1  0  0  0  0  0  0  0   1    1  0  0  0
6  34  2  0  0  1 -1  0  1  0  0  0   1    1  0  0  0

> mneu1 <- gnm(y~ EC + MS + FS + g1, elim=mu:SEX:WORK,
+            family=poisson, data=des2)
```


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)

Example: Make Design Matrix

- Preparation

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

```
> 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
```

21 responses where both responses $0, 1$ occur

and one covariate `cov` with levels $(1, 2)$

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

```
> des5 <- llbt.design(d, nitems = 7, cov.sel = "cov")
```

3. make a factor `mm` for μ of length 42 repeated two times (number of different comparisons * 2)

```
> des5$mm <- gl(42, 2)
```

4. make a variate γ for positition effect
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 $= 0$

$10\,10\,10\,10 \cdots$

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 $= 1$

$01\,01\,01\,01 \cdots$

```
> des5$pos <- c(rep(1:0, 21), rep(0:1, 21))
```


5. read in number of wins and losses and assign it to yy
(dont use y as **R** gives priority to data.frame specifications)

```
> data(baseball)
> des5$yy <- baseball
```

6. Give names to the teams

```
> 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
```

The columns `y`,`mu` `g0`,`g1` and `cov` are only auxiliary variables

7. fit the basic model including a position effect δ using `gnm()`

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

8. Calculate the worth

Note: `llbt.worth` can not be used because model was not fitted by `llbtPC.fit` !

We get the par. est. for the teams (when play away) and
the position effect by

```
> est <- coef(res5)
> est
Coefficients of interest:
    MIL     DET     TOR      NY     BOS     CLE     BAL     pos
0.80978 0.73768 0.66355 0.64067 0.57190 0.35235      NA 0.30226
> team <- est[1:7]
> team[7] <- 0
> pos <- est[8]
```

```
> nom <- exp(2 * team)
> den <- sum(exp(2 * team))
> worth_away <- nom/den
> worth_away
     MIL      DET      TOR       NY      BOS      CLE      BAL
0.220014 0.190469 0.164226 0.156879 0.136720 0.088131 0.043560
```

9. How can we interpret the "home effect" ?

- Parameter estimates are log Odds and
- Odds are exp(parameter estimates)
- Playing at home increases the odds to win a game by

$$\exp(0.3023) = 1.35$$

9.a Now compare two teams (TOR and NY) when playing each other (both away)

- odds for TOR (Toronto) to win against NY (New York) (both teams play away e.g. in a third city):

```
> exp(2 * team[3] - 2 * team[4])
   TOR
1.0468
```

odds is close to one (1.0468)

About the same chance to win when play against each other (both not playing at home)

9.b Compare TOR and NY when playing against each other; TOR playing at home

- if Toronto is playing away and NY at home

the odds for TOR to win against NY is:

```
> oddsTOR <- exp(2 * team[3] - 2 * team[4] + pos)
```

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

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 ×
number of possible decisions (response categories) ×
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