Paired Comparison Preference Models

The prefmod Package: Day 4

Pattern Models

Regina Dittrich & Reinhold Hatzinger
Institute for Statistics and Mathematics, WU Vienna

Paired Comparison Response Patterns

What are paired comparison response patterns?

comparison	(12)	(13)	(23)	...
response	$(1 \succ 2)$	$(3 \succ 1)$	$(2 \succ 3)$	...
random variable	Y_{12}	Y_{13}	Y_{23}	...

we model the complete responses $\mathbf{Y}$ simultanously

$$\mathbf{Y} = (Y_{12}, Y_{13}, \dots, Y_{1J}, \dots, Y_{J-1,J})$$

The BT Model as a Pattern Model

$$Y_{jk} = \begin{cases} 1 & \text{if object } O_j \text{ is preferred to } O_k \quad (j \succ k) \\ -1 & \text{if object } O_k \text{ is preferred to } O_j \quad (k \succ j) \end{cases}$$

$$P(j \succ k) = P(Y_{jk} = 1) = \frac{\pi_j}{\pi_j + \pi_k} = c\left(\frac{\sqrt{\pi_j}}{\sqrt{\pi_k}}\right)^{y_{jk}}$$

for 3 objects we have 3 comparisons: (12) (13) (23)

the probability for a specific response pattern e.g. (1, 1, 1) which means $(1 \succ 2)$, $(1 \succ 3)$, $(2 \succ 3)$ is given by:

$$p(1,\,1,\,1) = \delta\left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_2}}\right)\left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_3}}\right)\left(\frac{\sqrt{\pi_2}}{\sqrt{\pi_3}}\right)$$

the log-linear pattern model can be written as:
$\ln m(1,\,1,\,1) = \ln\delta + 2\lambda_1 - 2\lambda_3$

The BT Model as a Pattern Model

in general: we model the probability for a pattern by

$$\boxed{p(y_{12}, y_{13}, y_{23}) = \delta\left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_2}}\right)^{y_{12}}\left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_3}}\right)^{y_{13}}\left(\frac{\sqrt{\pi_2}}{\sqrt{\pi_3}}\right)^{y_{23}}}$$

the log-linear pattern model can be written as:

$$\begin{aligned} \ln m(y_{12}, y_{13}, y_{23}) = \\ \ln\delta + (y_{12} + y_{13})\lambda_1 + (y_{23} - y_{12})\lambda_2 + (-y_{13} - y_{23})\lambda_3 \end{aligned}$$

for a certain response pattern $(1,1,1)$ it is:

$$\ln m(1,\,1,\,1) = \ln\delta + 2\lambda_1 - 2\lambda_3$$

Design structure - two responses

					$\ln\delta$	λ_1	λ_2	λ_3
pattern	y_{12}	y_{13}	y_{23}	counts	const	x_1	x_2	x_3
s_1	1	1	1	n_1	1	2	0	-2
s_2	1	1	−1	n_2	1	2	-2	0
s_3	1	−1	1	n_3	1	0	0	0
s_4	1	−1	−1	n_4	1	0	-2	2
s_5	−1	1	1	n_5	1	0	2	-2
s_6	−1	1	−1	n_6	1	0	0	0
s_7	−1	−1	1	n_7	1	-2	2	0
s_8	−1	−1	−1	n_8	1	-2	0	2

x_i how often O_i is preferred in a pattern minus
how often O_i not preferred in a pattern

▹ the number of patterns is $2^{\binom{3}{2}} = 8$

$$\boxed{\#\text{possible response categories to the power of }\#\text{comparisons}}$$

Pattern Model Extensions: Overview

all extensions as before also apply to pattern models

- undecided ($3^{\binom{J}{2}}$ different patterns)
- subject covariates
- object specific covariates

but we can give up the assumption of independent decisions

- by introducing between-comparisons dependencies

and we can also deal with various response formats

- (real) paired comparisons
- ranking data
- rating data

Between-comparisons Dependencies

one important feature of the pattern models is

- we can give up the (unrealistic) assumption of independent decisions
- we assume that dependencies between responses come from repeated evaluation of the same objects in PC comparing (j with k) and comparing (j with l) the assessment of common object j might be similar in both comparisons

we can now include dependence terms of the form:

$$\boxed{\theta_{(jk),(jl)}}$$

for pairs of comparisons with one object in common

Dependencies are nuisance parameters

but have interpretation of • pairwise "coherent decisions"
we look at two **comparisons** y_{12}, y_{13} given $y_{(2>3)}$

expected numbers	y_{12}	y_{13}	y_{23}
$m_{11\|2}$	1	1	1
$m_{13\|2}$	1	-1	1
$m_{21\|2}$	-1	1	1
$m_{23\|2}$	-1	-1	1

$y_{(2>3)}$		y_{13}	
		$y_{(1>3)}$	$y_{(3>1)}$
y_{12}	$y_{(1>2)}$	$m_{11\|2}$	$m_{13\|2}$
	$y_{(2>1)}$	$m_{21\|2}$	$m_{23\|2}$

$$\boxed{OR = \frac{m_{11|2}m_{23|2}}{m_{21|2}m_{13|2}}}$$

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

"coherent" decisions: common object is preferred or not preferred in both comparisons

$$\boxed{\ln OR = 4\theta_{(jk),(jl)}}$$

dependence parameters
(option: `ia = T` in `prefmod`)

Design structure - with dependencies

				γ	λ_1	λ_2	λ_3	$\theta_{12.13}$	$\theta_{12.23}$	$\theta_{13.23}$
y_{12}	y_{13}	y_{23}	counts	c	x_1	x_2	x_3	$y_{12}y_{13}$	$y_{12}y_{23}$	$y_{13}y_{23}$
1	1	1	n_1	1	2	0	-2	1	1	1
1	1	−1	n_2	1	2	-2	0	1	-1	-1
1	−1	1	n_3	1	0	0	0	-1	1	-1
1	−1	−1	n_4	1	0	-2	2	-1	-1	1
−1	1	1	n_5	1	0	2	-2	-1	-1	1
−1	1	−1	n_6	1	0	0	0	-1	1	-1
−1	−1	1	n_7	1	-2	2	0	1	-1	-1
−1	−1	−1	n_8	1	-2	0	2	1	1	1

What makes a good teacher ?

239 education students at Vienna were asked to compare qualities of a good teacher in 2006 through a complete paired comparison experiment

Quality of the teachers are:

ST Structure of instruction

CM Class Management: productive environment - not wasting time

AC Activity: Success in getting students to participate

SU Support: Looking after every single pupil

Data

- 4 items ST, CM, AC, SU
- 3 subject covariates
- SEX ... (1 = female) (2 = male)
- SCH ... school (1 = secondary) (2 = vocational) (3 = university)
- LEI ... achievement test (0 – 100 test points)
- no undecided
- but missing values (`NA`)

Coding for each comparison (V1, ... V6) is

$$(jk) = \begin{cases} 0 & \text{if first item is preferred to second item} \quad (j > k) \\ 1 & \text{if second item is preferred to first itemt} \quad (k > j) \end{cases}$$

First respondent (subject covariates always at the end)

V1	V2	V3	V4	V5	V6	SEX	SCH	LEI
(12) (ST,CM)	(13) (ST,AC)	(23) (CM,AC)	(14) (ST,SU)	(24) (CM,SU)	(34) (AC,SU)			
0	0	1	1	1	1	2	1	30

file = "teacher4items.dat"

REMARK: if coding is not 1, –1 but e.g. 0, 1 the smaller number means first item in comparison is preferred

---

Function: `pattPC.fit()`

user friendly function (restricted functionality)

Read in data:

```
> teacher4 <- read.table("../data/teacher4items.dat",
+     header = TRUE)
> teacher4 <- na.omit(teacher4)
> it4 <- c("ST", "CM", "AC", "SU")
```

basic model (no dependencies)

```
> mtp <- pattPC.fit(teacher4, nitems =4, undec = F, ia = F,
+                  formel =~ 1,  elim = ~SEX*SCH, obj.names  =  it4)
```

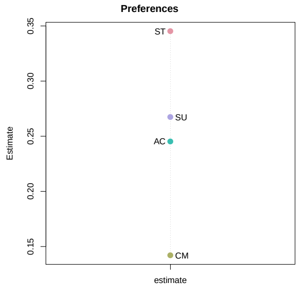
$$\text{Options} = \left\{ \begin{array}{ll} \text{teacher4} & \text{data.frame} \\ \text{nitems} = 4 & 4 \text{ items} \\ \text{undec} = \text{F} & \text{no undecided} \\ \text{ia} = \text{F} & \text{no dependencies} \\ \text{formel} =\sim 1 & \text{model is ST+CM+AC+SU} \\ \text{elim} =\sim \ \text{SEX} * \ \text{SCH} & \text{defines maximal table} \\ \text{obj.names} = \text{it4} & \text{names of items} \end{array} \right.$$

many other options: ▹ see ?pattPC.fit

Functions: `patt.worth()`, `plotworth()`

- Calculate worth and plot for basic model `mtp`

```
> wp <-  patt.worth(mtp)
> #wl <-  patt.worth(mtp, outmat = "lambda")
> plotworth(wp)
```



- compare results of llbt model and pattern model

llbt model: `llbtPC.fit()`

```
> ml  <- llbtPC.fit(teacher4, nitems =4, undec = F,
+                   formel =~ 1,  elim = ~SEX*SCH, obj.names  =  it4)
> coef(ml)
Coefficients of interest:
     ST      CM      AC      SU
 0.1277 -0.3166 -0.0428      NA
```

pattern model: `pattPC.fit()`

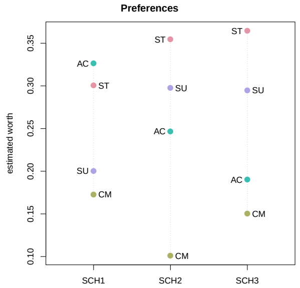
```
> mp  <- pattPC.fit(teacher4, nitems =4, undec = F, ia = F,
+                   formel =~ 1,  elim = ~SEX*SCH, obj.names  =  it4)
> coef(ml)
Coefficients of interest:
     ST      CM      AC      SU
 0.1277 -0.3166 -0.0428      NA
```

estimates are the same

---

include categorical subject covariate e.g. schooltype `SCH` using `pattPC.fit`

```
> mtp_sch <- pattPC.fit(teacher4, nitems =4, undec = F, ia = F,
+                        formel =~ SCH,  elim = ~SEX*SCH, obj.names  =  it4)

> w_sch <- patt.worth(mtp_sch)
> plotworth(w_sch, ylab = "estimated worth")
```



---

basic pattern model: without dependencies

```
> mia0 <- pattPC.fit(teacher4, nitems =4, undec = F, ia = F,
+                       formel =~ 1,  elim = ~1, obj.names  =  it4)

> coef(mia0)
[1]  0.1277 -0.3166 -0.0428
```

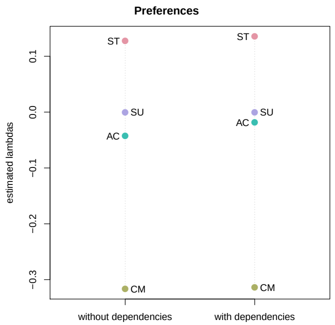
- include dependencies - option: `ia = TRUE`

```
> mia1 <- pattPC.fit(teacher4, nitems = 4, undec = F, ia = T,
+                       formel = ~1,  elim = ~1, obj.names  =  it4)
```

```
> mia1
Results of pattern model for paired comparison

Call:
pattPC.fit(obj = teacher4, nitems = 4, formel = ~1, elim = ~1,
    obj.names = it4, undec = F, ia = T)

Deviance:  103
log likelihood:  -820

no of iterations:  22  (Code: 1 )

       estimate     se      z p-value
ST       0.1356 0.0594  2.284  0.0224
CM      -0.3140 0.0606 -5.179  0.0000
AC      -0.0184 0.0513 -0.359  0.7196
I12.13  -0.1512 0.0783 -1.932  0.0534
I12.14   0.1789 0.0815  2.195  0.0282
I13.14  -0.0765 0.0733 -1.044  0.2965
I12.23   0.3016 0.0845  3.569  0.0004
I12.24  -0.3604 0.0849 -4.246  0.0000
I23.24  -0.0191 0.0815 -0.234  0.8150
I13.23   0.0590 0.0714  0.826  0.4088
I13.34  -0.2062 0.0704 -2.930  0.0034
I23.34   0.0184 0.0702  0.261  0.7941
I14.24  -0.0323 0.0793 -0.407  0.6840
I14.34   0.1221 0.0715  1.707  0.0878
I24.34   0.0246 0.0733  0.336  0.7369
```

- Plot basic model without and with dependencies

```
> w_mia0 <- patt.worth(mia0, outmat = "lambda")
> w_mia1 <- patt.worth(mia1, outmat = "lambda")
> w_01 <- cbind(w_mia0, w_mia1)
> colnames(w_01) <- c("without dependencies", "with dependencies")
> plotworth(w_01, ylab = "estimated lambdas")
```



Model fitting for more specialised models

use `patt.design()` and `gnm()`

- categorical subject covariates – use ♠ `cat.scovs = c(" ")`
- no dependencies – ♠ `ia = F`

(1) generate the design matrix with `patt.design()`

```
> des_t0<-patt.design(teacher4, 4,ia = F, cat.scovs = c("SCH"), objnames = it4)
> head(des_t0)
  y ST CM AC SU SCH
1 2 -3 -1  1  3   1
2 0 -3 -1  3  1   1
3 0 -3  1  1  1   1
4 0 -3  1  3 -1   1
5 2 -1 -1  1  1   1
6 1 -1 -1  3 -1   1
```

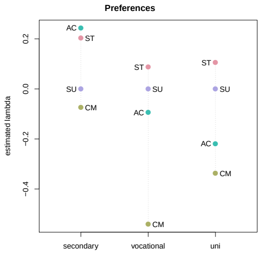
---

(2) fit model using `gnm()`

```
> t0_sch <- gnm(y ~ ST+CM+AC+SU + (ST+CM+AC+SU):SCH,
+               elim = SCH,
+               family = poisson,
+               data = des_t0)
```

(3) to plot the results we can use ♠ `patt.worth` and `plotworth()`

```
> est0 <- patt.worth(t0_sch, outmat = "lambda")
> rownames(est0) <- it4
> colnames(est0) <- c("secondary", "vocational", "uni")
> plotworth(est0, ylab = "estimated lambda")
```



▷ categorical subject covariates and dependencies

- for categorical subject covariates – use ♠ `cat.scovs = c(" ")`
- for dependencies – use ♠ `ia = T`

(1) generate the design matrix with `patt.design()`

```
> des_t1<-patt.design(teacher4, 4, ia = T, cat.scovs = c("SCH"), objnames = it4)


> des_t1[1,]
  y ST CM AC SU I12.13 I12.14 I13.14 I12.23 I12.24 I23.24 I13.23
1 2 -3 -1  1  3      1      1      1      1      1      1      1
  I13.34 I23.34 I14.24 I14.34 I24.34 SCH
1      1      1      1      1      1   1
```

(2) fit model using `gnm()`

```
> t1_sch <- gnm(y ~ ST+CM+AC+SU + (ST+CM+AC+SU):SCH +
+               I12.13+I12.14+I13.14+I12.23+I12.24+I23.24+I13.23+I13.34+
+               I23.34+I14.24+I14.34+I24.34,
+               elim = SCH,
+               family = poisson,
+               data = des_t1)
```

---

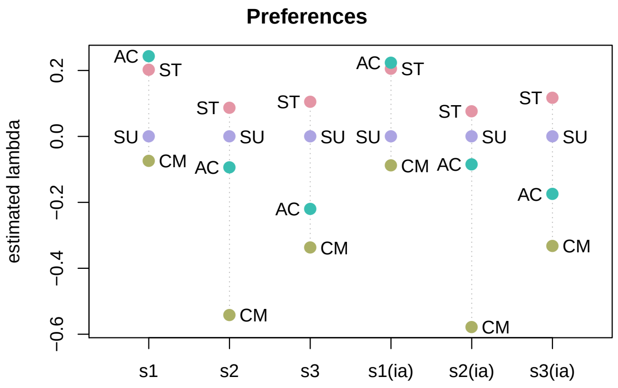
(3) comparing results with and without dependencies

```
> est_t0 <- patt.worth(t0_sch, outmat = "lambda")
> est_t1 <- patt.worth(t1_sch, outmat = "lambda")
> e_01 <- cbind(est_t0, est_t1)
> colnames(e_01) <- c("s1", "s2", "s3", "s1(ia)", "s2(ia)",
+     "s3(ia)")
> rownames(e_01) <- it4
```

(4) compare results using `anova()`

```
> anova(t0_o1, t0_o2)
Analysis of Deviance Table

Model 1: y ~ focus
Model 2: y ~ ST + CM + AC + SU
  Resid. Df Resid. Dev Df Deviance
1        62        228
2        60        175  2       53
```

```
> plotworth(e_01, ylab = "estimated lambda")
```



▷ Numerical Subject Covariates

basic pattern-model has to be extended for each individual i

$$\ln m(1,1,1) = \ln \delta + 2\lambda_1 - 2\lambda_3$$

We model $\lambda_{i,j}$ for one subject covariate x through the relationship

$$\boxed{\lambda_{i,j} = \lambda_j + \beta_j x_i}$$

where x_i is the covariate value for individual i

For each object j, there is one β-parameter which describes the effect of the covariate on that item.

† the design matrix has to be replicated for each different value of subject covariate – it might become rather large - use with care

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

♠*option:* `num.scovs = " "`

```
> des_n <- patt.design(teacher4, 4, num.scovs = "LEI", objnames = it4 )
> des_n[1:8,]
  y ST CM AC SU LEI CASE
1 0 -3 -1  1  3  30    1
2 0 -3 -1  3  1  30    1
3 0 -3  1  1  1  30    1
4 0 -3  1  3 -1  30    1
5 0 -1 -1  1  1  30    1
6 0 -1 -1  3 -1  30    1
7 0 -1  1  1 -1  30    1
8 0 -1  1  3 -3  30    1
```

▷ **step 2:** fit model

CASE is a subject covariate, therefore ♠ `eliminate = CASE`

```
> t_n <- gnm(y ~ ST+CM+AC+SU + (ST+CM+AC+SU):LEI,
+               elim = CASE,
+               family = poisson,
+               data = des_n)
```

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

- extract coefficients

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

- 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 LEI) to be plotted

```
> attach(teacher4)
The following object(s) are masked from 'teacher4 (position 3)':

    LEI, SCH, SEX, V1, V2, V3, V4, V5, V6
> s <- seq(min(LEI),max(LEI),0.01)
```

- we write a function to calculate worth

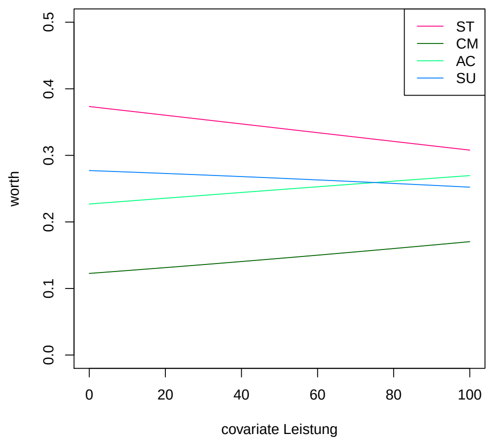
```
> ww <- function(x,a,b){exp(2*(a+b*x))/sum(exp(2*(a+b*x)))}
```

- calculate worth matrix

```
> res <- sapply(s, ww, a, b)
> res[, 1]
   ST    CM    AC    SU
0.373 0.123 0.227 0.277
```

- plot the worth

```
> plot(s,res[1,],type="l",ylim=c(0,0.5),xlim=range(LEI),
+     col= farbe[12],ylab = "worth",xlab = "covariate Leistung")
> lines(s,res[2,],col="darkgreen")
> lines(s,res[3,],col=farbe[6])
> lines(s,res[4,],col=farbe[8])
> legend("topright",rownames(res),lty=1,
+                    col=c(farbe[12],"darkgreen",farbe[c(6,8)]) )
```



Object covariate in pattern models

- We are interested if teaching qualities with a common attribute can be regarded as a group having the same rank
- consider the attribute `focus` (with two levels): universities are either located south or north
- the quality `ST`, `AC` are focused on achievement: values of focus are 1
- the quality `CM`, `SU` are focused on social aspects: values of focus are 0

The values for `focus` are given as follows:

Objects	ST	CM	AC	SU
focus	1	0	1	0

---

(1) setup object covariate – make data.frame

```
> focus <- c(1,0,1,0)
> objs  <- data.frame(focus=focus)
```

(2) generate design matrix – use ♠ `objcovs = objs`

```
> des_o1<-patt.design(teacher4, 4,ia = F, objcovs = objs, objnames = it4)
```

we look at design matrix

```
> head(des_o1)
  y ST CM AC SU focus
1 3 -3 -1  1  3    -2
2 2 -3 -1  3  1     0
3 0 -3  1  1  1    -2
4 1 -3  1  3 -1     0
5 2 -1 -1  1  1     0
6 2 -1 -1  3 -1     2
```

vector multiplication $(3,1,-1,-3)(1,0,1,0)^T = -2$

```
> o <- des_o1[2:5]
> focus <- o %*% focus
```

---

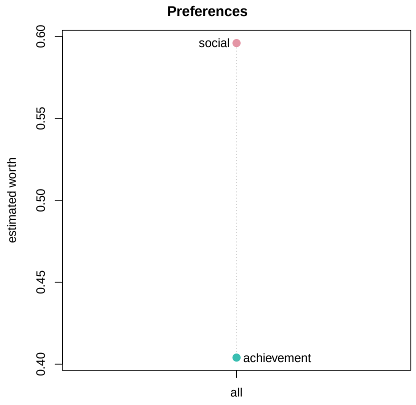
(3) fit model for object covariate

```
> t0_o1 <- gnm(y ~ focus, family = poisson, data = des_o1)
> t0_o1
Call:
gnm(formula = y ~ focus, family = poisson, data = des_o1)

Coefficients:
(Intercept)        focus
      1.141        0.195

Deviance:            228
Pearson chi-squared: 292
Residual df:         62
```

(4) calculate the lambdas and plot

```
> e_o1 <- patt.worth(t0_o1)
> colnames(e_o1)<-c("all" )
> rownames(e_o1)<- c("social", "achievement")
```

```
> plotworth(e_o1, ylab = "estimated worth")
```



▷ compare models:

- model with object covariate is `t0_o1` : deviance = 228, df = 62
- calculate model for all objects: `t0_o2`

```
> des_o2<-patt.design(teacher4, 4,ia = F, objnames = it4)
> t0_o2 <- gnm(y ~ ST + CM + AC + SU, family = poisson, data = des_o2)
> t0_o2
Call:

gnm(formula = y ~ ST + CM + AC + SU, family = poisson, data = des_o2)


Coefficients:
(Intercept)           ST           CM           AC           SU
     1.0110       0.1277      -0.3166      -0.0428           NA

Deviance:             175
Pearson chi-squared:  212
Residual df:          60
> p<- 1-pchisq(53,2)
```

p-value $<$ 0.00000001 – reduction not feasible!

Responsetype: Rankings

full rankings:

- people are asked to rank objects (items) regarding a certain aspect
  (e.g. alcohol of beers)
- all possible pairs are constructed afterwards
- no undecided category !

ordinal responses are transformed into paired comparisons

resulting PCs are called derived PC patterns

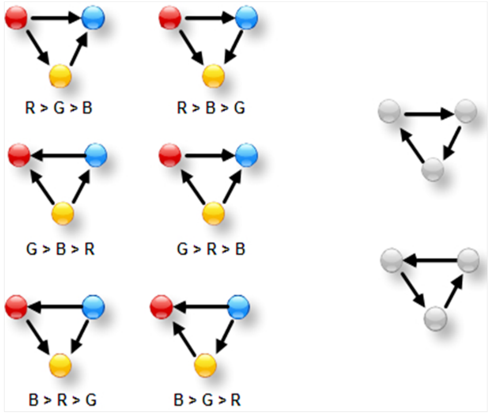
Transformation: Ranking to PC

Data				comparison		
R	G	B	Response	RG	RB	GB
1	2	3	R>G>B	1	1	1
1	3	2	R>B>G	1	1	-1
-	-	-	-	1	-1	1
2	3	1	B>R>G	1	-1	-1
2	1	3	G>R>B	-1	1	1
-	-	-	-	-1	1	-1
3	1	2	G>B>R	-1	-1	1
3	2	1	B>G>R	-1	-1	-1

- number of possible patterns is $3! = 6$ compared to $2^{\binom{3}{2}} = 8$

Transitive – Intransitive Patterns

Transitive
Patterns



Intransitive
Patterns

bei Rankings
nicht möglich

Pattern Model: Rankings

The probability for the ranking $R = 2, G = 3, B = 1$ transformed into pattern 1, −1, −1 is given by:

$$p(s_k) \Rightarrow p(y_{12}, y_{13}, y_{23}) = \delta \left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_2}}\right)^{1} \left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_3}}\right)^{-1} \left(\frac{\sqrt{\pi_2}}{\sqrt{\pi_3}}\right)^{-1}$$

$$p(2,3,1) \Rightarrow p(s_4) = p(1,-1,-1) = \delta \left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_2}}\right) \left(\frac{\sqrt{\pi_3}}{\sqrt{\pi_1}}\right) \left(\frac{\sqrt{\pi_3}}{\sqrt{\pi_2}}\right)$$

The log expected number for the ranking can be rewritten as

$$\boxed{\ln m(1,-1,-1) = \ln \delta - 2\lambda_2 + 2\lambda_3}$$

▹ number of ranked objects $J!$

Example: Rankings

Vargo (1989) collected a ranking data set
which was analysed by Critchlow, Fligner (Psychometrika, 1991)

- 32 judges were asked to rank four salad dressings according tartness.
- A low rank means very tart.

---

data file: `salad` is in `prefmod`

```
> data(salad)
```

the first and last six rankings are:

```
> head(salad)
  A B C D
1 1 2 3 4
2 1 2 3 4
3 2 1 3 4
4 2 1 4 3
5 2 1 4 3
6 2 3 1 4
> tail(salad)
   A B C D
27 4 1 3 2
28 4 2 1 3
29 4 2 1 3
30 4 2 1 3
31 4 3 1 2
32 4 3 2 1
```

▹ User friendly function `pattR.fit()`(restricted functionality)

```
> salmod <- pattR.fit(salad, nitems = 4)
> salmod
Results of pattern model for rankings

Call:
pattR.fit(obj = salad, nitems = 4)

Deviance:  22.2
log likelihood:  -77.4

no of iterations:  10  (Code: 1 )

  estimate    se     z p-value
A   -0.277 0.125 -2.23  0.0259
B    0.591 0.143  4.13  0.0000
C    0.189 0.111  1.69  0.0903
```

Does the model fit?

deviance is: 22.2 and df = #patterns − #estimates = 4! − 4 = 24 − 4

`1 - pchisq(22.2,20) = 0.33`

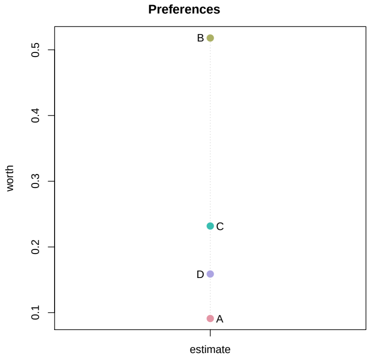
Model fitting with `patt.design()` and `gnm()`

- use ♠ `resptype = "ranking"`

```
> saldes <- patt.design(salad, nitems = 4, resptype = "ranking")
> salmod1 <- gnm(y ~ A + B + C + D, family = poisson, data = saldes)
> salmod1
Call:

gnm(formula = y ~ A + B + C + D, family = poisson, data = saldes)


Coefficients:
(Intercept)            A            B            C            D
     -0.802       -0.277        0.591        0.189           NA

Deviance:            22.2
Pearson chi-squared: 27.4
Residual df:         20
```

Functions: `patt.worth()`, `plotworth()`

```
> worth <- patt.worth(salmod1)
> plotworth(worth, ylab = "worth")
```



---

object specific covariates: use `patt.design()`, `gnm()`

- salads A - D have varying concentrations

the four pairs of concentrations of acetic and gluconic acid are:
`A = (.5, 0), B = (.5, 10.0), C = (1.0, 0), and D = (0, 10.0)`
▷ substitute each pattern of the design matrix by acetic / gluconic concentration

(1) make a data frame with 2 object covariates

```
> acid <- c(0.5,0.5,1,0)
> gluc <- c(0,10,0,10)
> conc <- data.frame(acid = acid, gluc = gluc)
> conc
  acid gluc
1  0.5    0
2  0.5   10
3  1.0    0
4  0.0   10
```

---

(2) make a design matrix –
use options ♠ `objcovs = conc` and ♠ `resptype = "ranking"`

```
> saldes2 <- patt.design(salad, nitems = 4,
+                        objcovs = conc,resptype = "ranking")
> head(saldes2)
  y  A  B  C  D acid gluc
1 2  3  1 -1 -3    1  -20
2 1  1  3 -1 -3    1    0
3 1  1 -1  3 -3    3  -40
4 0  3 -1  1 -3    2  -40
5 2 -1  3  1 -3    2    0
6 0 -1  1  3 -3    3  -20
```

(3) fit model

```
> salmod2 <- gnm(y ~ acid + gluc, family = poisson, data = saldes2)
```

deviance, df: all objects model `salmod1` : 22.2, 20
deviance, df: object covs model `salmod2` : 22.7, 21

can we simplify? YES!

```
> anova(salmod1, salmod2)
Analysis of Deviance Table

Model 1: y ~ A + B + C + D
Model 2: y ~ acid + gluc
  Resid. Df Resid. Dev Df Deviance
1        20       22.2
2        21       22.8 -1   -0.499
```

---

model `salmod2` with object covariates

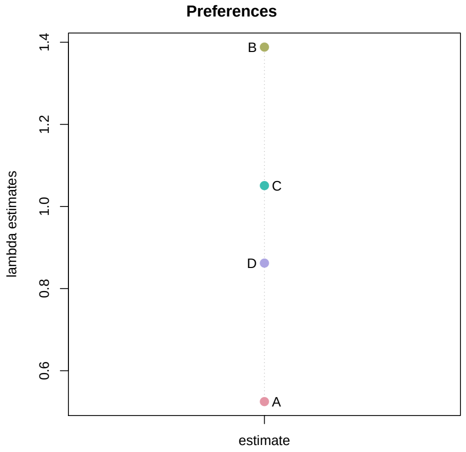
```
> summary(salmod2)
Call:

gnm(formula = y ~ acid + gluc, family = poisson, data = saldes2)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.952  -0.581  -0.335   0.182   2.250

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.7815     0.3811   -2.05     0.04
acid          1.0506     0.2074    5.07  4.1e-07
gluc          0.0863     0.0164    5.27  1.3e-07

(Dispersion parameter for poisson family taken to be 1)

Residual deviance: 22.747 on 21 degrees of freedom
AIC: 59.49

Number of iterations: 5
```

---

calculate reparameterised estimates

```
> est <- patt.worth(salmod2, outmat="lambda")
> est
  estimate
A    0.525
B    1.388
C    1.051
D    0.863
attr(,"objtable")
  acid gluc x
1  0.5    0 A
2  1.0    0 C
3  0.0   10 D
4  0.5   10 B
```

```
> plotworth(est, ylab = "lambda estimates")
```



Example: Ratings

we used a data set collected by the British Household Panel Study in 1996 where we have chosen three Likert items which ask respondents for their concern about:

- the destruction of the ozone layer (OZ)
- the high rate of unemployment (UN)
- declining moral standards (MO)

the possible answers are:

- A great deal 1
- A fair amount 2
- Not very much ... 3
- Not at all 4

low numbers mean a high concern and higher number lower concern!

Transformation: Ratings to PC

for example the Likert response pattern was

$OZ = 1, UN = 4, MO = 4$

we have 3 items and therefore 3 comparisons:
(12) =(OZ, UN)   (13) =(OZ, MO) (23) =(UN, MO)

- as OZ $\succ$ UN we assign $y_{12} = 1$
- as OZ $\succ$ MO we assign $y_{13} = 1$
- as UN = MO we assign $y_{23} = 0$ which is undecided

so we get the following (derived) PC pattern:
1, 1, 0

Pattern Model: Ratings

the probability for the rating $OZ = 1, UN = 4, MO = 4$ transformed into pattern (1, 1, 0) is given by:

$$p(1,1,0) = \delta \left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_2}}\right)\left(\frac{\sqrt{\pi_1}}{\sqrt{\pi_3}}\right)\gamma_{23}$$

the log expected number for the rating can be rewritten as

$$\ln m(1,1,0) = \ln \delta + 2\lambda_1 - 1\lambda_2 - 1\lambda_3 + u_{23}$$

where the γs are the undecided parameter

Transformation: Rating to PC

restricted example for 3 items, only 2 response categories
e.g., concern yes= 1 and concern no= 2

Rating patterns			derived PC-patterns			unique PC-patterns		
i_1	i_2	i_3				y_{12}	y_{13}	y_{23}
1	1	1	0	0	0	0	0	0
1	1	2	0	1	1	0	1	1
1	2	1	1	0	-1	1	0	-1
1	2	2	1	1	0	1	1	0
2	1	1	-1	-1	0	-1	-1	0
2	1	2	-1	0	1	-1	0	1
2	2	1	0	-1	-1	0	-1	-1
2	2	2	0	0	0			

▹ for 3 items only 7 possible patterns (instead of $9 = 3^3$ possible patterns)

Design structure: Rating

restricted example (cont.)

derived patterns			unique patterns			design structure X							
						counts	$\ln\delta$	λ_1	λ_2	λ_3	γ_{12}	γ_{13}	γ_{23}
			y_{12}	y_{13}	y_{23}			x_1	x_2	x_3	u12	u13	u23
0	0	0	0	0	0	n_1	1	0	0	0	1	1	1
0	1	1	0	1	1	n_2	1	1	1	-2	1	0	0
1	0	-1	1	0	-1	n_3	1	1	-2	1	0	1	0
1	1	0	1	1	0	n_4	1	2	-1	-1	0	0	1
-1	-1	0	-1	-1	0	n_5	1	-2	1	1	0	0	1
-1	0	1	-1	0	1	n_6	1	-1	2	-1	0	1	0
0	-1	-1	0	-1	-1	n_6	1	-1	-1	2	1	0	0
0	0	0											

▷ additionally we have undecided parameters for each comparison

---

▷ User friendly function (restricted functionality)

Function: `pattL.fit()`

data preparation

```
> t3dat<-read.table("../data/t3dat.dat", header=TRUE)
> #attach(t3dat)
> t3dat$sex<-factor(t3dat$sex)
> t3dat$age4k<-factor(t3dat$age4k)
```

- We fit pattern rating models
with undecided parametes ♠ `undec = T`

```
> lm1 <- pattL.fit(t3dat, 3,  undec = T, elim = ~sex * age4k)
```

---

result:

```
> lm1
Results of pattern model for ratings

Call:
pattL.fit(obj = t3dat, nitems = 3, elim = ~sex * age4k, undec = T)


Deviance:  955
log likelihood:  -8843
eliminated term(s):  ~age4k + age4k:sex + sex

no of iterations:  8  (Code: 1 )

   estimate     se      z p-value
OZ  -0.3108 0.0161 -19.28       0
UN  -0.0663 0.0155  -4.27       0
U    0.8457 0.0137  61.69       0
```

▷ Note: only one general undecided parameter `U` when using `pattL.fit()`

- we fit model with and without undecided

```
> lm0 <- pattL.fit(t3dat, 3, elim = ~sex * age4k)
> lm1 <- pattL.fit(t3dat, 3,  undec = T, elim = ~sex * age4k)
```

Compare models: we need to calculate deviance change
(2* difference of the log likelihoods)

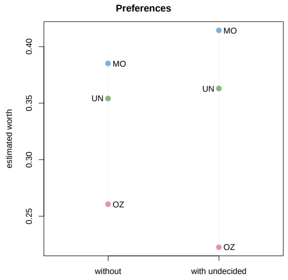
```
> dch <-2*abs(lm1$ll-lm0$ll)
>  # deviance change - (2* difference of the log likelihoods)
> df1<-length(lm1$coef)
> df2<-length(lm0$coef)
> dfc <- abs(df1-df2)
> # difference of df
> p <-1 - pchisq(dch,dfc)
> p
[1] 0
```

- the change in deviance is 3350.2 on 1 df – highly significant!

- Note: we **can not** use `anova()` when using `pattL.fit()`

calculate the worth and plot

```
> w0  <- patt.worth(lm0)
> w1  <- patt.worth(lm1)
> w01 <- cbind(w0, w1)
> colnames(w01) <- c("without","with undecided")

> plotworth(w01, ylab = "estimated worth")
```



Model fitting with `patt.design()` and `gnm()`

(1) generate the design matrix with `patt.design()`

use ♠*option:* `resptype="rating"`

```
> dlm1 <- patt.design(t3dat, nitems=3, resptype="rating",
+                      cat.scovs=c("sex","age4k"))

> head(dlm1)
   y OZ UN MO u12 u13 u23 sex age4k
1 41  0  0  0   1   1   1   1     1
2 17  1  1 -2   1   0   0   1     1
3  9  1 -2  1   0   1   0   1     1
4 11  2 -1 -1   0   0   1   1     1
5  0  2  0 -2   0   0   0   1     1
6  2  2 -2  0   0   0   0   1     1
```

▷ Note: we get undecided parameters for each comparison

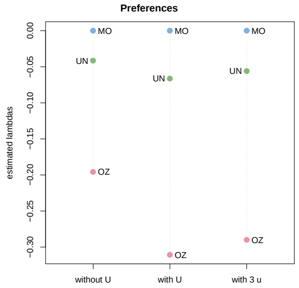
---

(2) fit model with undecided using gnm()

```
> plm1 <-gnm( y ~ OZ+UN+MO + u12+u13+u23,
+                 eliminate = sex:age4k,
+                 data=dlm1,family=poisson)

> plm1
Call:

gnm(formula = y ~ OZ + UN + MO + u12 + u13 + u23, eliminate = sex:age4k,
    family = poisson, data = dlm1)

Coefficients of interest:
     OZ        UN        MO       u12       u13       u23
-0.2901   -0.0557        NA    0.9004    0.6372    0.9964

Deviance:            918
Pearson chi-squared: 940
Residual df:         91
```

- Plot λs for the three models `lm0` `lm1` `plm1` (no dependencies)

```
> w0 <-  patt.worth(lm0,  outmat="lambda")
> w1 <-  patt.worth(lm1,  outmat="lambda")
> w2 <-  patt.worth(plm1, outmat="lambda")
> w_012 <- cbind(w0,w1,w2)
> colnames(w_012)<-c("without U", "with U", "with 3 u" )

> plotworth(w_012, ylab = "estimated lambdas")
```



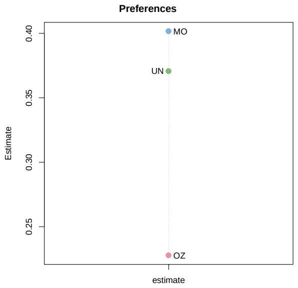
Pattern Models

- We fit model with dependencies and undecided parameter

`pattL.fit()` – use ♠ `ia = T`, `undec = T`,

```
> lm2 <- pattL.fit(t3dat, 3, undec = T, ia = T, elim = ~sex * age4k)
```

```
> w2 <- patt.worth(lm2)
> plotworth(w2)
```



Pattern Models

- We fit model with subject covariates

`pattL.fit()` – use ♠ `formel = ~ ..., elim = ~ ...`

two-way interaction model: `sex * age4k`

```
> lm4 <- pattL.fit(t3dat, 3, undec = T, ia = T,
+                  formel =~ sex * age4k, elim = ~sex * age4k)
```

two main effects model: `sex + age4k`

```
> lm3 <- pattL.fit(t3dat, 3, undec = T, ia = T,
+                  formel =~ sex + age4k, elim = ~sex * age4k)
```

Pattern Models

- compare models `sex*age4k` and `sex+age4k`:

we need to calculate deviance change using 2* difference of the log likelihoods

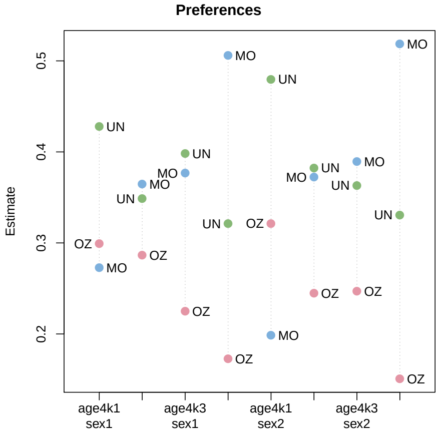
```
> dch <-2*abs(lm4$ll-lm3$ll)
>  # deviance change - (2* difference of the log likelihood)
> df1<-length(lm4$coef)
> df2<-length(lm3$coef)
> dfc <- abs(df1-df2)
> # difference of df
> p <-1 - pchisq(dch,dfc)
> p
[1] 0.0379
```

- the change in deviance is 13.3 on 6 df is significant!

Can not reduce model to main effects model or smaller model !

Pattern Models

```
> worth4 <- patt.worth(lm4)
> plotworth(worth4)
```



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

Further Extensions

- multidimensional PC pattern models
  when objects are compared on more than one attribute
- repeated evaluation of the same objects by the same judges (panel data)
- missing values in pattern models
- mixture models (latent class) for all extensions
- further response formats
  e.g. partial rankings, *piling*, *best to worst scaling*
- combinations of these options

Some References

Bradley, R. and Terry, M. (1952). Rank Analysis of Incomplete Block Designs. I. The Method of Paired Comparisons. *Biometrika*, 39:324–345.

Critchlow, D. and Fligner, M. (1991). Paired comparison, triple comparison, and ranking experiments as generalized linear models, and their implementation in GLIM. *Psychometrika*, 56:517–533.

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

Dittrich, R., Francis, B., Hatzinger, R., and Katzenbeisser, W. (2007). A paired comparison approach for the analysis of sets of Likert scale responses. *Statistical Modelling*, 7:3–28.

Dittrich, R., Hatzinger, R., and 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*, 47:511–525.

Dittrich, R., Hatzinger, R., and Katzenbeisser, W. (2002). Modelling dependencies in paired comparison experiments . *Computational Statistics and Data Analysis*, 40:39–57.

Sinclair, C. (1982). GLIM for preference. In Gilchrist, R., editor, *Proceedings of the International Conference on Generalised Linear Models*, volume 14, pages 164–178. Springer Lecture Notes in Statistics.

Turner, H., Firth, D.(2009). *Generalized nonlinear models in R: An overview of the gnm package., R package version 0.10-0, url = http://CRAN.R-project.org/package=gnm*