Paired Comparison Preference Models

The prefmod Package: Day 5 / Part 1

Modelling Subject Information:

– Parameterization and Design Matrices

– Unobserved Subject Effects (Latent Classes)

Regina Dittrich & Reinhold Hatzinger

Institute for Statistics and Mathematics, WU Vienna

Parameterization and Design Matrices

loglinear pattern model *(Dittrich et al., CSDA, 2002)*

$$\ln m(y_{12}, \ldots, y_{J-1,J}) = \ln m_\ell = \mu + \sum_{j=1}^{J} \lambda_j^O x_j = \eta_\ell$$

design structure for 3 objects:

pattern	y_{12}	y_{13}	y_{23}	counts	μ const	λ_1^O x_1	λ_2^O x_2	λ_3^O x_3
ℓ_1	1	1	1	n_1	1	2	0	-2
ℓ_2	1	1	-1	n_2	1	2	-2	0
ℓ_3	1	-1	1	n_3	1	0	0	0
ℓ_4	1	-1	-1	n_4	1	0	-2	2
ℓ_5	-1	1	1	n_5	1	0	2	-2
ℓ_6	-1	1	-1	n_6	1	0	0	0
ℓ_7	-1	-1	1	n_7	1	-2	2	0
ℓ_8	-1	-1	-1	n_8	1	-2	0	2

$x_j = \#(O_j$ is preferred in ℓ) - $\#(O_j$ not preferred in ℓ)

Pattern model with categorical subject covariates

general expression

$$\ln m(y_{12}, \ldots, y_{J-1,J} \,|\, s) = \ln m_{\ell s} = \mu_s + \sum_{j=1}^{J} (\lambda_j^O + \lambda_{js}) x_j$$

$\lambda_j^O \ldots$ location of object j on preference scale
    here defined for covariate level $s = 1$ (reference group)

$s \ldots$ level of the subject covariate or level of the combination of
    subject covariates, e.g.,
       SEX with two categories (1 male, 2 female)
       AGE with three categories (1 young, 2 middle, 3 old)
       $s = 1$ (male/young), $s = 2$ (male/middle), $\ldots$, $s = 6$ (female/old)

$\lambda_{js} \ldots$ interaction term between the object and the covariate(s)
    defined on level s

Examples for λ_{js}

$\lambda_{j2} = \lambda_{j2}^{OS_1}$
    S_1 is SEX, $s = 2$ is male

$\lambda_{j5} = \lambda_{j2}^{OS_1} + \lambda_{j3}^{OS_2}$
    S_2 is AGE, s is combination of levels for SEX and AGE
    e.g., $s = 5$ could be female/middle

$\lambda_{j5} = \lambda_{j2}^{OS_1} + \lambda_{j3}^{OS_2} + \lambda_{j23}^{OS_1 S_2}$
    same as above, but now additional interaction effect for
    SEX and AGE

λ_{js} is the change of location of object j on the preference scale
    between reference group and covariate group s
    (the shift up or down)

One categorical subject covariate with 2 categories

$$\ln m_{\ell s} = \mu_s + (\lambda_1^O + \lambda_{1s})x_1 + (\lambda_2^O + \lambda_{2s})x_2 + (\lambda_3^O + \lambda_{3s})x_3$$

pattern	y_{12}	y_{13}	y_{23}	s	μ_1 s_1	μ_2 s_2	λ_1^O x_1	λ_2^O x_2	λ_3^O x_3	λ_{12} x_1s_2	λ_{22} x_2s_2	λ_{32} x_3s_2
ℓ_1	1	1	1	1	1	0	2	0	-2	0	0	0
ℓ_2	1	1	-1	1	1	0	2	-2	0	0	0	0
ℓ_3	1	-1	1	1	1	0	0	0	0	0	0	0
ℓ_4	1	-1	-1	1	1	0	0	-2	2	0	0	0
ℓ_5	-1	1	1	1	1	0	0	2	-2	0	0	0
ℓ_6	-1	1	-1	1	1	0	0	0	0	0	0	0
ℓ_7	-1	-1	1	1	1	0	-2	2	0	0	0	0
ℓ_8	-1	-1	-1	1	1	0	-2	0	2	0	0	0
ℓ_1	1	1	1	2	0	1	2	0	-2	2	0	-2
ℓ_2	1	1	-1	2	0	1	2	-2	0	2	-2	0
ℓ_3	1	-1	1	2	0	1	0	0	0	0	0	0
ℓ_4	1	-1	-1	2	0	1	0	-2	2	0	-2	2
ℓ_5	-1	1	1	2	0	1	0	2	-2	0	2	-2
ℓ_6	-1	1	-1	2	0	1	0	0	0	0	0	0
ℓ_7	-1	-1	1	2	0	1	-2	2	0	-2	2	0
ℓ_8	-1	-1	-1	2	0	1	-2	0	2	-2	0	2

---

why μ_1 and μ_2 on previous slide?

if we have subject covariates: we have to fit their highest interaction term – this defines the design blocks

μ_s is normalising constant for level (block) s

formula: `a*b`

s	μ	a2	b2	a2:b2
1	1			
1	1			
1	1			
2	1	1		
2	1	1		
2	1	1		
3	1		1	
3	1		1	
3	1		1	
4	1	1	1	1
4	1	1	1	1
4	1	1	1	1

formula: `a:b - 1`

s	a1:b1	a2:b1	a1:b2	a2:b2
1	1			
1	1			
1	1			
2		1		
2		1		
2		1		
3			1	
3			1	
3			1	
4				1
4				1
4				1
	μ_1	μ_2	μ_3	μ_4

this happens in gnm() with:
`eliminate = a:b`

Unobserved Subject Effects (Latent Classes)

Heterogeneity in paired comparisons

(Francis, Dittrich, Hatzinger; Annals of Applied Statistics, 2010)

- responses vary between respondents
- measured covariates can be taken into account
- other unmeasured or unmeasurable characteristics of the respondents might affect the response

in practice mainly 2 situations:

- unknown or not available subject variables
- very complex situations make model fit untractable

Categorical subject effects

	fixed effects	random effects
how many groups	known	?
group size	known	?
location of worth	?	?
probability of patterns	?	?

The two models for subject effects

model with fixed effects:

$$m_{\ell s} = \exp\!\Big(\mu_s + \sum_j (\lambda_j^O + \lambda_{js})x_j\Big)$$

the latent class model (K latent classes):

$$m_\ell = \sum_{k=1}^{K} q_k \, \exp\!\Big(\mu_k + (\lambda_j^O + \delta_{jk})x_j\Big)$$

additionally to λ_j^O, we have to estimate q_k and δ_{jk}

δ_{jk} describes the location of object j in class k (*mass points*)
q_k is probability for class k (*masses*)

Estimation (details in the paper)

the estimation is performed using the EM-algorithm

setup:
- as many blocks as latent classes
- each pattern has a probability (weight) $w_{\ell k}$

for any iteration of the EM algorithm, there are two steps:
- **Estimation-Step:**
  calculate $w_{\ell k}$ from current pattern probabilities
- **Maximisation-Step:**
  calculate model parameters using a weighted ML-estimation

probability for class k (mean of weights): $q_k = \sum_\ell w_{\ell k}/L$

expected number of respondents in class k: $\sum_\ell n_\ell w_{\ell k}$

Example

(1) we generate some data (for 2 groups) and design structure
```
> dat1 <- simPC(3, 400, c(1, 5, 9), seed = 87)
> dat2 <- simPC(3, 600, c(5, 1, 9))
> dat <- data.frame(rbind(dat1, dat2))
> des <- patt.design(dat, 3)
```
(2) fit random effects model (with $k = 2$ mass points)
```
> (mnp <- pattnpml.fit(y ~ 1, random = ~o1 + o2 + o3, k = 2, design = des))
Call:  pattnpml.fit(formula = y ~ 1, random = ~o1 + o2 + o3, k = 2,
    design = des)

Coefficients:
(Intercept)      o1:MASS1      o1:MASS2      o2:MASS1      o2:MASS2      o3:MASS1
     3.7361       -0.8841       -0.3826       -0.3746       -1.0630            NA

Mixture proportions:
    MASS1        MASS2
0.5016344    0.4983656

Deviance:             9.17
-2 log L:            58.52      EM convergence at iteration  7
```

Example

(3) now we fit a model using a fixed effect (e.g., sex)
```
> sex <- factor(rep(1:2, c(400, 600)))
> dat.s <- data.frame(rbind(dat1, dat2), sex)
> des.s <- patt.design(dat.s, 3, cat.scovs = "sex")
> mpc <- gnm(y ~ (o1 + o2 + o3):sex, elim = sex, family = poisson,
+     data = des.s)
> mpc
Call:
gnm(formula = y ~ (o1 + o2 + o3):sex, eliminate = sex, family = poisson,
    data = des.s)

Coefficients of interest:
o1:sex1  o1:sex2  sex1:o2  sex2:o2  sex1:o3  sex2:o3
-1.0854  -0.3263  -0.2929  -1.1716       NA       NA

Deviance:             14.28733
Pearson chi-squared: 14.241
Residual df:          10
```
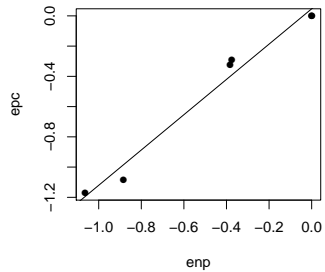
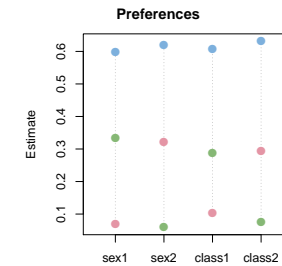## (4a) comparison of estimates

```
> enp <- c(mnp$coefficients[2:5], 0, 0)
> epc <- c(mpc$coefficients[1:4], 0, 0)
> plot(enp, epc, pch = 16)
> abline(lm(epc ~ enp))
```
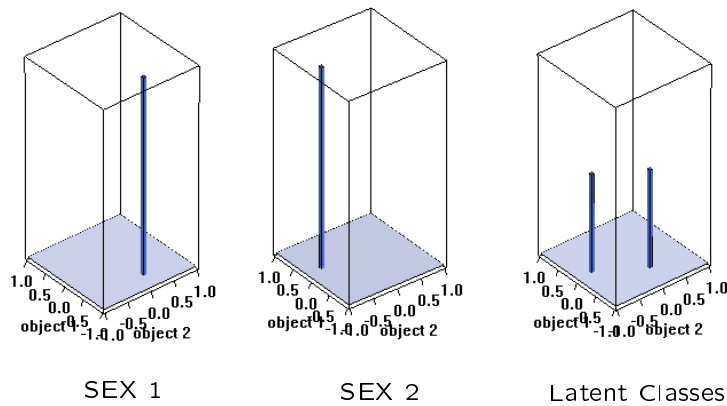

(4b) comparison of worth

```
> epcm <- matrix(epc, 3, byrow = TRUE)
> enpm <- matrix(enp, 3, byrow = TRUE)
> w <- cbind(epcm, enpm)
> w <- apply(w, 2, function(x) exp(2 * x)/sum(exp(2 * x)))
> colnames(w) <- c("sex1", "sex2", "class1", "class2")
> plotworth(w)
```


Preferences

Locations (centered) and masses

SEX 1   SEX 2   Latent Classes

The NPML model in prefmod

```
pattnpml.fit(
        formula,                # formula for fixed effects
        random = ~1,            # formula for random effects
        k = 1,                  # number of mass-points (classes)
        design,                 # design matrix
        tol = 0.5,              # to control the EM-algorithm
        startp = NULL,
        EMmaxit = 500,
        EMdev.change = 0.001,
        pr.it = FALSE
        )
```

Notes:

formula ... without fixed effects: `formula = y ~ 1`
         with fixed effects, e.g., `y ~ sex:(o1 + o2 + o3)`

random ... usually the objects, e.g., `random = ~ o1 + o2 + o3`

design ... the design matrix, generated using `patt.design()`

startp ... starting values for mixture proportions (length k)

NAs are not allowed

Steps to fit and interpret a PC mixture model

(1) generate a design matrix using `patt.design()`

(2) fit a sequence of models using $k = 1, 2, \ldots$

(3) choose a reasonable model based on:

disparity changes, BIC, graphical display of these and/or worths

(4) if subject variables are available, try to explain latent classes

Notes:

ad (2): refinement

mixture likelihood may have various local maxima

EM may find only a local maximum

in applications: for each k, fit several models with different starting values (`startp`)

choose the one with minimum disparity

ad (3):

disparity is -2 ln Likelihood, where the likelihood is computed with all (combinatorial) constants that are usually omitted

Sample Analysis

Data: A study of $n = 279$ persons in the labour market training (AMS) to explore preferences for training delivery modes like:

- computer-based (CO)
- TV-based (TV)
- paper-based (GL)
- audio-based (AU)
- classroom-based (UV)

Step (1): data and design matrix

```
> load("trdeliv2.RData")
> data <- na.omit(trdeliv2)
> nam <- c("CO", "TV", "GL", "AU", "UV")
> des <- patt.design(trdata, 5, objnames = nam)
```

Step (2): fit models with $k = 1, \ldots, 5$

```
> m1 <- pattnpml.fit(y ~ CO + TV + GL + AU, k = 1, design = des)
> m2 <- pattnpml.fit(y ~ 1, random = ~CO + TV + GL + AU, k = 2,
+     design = des)
> m3 <- pattnpml.fit(y ~ 1, random = ~CO + TV + GL + AU, k = 3,
+     design = des)
> m4 <- pattnpml.fit(y ~ 1, random = ~CO + TV + GL + AU, k = 4,
+     design = des)
> m5 <- pattnpml.fit(y ~ 1, random = ~CO + TV + GL + AU, k = 5,
+     design = des)
```

Step (3a): choose a reasonable model based on disparity changes

```
> di1 <- m1$disparity
> di2 <- m2$disparity
> di3 <- m3$disparity
> di4 <- m4$disparity
> di5 <- m5$disparity
> di <- c(di1, di2, di3, di4, di5)
```
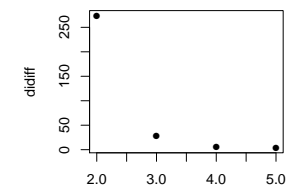
Step (3a): (cont'd)

```
> (didiff <- sapply(1:4, function(i) di[i] - di[i + 1]))
[1] 273.410186  28.403143   5.292555   4.402321
```

change from $k = 2$ to $k = 3$ is large (28.4), further changes small

"Scree Plot"

```
> plot(2:5, didiff, pch = 16, xlab = "")
```

Step (3b): choose a reasonable model based on BIC

BIC (Bayesian information criterion)

$$BIC = -2 \ln \text{Likelihood} + \#(\text{parameters}) \cdot \ln(N)$$

criterion for model selection, models with smaller BIC preferred

```
> bic1 <- m1$disparity + length(m1$coefficients) * log(sum(des$y))
> bic2 <- m2$disparity + length(m2$coefficients + 1) * log(sum(des$y))
> bic3 <- m3$disparity + length(m3$coefficients + 2) * log(sum(des$y))
> bic4 <- m4$disparity + length(m4$coefficients + 3) * log(sum(des$y))
> bic5 <- m5$disparity + length(m5$coefficients + 4) * log(sum(des$y))
> bic <- c(bic1, bic2, bic3, bic4, bic5)
> bic
[1] 1028.2354  775.9783  768.7282  784.5888  801.3395
```

again model with 3 classes (BIC= 768.7)

Step (3c): graphical display - worth parameters

```
> worth <- function(obj) {
+     lambda <- obj$coefficients[-1]
+     k <- length(obj$masses)
+     lmat <- matrix(lambda, ncol = k, byrow = TRUE)
+     lmat <- rbind(lmat, rep(0, k))
+     w <- apply(lmat, 2, function(x) exp(2 * x)/sum(exp(2 * x)))
+     w
+ }
> w3 <- worth(m3)
> w4 <- worth(m4)
> rownames(w3) <- rownames(w4) <- nam
> colnames(w3) <- round(m3$masses, 3)
> colnames(w4) <- round(m4$masses, 3)
> oldpar <- par(mfrow = c(1, 2))
> plotworth(w3, main = "", log = "y", ylab = "worth")
> plotworth(w4, main = "", log = "y", ylab = "worth")
> par(oldpar)
```

Step (3c):(cont'd) graphical display:

worth parameters and mixture probabilities
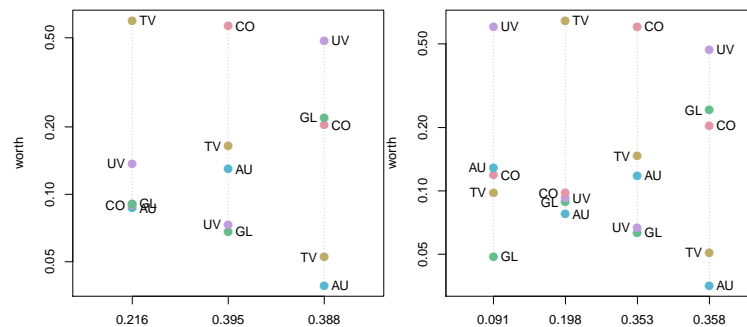
Step (4): explain latent classes

if additional information (subject variables) are available, we can try to relate them to the latent classes

from the fit object, we can obtain the posterior probabilities w_{lk} for the patterns belonging to a certain class (obj$post.prob)

```
> head(round(m3$post.prob, 3))
      1     2     3
1 0.222 0.409 0.369
2 0.216 0.397 0.387
3 0.219 0.402 0.380
4 0.216 0.396 0.388
5 0.212 0.399 0.389
6 0.213 0.397 0.390
```

the matrix obj$post.prob has L (#patterns) rows and k columns

Step (4):(cont'd) explain latent classes

first we have to find out the pattern for each person

```
> # generate strings for all patterns and data
> allp <- prefmod:::all_patterns(1, 2, 10)    # generate pattern matrix
> apstr <- apply(allp, 1, paste, collapse="") # convert each row to a string
> dpstr <- apply(trdata[,1:10], 1, paste, collapse="") # same for data
> #
> # which pattern has a person (idx contains pattern number)
> idx<-match(dpstr,apstr)
```

we have probabilities for each person to belong to a certain class, e.g., for person 5

```
> m3$post.prob[idx[5], ]
        1         2         3
0.2154712 0.3943123 0.3902165
```

Step (4):(cont'd)

assign each person to a latent class
persons are assumed to belong to the class for which their response pattern has highest probability

```
> # what is highest class probability for pattern
> patt.class <- apply(m3$post.prob, 1, which.max)
> #
> # which person belongs to which class
> class.membership <- patt.class[idx]
> head(class.membership)
336 400 772 166  16 784
  3   3   2   1   2   2
> #
> # add class.membership to data frame
> trdata$class.membership <- class.membership
```

now various analyses possible (multinomial regression, . . .)
alternatively: Dirichlet regression on posterior probabilities

---

Step (4):(cont'd) Example: mosaic plot with sex

```
> library(vcd)
> mosaic(~sex + class.membership, data = trdata, highlighting = 1,
+     highlighting_fill = c("lightblue", "pink"))
```
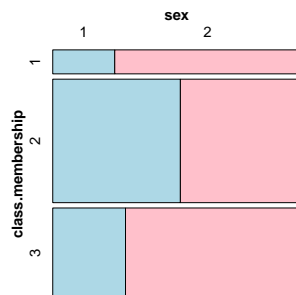
The model with fixed and random effects

- sometimes we have subject information and want to investigate if there is further heterogeneity in the data

- we have subject information which is too complex and/or leads to intractable models

the latent class model becomes:

$$m_\ell = \sum_{k=1}^{K} q_k \, \exp\Big(\mu_k + (\lambda_j^O + \lambda_{js} + \delta_{jk})x_j\Big)$$

in `pattnpml.fit()` we have to specify the fixed part
        `formula=` . . . as in `gnm()` for pattern models
and the random part
        `random=` . . . as before

Model with sex as fixed effect

Step (1): data and design matrix

```
> des2 <- patt.design(trdata, 5, objnames = nam, cat.scovs = "sex")
```

Step (2): fit models with $k = 1, \ldots, 5$

```
> fixed <- y ~ sex:(CO + TV + GL + AU)
> random <- ~CO + TV + GL + AU
> mm0 <- pattnpml.fit(y ~ CO + TV + GL + AU, k = 1, design = des2)
> mm1 <- pattnpml.fit(fixed, k = 1, design = des2)
> mm2 <- pattnpml.fit(fixed, random = random, k = 2, design = des2)
> mm3 <- pattnpml.fit(fixed, random = random, k = 3, design = des2)
> mm4 <- pattnpml.fit(fixed, random = random, k = 4, design = des2)
> mm5 <- pattnpml.fit(fixed, random = random, k = 5, design = des2)
```

do we need sex at all? yes

```
> mm0$deviance - mm1$deviance
[1] 48.1424
```

Step (3a): choose a reasonable model based on disparity changes

```
> di1 <- mm1$disparity
> di2 <- mm2$disparity
> di3 <- mm3$disparity
> di4 <- mm4$disparity
> di5 <- mm5$disparity
> di <- c(di1, di2, di3, di4, di5)
> (didiff <- sapply(1:4, function(i) di[i] - di[i + 1]))
[1] 220.04357  24.73769  16.79865  14.24057
```

Step (3b): choose a reasonable model based on BIC

```
> bic1 <- di1 + length(mm1$coefficients) * log(sum(des2$y))
> bic2 <- di2 + length(mm2$coefficients + 1) * log(sum(des2$y))
> bic3 <- di3 + length(mm3$coefficients + 2) * log(sum(des2$y))
> bic4 <- di4 + length(mm4$coefficients + 3) * log(sum(des2$y))
> bic5 <- di5 + length(mm5$coefficients + 4) * log(sum(des2$y))
> (bic <- c(bic1, bic2, bic3, bic4, bic5))
[1] 1190.718 1007.692 1004.108 1008.462 1015.375
```

again model with 3 classes (BIC= 1004.11)

---

the estimates for $k = 3$ are

```
> mm3summ <- summary(mm3)
> mm3summ$coefficients
```

```
              Estimate Std. Error    t value
(Intercept) -4.7892674 0.21076505 -22.723252
sex1:CO     -0.2734693 0.08663993  -3.156388
sex2:CO     -0.4971789 0.07315839  -6.795924
sex1:TV     -1.0685175 0.08086429 -13.213712
sex2:TV     -1.1278920 0.07114550 -15.853313
sex1:GL     -0.4053989 0.09064245  -4.472507
sex2:GL     -0.3868366 0.07467515  -5.180259
sex1:AU     -1.1351014 0.07982666 -14.219578
sex2:AU     -1.3921122 0.06946480 -20.040542
CO:MASS1     0.9387036 0.11717737   8.010963
CO:MASS2     1.1329212 0.15769264   7.184363
TV:MASS1     2.0586947 0.13553016  15.189938
TV:MASS2     0.8932241 0.16280404   5.486498
GL:MASS1     0.2502708 0.11975733   2.089817
GL:MASS2     0.4828529 0.16484783   2.929083
AU:MASS1     1.2021824 0.11861579  10.135096
AU:MASS2     2.0394562 0.16713687  12.202312
```

Step (3c): graphical display - worth parameters for sex1 and sex2

```
> lambda <- mm3$coefficients[-1]
> lambda <- c(lambda, NA)
> fix.lmat <- matrix(lambda[1:8], ncol = 2, byrow = TRUE)
> rand.lmat <- matrix(lambda[9:20], ncol = 3, byrow = TRUE)
> all.lmat <- fix.lmat
> for (class in 1:2) for (sex in 1:2) all.lmat <- cbind(all.lmat,
+     fix.lmat[, sex] + rand.lmat[, class])
> all.lmat <- rbind(all.lmat, rep(0, 6))
> w <- apply(all.lmat, 2, function(x) exp(2 * x)/sum(exp(2 * x)))
> rownames(w) <- nam
> colnames(w) <- rep(round(mm3$masses, 3), each = 2)
```

```
> oldpar <- par(mfrow = c(1, 2))
> plotworth(w[, c(1, 3, 5)], log = "y", ylab = "worth male")
> plotworth(w[, c(2, 4, 6)], log = "y", ylab = "worth female")
> par(oldpar)
```