HUDM6052 Psychometric II Homework_03

Chenguang Pan (cp3280@tc.columbia.edu)

2023-10-26

Contents

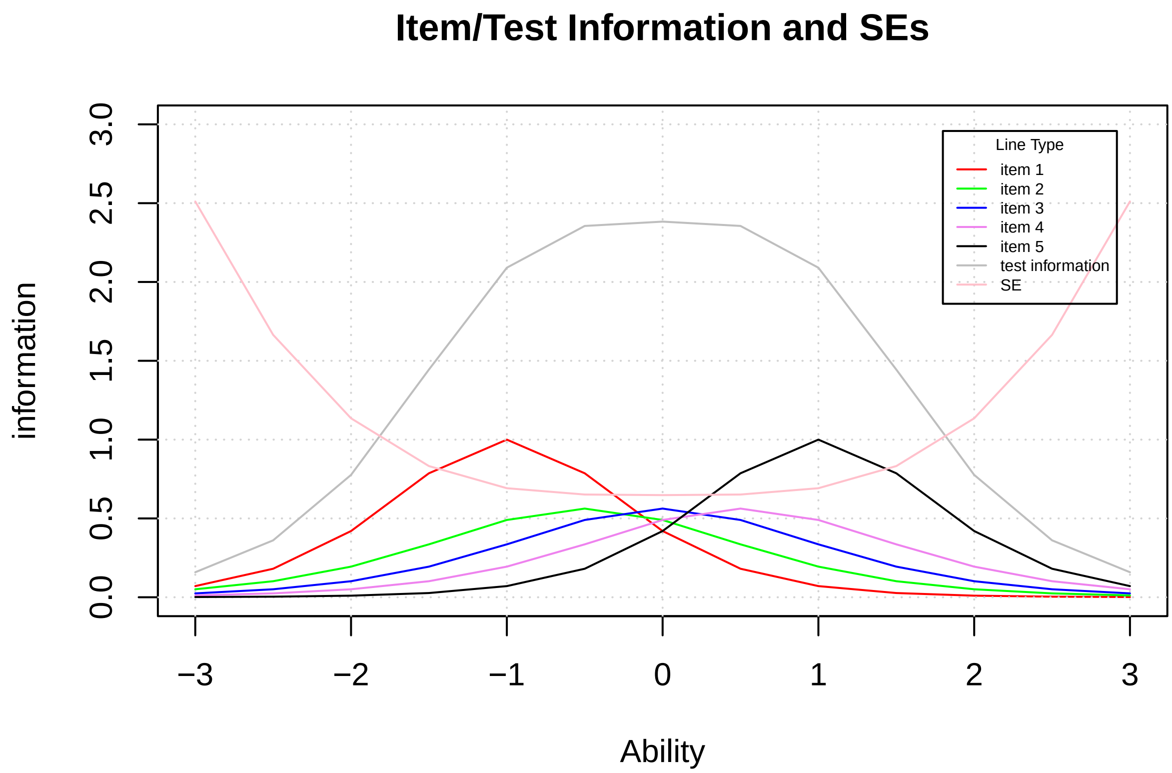
Q1

Using the item parameters given in the Table...

My Solution:

```
> # write the corresponding information function
> iif <- function(theta, a, b){
+   # get the logit
+   Z <- a*(theta-b)
+   # get the probability
+   out <- 1/(1 + exp(-Z))
+   info <- out*(1-out)*(a^2)
+   return(info)
+ }
>
> # set the ability range
> theta <- seq(-3,3, by=0.5)
> # set the item parameters
> a <- c(2, 1.5, 1.5, 1.5, 2)
> b <- c(-1, -0.5, 0, 0.5, 1)
> # define the color
> color_set <- c("red", "green", "blue","violet","black")
>
> # plot the item information function
> info_out <- iif(theta, a[1], b[1])
> # create a vector to sum up all the information function
> test_info <- info_out
>
```

```r
> # initialize the plot by plotting the first item
> plot(theta, info_out,type = "l", col=color_set[1],
+      main = "Item/Test Information and SEs",
+      xlab = "Ability", ylab = "information",
+      ylim = c(0,3))
> grid()
>
> # plot the rest item using a for loop
> for (i in 2:5) {
+   info_out_i <- iif(theta,a[i],b[i])
+   lines(theta, info_out_i, type ="l", col=color_set[i])
+   test_info <- test_info +info_out_i
+ }
>
> # draw the test information function
> lines(theta, test_info, type = "l", col="gray")
> # plot the SE
> SE <- c()
> for (j in 1:length(theta)) {
+   se_j <- 1/sqrt(sum(iif(theta[j],a,b)))
+   SE[j] <- se_j
+ }
> lines(theta, SE, type = "l", col="pink")
>
> # add a legend
> legend('topright',inset=0.05,c("item 1","item 2","item 3","item 4","item 5",
+                               "test information","SE"),
+        lty=1,col=c("red", "green","blue","violet","black","gray","pink"),
+        title="Line Type", cex = 0.5)
```



Q2.a

a. For each of the six items given in the Table below...

My Solution:
The maximum value of a 3PL's information function is at

$$\theta_{max} = \beta_i + \frac{1}{a_i} log[\frac{1+\sqrt{1+8c_i}}{2}].$$

Therefore, I write a function to get the optimal θ_{max} first. And then send this value together with the parameters into the information function for 3PL model to get the results.

```
> # write a function to get the theta_max
> get_theta <- function(a,b,c){
+   out <- b + log(0.5+0.5*sqrt(1+8*c))/a
+   return(out)
+ }
>
> # write the 3pl information function
> iif_3pl <- function(theta, a, b, c){
+   z <- a*(theta - b)
+   p <- c + (1-c)/(1+exp(-z))
+   p_star <- 1/(1 + exp(-z))
+   I <- (a^2)*p*(1-p)*(p_star/p)^2
```

```
+   return(I)
+ }
```

Next, plug the given paramters into the functions to get the results.

```
> # load the given parameters vectors
> b <- c(1,1,1,-1.5,-0.5,0.5)
> a <- c(1.8,0.8,1.8,1.8,1.2,0.4)
> c <- c(0,0,0.25,0,0.1,0.15)
>
> # using a for loop to get all the required values
> theta_vec <- c()
> info_vec <- c()
> for (i in 1:length(b)) {
+   # get the optimal theta value
+   theta_max <- get_theta(a = a[i],b = b[i],c = c[i])
+   theta_vec[i] <- theta_max
+   # get the maximum value of information of item i
+   info_max <- iif_3pl(theta = theta_max,a = a[i],b = b[i],c = c[i])
+   info_vec[i] <- info_max
+ }
>
> # Merge all the values as a dataframe
> df_out <- data.frame(
+   item = seq(1,6),
+   theta_max = theta_vec,
+   info_max = info_vec
+ )
>
> df_out
  item  theta_max   info_max
1    1  1.0000000 0.81000000
2    2  1.0000000 0.16000000
3    3  1.1732808 0.50122974
4    4 -1.5000000 0.81000000
5    5 -0.3685794 0.29665631
6    6  1.0410421 0.02998276
```

The maximum values of information and corresponding θs are shown at end of the code chunk above.

Q2.b

b. Which item would you choose to make up a two-item…

My Solution:
I will choose the `item 1` and `item 2` to make a two-item test since they have the maximum information at the $\theta = 1.0$, which means this test can more accurately measure this given test-taker's ability. The test information at $\theta = 1.0$ is

$$0.81 + 0.16 = 0.97.$$

Q3.a

a. Determine the standard error of the estimate…

My Solution:

```
> # load the given paramters
> a <- c(1,1,2,2)
> b <- c(0,1,1,1.5)
> theta <- 1.5
>
> # using the information function created in the Q1 to get the
> # vector of information at the theta=1.5 for each item
> info_vec <- iif(theta=1.5, a, b)
>
> # get the SE
> SE_j <- 1/sqrt(sum(info_vec))
> SE_j
[1] 0.6787507
```

Therefore, the SE for the test-taker with estimated trait $\theta = 1.5$ is 0.679.

Q3.b

b. Construct a 95% confidence interval for...

My Solution:
The 95% confidence interval of the estimated $\theta = 1.5$ is

$$95\%CI = 1.5 \pm 1.96 \times 0.679.$$

Therefore, $95\%CI$ is $[.169, 2.831]$.

Q4

Fit 3PL model to the dataset...

My Solution:
For estimating the 3PL model,

```
> # load the data
> widths <- c(4, rep(1,40))
> df <- read.fwf("~/Desktop/PhD_Learning/HUDM6052 Psychometric II/HUDM6052_Psychometic_II/Assignment 3/
+                header = F, stringsAsFactors=F)
>
> # rename the columns
> items_name <- paste0("item_", seq(1,40))
> col_names <- c("ID", items_name)
> names(df) <- col_names
>
> # fit the 3PL model through MIRT package
> library(mirt)
> # step 1: specify the model, 40 items load on one factor
> #         set a prior distribution for the pseudo-guessing parameter to
> #         reduce problems of estimation convergence
> spec <- 'F = 1-40
```

```
+ PRIOR = (1-40,g,norm, -1.1,2)'
>
> # step 2: fit the model
> mod_3pl <- mirt(df[,-1], model=spec,itemtype = "3PL", SE=T)
Iteration: 1, Log-Lik: -11855.215, Max-Change: 1.43351Iteration: 2, Log-Lik: -11673.391, Max-Change: 0.9

Calculating information matrix...
> mod_3pl

Call:
mirt(data = df[, -1], model = spec, itemtype = "3PL", SE = T)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 46 EM iterations.
mirt version: 1.40
M-step optimizer: BFGS
EM acceleration: Ramsay
Number of rectangular quadrature: 61
Latent density type: Gaussian

Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  488.9912

Log-posterior = -11592.2
Estimated parameters: 120
AIC = 23290.18
BIC = 23795.94; SABIC = 23415.05
G2 (1e+10) = 16835.58, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

Next, to get the estimated results for the parameters.

```
> coef(mod_3pl, IRTpars = T, printSE=T)
$item_1
        a      b     g  u
par 0.846 -1.216 0.226  1
SE  0.240  0.950 0.294 NA

$item_2
        a      b     g  u
par 0.571 -1.680 0.282  1
SE  0.226  1.991 0.410 NA

$item_3
        a      b     g  u
par 0.850 -1.686 0.119  1
SE  0.159  0.449 0.150 NA

$item_4
        a      b     g  u
par 0.347 -1.118 0.223  1
SE  0.160  2.401 0.318 NA
```

```
$item_5
        a     b     g  u
par 3.690 0.323 0.767  1
SE  1.958 0.209 0.038 NA

$item_6
        a      b     g  u
par 0.947 -1.296 0.156  1
SE  0.192  0.538 0.198 NA

$item_7
        a      b     g  u
par 0.602 -2.458 0.187  1
SE  0.159  1.036 0.255 NA

$item_8
        a      b     g  u
par 0.686 -1.667 0.249  1
SE  0.212  1.345 0.349 NA

$item_9
       a     b     g  u
par 1.11 0.404 0.501  1
SE  0.60 0.649 0.142 NA

$item_10
        a      b     g  u
par 1.786 -0.271 0.143  1
SE  0.331  0.181 0.085 NA

$item_11
        a      b     g  u
par 1.161 -0.709 0.110  1
SE  0.204  0.295 0.119 NA

$item_12
        a      b     g  u
par 1.388 -0.439 0.296  1
SE  0.393  0.441 0.163 NA

$item_13
        a      b     g  u
par 0.832 -1.551 0.163  1
SE  0.177  0.637 0.212 NA

$item_14
        a      b     g  u
par 0.931 -1.602 0.140  1
SE  0.177  0.482 0.178 NA

$item_15
        a      b     g  u
par 1.589 -0.732 0.475  1
```

```
SE  0.875  0.984 0.313 NA

$item_16
        a      b     g  u
par 0.842 -0.545 0.073  1
SE  0.144  0.272 0.084 NA

$item_17
        a      b     g  u
par 1.371 -0.145 0.308  1
SE  0.456  0.448 0.156 NA

$item_18
        a      b     g  u
par 0.853 -0.439 0.202  1
SE  0.258  0.718 0.212 NA

$item_19
        a      b     g  u
par 1.471 -0.376 0.238  1
SE  0.365  0.341 0.138 NA

$item_20
        a     b     g  u
par 1.388 0.322 0.263  1
SE  0.459 0.313 0.112 NA

$item_21
        a      b     g  u
par 0.744 -0.420 0.122  1
SE  0.167  0.485 0.138 NA

$item_22
        a      b     g  u
par 1.154 -0.421 0.103  1
SE  0.212  0.276 0.108 NA

$item_23
        a     b     g  u
par 1.182 0.376 0.263  1
SE  0.414 0.386 0.127 NA

$item_24
        a     b     g  u
par 0.689 0.491 0.124  1
SE  0.205 0.518 0.135 NA

$item_25
        a     b     g  u
par 1.246 0.603 0.182  1
SE  0.355 0.244 0.088 NA

$item_26
```

```
        a      b     g  u
par 0.727 -1.099 0.128  1
SE  0.152  0.550 0.160 NA

$item_27
        a     b     g  u
par 1.775 0.560 0.238  1
SE  0.542 0.189 0.075 NA

$item_28
        a     b     g  u
par 0.484 0.430 0.260  1
SE  0.342 2.092 0.355 NA

$item_29
        a     b     g  u
par 1.378 0.034 0.149  1
SE  0.356 0.292 0.119 NA

$item_30
        a     b     g  u
par 1.864 0.209 0.194  1
SE  0.456 0.181 0.078 NA

$item_31
        a    b     g  u
par 1.679 0.65 0.370  1
SE  0.599 0.23 0.075 NA

$item_32
        a     b     g  u
par 0.900 1.106 0.323  1
SE  0.513 0.497 0.137 NA

$item_33
        a     b    g  u
par 0.920 0.275 0.26  1
SE  0.435 0.741 0.21 NA

$item_34
        a     b     g  u
par 1.863 0.459 0.201  1
SE  0.432 0.148 0.059 NA

$item_35
        a     b     g  u
par 1.289 0.247 0.153  1
SE  0.331 0.267 0.103 NA

$item_36
        a     b     g  u
par 1.565 0.282 0.220  1
SE  0.446 0.239 0.095 NA
```

```
$item_37
        a     b     g  u
par 1.240 0.610 0.221  1
SE  0.412 0.277 0.098 NA

$item_38
       a     b     g  u
par 0.29 0.460 0.197  1
SE  0.16 2.336 0.262 NA

$item_39
        a     b     g  u
par 2.327 0.233 0.078  1
SE  0.419 0.100 0.043 NA

$item_40
        a     b     g  u
par 1.286 0.299 0.126  1
SE  0.304 0.229 0.089 NA

$GroupPars
    MEAN_1 COV_11
par      0      1
SE      NA     NA
```

Then, to get the overall model fit indicies.

```
> # get the overall model fit
> M2(mod_3pl)
            M2  df         p RMSEA RMSEA_5    RMSEA_95     SRMSR      TLI CFI
stats 645.5297 700 0.9302178     0       0 0.004300395 0.0386178 1.011112   1
```

The `M2 test` shows that the we fail to reject the 3PL model, $p = .930$. The rest of the fit indices show that overall model fits well.

Finally, to get the item level fit metric.

```
> itemfit(mod_3pl)
      item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   item_1 30.810      23      0.026  0.128
2   item_2 23.825      24      0.000  0.472
3   item_3 35.526      22      0.035  0.034
4   item_4 17.174      25      0.000  0.875
5   item_5 26.809      20      0.026  0.141
6   item_6 15.803      22      0.000  0.826
7   item_7 19.200      24      0.000  0.741
8   item_8 27.442      24      0.017  0.284
9   item_9 17.275      23      0.000  0.796
10 item_10 15.208      21      0.000  0.812
11 item_11 20.931      22      0.000  0.525
12 item_12 20.207      22      0.000  0.570
13 item_13 23.317      23      0.005  0.442
14 item_14 27.513      21      0.025  0.155
```

```
15 item_15 17.237       19      0.000  0.574
16 item_16 27.914       23      0.021  0.219
17 item_17 21.929       23      0.000  0.525
18 item_18 23.685       23      0.008  0.421
19 item_19 18.783       22      0.000  0.659
20 item_20 26.694       22      0.021  0.223
21 item_21 27.030       24      0.016  0.303
22 item_22 17.046       22      0.000  0.761
23 item_23 21.257       24      0.000  0.624
24 item_24 35.760       24      0.031  0.058
25 item_25 22.675       23      0.000  0.480
26 item_26 24.753       23      0.012  0.363
27 item_27 35.853       22      0.036  0.031
28 item_28 27.224       25      0.013  0.345
29 item_29 15.176       22      0.000  0.855
30 item_30 12.615       22      0.000  0.943
31 item_31 25.776       23      0.016  0.312
32 item_32 19.039       24      0.000  0.750
33 item_33 14.752       24      0.000  0.928
34 item_34 15.478       22      0.000  0.841
35 item_35 21.033       22      0.000  0.519
36 item_36 27.446       22      0.022  0.195
37 item_37 19.960       23      0.000  0.644
38 item_38 15.718       25      0.000  0.923
39 item_39 21.917       19      0.018  0.288
40 item_40 14.849       22      0.000  0.869
```

Only `item 3` and `item 27` are statistically flagged under the family-wise Type I error rate of .05, with the Bonferroni correction being poorly fitted items. [*I found this type of interpretation from a textbook. To my current knowledge, I am not very clear about this test. I will dive in later.*]