HUDM6052 Psychometric II Homework_04

Chenguang Pan

2023-11-14

Contents

Q1-a-Parameters-Estimation

Fit the 1PL, 2PL, and 3PL models…report the estimated item parameters in separated tables

My Solution:
To make the layout concise and good-looking, I intentionally omitted the codes for data cleaning and some instant display of running outcomes. I attached the estimated item parameters from all three models into one table to save space.

```
> # load the binary response dataset
> library(mirt)
> df <- read.csv("/Users/panpeter/Desktop/PhD_Learning/HUDM6052 Psychometric II/HUDM6052_Psychometic_II/
> df <- df[,-1]
>
> # ---------------------------------------------------
> #                        Run 1PL
> # ---------------------------------------------------
>
> # specify the model for 33 items loading on 1 dimension
> # and constrain all the item slope to be equal for 1PL estimation
> spec <- 'F = 1-33
+ CONSTRAIN = (1-33, a1)'
>
> # estimated the model
> # since I constrained all slopes to be equal, here the argument "2PL" is safe
> irt_1pl <- mirt(df, model = spec, itemtype = "2PL", SE=T)
Iteration: 1, Log-Lik: -20276.377, Max-Change: 0.18304Iteration: 2, Log-Lik: -20265.381, Max-Change: 0.0

Calculating information matrix...
> irt_1pl
```

```
Call:
mirt(data = df, model = spec, itemtype = "2PL", SE = T)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 25 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 =  50.40657

Log-likelihood = -20260.22
Estimated parameters: 66
AIC = 40588.44
BIC = 40756.74; SABIC = 40648.75
```

```
> # --------------------------------------------------
> #                       Run 2PL
> # --------------------------------------------------
> irt_2pl <- mirt(df, model = 1, itemtype = "2PL", SE=T)
Iteration: 1, Log-Lik: -20276.377, Max-Change: 0.63072Iteration: 2, Log-Lik: -20095.717, Max-Change: 0.

Calculating information matrix...
> irt_2pl

Call:
mirt(data = df, model = 1, itemtype = "2PL", SE = T)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 32 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 =  11.16451

Log-likelihood = -20078.62
Estimated parameters: 66
AIC = 40289.23
BIC = 40615.92; SABIC = 40406.3
```

```
> # --------------------------------------------------
> #                       Run 3PL
> # --------------------------------------------------
> # specify the model
```

```r
> spec <- 'F = 1-33
+ PRIOR = (1-33, g, norm, -1.1, 2)'
> irt_3pl <- mirt(df, model = spec, itemtype = "3PL", SE = T)
Iteration: 1, Log-Lik: -20990.083, Max-Change: 1.61517Iteration: 2, Log-Lik: -20300.703, Max-Change: 1.
```

Items	1PL				2PL				3PL			
	a	b	g	u	a	b	g	u	a	b	g	u
mathc1	0.960	-0.881	0	1	1.078	-0.816	0	1	1.385	-0.250	0.238	1
mathc2	0.960	-1.706	0	1	1.135	-1.516	0	1	1.198	-1.170	0.198	1
mathc3	0.960	-0.162	0	1	1.307	-0.145	0	1	1.529	0.080	0.095	1
mathc4	0.960	-0.412	0	1	1.308	-0.349	0	1	2.095	0.170	0.235	1
mathc5	0.960	1.104	0	1	0.627	1.556	0	1	2.282	1.443	0.187	1
mathc6	0.960	-0.696	0	1	1.352	-0.569	0	1	1.428	-0.432	0.060	1
mathc7	0.960	-0.457	0	1	1.044	-0.434	0	1	1.247	-0.081	0.141	1
mathc8	0.960	-0.562	0	1	0.897	-0.589	0	1	0.951	-0.395	0.071	1
mathc9	0.960	-2.089	0	1	1.415	-1.609	0	1	1.464	-1.443	0.129	1
mathc11	0.960	-0.668	0	1	1.243	-0.571	0	1	1.956	0.092	0.293	1
mathc12	0.960	0.470	0	1	0.856	0.514	0	1	1.130	0.802	0.114	1
mathc13	0.960	-1.212	0	1	1.535	-0.910	0	1	2.546	-0.207	0.360	1
mathc14	0.960	0.552	0	1	0.775	0.655	0	1	0.887	0.819	0.060	1
mathc15	0.960	0.498	0	1	0.726	0.624	0	1	0.858	0.838	0.075	1
mathc16	0.960	-0.216	0	1	0.829	-0.237	0	1	0.942	0.027	0.090	1
mathc17	0.960	0.715	0	1	0.837	0.795	0	1	1.254	1.050	0.125	1
mathc18	0.960	0.767	0	1	1.107	0.688	0	1	1.253	0.757	0.034	1
mathc19	0.960	0.290	0	1	0.551	0.475	0	1	1.432	1.302	0.297	1
mathc21	0.960	-0.004	0	1	0.607	0.011	0	1	3.411	1.149	0.407	1
mathc22	0.960	-0.253	0	1	0.990	-0.250	0	1	1.401	0.295	0.212	1
mathc23	0.960	-0.085	0	1	0.688	-0.099	0	1	1.428	0.863	0.316	1
mathc24	0.960	-0.052	0	1	0.986	-0.053	0	1	1.460	0.460	0.202	1
mathc25	0.960	-0.170	0	1	1.230	-0.155	0	1	1.517	0.132	0.122	1
mathc26	0.960	0.332	0	1	0.893	0.350	0	1	1.177	0.672	0.125	1
mathc27	0.960	-0.282	0	1	1.479	-0.232	0	1	1.719	-0.018	0.094	1
mathc28	0.960	-0.026	0	1	1.949	-0.044	0	1	2.260	0.100	0.058	1
mathc29	0.960	0.119	0	1	1.229	0.092	0	1	1.928	0.476	0.173	1
mathc31	0.960	1.250	0	1	0.723	1.564	0	1	1.699	1.576	0.150	1
mathc32	0.960	-0.476	0	1	0.837	-0.522	0	1	1.600	0.481	0.350	1
mathc33	0.960	0.176	0	1	1.284	0.134	0	1	2.790	0.582	0.217	1
mathc34	0.960	0.165	0	1	0.678	0.229	0	1	0.863	0.645	0.131	1
mathc35	0.960	0.667	0	1	0.360	1.577	0	1	1.480	1.899	0.288	1
mathc36	0.960	1.449	0	1	1.060	1.352	0	1	1.601	1.359	0.070	1

Due to the limitation of `mirt package`, I can't constrain all the α to be 1. Rather, I can only set them to be equal across all the items. Therefore, in the estimation for the 1PL, the estimated universal α is .96 here.

Q1-a-(1)

Does it appear reasonable to assume all the items having an equal slope...

My Solution:
No.

From a aspect of test development, since this is a test about math placement, we should expect that items can discriminate students with different traits well. In addition, comparing the estimated parameters from `2PL` model versus `1PL`, these items' levels of discrimination spread along a wide range. It is reasonable to have items with higher levels of discrimination than others.

From a mathematics perspective, since the 1PL model is nested in the 2PL model, I conducted the `Likelihood Ratio Test` to compare the two models as followed:

$$D = -2[ln(L_{1pl}) - ln(L_{2pl})].$$

Plug the log likelihood estimated from the above code chunck, then one can have $D = 363.2$ at the degree of freedom of $df = df_{2pl} - df_{1pl} = 66 - 34 = 32$. Based on the Chi-squared distribution, the p value is lower than .001. Therefore, 2PL is better than 1PL, which means the discrimination is preferred.

Q1-a-(2)

Does it appear useful to include a guessing parameter in the model...

My Solution:

Yes, it is useful.

Intuitively, it is reasonable to include a guessing parameter since this a test with multiple choice and guessing is very possible. In addition, by looking through all the guessing parameters, one can find that the `matchc13`, `matchc21`, and `matchc32` do have quite high guessing rate, i.e., all above .30.

However, in terms of model comparison, when using the LRT test again to compare the 2PL vs 3PL model, one can have $D = 28.28$ at 33 degree of freedom, $P = .701$. Based on the parsimony rule, one should endorse the simpler model, i.e., the 2PL.

Therefore, my overall conclusion is including a guessing parameter is useful in this scenario. A practitioner should choose the either model based on their purpose since these two models do not differ a lot.

Q1-a-(3)

Evaluate the goodness of fit of the items with the option of the chi-square test...

My Solution:

I conduct the item fit analysis on each model and summarize the results into one table to make the layout concise.

```
> # get the item fit indices for each model
> item_fit_1pl <- itemfit(irt_1pl, na.rm = T)
> item_fit_2pl <- itemfit(irt_2pl, na.rm = T)
> item_fit_3pl <- itemfit(irt_3pl, na.rm = T)
```

```
>
> # combine all the outputs into one table
> item_fit_all <- cbind(item_fit_1pl[,c("item")],
+                       round(item_fit_1pl[,c("S_X2", "p.S_X2")],3),
+                       round(item_fit_2pl[,c("S_X2", "p.S_X2")],3),
+                       round(item_fit_3pl[,c("S_X2", "p.S_X2")],3))
> names(item_fit_all)[1] <- "item"
>
> # get all the item fit indices for 1PL, 2PL, and 3PL model
> item_fit_all
      item    S_X2 p.S_X2   S_X2 p.S_X2   S_X2 p.S_X2
1   mathc1  24.871  0.413 23.057  0.457 23.880  0.354
2   mathc2  28.123  0.211 30.474  0.107 28.602  0.124
3   mathc3  27.272  0.292 22.457  0.493 21.706  0.478
4   mathc4  26.545  0.326 19.225  0.631 17.456  0.683
5   mathc5  97.340  0.000 52.836  0.001 39.324  0.018
6   mathc6  35.284  0.064 26.846  0.217 23.995  0.293
7   mathc7  30.637  0.165 30.432  0.171 30.827  0.127
8   mathc8  31.167  0.149 31.523  0.139 23.075  0.456
9   mathc9  31.551  0.085 16.430  0.690 12.867  0.799
10 mathc11  24.730  0.421 17.479  0.736 16.875  0.719
11 mathc12  23.575  0.486 22.830  0.530 20.698  0.600
12 mathc13  35.908  0.042 24.648  0.263 19.390  0.368
13 mathc14  27.538  0.280 23.119  0.513 20.507  0.611
14 mathc15  33.526  0.093 21.041  0.690 17.235  0.798
15 mathc16  26.681  0.320 25.327  0.388 22.755  0.534
16 mathc17  21.267  0.623 17.968  0.805 17.158  0.801
17 mathc18  21.575  0.605 22.317  0.501 19.125  0.638
18 mathc19  54.048  0.000 20.763  0.706 20.199  0.685
19 mathc21  61.763  0.000 33.258  0.125 23.345  0.441
20 mathc22  20.404  0.674 20.405  0.674 20.811  0.593
21 mathc23  20.837  0.648 14.494  0.952 13.527  0.957
22 mathc24  20.931  0.643 21.081  0.634 21.063  0.577
23 mathc25  24.945  0.409 21.333  0.561 21.436  0.494
24 mathc26  19.354  0.733 17.673  0.819 18.224  0.745
25 mathc27  45.053  0.006 29.119  0.111 27.443  0.157
26 mathc28  65.450  0.000 24.167  0.235 23.947  0.198
27 mathc29  20.699  0.656 16.372  0.839 14.253  0.892
28 mathc31  38.201  0.024 24.738  0.477 22.294  0.503
29 mathc32  26.751  0.316 22.572  0.545 21.635  0.542
30 mathc33  29.067  0.218 21.204  0.569  8.064  0.995
31 mathc34  37.352  0.040 26.872  0.362 24.880  0.412
32 mathc35 109.636  0.000 32.770  0.205 35.397  0.103
33 mathc36  28.999  0.220 25.182  0.341 23.768  0.360
```

The results show that all items fit well in all models except the `mathc5` item.

Q1-a-(4)

Evaluate the overall fit of the model. Which model do you prefer for this data?...

My Solution:

```
> # get the fit indices for 1PL model
> M2(irt_1pl, na.rm = T)
            M2  df p      RMSEA    RMSEA_5   RMSEA_95      SRMSR       TLI
stats 992.2492 527 0 0.03104487 0.02805368 0.03398001 0.05781072 0.9632371
            CFI
stats 0.9633067
>
> # get the fit indices for 2PL model
> M2(irt_2pl, na.rm = T)
            M2  df            p      RMSEA    RMSEA_5   RMSEA_95      SRMSR
stats 678.0246 495 7.852727e-08 0.02009113 0.01617592 0.02371771 0.03274386
            TLI       CFI
stats 0.9846029 0.9855652
>
> # get the fit indices for 3PL model
> M2(irt_3pl, na.rm = T)
            M2  df            p      RMSEA    RMSEA_5   RMSEA_95      SRMSR
stats 604.7095 462 8.359281e-06 0.01836359 0.01400902 0.02228458 0.03106851
            TLI       CFI
stats 0.9871369 0.9887448
```