Item Response Theory in R: Model Selection and Diagnostics

Dr. Matthew Zeigenfuse

Lehrstuhl für Psychologisches Methodenlehre, Evaluation und Statistik
Psychologisches Institut
Universität Zürich

January 19, 2016

Model Selection

- A common strategy in IRT modeling is to fit multiple nested models of increasing complexity and select the one providing the best fit to estimate abilities
- An example of nested models is the Rasch and 2PL models (the Rasch model is nested in the 2PL)
- The `mirt` package provides an `anova` method for this purpose
- The function takes two fitted models resulting from calls to the `mirt` function

anova Example

```
> X <- expand.table(LSAT6)
> fitRM <- mirt(X, 1, "Rasch")
> fit2PL <- mirt(X, 1, "2PL")

> anova(fitRM, fit2PL)

Model 1: mirt(data = X, model = 1, itemtype = "Rasch")
Model 2: mirt(data = X, model = 1, itemtype = "2PL")

       AIC     AICc    SABIC      BIC    logLik
1 4945.875 4945.960 4956.266 4975.322 -2466.938
2 4953.307 4953.529 4970.624 5002.384 -2466.653
     X2  df      p
1   NaN NaN    NaN
2 0.569   4 0.9664
```

Information Criteria

- The AIC, AICc, SABIC and BIC report the AIC and BIC and their sample-size corrected counterparts
- These criteria can be used to compare nested models
- They balance fit against model complexity by penalizing overall fit by a function of the number of parameters
- The function differs by criterion
- In all cases, the model with the lowest score provides the best balance of fit against complexity

Likelihood Ratio Test

- In addition to computing information criteria, the `anova` function also performs a likelihood ratio test (LRT)
- LRTs test the null hypothesis that the simpler model is sufficient to account for the data
- It does so by comparing the fit of the simpler model to that of a more complex model in which the simpler model is nested
- For example, an LRT involving the Rasch and 2PL models tests the null hypothesis that the Rasch model is sufficient to account for the observed test responses by comparing its fit to that of the 2PL
- A low *p*-value suggests that the simpler model (e.g., Rasch) does not provide a good account of the data

Test Fit

- In addition to comparative measures of model fit, the `mirt` package also provides the M2 and M2* statistics for assessing the fit of an IRT model (Maydeu-Olivares & Joe, 2006)
- These statistics deal with the sparsity problems that can arise when computing G^2 statistics
- The M2 statistic is appropriate for dichotomous data and the M2* is appropriate for polytomous data
- The `M2` function computes the appropriate statistic
- It also computes the RMSEA (along with its 90% confidence interval), the SRMSR and the TLI and CFI (if `calcNull = TRUE`, as it is by default)

M2 Example

```
> M2(fitRM)
            M2 df         p RMSEA RMSEA_5
stats 5.258913  9 0.8111793     0       0
        RMSEA_95      TLI CFI      SRMSR
stats 0.02237086 1.076889   1 0.02242576
> M2(fitRM, calcNull=FALSE)
            M2 df         p RMSEA RMSEA_5
stats 5.258913  9 0.8111793     0       0
        RMSEA_95      SRMSR
stats 0.02237086 0.02242576
```

Item Fit

- `mirt` can compute a number of item fit statistics via the `itemfit` function
- By default, this function will compute
  - Z_h (Drasgow, Levine, & Williams, 1985)
  - Infit, outfit and their Z-scores (Rasch model only)
  - S-X^2 (Orlando & Thissen, 2000)
- It will also compute the χ^2 statistic when `X2 = TRUE`
- These statistics are used to assess the degree to which the IRT model captures the observed patterns of responses for an item
- If ability estimates have already been computed, they can be provided to `itemfit` via the `Theta` argument
- Otherwise they will computed using `fscores` with `method = "EAP"`

itemfit Example 1

```
> itemfit(fitRM)
    item       Zh outfit z.outfit  infit  z.infit
1 Item_1 -0.0206 0.8189  -1.5570 1.0772   0.7264
2 Item_2  3.9994 0.8203  -5.2300 0.8845  -3.6637
3 Item_3 12.0121 0.8145 -12.9393 0.8267 -12.7176
4 Item_4  2.5170 0.8231  -3.9262 0.9183  -1.9600
5 Item_5  0.5527 0.8264  -2.1871 1.0132   0.2006
    S_X2 df.S_X2 p.S_X2
1 0.4363       3 0.9326
2 1.5763       3 0.6648
3 0.8715       2 0.6468
4 0.1900       3 0.9792
5 0.1904       3 0.9791
```

itemfit Example 1

```
> itemfit(fitRM, X2=TRUE)
    item      Zh outfit z.outfit  infit  z.infit
1 Item_1 -0.0206 0.8189  -1.5570 1.0772   0.7264
2 Item_2  3.9994 0.8203  -5.2300 0.8845  -3.6637
3 Item_3 12.0121 0.8145 -12.9393 0.8267 -12.7176
4 Item_4  2.5170 0.8231  -3.9262 0.9183  -1.9600
5 Item_5  0.5527 0.8264  -2.1871 1.0132   0.2006
        X2 df   p.X2   S_X2 df.S_X2 p.S_X2
1   9.4056  6 0.1520 0.4363       3 0.9326
2  82.1440  6 0.0000 1.5763       3 0.6648
3 100.2582  7 0.0000 0.8715       2 0.6468
4  28.8998  6 0.0001 0.1900       3 0.9792
5   7.8539  6 0.2490 0.1904       3 0.9791
```

Person Fit

- `mirt` will also compute the Z_h and the infit and outfit statistics (Rasch model only) using the `personfit` function
- If ability estimates have already been computed, they can be provided to `itemfit` via the `Theta` argument
- Otherwise they will computed using `fscores` with `method = "EAP"`

personfit Example

```
> abilRM <- fscores(fitRM, method="WLE")
> head(personfit(fitRM, Theta=abilRM))
```

```
    outfit   z.outfit     infit     z.infit
1 0.0875955 -0.7078247 0.1365218 -0.64382107
2 0.0875955 -0.7078247 0.1365218 -0.64382107
3 0.0875955 -0.7078247 0.1365218 -0.64382107
4 0.6844327 -0.1226670 0.9162247  0.02768334
5 0.6844327 -0.1226670 0.9162247  0.02768334
6 0.6844327 -0.1226670 0.9162247  0.02768334
         Zh
1 0.6122874
2 0.6122874
3 0.6122874
4 0.2392340
5 0.2392340
6 0.2392340
```

Differential Item Functioning

- Differential item function (DIF), or measurement bias, occurs when people from different groups with the same ability have different response probabilities for a test item
- The presence of DIF items can result in biased ability estimates for the affected group, leading to unfair tests
- DIF is typically subdivided into two types
  - Uniform: The effect of DIF does not depend on ability level
  - Non-uniform: The effect of DIF depends on the ability of the test taker

The Effect of DIF on Model Parameters

- Suppose we are interested in checking for DIF across men and women in a test
- We fit separate IRT models for men and women
- Dichotomous items
  - Uniform DIF will result in different estimates of the difficulty parameter for each of the two groups
  - Non-uniform DIF will result in different discrimination parameter estimates
- Polytomous items
  - Uniform DIF will result in different category threshold estimates for each of the two groups
  - Non-uniform DIF will result in different discrimination estimates

Testing for DIF

- Testing for DIF using the `mirt` package is a multi-step process
- The first step involves fitting an IRT model for each of the groups using the `multipleGroup` function
- This function is a wrapper to `mirt` which takes a grouping parameter, a grouping variable and any invariances across the groups
- This is illustrated using the FIMS data set from the `TAM` package

Example: Fitting Multiple Groups

```
> library(TAM)
> data("data.fims.Aus.Jpn.scored")
> fims <- data.fims.Aus.Jpn.scored
> X <- fims[, -c(1, 16)]
> country <- factor(fims[, 16], 1:2,
+                   c("Australia", "Japan"))
> fit2Group <- multipleGroup(X, 1, group = country,
+                            itemtype = "Rasch")
```

Global LRT

Does a model with two groups fit better than a single model?

```
> fit1Group <- mirt(X, 1, itemtype = "Rasch",
+                          verbose = FALSE)
> anova(fit1Group, fit2Group)
```

```
Model 1: mirt(data = X, model = 1, itemtype = "Rasch", verb
Model 2: multipleGroup(data = X, model = 1, group = country

       AIC     AICc    SABIC      BIC    logLik
1 94269.78 94269.85 94323.51 94371.17 -47119.89
2 91845.23 91845.52 91952.68 92048.01 -45892.61
        X2  df   p
1      NaN NaN NaN
2 2454.552  15   0
```

Testing Individual Items for DIF

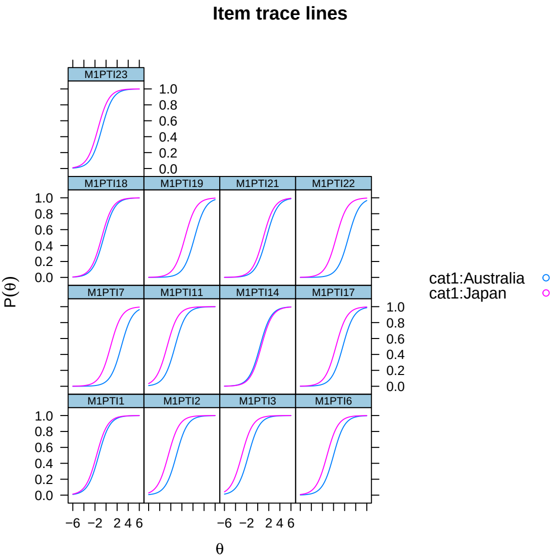
- Individual items can be tested for DIF using the `DIF` function
- At a minimum, this function must be supplied the result from `multipleGroup` (`MGmodel`) and the parameter(s) to be tested (`which.par`)
- By default, `DIF` computes a number of information criteria and performs a LRT between a model where `which.par` for item i is invariant across groups and a model where it is not
- By setting `Wald = TRUE`, `DIF` will instead perform Wald tests
- It can also automatically produce ICC or category probability plots for items exhibiting DIF by setting `plotdif = TRUE`

DIF in FIMS (1)

To test for uniform DIF across countries in the FIMS data using LRTs

```
> difres <- DIF(fit2Group, "d", plotdif = TRUE)
```

DIF in FIMS (2)


Item trace lines
M1PTI23
M1PTI18
M1PTI19
M1PTI21
M1PTI22
M1PTI7
M1PTI11
M1PTI14
M1PTI17
M1PTI1
M1PTI2
M1PTI3
M1PTI6
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−6 −2 2 4 6
θ
cat1:Australia
cat1:Japan

DIF in FIMS (3)

```
> difres$M1PTI1

       AIC     AICc    SABIC      BIC    logLik
1 91869.95 91870.22 91973.82 92065.97 -45905.97
2 91845.23 91845.52 91952.68 92048.01 -45892.61
      X2  df   p
1    NaN NaN NaN
2 26.719   1   0
```

Anchor Items

- Anchor items are typically used to equate the ability distributions of the two groups when testing for DIF
- These items are assumed not to contain DIF
- Test takers of the same ability level will have the same probability of answering an anchor item regardless of group
- These items anchor the ability distributions, allowing differences in item parameters to be distinguished from group difference in ability

Anchor Items in `mirt`

- We can set anchor items using the `invariance` argument to `multipleGroup`
- To do this, we should first provide a character vector giving the item names that we would like to fix across groups
- We should also free the mean and variance parameters to vary across groups

FIMS with Anchors (1)

Fit the FIMS data with the first four items as anchors

```
> itemnames <- names(X)
> fit2GroupAnchor <-
+   multipleGroup(X, 1, group = country,
+                 invariance = c(itemnames[1:4],
+                                "free_means",
+                                "free_var"),
+                 itemtype = "Rasch",
+                 verbose = FALSE)
```

FIMS with Anchors (2)

```
> difWithAnchor <- DIF(fit2GroupAnchor, "d",
+                      items2test = itemnames[-(1:4)],
+                      plotdif = TRUE)
```

FIMS with Anchors (3)