IRT Scale Constraints Using the mirt Package

Derek Briggs

EDUC 8720

Contents

Introduction and Background

The purpose of this document is both educational and practical. Educationally speaking, I will demonstrate the equivalent relationships between constraints that can be imposed to identify or “anchor” the scale of an IRT model. I’ll mostly focus on the Rasch Model. Practically speaking, I will show the syntax that needs to be employed to accomplish all this using functions with Phil Chalmer’s mirt package.

The indeterminacy of the logit scale

When we measure things like length, duration and temperature, a key decision point for the measuring scale is the definition of the scale’s origin or zero point, and a unit. For height and duration, the origin for most measuring scales is pretty straightforward: it represents no length, or no passage of time. For temperature, this is more complicated, because for the two most commonly used scales, Celsius and Fahrenheit, zero on the scale doesn’t mean no temperature. In other words, temperature is typically measured on an interval scale, while height and duration are measured on ratio scales. In the context of educational and psychological attributes, we typically only aspire to a measuring scale with interval properties. But this means we need to purposefully define what a “0” means, or to transform the scale such that there is no 0.

We also need to define the unit of the scale. For length we could choose inches or feet, millimeters or centimeters. For temperature we could choose degrees C and F. In some sense the choice is arbitrary (e.g., should we use centimeters or inches?) But in another sense the choice is critical because it depends on the precision that is desired and our ability to enact a measuring procedure that accurately record these distinctions (e.g., should we use meters, centimeters or millimeters?)

It is easy to see the arbitrary nature of the logit scale in the context of an IRT model. Consider the logit formulation of the 2PL

$$logit(X_{pi}) = a_i(\theta_p - b_i)$$

There are many many different combinations of a_i, b_i and θ_p that will produce the same log odds of a correct item response. A key feature of IRT models is that they make it possible to locate persons and items on a common scale. But without imposing constraints, there is no way to establish an unambiguous interpretation for the distance between the location of any item and the location of any person.

Scale constraints in IRT models

Mathematically, the Rasch Model can be viewed as a constrained version of the 2PL in which

$$logit(X_{pi}) = a(\theta_p - b_i)$$

OR

$$logit(X_{pi}) = \theta_p - b_i$$

In either case, we are imposing the constraint that all items are equally discriminating, which means that if you think of each item as a replication of the basic measurement procedure, then the procedure always involves the same slope parameter, which is a. In the second formulation, we have simply set $a = 1$.

It would be an understandable mistake to think that we learn something new by getting an estimate of a instead of constraining it to be 1. To see why this is a mistake I will apply some code Phil Chalmers includes on p. 96 of his mirt manual.

The following illustration is based on 5 LSAT items taken by 1000 examinees. This little data set is contained within the mirt package. Let's open mirt and load the data into an active dataframe. Note that the function **itemstats** within mirt computes classical item stats much like we get using the function **alpha** in the psych package.

```r
library(mirt)
```

```
## Loading required package: stats4
```

```
## Loading required package: lattice
```

```r
dat <- expand.table(LSAT6)
itemstats(dat)
```

```
## $overall
##     N mean_total.score sd_total.score ave.r sd.r alpha SEM.alpha
##  1000            3.819          1.035 0.077 0.03 0.295     0.869
##
## $itemstats
##           N K  mean    sd total.r total.r_if_rm alpha_if_rm
## Item_1 1000 2 0.924 0.265   0.362         0.113       0.275
## Item_2 1000 2 0.709 0.454   0.567         0.153       0.238
## Item_3 1000 2 0.553 0.497   0.618         0.173       0.217
## Item_4 1000 2 0.763 0.425   0.534         0.144       0.246
## Item_5 1000 2 0.870 0.336   0.435         0.122       0.266
##
## $proportions
##            0     1
## Item_1 0.076 0.924
## Item_2 0.291 0.709
## Item_3 0.447 0.553
## Item_4 0.237 0.763
## Item_5 0.130 0.870
```

Now, let's calibrate these items using mirt and choose the default specification for the Rasch Model where $a = 1$. (Note: below the **verbose=FALSE** option suppresses info about the iterations of the MMLE estimation algorithm. I'm doing this to keep it from cluttering up the R Markdown document.)

```
m1 <- mirt(dat, 1, itemtype = 'Rasch',verbose=FALSE)
#Item parameter estimates
coef(m1, simplify=TRUE)
```

```
## $items
##         a1     d g u
## Item_1   1 2.731 0 1
## Item_2   1 0.999 0 1
## Item_3   1 0.240 0 1
## Item_4   1 1.307 0 1
## Item_5   1 2.100 0 1
##
## $means
## F1
##  0
##
## $cov
##       F1
## F1 0.572
```

In looking at the mirt output, recall that mirt parameterizes IRT models in slope intercept form, $a_i(\theta_p + d_i)$ instead of $a_i(\theta_p - b_i)$. In this case since all $a_i = 1$ to get item difficulty instead of easiness, you would just multiply d_i by -1. Or we could have invoked the option **IRTpars=TRUE**.

Notice in this specification the standard deviation of the ability distribution is 0.76.

```
#SD of theta distribution
sqrt(coef(m1)$GroupPars[2])
```

```
## [1] 0.7561932
```

Also, the default in mirt is to fix the location of the scale such that the mean of the ability distribution is 0.

```
#Mean of theta distribution
coef(m1)$GroupPars[1]
```

```
## [1] 0
```

Now, let's see what happens when we use the alternate specification where a is constrained to be the same for each item (but not necessarily 1).

```
model <- 'F = 1-5
CONSTRAIN = (1-5, a1)'
m2 <- mirt(dat, model,verbose=FALSE)
coef(m2, simplify=TRUE)
```

```
## $items
##          a1     d g u
## Item_1 0.755 2.730 0 1
## Item_2 0.755 0.999 0 1
## Item_3 0.755 0.240 0 1
## Item_4 0.755 1.307 0 1
## Item_5 0.755 2.100 0 1
##
## $means
## F
## 0
##
## $cov
##   F
## F 1
```

```
#SD of theta distribution
sqrt(coef(m2)$GroupPars[2])
```

```
## [1] 1
```

```
#Mean of theta distribution
coef(m2)$GroupPars[1]
```

```
## [1] 0
```

Notice the key difference in syntax: the command "CONSTRAIN" above accomplishes the following "for items 1-5, constrain the parameter a1 to be the same". We now get an estimate of $a1 = .76$. Does this value seem familiar? Yes! It is the same value we got for the SD of the ability distribution in our original mirt specification. So the common slope (i.e., discrimination) parameter for items is not just related to the SD of the ability distribution in the original model specification, it is identical.

To sum up.

We fit two equivalent versions of the Rasch Model.

- In the first specification $logit(X_{pi}) = (\theta_p - b_i)$: $a1 = 1$ and the $SD(\theta) = .76$
- In the second specification $logit(X_{pi}) = a(\theta_p - b_i)$: $a1 = .76$ and the $SD(\theta) = 1$

Now let's see what happens if we specify the 2PL with this same data. We identify the scale by fixing the mean and SD of the ability distribution to 0,1.

```
m3 <- mirt(dat, 1, itemtype = '2PL',verbose=FALSE)
#Item parameter estimates
coef(m3, simplify=TRUE, IRTpars=TRUE)
```

```
## $items
##           a      b g u
## Item_1 0.825 -3.361 0 1
## Item_2 0.723 -1.370 0 1
## Item_3 0.890 -0.280 0 1
## Item_4 0.689 -1.866 0 1
## Item_5 0.658 -3.123 0 1
##
## $means
## F1
##  0
##
## $cov
##    F1
## F1  1
```

What if we take the mean of the 5 item discrimination parameter estimates? You guessed it: 0.76

The logit as a unit of measurement

For any IRT model, the default unit of measurement is a logit. The interpretation of this unit is not straightforward! A starting point is to think in terms of a one unit difference between person ability and item difficulty, which is the central kernel of the logit formulation of an IRT model.

If we take exp(1) then we can express a one unit difference between ability and difficulty in terms of a change in the odds of an event occurring (e.g., the event of a student answering a test item correctly) starting from some baseline. Here we see that exp(1) = 2.72. So relative to a baseline of even odds, an increase of 1 logit increases our odds ratio to 2.7. Or, we can express a one unit difference between ability and difficulty in terms of a change of probability. For example, if we go from a difference of 0 to 1 logit, the probability of a correct response goes up from plogis(0) = 0.5 to plogis(1) = 0.73.

How should we interpret a relative to the logit as a unit of measurement? For every one unit change in the *difference* between the ability of person p and the difficulty of item i, the log odds of a correct response increases by a. In the default specification of the Rasch Model, where $a = 1$, all differences between the locations of people and items on the scale have a direct interpretation in terms of the log odds of a correct response.

In the alternative specification, where a is some constant and need not equal 1, these differences $(\theta_p - b_i)$ get "filtered" (multiplied) by the value of a. However, a convenient thing in the alternate specification is that since SD(θ) is constrained to be 1, the unit of measurement of one logit is equivalent to the SD of the person ability distribution.

Anchoring on item difficulty

There is no simple option within the mirt function that will set the sum (equivalently, the mean) of item difficulty estimates to 0. Phil Chalmers provides some code that can supposedly be used to make this happen via "approach 3" described in https://philchalmers.github.io/mirt/html/Three-Rasch.html but I frankly can't make sense of what he's doing. If you can figure this out, then perhaps you can tailor his code accordingly.

That said, it is not hard to enact this post hoc. That is, you can estimate item parameters using mirt's default constraint that the person ability distribution has a mean of 0, and then after extracting the item parameters and estimating (i.e., scoring) person parameters, you can simply change set the location of 0 to represent the mean item difficulty. Here's an example. In this example I use eap scoring to get theta values.

```
#Pull item parameter estimates from earlier Rasch Model calibration
rasch.diffs.m<-coef(m1, simplify=TRUE, IRTpars=TRUE)$items[,2]
#Generate person ability estimates
rasch.theta.eap.0<-fscores(m1,method="EAP",fullscores.SE = TRUE)[,1]
#Recenter item parameters by subtracting the mean item difficulty from each value
```

```r
rasch.diffs.0<-rasch.diffs.m-mean(rasch.diffs.m)
#Now shift person ability by also subtracting mean item difficulty from each value
rasch.theta.eap.m<-rasch.theta.eap.0-mean(rasch.diffs.m)
```

- For the default mirt scale, we characterize the locations of items and persons using rasch.diffs.m and rasch.theta.eap.0. On this scale, the mean of rasch.diffs.m is -1.47 and mean of rasch.theta.eap.0 is 0 which can be interpreted as follows: relative to the location on the scale of the average person taking this test ($\theta_p = 0$), the location of the average item is 1.47 logits lower. And since plogis$(0-(-1.47)) = 0.81$ we can predict that if the average person was given an item of average difficulty for this test, they would have an 81% chance of answering it correctly.

- For our alternate mirt scale, we characterize the locations of items and persons using rasch.diffs.0 (item difficulty parameters) and rasch.theta.eap.m. On this scale, the mean of rasch.diffs.0 is 0 and mean of rasch.theta.eap.m is 1.47 which can be interpreted as follows: relative to the location on the scale of the average item on this test ($b_i = 0$), the average person is 1.47 logits higher. But notice that since plogis$(1.47-(0)) = 0.81$ we can predict that if the average person was given an item of average difficulty for this test, they would have an 81% chance of answering it correctly.

What I've shown in this last section about anchoring the scale (e.g., choosing a location for 0 on the scale) would also apply if we were fitting the 2PL or 3PL models. There is also nothing to stop you from anchoring the scale by choosing a specific item (or person!) and constraining the value of that parameter to be 0. I will show you how to fix item parameters to specific values in mirt in another R Markdown document.