IRT Observed-Score Kernel Equating with the R Package kequate

Björn Andersson
Uppsala University

Marie Wiberg
Umeå University

Abstract

The R package **kequate** enables observed-score equating using the kernel method of test equating. We present the recent developments of **kequate**, which provide additional support for item-response theory observed score equating using 2-PL and 3-PL models in the equivalent groups design and non-equivalent groups with anchor test design using chain equating. The implementation also allows for local equating using IRT observed-score equating. Support is provided for the R package **ltm**.

Keywords: kernel equating, observed-score test equating, item-response theory, R.

1. Introduction

The kernel method of test equating (von Davier, Holland, and Thayer 2004) is a flexible observed-score equating framework which enables the equating of two tests using all common equating designs. Kernel equating has usually been described using pre-smoothing through log-linear models but the framework provides support for input data of various types, such as observed data and data derived from IRT models. Here, we focus on IRT observed-score equating in the kernel method of test equating. We introduce IRT observed-score kernel equating in the equivalent groups (EG) design and non-equivalent groups with anchor test (NEAT) design using chain equating (CE) and illustrate how to conduct these equating methods using the R (R Development Core Team 2013) package **kequate** (Andersson, Bränberg, and Wiberg 2013). It is also shown how local equating using IRT observed-score equating van der Linden (2011) can be conducted in **kequate**.

This document has the following structure. In Section 2, IRT observed-score equating in the kernel equating framework is described and in Section 3 the implementation of IRT observed-score equating in **kequate** is introduced. In Section 4 examples of the available methods of IRT observed-score equating in **kequate** are given and in Section 5 future additions to the package are presented.

2. IRT observed-score kernel equating

The kernel equating framework enables the usage of score probabilities which are either observed or estimated using a statistical model. Typically the kernel equating framework has utilized score probabilities derived from log-linear models (Holland, King, and Thayer 1989;

von Davier *et al.* 2004; Lee and von Davier 2011). The usage of score probabilities derived from IRT models, which would enable IRT observed-score equating, has been suggested (von Davier 2010) but has not been described in the literature. IRT observed-score equating has however been described in traditional equipercentile equating using linear interpolation (Lord and Wingersky 1984; Kolen and Brennan 2004). The asymptotic standard errors of equating for IRT observed-score equating in various NEAT designs were given in Ogasawara (2003). For kernel equating, the necessary components are the covariance matrices of the score probabilities which are needed to calculate the asymptotic standard errors of equating. In this section we show how the results of Ogasawara (2003) can be applied in the kernel equating framework for the NEAT CE design in the case of an external anchor test under the three parameter logistic model (3-PL). The results for the EG design and when using the two parameter logistic model (2-PL) are similar, but simpler, and are therefore omitted.

2.1. IRT observed-score kernel equating in the NEAT CE design

Let X and Y denote two tests, each with k number of items. For the sake of simplicity we assume an equal number of items on the tests in this section but the results apply to the case where the number of items are not equal and the implementation in **kequate** allows for a non-equal number of items. The tests consist of k^* unique items and k_A common items. Denote the subtests of unique items X^* and Y^* and the subtest of common items A. Each test is administered to a separate group of test takers each from a separate population. Denote the populations P and Q, respectively, with samples sizes n and m for the respective test groups.

Let Θ_P and Θ_Q be the random variables corresponding to the ability level of a member of the population from which each test taker for tests X and Y is taken. Now, let $P_{Xl}(\theta_P)$ and $P_{Yl}(\theta_Q)$ be the probabilities to answer item l of tests X and Y correctly, viewed as a functions of the ability levels θ_P and θ_Q. With the 3-PL model we have that

$$P_{Xl}(\theta_P) = c_{Xl} + \frac{1 - c_{Xl}}{1 + \exp[-a_{Xl}(\theta_P - b_{Xl})]}, \tag{1}$$

where a_{Xl} is the discrimination parameter for item l, b_{Xl} is the difficulty parameter for item l and c_{Xl} is the guessing parameter for item l (Ogasawara 2003). $P_{Yl}(\theta_Q)$ is defined analogously. The 2-PL model is also defined by Equation 1, if $c_{Xl} = 0$. Hence with the 3-PL model we have a total of $3k$ number of parameters across all items for tests X and Y respectively. Let $\boldsymbol{\alpha}_X$ and $\boldsymbol{\alpha}_Y$ denote the $1 \times 3k$ vectors of all item parameters for tests X and Y.

We define $\beta_{X,x}(\theta_P)$ and $\beta_{Y,y}(\theta_Q)$ as the probabilities to obtain score values $x, y \in \{0, 1, \ldots, k\}$ on tests X and Y, respectively, as a function of the ability levels θ_P and θ_Q. Similarly, we define $\beta_{X^*,x^*}(\theta_P)$ and $\beta_{Y^*,y^*}(\theta_Q)$ as the probabilities to obtain the score values $x^*, y^* \in \{0, 1, \ldots, k^*\}$ and $\beta_{A_P,a}(\theta_P)$ and $\beta_{A_Q,a}(\theta_Q)$ as the probabilities to obtain the score values $a \in \{0, 1, \ldots k_A\}$. These probabilities can be obtained by using the procedure outlined in Lord and Wingersky (1984).

Now, let β_{X^*,x^*}, β_{Y^*,y^*}, $\beta_{A_P,a}$ and $\beta_{A_Q,a}$ be the probabilities to obtain score values x^*, y^* and a across all ability levels and let $\boldsymbol{\beta}_{X^*}$ and $\boldsymbol{\beta}_{Y^*}$ be the $1 \times (k^* + 1)$ vectors of probabilities β_{X^*,x^*} and β_{Y^*,y^*} to obtain each of the score values $x^*, y^* \in \{0, 1, \ldots, k^*\}$ on the tests X^* and Y^* and let $\boldsymbol{\beta}_{A_P}$ and $\boldsymbol{\beta}_{A_Q}$ be the $1 \times (k_A + 1)$ vectors of probabilities $\beta_{A_P,a}$ and $\beta_{A_Q,a}$ to obtain each of the score values $a \in \{0, 1, \dots k_A\}$ on test A. We have that

$$\beta_{X^*,x^*} \approx \sum_{r=1}^{R} \beta_{X,x^*}(t_r) W(t_r), \tag{2}$$

where t_r denotes the ability level for the r-th quadrature point, $r \in \{1, 2, \dots, R\}$, and where $W(\cdot)$ is a weight function such that each quadrature point is weighted in accordance with the assumptions made about the distribution of the ability level. Corresponding expressions apply for β_{Y^*,y^*}, $\beta_{A_P,a}$ and $\beta_{A_Q,a}$. We are interested in finding $\mathbf{\Sigma}_{(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'}$ and $\mathbf{\Sigma}_{(\boldsymbol{\beta}_Y^*, \boldsymbol{\beta}_{A_Q})'}$. The results are of the same form for both $(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'$ and $(\boldsymbol{\beta}_Y^*, \boldsymbol{\beta}_{A_Q})'$ so we consider only $(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'$ hereafter. The vector $(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'$ is a function of parameters $\boldsymbol{\alpha}_X$ which are estimated using marginal maximum likelihood. We thus have that $\sqrt{n}(\hat{\boldsymbol{\alpha}_X} - \boldsymbol{\alpha}_X) \to N(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\alpha}_X})$ as $n \to \infty$. Since $(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'$ is a differentiable function of the item parameters, the variance of $(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'$ can be derived using Cramer's theorem, retrieving

$$\sqrt{n}\left[(\boldsymbol{\beta}_X^*, \hat{\boldsymbol{\beta}}_{A_P})' - (\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'\right] \to N\left\{\mathbf{0}, \frac{\partial(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'}{\partial \boldsymbol{\alpha}_X} \mathbf{\Sigma}_{\boldsymbol{\alpha}_X} \left[\frac{\partial(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'}{\partial \boldsymbol{\alpha}_X}\right]'\right\}, \tag{3}$$

where, following Ogasawara (2003), $\frac{\partial(\boldsymbol{\beta}_X^*, \boldsymbol{\beta}_{A_P})'}{\partial \boldsymbol{\alpha}_X}$ is a $(k+1) \times 3k$ matrix of partial derivatives with 1×3 vector entries $\frac{\partial \beta_{X^*,x^*}}{\partial \boldsymbol{\alpha}_{Xl}}$ and $\frac{\partial \beta_{A_P,a}}{\partial \boldsymbol{\alpha}_{l_A}}$, $x^* \in \{0, 1, \dots, k^*\}$, $l \in \{1, 2, \dots, k^*\}$, $a \in \{0, 1, \dots, k_A\}$, $l_A \in \{1, 2, \dots, k_A\}$, such that

$$\begin{aligned}
&\frac{\partial \beta_{X^*,x^*}}{\partial \boldsymbol{\alpha}_{Xl}} = \frac{\partial}{\partial \boldsymbol{\alpha}_{Xl}} \sum_{r=1}^{R} \beta_{X^*,x^*}(t_r) W(t_r) \\
&= \sum_{r=1}^{R} W(t_r) \left\{ \Pr\left[\left(\sum_{p=1}^{k^*} u_{X^*p}\right) = x^*, u_{X^*l} = 1 | \boldsymbol{\alpha}_X, \theta_P = t_r\right] - \beta_{X^*,x^*}(t_r) P_{X^*l}(t_r) \right\} \\
&\times \frac{\langle [P_{X^*l}(t_r) - c_{X^*l}] D(t_r - b_{X^*l}), -[P_{X^*l}(t_r) - c_{X^*l}] D a_{X^*l}, 1 \rangle}{P_{X^*l}(t_r)(1 - c_{X^*l})},
\end{aligned} \tag{4}$$

and

$$\begin{aligned}
&\frac{\partial \beta_{A_P,a}}{\partial \boldsymbol{\alpha}_{Xl_A}} = \frac{\partial}{\partial \boldsymbol{\alpha}_{Xl_A}} \sum_{r=1}^{R} \beta_{A_P,a}(t_r) W(t_r) \\
&= \sum_{r=1}^{R} W(t_r) \left\{ \Pr\left[\left(\sum_{p_A=1}^{k_A} u_{Ap_A}\right) = a, u_{Al_A} = 1 | \boldsymbol{\alpha}_X, \theta_P = t_r\right] - \beta_{A_P,a}(t_r) P_{Al_A}(t_r) \right\} \\
&\times \frac{\langle [P_{Al_A}(t_r) - c_{Al_A}] D(t_r - b_{Al_A}), -[P_{Al_A}(t_r) - c_{Al_A}] D a_{Al_A}, 1 \rangle}{P_{Al_A}(t_r)(1 - c_{Al_A})},
\end{aligned} \tag{5}$$

where u_{X^*l}, u_{X^*p}, u_{Al_A} and u_{Ap_A} take value 0 for an incorrect answer and 1 for a correct answer to each item $l, p \in \{1, 2, \dots k^*\}$, $l_A, p_A \in \{1, 2, \dots k_A\}$. As such,

$$\Pr\left[\left(\sum_{p=1}^{k} u_{Xp}\right) = x^*, u_{Xl} = 1 | \boldsymbol{\alpha}_P, \theta_P = t_r\right] \tag{6}$$

is the probability to achieve score x^* while answering item l correctly for the ability $\theta_P = t_r$. By Bayes theorem we then have

$$\Pr\left[\left(\sum_{p=1}^{k^*} u_{X^*p}\right) = x^*, u_{X^*l} = 1|\boldsymbol{\alpha}_X, \theta_P = t_r\right] =$$
$$\Pr\left[\left(\sum_{p=1}^{k^*} u_{X^*p}\right) = x^*|\boldsymbol{\alpha}_X, \theta_P = t_r, u_{X^*l} = 1\right] \times \Pr\left[u_{X^*l} = 1|\boldsymbol{\alpha}_X, \theta_P = t_r\right]. \qquad (7)$$

From Equation 1 it follows that

$$\Pr\left[u_{X^*l} = 1|\boldsymbol{\alpha}, \theta_P = t_r\right] = P_{X^*l}(t_r) \qquad (8)$$

and we note that

$$\Pr\left[\left(\sum_{p=1}^{k^*} u_{X^*p}\right) = x^*|\boldsymbol{\alpha}, \theta_P = t_r, u_{X^*l} = 1\right] \qquad (9)$$

can be found with the algorithm in Lord and Wingersky (1984) by fixing the probability to answer item l correctly to 1.

Since Equation 3 defines the asymptotic distribution of the score probabilities the results can be directly applied in the kernel equating framework by the derivations provided in von Davier *et al.* (2004).

3. Implementation of IRT observed-score equating in kequate

The package **kequate** for R supports IRT observed-score equating for the EG and NEAT CE designs with the 2-PL or 3-PL IRT models. Asymptotic or bootstrap standard errors are calculated for each of the methods. The input used can either be matrices of observed item responses for each individual or objects containing IRT models which have been estimated using the R package **ltm** (Rizopoulos 2006).

To conduct an IRT observed-score equating in **kequate**, the function `irtose()` is used. The function `irtose()` has the following formal function call:

```
irtose(design="CE", P, Q, x, y, a=0, qpoints, model="2pl", see="analytical",
replications=50, kernel="gaussian", h=list(hx=0, hy=0, hxP=0, haP=0, hyQ=0,
haQ=0), hlin=list(hxlin=0, hylin=0, hxPlin=0, haPlin=0, hyQlin=0, haQlin=0),
KPEN=0, wpen=0.5, linear=FALSE, slog=1, bunif=1, altopt=FALSE)
```

Explanations of each of the arguments supplied to `irtose()` are given in Table 1.

If matrices of responses are provided as input to `irtose()`, the IRT models will be estimated using the R package **ltm**. The settings used in **ltm** will then be the default ones, except for the case of the 3-PL model where the `nlminb` optimizer is used instead of the default. Note that the 3-PL model has issues with convergence, hence it will not always be possible to get stable estimates of item parameters using this model. It is recommended to estimate the 3-PL models separately using the package **ltm**. Currently, **kequate** only provides support for IRT models without particular restrictions on the parameters.

Argument(s)	**Designs**	**Description**
`design`	ALL	A character vector indicating which design to use. Possible designs are `"CE"` and `"EG"`.
`P`, `Q`	ALL	Matrices or objects created by the R package **ltm** containing either the responses for each question in groups P and Q or the estimated IRT models in groups P and Q.
`x`, `y`	ALL	Score value vectors for test X and test Y.
`a`	CE	Score value vector for the anchor test A.
`qpoints`	ALL	A numeric vector containing the quadrature points used in the equating. If not specified, the quadrature points from the IRT models will be used.
`model`	ALL	A character vector indicating which IRT model to use. Available models are 2PL and 3PL. Default is `"2PL"`.
`see`	ALL	A character vector indicating which standard errors of equating to use. Options are `"analytical"` and `"bootstrap"`, with default `"analytical"`.
`replications`	ALL	The number of bootstrap replications if using the bootstrap standard error calculations. Default is 50.
`kernel`	ALL	A character vector denoting which kernel to use, with options `"gaussian"`, `"logistic"`, `"stdgaussian"` and `"uniform"`. Default is `"gaussian"`.
`h`	ALL	Optional argument to specify the continuization parameters manually as a list with suitable bandwidth parameters. In an EG design design: `hx` and `hy`, in a NEAT CE design: `hxP`, `haP`, `hyQ` and `haQ`. (If `linear=TRUE`, then these arguments have no effect.)
`hlin`	ALL	Optional argument to specify the linear continuization parameters manually as a list with suitable bandwidth parameters. In an EG design: `hxlin` and `hylin`, in a NEAT CE design: `hxPlin`, `haPlin`, `hyQlin` and `haQlin`.
`slog`	ALL	The parameter used in the logistic kernel. Default is 1.
`bunif`	ALL	The parameter used in the uniform kernel. Default is 0.5.
`KPEN`	ALL	Optional argument to specify the constant used in deciding the optimal continuization parameter. Default is 0.
`wpen`	ALL	An argument denoting at which point the derivatives in the second part of the penalty function should be evaluated. Default is 1/4.
`linear`	ALL	Logical denoting if a linear equating only is to be performed. Default is `FALSE`.
`altopt`	ALL	Logical which sets the bandwidth parameter equal to a variant of Silverman's rule of thumb. Default is `FALSE`.

Table 1: Arguments supplied to `irtose()`.

4. Examples

For these examples, data was simulated using R in accordance with the 2-PL and 3-PL IRT models. The simulated data for both the 2-PL model and the 3-PL model have the same ability level for each individual and the same discrimination and difficulty parameters for each item. The simulation procedure is identical to that for the 2-PL and 3-PL IRT models described in Ogasawara (2003). The R code which generated the data is given below.

```
R> library(kequate)
R> set.seed(7)
R> akX <- runif(15, 0.5, 2)
R> bkX <- rnorm(15)
R> ckX <- runif(15, 0.1, 0.2)
R> akY <- runif(15, 0.5, 2)
R> bkY <- rnorm(15)
R> ckY <- runif(15, 0.1, 0.2)
R> akA <- runif(15, 0.5, 2)
R> bkA <- rnorm(15)
R> ckA <- runif(15, 0.1, 0.2)
R> dataP <- matrix(0, nrow=1000, ncol=30)
R> dataQ <- matrix(0, nrow=1000, ncol=30)
R> data3plP <- matrix(0, nrow=1000, ncol=30)
R> data3plQ <- matrix(0, nrow=1000, ncol=30)
R> for(i in 1:1000){
+  ability <- rnorm(1)
+  dataP[i,1:15] <- (1/(1+exp(-akX*(ability-bkX)))) > runif(15)
+  dataP[i,16:30] <- (1/(1+exp(-akA*(ability-bkA)))) > runif(15)
+  data3plP[i,1:15] <- (ckX+(1-ckX)/(1+exp(-akX*(ability-bkX)))) > runif(15)
+  data3plP[i,16:30] <- (ckA+(1-ckA)/(1+exp(-akA*(ability-bkA)))) > runif(15)
+  }
R> for(i in 1:1000){
+  ability <- rnorm(1, mean=0.5)
+  dataQ[i,1:15] <- (1/(1+exp(-akY*(ability -bkY)))) > runif(15)
+  dataQ[i,16:30] <- (1/(1+exp(-akA*(ability -bkA)))) > runif(15)
+  data3plQ[i,1:15] <- (ckY+(1-ckY)/(1+exp(-akY*(ability-bkY)))) > runif(15)
+  data3plQ[i,16:30] <- (ckA+(1-ckA)/(1+exp(-akA*(ability-bkA)))) > runif(15)
+  }
```

4.1. IRT observed-score kernel equating with the 2-PL model

For the 2-PL model data was simulated in a non-equivalent groups with anchor test design for two populations of size 1000 with differing ability levels. The main tests had 15 items each and the anchor test had 15 items. The simulated data were stored in matrices `dataP` for group P and `dataQ` for group Q. To equate the two main tests using chain equating, we then call the function `irtose()` as follows:

```
R> eq2pl <- irtose("CE", dataP, dataQ, 0:15, 0:15, 0:15)
```

To display a summary of the equating we write:

```
R> summary(eq2pl)

 Design: IRT-OSE CE

 Kernel: gaussian

 Sample Sizes:
   Test X: 1000
   Test Y: 1000

 Score Ranges:
   Test X:
       Min = 0 Max = 15
   Test Y:
       Min = 0 Max = 15
   Test A:
       Min = 0 Max = 15

 Bandwidths Used:
        hxP       hyQ       haP       haQ   hxPlin   hyQlin  haPlin
1 0.5659422 0.5569558 0.5174676 0.5366143 2989.264 3284.912 3438.16
   haQlin
1 3730.23

 Equating Function and Standard Errors:
   Score       eqYx     SEEYx
1      0 -0.4102940 0.1125946
2      1  0.3508717 0.1492459
3      2  1.1151089 0.1658637
4      3  1.9187788 0.1817399
5      4  2.7824709 0.1880102
6      5  3.6874286 0.1830661
7      6  4.6417963 0.1724347
8      7  5.6350849 0.1573441
9      8  6.6721176 0.1407003
10     9  7.7570920 0.1269212
11    10  8.9026459 0.1205589
12    11 10.1082113 0.1231765
13    12 11.3587737 0.1348788
14    13 12.6381279 0.1445085
15    14 13.8698705 0.1377295
16    15 14.9852291 0.1028491

 Comparing the Moments:
          PREAx       PREYa
```

```
1    0.04021784  0.02397344
2   -0.12154177 -0.05919980
3   -0.89229394 -0.00891477
4   -1.94551884  0.14061826
5   -3.19804894  0.37008459
6   -4.59375169  0.67030506
7   -6.09860794  1.03720124
8   -7.68957605  1.46915126
9   -9.35018326  1.96571350
10 -11.06810250  2.52703115
```

The equating shows that the tests are similar in difficulty but that test Y is slightly more difficult than test X.

When supplying matrices of responses to each item as input to `irtose()`, the IRT models are estimated using the package **ltm**. An equating is then conducted using the estimated IRT models. The objects created by **ltm** are stored in the output from `irtose()`. To access the objects we write:

```
R> irtobjects <- eq2pl@irt
```

This will create a list of the objects created by **ltm** and the adjusted asymptotic covariance matrices of the item parameters. We save the objects from **ltm** for future usage:

```
R> sim2plP <- irtobjects$ltmP
R> sim2plQ <- irtobjects$ltmQ
```

4.2. IRT observed-score kernel equating with the 3-PL model

For the 3-PL model data was again simulated in a non-equivalent groups with anchor test design for two populations of size 1000 with differing ability levels. As before, the main tests had 15 items each and the anchor test had 15 items. In this example, the IRT models were estimated using the function`tpm()` in the package **ltm**, creating the objects `sim3plP` and `sim3plQ` containing the IRT models. For details of IRT model estimation using **ltm**, see Rizopoulos (2006). The resulting objects are then given as input to the function `irtose()` to conduct an equating:

```
R> eq3pl <- irtose("CE", sim3plP, sim3plQ, 0:15, 0:15, 0:15, model="3pl")
R> summary(eq3pl)

 Design: IRT-OSE CE

 Kernel: gaussian

 Sample Sizes:
   Test X: 1000
   Test Y: 1000
```

```
 Score Ranges:
   Test X:
       Min = 0 Max = 15
   Test Y:
       Min = 0 Max = 15
   Test A:
       Min = 0 Max = 15

 Bandwidths Used:
        hxP      hyQ       haP      haQ   hxPlin  hyQlin   haPlin
1 0.5557718 0.541747 0.5475796 0.554931 2751.929 2854.93 3065.727
    haQlin
1 3397.708

 Equating Function and Standard Errors:
   Score       eqYx     SEEYx
1      0  0.3276559 0.2735128
2      1  1.2794952 0.3000089
3      2  2.1655439 0.2951530
4      3  3.0044761 0.2747276
5      4  3.8187669 0.2519666
6      5  4.6197037 0.2295636
7      6  5.4251176 0.2048256
8      7  6.2483820 0.1795047
9      8  7.1033181 0.1566583
10     9  8.0032170 0.1394791
11    10  8.9643124 0.1289517
12    11 10.0057868 0.1242661
13    12 11.1381010 0.1244624
14    13 12.3486810 0.1333691
15    14 13.5779339 0.1444042
16    15 14.7432503 0.1317769

 Comparing the Moments:
          PREAx        PREYa
1    0.006831737  0.005300371
2   -0.136499282 -0.020130677
3   -0.692544150  0.003341095
4   -1.605509643  0.088587650
5   -2.798851142  0.235472154
6   -4.207733210  0.441460694
7   -5.785000345  0.704211100
8   -7.497598584  1.022064319
9   -9.322150367  1.393994453
10 -11.241520639  1.819456386
```

We plot the results with the method for the function `plot()` for the class `keout` created by `itrtose()`.

```
R> plot(eq3pl)
```

The plot is seen in Figure 1.

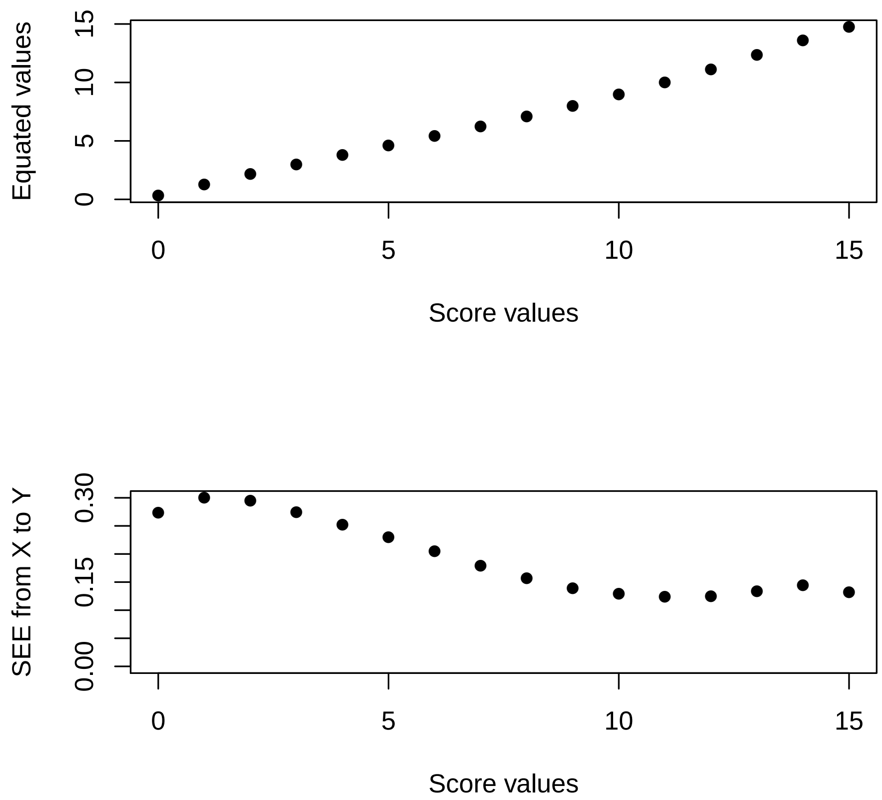

Figure 1: The equated values and standard errors of equating for the IRT observed-score equating using the 3-PL model.

4.3. IRT observed-score local equating

IRT observed-score equating can be utilized when conducting what is called local equating, where different equating functions are calculated based on the ability level or a proxy of the ability level of the individuals taking the tests to be equated. Local equating using IRT observed-score equating is conducted by fixing the ability level to a particular single value or a sequence of values and then only considering this value or sequence of values when calculating the score probabilities. These score probabilities are then used for the equating just as in a regular IRT observed-score equating.

In **kequate**, local equating using IRT observed-score equating can be conducted by adjusting the optional argument `qpoints` in the `irtose()` function call. For example, by specifying `qpoints=1` a local equating for the individuals with the ability level equal to 1 is conducted. The argument `qpoints` can be set to a numeric vector of any length.

As an example, we conduct a local equating for individuals with ability level equal to -1, 0 and 1, respectively, using the simulated 2-PL data previously described. We then call `irtose()` as follows:

```
R> eq2plLOW <- irtose("CE", sim2plP, sim2plQ, 0:15, 0:15, 0:15, qpoints=-1)
R> eq2plAVG <- irtose("CE", sim2plP, sim2plQ, 0:15, 0:15, 0:15, qpoints=0)
R> eq2plHIGH <- irtose("CE", sim2plP, sim2plQ, 0:15, 0:15, 0:15, qpoints=1)
```

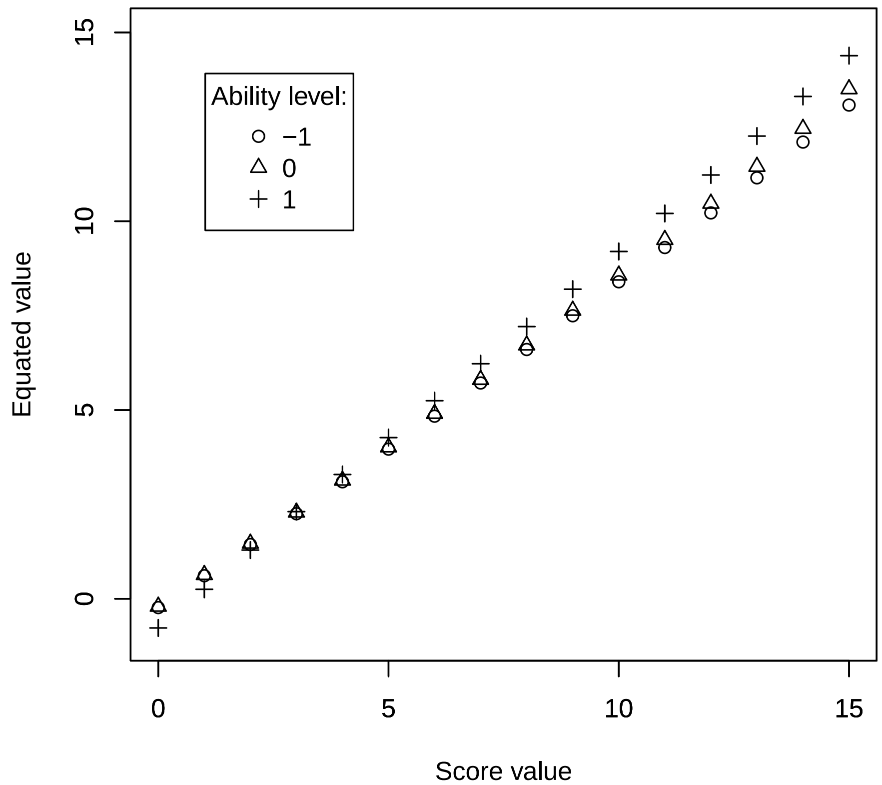

Figure 2: The equated values for each score value for three different ability levels in a local equating in the NEAT CE design.

The results of these equatings are displayed in Figure 2, showing that the equating function is somewhat different for the three different ability levels.

5. Future developments

In the present implementation, only the 2-PL and 3-PL IRT models without parameter restrictions are supported in **kequate**. Future work will include support for the additional IRT models available in **ltm** such as the Rasch model and the 1-PL model and the ability to use the features of parameter restrictions available in **ltm** when conducting IRT observed-score equating. Additionally, the NEAT design using post-stratification equating (PSE) with support for various ways of estimating the equating coefficients is planned to be included in the package.

References

Andersson B, Bränberg K, Wiberg M (2013). ***kequate**: The Kernel Method of Test Equating.* R Package Version 1.3.1, URL http://CRAN.R-project.org/package=kequate.

Holland P, King BF, Thayer DT (1989). "The Standard Error of Equating for the Kernel Method of Equating Score Distributions." *Technical Report 89-83*, Princeton, NJ: Educational Testing Service.

Kolen MJ, Brennan RJ (2004). *Test Equating: Methods and Practices (2nd ed.).* New York: Springer-Verlag.

Lee YH, von Davier AA (2011). "Equating Through Alternative Kernels." In AA von Davier (ed.), *Statistical Models for Test Equating, Scaling, and Linking*. New York: Springer-Verlag.

Lord F, Wingersky M (1984). "Comparison of IRT true-score and equipercentile observed-score "equatings"." *Applied Psychological Measurement*, **8**, 452–461.

Ogasawara H (2003). "Asymptotic standard errors of IRT observed-score equating methods." *Psychometrika*, **68**, 193–211.

R Development Core Team (2013). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rizopoulos D (2006). "ltm: An R Package for Latent Variable Modeling and Item Response Analysis." *Journal of Statistical Software*, **17**(5), 1–25. ISSN 1548-7660. URL http://www.jstatsoft.org/v17/i05.

van der Linden WJ (2011). "Local Observed-Score Equating." In AA von Davier (ed.), *Statistical Models for Test Equating, Scaling, and Linking*. Springer.

von Davier AA (2010). "Equating Observed-Scores: The Percentile Rank, Gaussian Kernel, and IRT Observed-Score Equating Methods." In *International Meeting of Psychometric Society*.

von Davier AA, Holland PW, Thayer DT (2004). *The Kernel Method of Test Equating.* New York: Springer-Verlag.

Affiliation:

Björn Andersson
Department of Statistics
Uppsala University, Box 513
SE-751 20 Uppsala, Sweden
E-mail: bjorn.andersson@statistik.uu.se
URL: http://katalog.uu.se/empInfo?id=N11-1505

Marie Wiberg
Department of Statistics, USBE
Umeå University
SE-901 87 Umeå, Sweden
E-mail: marie.wiberg@stat.umu.se
URL: http://www.usbe.umu.se/om-handelshogskolan/personal/maewig95