Journal of Statistical Software

January 2014, Volume 56, Issue 10. *http://www.jstatsoft.org/*

Classification Accuracy and Consistency under Item Response Theory Models Using the Package classify

Christopher Wheadon
No More Marking Ltd.

Abstract

The R package **classify** presents a number of useful functions which can be used to estimate the classification accuracy and consistency of assessments. Classification accuracy refers to the probability that an examinee's achieved grade classification on an assessment reflects their true grade. Classification consistency refers to the probability that an examinee will be classified into the same grade classification under repeated administrations of an assessment. Understanding the classification accuracy and consistency of assessments is important where key decisions are being taken on the basis of grades or classifications. The study of classification accuracy can help to improve the design of assessments and aid public understanding and confidence in those assessments.

Keywords: item response theory, classification accuracy, classification consistency, model fit, true score, Bayesian IRT, R.

1. Background

Test scores are often reported as classifications or grades. For example, a test may have two classifications, pass or fail, or several grades, from A to E. Wherever scores are reported as grades, users have an interest in understanding how accurate those grades may be. If a candidate were to take a slightly different sample of items from the same domain, for example, would they achieve the same classification or grade? The study of classification accuracy has been used to help to improve the design of assessments (Douglas and Mislevy 2010) and aid public understanding and confidence in those assessments (Wheadon and Stockford 2010).

The study of classification accuracy began with classical test theory (Swaminathan, Hambleton, and Algina 1974) and has been refined over the last 40 years (Lee, Brennan, and Wan 2009; Lee, Hanson, and Brennan 2002; Livingston and Lewis 1995; Subkoviak 1976). There are many possible approaches to the estimation of classification accuracy which involve varying assumptions and have various restrictions. The package **classify** (Wheadon and Stockford 2014) for R (R Core Team 2013) uses the item response theory (IRT) approach (Lee 2008).

The IRT approach to estimating classification accuracy compares the degree to which observed classifications agree with those based on examinees' true scores (Lee, Hanson, and Brennan 2002; Livingston and Lewis 1995). The classification accuracy for an individual is defined as the probability that their observed score falls in the same grade classification as his or her true ability. The classification consistency for an individual is defined as the probability that he or she would be classified with the same grade over successive administrations of a test.

The main advantage of using an IRT approach is that it is extremely flexible. The package **classify** can be applied to tests composed of dichotomous or polytomous items and has built-in support for the Rasch model (Rasch 1960), the partial credit model (Masters 1982), the generalized partial credit model (Muraki 1992) and the two parameter logistic model (Lord and Novick 1968). As **classify** interfaces with **OpenBUGS** (Thomas, O'Hara, Ligges, and Sturtz 2006), **WinBUGS** (Lunn, Thomas, Best, and Spiegelhalter 2000; Spiegelhalter, Thomas, Best, and Lunn 2003), and **JAGS** (Plummer 2003, 2013a) it is relatively straightforward to implement a full range of flexible IRT models using the functionality provided by the package. A range of statistical outputs and plots are provided, including summary statistics, classification accuracy and consistency plots and Cohen's kappa measure of consistency.

This paper presents an in-depth explanation of the algorithm that underpins the calculation of the classification estimations as well as a fully worked example that illustrates the statistical outputs and plots that can be produced from the package. Brief references are made to alternative estimations, but the interested reader should turn to Lee (2008) for a full discussion of the relative strengths and weaknesses of different estimation procedures.

2. Estimation

IRT models yield probabilities of correct responses or, in the case of polytomous items, probabilities of responses in various categories. The probabilities are a function of proficiency in the latent variable under consideration and estimable parameters of the items. Once derived, the probabilities can be used to estimate the classification accuracy and consistency of decisions made on the basis of assessments (Lee 2008).

Although it is a trivial task to compute the probability of a score on a single item, it is more complex to compute the probability of achieving a particular score from multiple items as there are many ways in which each score can be achieved. For example, there are 560 ways to score 3 on a test with 16 dichotomous items. The Wingersky-Lord recursive algorithm (Lord and Wingersky 1984) simplifies the calculation by combining the probabilities of responding in any particular category of an item with the multiple ways in which any test score can be achieved. The Wingersky-Lord algorithm is computationally intensive so the implementation in **classify** takes advantage of the processing efficiency gained by interfacing R with C++ via **Rcpp** (Eddelbuettel and François 2011).

3. The Wingersky-Lord recursion formula

Kolen and Brennan (2004, p. 32) give the example of a group of examinees who have all been administered a three item test, with a probability of success given as $p_{ij}(\vartheta, B)$. Equation 1 is used to find the probability that examinees of ability equal to ϑ_i will incorrectly answer all three items and score 0, where x is the summed score and i is the examinee.

$$\mathsf{P}\left(x = 0|\vartheta_i\right) = \left(1 - p_{i1}\right)\left(1 - p_{i2}\right)\left(1 - p_{i3}\right) \tag{1}$$

To earn a score of 1, an examinee could answer item 1 correctly and items 2 and 3 incorrectly, or the examinee could answer item 2 correctly and items 1 and 3 incorrectly, or item 3 correctly and items 1 and 2 incorrectly. Equation 2 is therefore used to find the probability of the examinee earning a score of 1.

$$\mathsf{P}\left(x = 1|\vartheta_i\right) = p_{i1}\left(1 - p_{i2}\right)\left(1 - p_{i3}\right) + \left(1 - p_{i1}\right)p_{i2}\left(1 - p_{i3}\right) + \left(1 - p_{i1}\right)\left(1 - p_{i2}\right)p_{i3} \tag{2}$$

Equation 3 is used to find the probability of correctly answering two items.

$$\mathsf{P}\left(x = 2|\vartheta_i\right) = p_{i1}p_{i2}\left(1 - p_{i3}\right) + p_{i1}\left(1 - p_{i2}\right)p_{i3} + \left(1 - p_{i1}\right)p_{i2}p_{i3} \tag{3}$$

Equation 4 is used to find the probability of correctly answering three items.

$$\mathsf{P}\left(x = 3|\vartheta_i\right) = p_{i1}p_{i2}p_{i3} \tag{4}$$

The sequence of calculations involved in calculating the probabilities can be generalised using a recursion formula given in Equation 5 (Lord and Wingersky 1984) where r represents the number of valid response categories.

$$\mathsf{P}_r\left(x|\vartheta_i\right) = \begin{cases} \mathsf{P}_{r-1}\left(x\right) = 1 & r = 1 \\ \mathsf{P}_{r-1}\left(x|\vartheta_i\right)\left(1 - p_{ir}\right) & x = 0, r \neq 1 \\ \mathsf{P}_{r-1}\left(x|\vartheta_i\right)\left(1 - p_{ir}\right) + \mathsf{P}_{r-1}\left(x - 1|\vartheta_i\right)p_{ir} & 0 < x < r \\ \mathsf{P}_{r-1}\left(x - 1|\vartheta_i\right)p_{ir} & x = r \end{cases} \tag{5}$$

The recursion formula is used to find the probability of a candidate of any given ability achieving any score.

Following the numerical solution given by Kolen and Brennan (2004, p. 183) the probabilities of success on each item are: 0.26, 0.27 and 0.18. Using the function `wlord` the probability of scoring 0, 1, 2, 3 can be estimated as follows:

```
R> library("classify")
R> probs <- matrix(c(0.74, 0.73, 0.82, 0.26, 0.27, 0.18),
+    nrow = 3, ncol = 2, byrow = FALSE)
R> wlord(probs, cats = c(2, 2, 2))

[1] 0.442964 0.416708 0.127692 0.012636
```

To obtain scores across various abilities the distribution of scores is found at each ability level and accumulated. This can be achieved by integration, or, as it is in the package **classify**, by summing over all abilities.

4. Classification accuracy and consistency

IRT classification uses the probability that candidates of a given ability, ϑ, will correctly answer items of a specified difficulty, β, to estimate how likely it is that a candidate would receive the same grade classification should they take a test again. The first stage in the procedure is to fit an IRT model to derive the probability of success on each item. Equation 6 for example gives the probability of success on an item under the Rasch model.

$$p_{ij} = \frac{e^{\vartheta_i - \beta_j}}{1 + e^{\vartheta_i - \beta_j}} \tag{6}$$

Once the probabilities of success on each item for each candidate have been derived the recursion formula can be used to calculate the expected score distribution for those candidates. The expected score distribution reveals the most likely score for a candidate, their true score or expected score, as well as the probability of achieving all other scores. From this information it is now possible to estimate how likely it is that a candidate would receive the same grade classification should they take the test again.

5. Classification consistency

The conditional classification consistency index is typically defined as the probability that examinees of a given ability are classified into the same category on independent administrations of two parallel forms of test. The summary statistic, the marginal classification index for all ability levels, can then be calculated by obtaining classification indices for every examinee and averaging them over all examinees. Another estimate of classification consistency is the kappa coefficient (Cohen 1960). It is possible that, even with random scores, candidates will achieve the same grade. The kappa coefficient adjusts for the proportion of random consistency that can be expected.

6. Classification accuracy

Classification accuracy is often evaluated by false positive and false negative error rates (Hanson and Brennan 1990; Lee, Hanson, and Brennan 2002). The conditional false positive error rate is defined as the probability that an examinee is classified into a category that is higher than the examinee's true category. The conditional false negative error rate is the probability that an examinee is classified into a category that is lower than an examinee's true category. The true category can be determined by comparing the expected summed score of a candidate with the actual boundaries applied to the overall test. The probability that a candidate of given ability will then be classified into another category allows the false positive and false negative rates to be assessed. The accuracy is then determined by subtracting the incidence of false positives and false negatives from 1. Once again a summary statistic can be calculated by obtaining classification indices for every examinee and averaging them over all examinees.

7. Model fit

Models are useful to the extent in which the predictions made by the models can be verified. If a test showed poor fit to an IRT model then the estimation of classification accuracy from an IRT model could be misleading. One method of assessing IRT model fit is to compare the observed score distribution with the predicted score distribution once the models have been fitted to the data (Béguin and Glas 2001). This comparison can be undertaken as a posterior predictive model checking (PPMC) method (Guttman 1967; Rubin 1984) in a Bayesian Markov chain Monte Carlo (MCMC) framework. The MCMC framework is appealing because of its simplicity, strong theoretical basis, and intuitive appeal (Sinharay 2005). The PPMC method primarily consists of comparing the observed data with replicated data (those predicted by the model). In the case of model fit, the comparison between the observed score distribution and the replicated score distributions can be done visually through plots of the distributions.

Comparing the observed score distribution and the replicated score distributions in a Bayesian framework can be challenging due to the number of posterior samples such a procedure can produce. For example, 1,000 simulations of 1,000 candidates on 21 items yields 21 million responses. For this reason package **classify** takes advantage of the efficiency gained through embedding C++ code with R to provide a comparison of the observed score distribution with the replicated score distributions for tests fitted under MCMC estimation methods.

8. Alternative estimators

While there are many alternative methods by which classification accuracy and consistency of assessments can be estimated, see Wheadon and Stockford (2010), there is an obvious alternative while working in a Bayesian framework. Wainer, Bradlow, and Wang (2007) provide an example where they sample from the posterior and count how many times the sample proficiency of a candidate is above the passing score. If 1,000 draws are taken independently and 900 are above the passing score, the probability of passing is estimated at 0.90. While it is to be expected that the results from the method described by Wainer, Bradlow, and Wang (2007) and those derived from the Wingersky-Lord algorithm are similar, the Wingersky-Lord approach is more mathematically elegant, and produces smoothed probability distributions which would require many thousands of draws to achieve the same smoothness.

9. A worked example

The example dataset `biology` in package **classify** provides polytomous responses from 200 candidates to 31 items. The items are taken from a test of achievement in biology which is typically taken by pupils in England at age 16. The items are open response and expert marked, and have a format such as the following four mark item: 'When a pathogen enters the body it may be destroyed by phagocytosis. Describe how.' While the full dataset consists of many thousands of responses it is recommended that a fairly small sample is taken as the MCMC simulation is computationally intensive.

```
R> data("biology", package = "classify")
R> biology <- biology[complete.cases(biology), ]
R> biology <- biology[sample(1:nrow(biology), 200, replace = FALSE), ]
R> biology <- as.matrix(biology)
```

The number of people and items are then extracted.

```
R> n <- nrow(biology)
R> p <- ncol(biology)
```

Along with the data, the grade boundaries (cut scores) should be entered, along with associated labels. There must be at least one boundary, and there must be one more label than boundaries. The order of boundary labels should reflect the order of the boundaries, which should be ascending.

```
R> bnds <- c(26, 30, 35, 40, 45)
R> lbls <- c("U", "E", "D", "C", "B","A")
```

Both the partial credit model (PCM; Masters 1982) and the generalized partial credit model (GPCM; Muraki 1992) are suitable for modelling these responses. The associated bugs files 'pcm.bug' and 'gpcm.bug' are specified according to Curtis (2010) and are included in the installation directory `bugs` in package **classify**. The PCM assumes that all items have a constant discrimination while the GPCM assumes that the discrimination of items varies, and models the discrimination of each item using the parameter α. These different assumptions may have an impact on the modelled score distribution (Swaminathan, Hambleton, and Rogers 2007). Much of the model fitting code that now follows is taken from Curtis (2010).

```
R> mdl <- "gpcm.bug"
```

For the PCM or the GPCM, category responses should be used instead of item responses. Categories are based on an index of 1, so that a dichotomous item scored 0 or 1 has category values of 1 or 2. Scores can be converted to categories by adding 1 to them. `Y` represents the categories for each response, while `K` represents the categories available in each item.

```
R> Y <- biology + 1
R> K <- as.numeric(apply(Y, 2, max))
```

Then prior values of parameters are set. The prior values specify our uncertainty about the β and α parameters in terms of their expected mean and standard deviation values.

```
R> m.beta <- 0.0
R> s.beta <- 1.0
R> m.alpha <- 0.0
R> s.alpha <- 1.0
```

Next, the data required by the models is supplied along with the parameters we wish to monitor and the BUGS file which specifies the MCMC model.

```
R> data <- list("Y", "n", "p", "K", "m.beta", "s.beta", "m.alpha", "s.alpha")
R> monitor <- c("beta", "theta", "alpha")
R> jags.file <- file.path(path.package("classify", quiet = FALSE),
+    "bugs", mdl)
```

Initial values may also be provided to MCMC models in order to aid convergence. For our purposes these can be set to zero. Where different items have different numbers of categories the matrix can be padded with missing values.

```
R> beta <- t(sapply(1:p, function(j)
+    rep(c(NA, 0.0), c(K[j], max(K) - K[j]))))
R> data <- c(data, "beta")
```

MCMC simulation requires a number of iterations in order to converge. Before converging there is no point in retaining the posterior samples so these 'burn in' samples are discarded. For the Rasch model around 1,000 iterations, with 500 burn in iterations, are usually enough, but for more complex models further iterations may be required. Finally, as there may be some autocorrelation between successive iterations, a certain process of sampling from the posterior is also usually employed. The process involves thinning and using separate chains. Thinning refers to the process of sampling one in n iterations to minimise the influence of successive iterations. Different chains have different starting points and are useful in checking convergence. If different chains give different estimates then the model has not converged successfully. The following code produces 100 sets of posterior samples from three chains, giving 300 sets in total.

```
R> iter <- 2000
R> burnin <- 1000
R> thin <- 10
```

Then the models are fitted in the Bayesian framework using either **WinBUGS** or **JAGS**. Both programs provide analysis of Bayesian hierarchical models using MCMC simulation.

```
R> estimation <- "jags"
R> jagsout <- jags(data = data, inits = NULL,
+    parameters.to.save = monitor, model.file = jags.file,
+    n.iter = iter, n.thin = thin, n.burnin = burnin)
```

At this stage the convergence of the model can be checked. The following steps provide some plots that reveal potential lack of convergence using package **mcmcplots** (Curtis 2012):

```
R> library("mcmcplots")
R> mcmc.jags <- as.mcmc(jagsout)
R> mcmc.jags.list <- as.mcmc.list(mcmc.jags)
R> mcmcplot(mcmc.jags.list, random = 20)
```

For a full explanation of convergence, see Curtis (2010). If the model appears to have settled on reasonable consistent values, the posterior samples can be extracted.

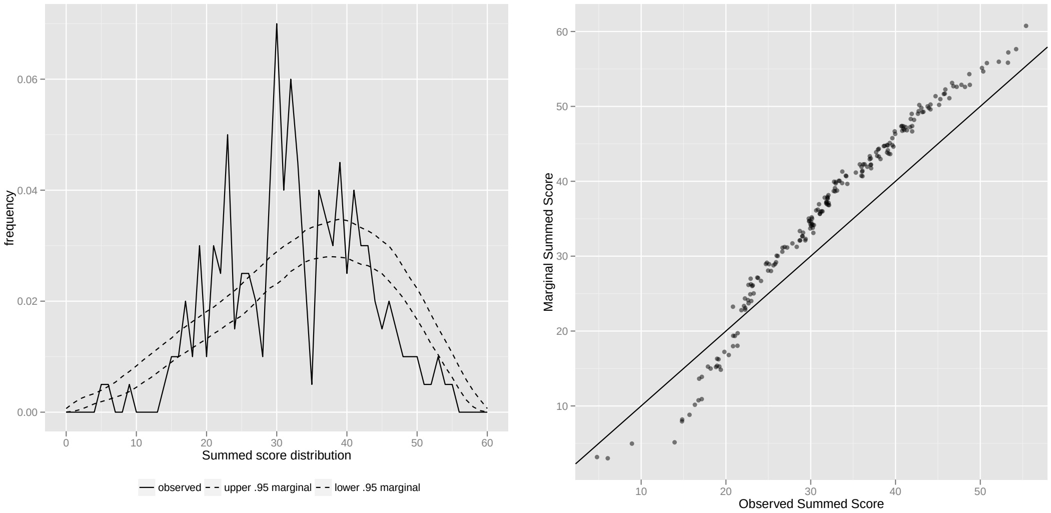
```
R> sims <- jagsout$BUGSoutput$sims.matrix
```

If package **rjags** (Plummer 2013b) has been used rather than package **R2jags** (Su and Yajima 2013) the posterior samples can also be extracted from one of the chains of an '`mcmc.list`' object as follows:

```
R> sims <- mcmc.list.object[[1]]
```

The first task of any IRT analysis is to check the model fit. At test level, the observed score distribution can be compared with the predicted score distribution. Firstly the predicted score distributions can be extracted from the posterior samples:

```
R> scores <- scores.gpcm.bug(biology, sims, mdl)
```



(a) Summed score distributions.

(b) Q-Q plot of score distributions.

Figure 1: Expected scores compared to observed scores.

Then the results can be visualised in different ways:

```
R> plot(scores)
R> plot(scores, type = "cond")
R> plot(scores, type = "qq", alpha = 0.5)
```

Figures 1a and 1b illustrate two of these different visualisations. Figure 1a shows the range of expected scores as confidence intervals around the observed score distribution. For each set of posterior samples produced by the model, the expected scores can be calculated. The confidence interval represents the range of scores in which 95 per cent of the expected scores fall. Clearly in the case where there is a small sample like this one the observed distribution is likely to be jagged, but the general shape of the distribution appears to be modelled relatively well.

Figure 1b plots quantiles of the expected and observed scores against each other. If the two distributions are identical, the Q-Q plot will be a straight line. Departures from the straight line are easier to spot than differences in the frequency distributions of the cumulative frequency distribution. In this case the Q-Q plot highlights deviation from the expected scores across a wide range of scores, a finding which should be borne in mind when interpreting the results. The classification accuracy estimates are derived from the model, not the observed data, so when the two diverge, results can be misleading. Obviously, testing fit should be a multi-faceted operation involving other tests than this one.

Once reasonable model fit has been ascertained the classification accuracy and consistency of the test can then be estimated from the posterior samples. Again, visualisations aid interpretation of the estimated values.

```
R> accs <- classify.bug(sims, scores, bnds, lbls)
R> plot(accs)
R> plot(accs, type = "kappa")
```

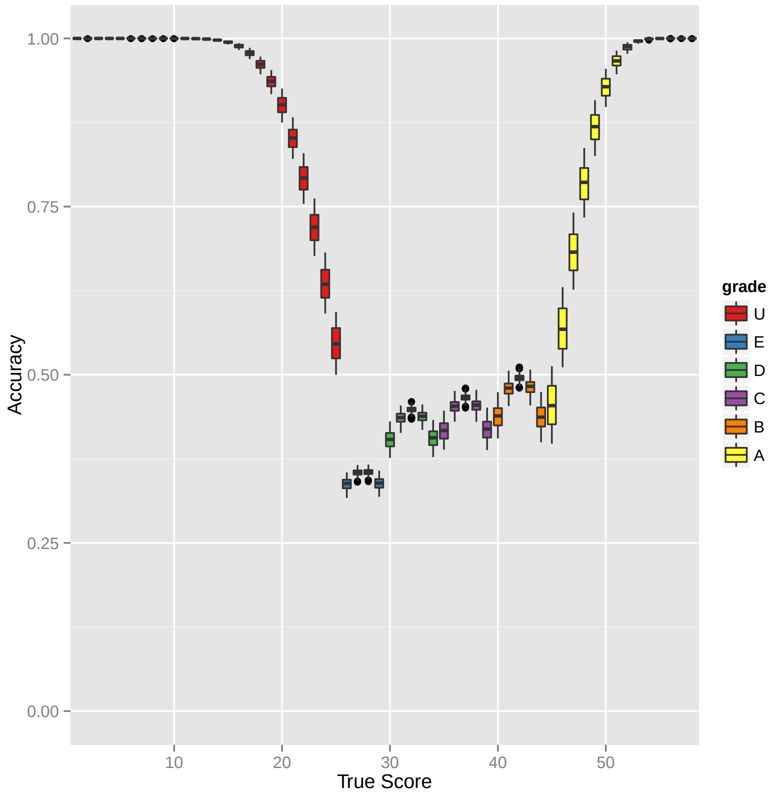


Figure 2: Classification accuracy.

Figures 2 and 3 illustrate these different visualisations. Figure 2 shows the classification accuracy of the test according to the expected or true score. This test is graded from A to U. The boxes are quartile plots of the range of classification accuracy values at specific score points. At any grade boundary the classification accuracy is likely to be near 0.5 as it is just as likely a candidate will be above the grade boundary as below it. In this case the plot shows accuracy values of below 0.5 near the grade boundaries. Very low values imply that candidates may be classified more than one grade away from their true grade. The plot clearly shows how the narrow grade boundary widths lead to poor classification accuracy on this test.

The classification consistency of the test is visualised in Figure 3. Each square displays the proportion of times candidates are classified into the same grade. The squares are shaded in proportion to the value in the square. The poor classification consistency of the test is revealed by the likelihood of achieving a grade C on successive administrations being only 0.05. The summary classification consistency figure is the sum of the diagonal, in other words, the proportion of times that candidates are classified into the same grade.

In addition to the visualisations, summary statistics are available. The summary statistics detail the classification accuracy and consistency values as well as the false positive and false positive rates across the grade range.

```
R> summary(accs)
```

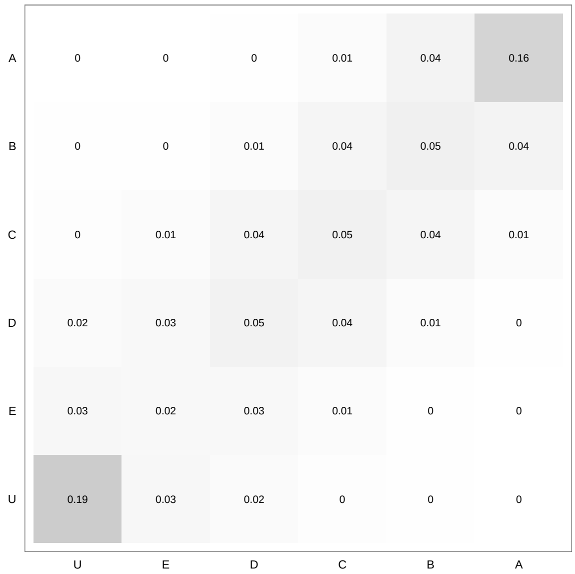


Figure 3: Classification consistency.

```
Marginal Classification Accuracy: 0.6(0.011)
Marginal Classification Consistency :0.529(0.012)
Kappa: 0.423(0.013)
Marginal False Negative Error Rate: 0.205(0.01)
Marginal False Positive Error Rate: 0.195(0.01)
Accuracy by grade:
   accuracy false.positive false.negative consistency
U 0.8680852      0.0000000      0.1319148   0.8003760
E 0.3464525      0.3179664      0.3355810   0.3149058
D 0.4261938      0.2763956      0.2974106   0.3100558
C 0.4414305      0.2772337      0.2813358   0.3315489
B 0.4666372      0.2640648      0.2692980   0.3690077
A 0.7721854      0.2278146      0.0000000   0.7615742
```

10. Comparison to Wainer *et al.* (2007)

Finally it may be useful to compare the estimate of classification accuracy derived from the Wingersky-Lord algorithm with the Wainer *et al.* (2007) method, which uses multiple draws from the posterior to estimate classification accuracy. Firstly the answer according to Wingersky-Lord.

```
R> data("physics", package = "classify")
R> n <- nrow(physics)
R> p <- ncol(physics)
R> bnds <- c(9, 11, 13, 15, 18, 21)
R> lbls <- c("U", "E", "D", "C", "B", "A", "A*")
```

```
R> mdl <- "rasch.bug"
R> Y <- physics
R> m.delta <- 0.0
R> s.delta <- 1.0
R> data <- list("Y", "n", "p", "m.delta", "s.delta")
R> monitor <- c("delta", "theta")
R> jags.file <- file.path(path.package("classify", quiet = FALSE),
+    "bugs", mdl)
R> system.time(jagsout <- jags(data = data, inits = NULL,
+    parameters.to.save = monitor, model.file = jags.file,
+    n.iter = 10000, n.thin = 10, n.burnin = 1000))
R> sims <- jagsout$BUGSoutput$sims.matrix
R> scores <- scores.gpcm.bug(Y, sims, mdl)
R> accs <- classify.bug(sims, scores, bnds, lbls)
R> print(accs@m.acc)
```

Now take draws from the posterior to estimate classification accuracy according to the Wainer *et al.* (2007) estimate.

```
R> iters <- nrow(sims)
R> delta.index <- grep("delta", colnames(sims))
R> theta.index <- grep("theta", colnames(sims))
R> scores.out <- matrix(ncol = iters, nrow = length(theta.index))
```

Calculate scores for each candidate:

```
R> cats <- rep(2, length(delta.index))
R> alpha <- rep(1, length(delta.index))
R> for(i in 1:iters) {
+    beta <- cbind(0, sims[i, delta.index])
+    theta <- sims[i, theta.index]
+    scores.out[, i] <- round(expected.rc(beta, theta, cats, alpha), 0)
+  }
```

Grade each candidate:

```
R> grades.out <- matrix(lbls[findInterval(scores.out, bnds) + 1],
+    nrow = nrow(scores.out))
```

Set the true grade to the candidate's observed grade:

```
R> obs.scores <- rowSums(physics)
R> obs.grades <- lbls[findInterval(obs.scores, bnds) + 1]
```

Take subsets from the simulations and calculate the proportion of times a candidate got their observed grade:

```
R> acc <- cumsum(colSums(grades.out == obs.grades)) /
+    cumsum(rep(200, ncol(grades.out)))
```

```
R> subset.sims <- seq(from = 100, to = iters, by = 100)
R> sampling.out <- cbind(subset.sims, acc[subset.sims])
R> print(sampling.out)
```

In this case the estimates are very similar: the Wingersky-Lord estimate of classification accuracy is 0.43, while the Wainer *et al.* (2007) estimate is 0.44.

11. Summary

It is hoped that the functions provided to estimate classification accuracy and consistency in the package **classify** will be useful in the study and enhancement of assessment design. Future developments to the package being considered include the extension to more models, the integration with non-Bayesian IRT estimation routines, the facility to calculate classification statistics across multiple test components and the inclusion of more tests of fit.

References

Béguin AA, Glas CAW (2001). "MCMC Estimation and Some Model-Fit Analysis of Multidimensional IRT Models." *Psychometrika*, **66**(4), 541–562.

Cohen J (1960). "A Coefficient of Agreement for Nominal Scales." *Educational and Psychological Measurement*, **20**(1), 37–46.

Curtis SM (2010). "BUGS Code for Item Response Theory." *Journal of Statistical Software, Code Snippets*, **36**(1), 1–34. URL http://www.jstatsoft.org/v36/c01/.

Curtis SM (2012). ***mcmcplots**: Create Plots from MCMC Output.* R package version 0.4.1, URL http://CRAN.R-project.org/package=mcmcplots.

Douglas KM, Mislevy RJ (2010). "Estimating Classification Accuracy for Complex Decision Rules Based on Multiple Scores." *Journal of Educational and Behavioral Statistics*, **35**(3), 280–306.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. URL http://www.jstatsoft.org/v40/i08/.

Guttman I (1967). "The Use of the Concept of a Future Observation in Goodness-of-Fit Problems." *Journal of the Royal Statistical Society B*, **29**(1), 83–100.

Hanson BA, Brennan RL (1990). "An Investigation of Classification Consistency Indexes Estimated under Alternative Strong True Score Models." *Journal of Educational Measurement*, **27**(4), 345–359.

Kolen MJ, Brennan RL (2004). *Test Equating, Scaling, and Linking.* Statistics in Social Science and Public Policy. Springer-Verlag, New York.

Lee WC (2008). "Classification Consistency and Accuracy for Complex Assessments Using Item Response Theory." *Technical Report 27*, Center for Advanced Studies in Measurement and Assessment, University of Iowa, Iowa City. URL http://www.education.uiowa.edu/casma/research_reports.htm.

Lee WC, Brennan RL, Wan L (2009). "Classification Consistency and Accuracy for Complex Assessments under the Compound Multinomial Model." *Applied Psychological Measurement*, **33**(5), 374–390.

Lee WC, Hanson BA, Brennan RL (2002). "Estimating Consistency and Accuracy Indices for Multiple Classifications." *Applied Psychological Measurement*, **26**(4), 412–432.

Livingston SA, Lewis C (1995). "Estimating the Consistency and Accuracy of Classifications Based on Test Scores." *Journal of Educational Measurement*, **32**(2), 179–197.

Lord FM, Novick MR (1968). *Statistical Theories of Mental Test Scores*. Addison-Wesley, Reading.

Lord FM, Wingersky MS (1984). "Comparison of IRT True-Score and Equipercentile Observed-Score "Equatings"." *Applied Psychological Measurement*, **8**(4), 452–461.

Lunn D, Thomas A, Best NG, Spiegelhalter DJ (2000). "**WinBUGS** – A Bayesian Modelling Framework: Concepts, Structure, and Extensibility." *Statistics and Computing*, **10**, 325–337.

Masters GN (1982). "A Rasch Model for Partial Credit Scoring." *Psychometrika*, **47**(2), 149–174.

Muraki E (1992). "A Generalized Partial Credit Model: Application of an EM Algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Plummer M (2003). "**JAGS**: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*. Vienna, Austria. ISSN 1609-395X. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/.

Plummer M (2013a). ***JAGS** Version 3.4.0 User Manual*. URL http://mcmc-jags.sourceforge.net/.

Plummer M (2013b). ***rjags**: Bayesian Graphical Models Using MCMC*. R package version 3-11, URL http://CRAN.R-project.org/package=rjags.

Rasch G (1960). *Probabilistic Models for Some Intelligence and Attainment Tests*. Danmarks Paedogogiske Institut, Copenhagen. Reprint, with Foreword and Afterword by BD Wright, Chicago: University of Chicago Press, 1980.

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

Rubin DB (1984). "Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician." *The Annals of Statistics*, **12**(4), 1151–1172.

Sinharay S (2005). "Assessing Fit of Unidimensional Item Response Theory Models Using a Bayesian Approach." *Journal of Educational Measurement*, **42**(4), 375–394.

Spiegelhalter DJ, Thomas A, Best NG, Lunn D (2003). "**WinBUGS** Version 1.4 User Manual." *Technical report*, Cambridge. URL http://www.mrc-bsu.cam.ac.uk/bugs/.

Su YS, Yajima M (2013). ***R2jags**: A Package for Running **JAGS** from R*. R package version 0.03-11, URL http://CRAN.R-project.org/package=R2jags.

Subkoviak MJ (1976). "Estimating Reliability from a Single Administration of a Criterion-Referenced Test." *Journal of Educational Measurement*, **13**(4), 265–276.

Swaminathan H, Hambleton RK, Algina J (1974). "Reliability of Criterion-Referenced Tests: A Decision-Theoretic Formulation." *Journal of Educational Measurement*, **11**(4), 263–267.

Swaminathan H, Hambleton RK, Rogers JH (2007). "Assessing the Fit of Item Response Models." In CR Rao, S Sinharay (eds.), *Handbook of Statistics*, volume 26. Elsevier B.V., Amsterdam.

Thomas A, O'Hara B, Ligges U, Sturtz S (2006). "Making BUGS Open." *R News*, **6**(1), 12–17. URL http://CRAN.R-project.org/doc/Rnews/.

Wainer H, Bradlow ET, Wang X (2007). *Testlet Response Theory and Its Applications*. Cambridge University Press, Cambridge.

Wheadon C, Stockford I (2010). "Classification Accuracy and Consistency in GCSE and A Level Examinations Offered by the Assessment and Qualifications Alliance (AQA) November 2008 to June 2009." Report for Ofqual by AQA Centre for Education Research and Policy.

Wheadon C, Stockford I (2014). ***classify**: Classification Accuracy and Consistency under IRT models*. R package version 1.2, URL http://CRAN.R-project.org/package=classify.

Affiliation:

Chris Wheadon
No More Marking Ltd.
18, Addision Road
Guildford, GU1 3QG, United Kingdom
E-mail: chris.wheadon@gmail.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 56, Issue 10 *Submitted:* 2012-08-23
January 2014 *Accepted:* 2013-08-22