CutScore

A Shiny App for the Cut-Score Operating Function

Christopher Runyon, The University of Texas at Austin
Irina Grabovsky, National Board of Medical Examiners

The Cut-Score Operating Function (Grabovsky & Wainer, 2017a; 2017b) is a graphical standard setting tool that provides the classification error rates associated with a given cut score c. The event that an examinee is misclassified by a Pass/Fail test can be described as a combination of two possible scenarios: a false positive misclassification (an examinee that is not truly proficient passes the exam, occurring with probability *FP(c)*), and false-negative misclassification (a proficient examinee does not pass the exam, occurring with probability *FN(c)*). The overall classification error rate *CE(c)* can be calculated by adding probabilities of these two non-overlapping events:

$$CE(c) = FP(c) + FN(c). \tag{1}$$

Alternatively, if either misclassification error is more important, positive weights adding up to 1 can be applied to probabilities above. For example, if false positive outcomes in certification is highly undesirable (e.g., certification of a non-competent pilot) then the user can impose a higher weight (w) on false negative outcome, resulting in a *weighted classification error rate, WCE(c)*:

$$WCE(c) = wFP(c) + (1-w)FN(c), \quad 0 \leq w \leq 1. \tag{2}$$

We refine the definition of the cut-score function to account for the mastery level of examinees on the particular exam. If we know the true cut score τ^* which divides truly competent examinees from those that are not competent, the *conditional false positive* can be defined as passing the test given the examinee is not competent (her true τ ability is lower than τ^*), and *conditional false negative* as failing the test given that examinee is competent (her true ability τ is equal to or larger than τ^*). The *conditional classification error rate CCE(c)* is defined as an average of the two conditional probabilities:

$$CCE(c) = \frac{1}{2}(P(passing|\tau < \tau^*) + P(failing|\tau \geq \tau^*)) \tag{3}$$

The advantage of the conditional classification error is that it is considerably more sensitive to the inherent relationship between the true abilities in the examinee population and the true cut score. For example, if the historical fail rate for the exam is small (e.g., 5%-10%) then the false negative rate will remain small, no matter how high the cut score. In contrast, this is similar to when a disease is so rare that those who have it are outnumbered by false positives.

The Method

Applying Bayes theorem to both conditional probabilities in (3), we get the expressions for conditional probabilities:

$$P(passing \mid \tau < \tau^*) = \frac{1}{P(\tau < \tau^*)} P(passing\, and\, \tau < \tau^*) \tag{4}$$

$$P(failing \mid \tau \geq \tau^*) = \frac{1}{P(\tau \geq \tau^*)} P(failing\, and\, \tau \geq \tau^*) \tag{5}$$

Thus, CCE is a weighted average of the two probabilities above defined by the value of the true cut score τ^*:

$$CCE(c) = \frac{1}{2}(\frac{P(passing\, and\, \tau < \tau^*)}{P(\tau < \tau^*)} + \frac{P(failing\, and\, \tau \geq \tau^*))}{P(\tau \geq \tau^*)}) \tag{6}$$

Of course, the true cut score is unknown. However, one can make reasonable assumptions about it based on historical cut score data or results of prior standard setting studies that use such methods as the Angoff and Hofstee method. The point of minimum found in (6) is always unique and can be calculated with the high degree of precision, along with associated error rates. The expressions in (6) rely on the normality of abilities distribution in the population, the Rasch model, and the approximation of the CDF given by the central limit theorem.

The CutScore App

The "CutScore" app was developed using the Shiny package (Chang, Cheng, Allaire, Xie, & McPherson, 2017) for the R statistical computing package (R Core Team, 2017) to give researchers and practitioners an accessible way to implement the weighted and/or conditional cut-score operating function. With CutScore, one inputs information relevant to the calculation of the cut-score function and the resulting minima of the function(s) is located. Specifically, the user enters:

- Information about examination item difficulties, either by .csv or user-specified values.
- Examinee ability information, either by .csv or user-specified values.
- The examination reliability.
- The desired cut-score function (either weighted, conditional, or both).
- The proposed cut score ($\tau*$) in the metric of the
- The weight that should be used for the ratio of $\frac{FP(c)}{FN(c)}$ if the weighted cut-score function is chosen.

The output is the minima of the cut-score function(s), with an option to view these minima on a graph with the classification error function (CE(c)) at the top, the FN(c) loss function in the middle, and the FP(c) loss function at the bottom, with a red line indentifying the minima of the classification error function.

Contact Information

One can download the current version of the Cutscore Shiny application from **https://github.com/runyoncr/NCME_CutScore**. Please note that the package is still under development. Technical errors and inefficient coding are the responsibility of the first author.

Chris may be reached at runyon.christopher@gmail.com. Please email him with suggestions on how to improve the Cutscore app, or with any general comments about the project.

Irina may be reached at igrabovsky@nbme.org. Please email her with any questions concerning the derivation and technical details of the cut-score operating function.

References

Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (20175). Shiny: web application framework for R. R package version 1.0.5. https://CRAN.R-project.org/package=shiny

Grabovsky, I., & Wainer, H. (2017ab). The cut-score operating function: A new tool to aid in standard setting. Journal of Educational and Behavioral Statistics, 42(3), 251-263.

Grabovsky, I., & Wainer, H. (2017). A Guide for setting the cut-scores to minimize weighted classification errors in test batteries. Journal of Educational and Behavioral Statistics, 42(3), 264-281.

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.