

Using `ruleR` as a framework for rule-based item generation

Maria Rafalak
m.rafalak@practest.com.pl

Philipp Doebler
philipp.doebler@googlemail.com

August 21, 2012

1 Introduction

The success of a psychological test is largely determined by the quality of its items. In classic linear testing, the number of items needed to measure a single latent trait with acceptable reliability is often in the range of 20 to 60 items. While it is certainly possible to construct this number of items with the help of heuristics it is often possible to identify rules governing the item construction process. For example for the classic Advanced Progressive Matrices Test ([Rav62]) five (broad) rules used in its construction have been identified ([CJS90]). These rules and their combinations are used on the rows and/or columns of a matrix resulting in many different stimuli. Also number sequences used in intelligence tests are often derived from basic rules like addition and subtraction (of constants or two previous numbers), multiplication and the digitsum (sum of the digits of a number).

Recently *automated item generation* (AIG; [IK02]) has been explored in various applied contexts ([Are05, ASGH06, HBZ09, HBKK10, Zeu10]). The main idea of AIG is to identify the underlying template or rule(s) that constitute an item and to generate new items (potentially infinitely many) using the template or rule(s). Two main approaches can be identified: *item cloning* (IC) and *rule-based item generation* (RIG). The first approach uses an existing item (a parent), typically with known psychometric qualities, and produces a clone (a child) of that item by changing its *surface features* (or incidentals). For example in a statistics exam for university students, the cover story of the example is changed, but the student is nevertheless to make the same calculations (e.g. [HBZ09]). RIG on the other hand focuses on the rules (sometimes called radicals) that govern the item construction process. Once the rules and their relations (and their relation to surface features of items) are known, a new item can be generated from a (combination of) rule(s). Often it is possible to predict the difficulty of an item by using a linear logistic test model or a relative (LLTM; [Fis73]; [GGvdL11]).

There are several situations in which automated item generation is favourable:

1. Linear tests, especially in high stakes situations like college admission, are often used only in one year since the test security can not be guaranteed once the test has been exposed to a large population ([AS12]). Here automated item generation leads to tests for which the answers can not be learnt by heart.
2. Computer adaptive testing (CAT; [vdLG10]; [VDLG00]; [WDF⁺00]) relies on large pools to cover a wide range of potential person abilities. It is often expensive to produce items, so automating the process is certainly desirable here. Also if the psychometric properties of parent items or rules are known, the CAT algorithm can generate items on the fly.

The identification of rules is not the same as an implementation of a rule-based item generation algorithm, but it is a necessary step. Besides providing a basis for RIG, an analysis of the cognitive task at hand is a result of this identification. Another side product is that a suitable psychometric model can sometimes be found after such an analysis.

In the following we identify some rules for number sequence items and matrix type items and explain details of their implementation in the `ruleR` package. We aim to provide a framework to

generate number sequence items and matrix type items with the ability to extend the system. The package uses S4 classes to represent rules and is written with the understanding that the user will eventually want to extend the existing possibilities.

While R itself is not a frontend for computer based testing, it's applicability has been successfully demonstrated, for example in the form of the concerto testing platform ([The12]). Several R packages are worth mentioning in this context: `catR` ([MM11]), which provides functionality for computer adaptive testing, `ltm` ([Riz06]), which can be used to perform a range of psychometric analyses and `RMySQL` ([JD12]) which handles the interaction of R and MySQL databases.

2 Number sequences

Before we showcase a sample R session in which `ruleR` is used to generate number sequences, we explain some of the ideas.

2.1 Number sequence items and their cognitive analysis

2.2 Using ruleR to generate number sequence items

3 Matrix items

4 Further development of ruleR

References

- [Are05] M. Arendasy. Automatic generation of Rasch-calibrated items: Figural matrices test GEOM and Endless-Loops Test EC. *International Journal of Testing*, 5:197–224, 2005.
- [AS12] M.E. Arendasy and M. Sommer. Using automatic item generation to meet the increasing item demands of high-stakes educational and occupational assessment. *Learning and Individual Differences*, 22:112–117, 2012.
- [ASGH06] M. Arendasy, M. Sommer, G. Gittler, and A. Hergovich. Automatic generation of quantitative reasoning items: A pilot study. *Journal of Individual Differences*, 27:2–14, 2006.
- [CJS90] P.A. Carpenter, M.A. Just, and P. Shell. What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological Review*, 97(3):404–431, 1990.
- [Fis73] G.H. Fischer. The linear logistic test model as an instrument in educational research. *Acta psychologica*, 37:359–374, 1973.
- [GGvdL11] H. Geerlings, C.A.W. Glas, and W.J. van der Linden. Modeling rule-based item generation. *Psychometrika*, pages 1–23, 2011.
- [HBKK10] H. Holling, H. Blank, K. Kuchenbacker, and J.T. Kuhn. Rule-based item design of statistical word problems: A review and first implementation. *Psychology Science Quarterly*, 50:363–378, 2010.
- [HBZ09] H. Holling, J.P. Bertling, and N. Zeuch. Automatic item generation of probability word problems. *Studies In Educational Evaluation*, 35:71–76, 2009.
- [IK02] S.H. Irvine and P.C. Kyllonen. *Item generation for test development*. Lawrence Erlbaum, 2002.

- [JD12] David A. James and Saikat DebRoy. *RMySQL: R interface to the MySQL database*, 2012. R package version 0.9-3.
- [MM11] David Magis and David Magis. *catR: an R package to generate IRT adaptive tests*, 2011. R package version 2.2.
- [Rav62] J.C. Raven. *Advanced Progressive Matrices*. London: H.K. Lewis, 1962.
- [Riz06] Dimitris Rizopoulos. ltm: An r package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, 17(5):1–25, 2006.
- [The12] The Psychometrics Centre. Concerto: Open-source Online R-based Adaptive Testing Platform. Software package, 2012. Available at: <http://code.google.com/p/concerto-platform>.
- [VDLG00] W.J. Van Der Linden and C.A.W. Glas. *Computerized adaptive testing: Theory and practice*. Springer, 2000.
- [vdLG10] W. J. van der Linden and C. A. W. Glas. *Elements of Adaptive Testing*. Springer, New York, 2010.
- [WDF⁺00] H. Wainer, N.J. Dorans, R. Flaugher, B.F. Bert Green, and R.J. Mislevy. *Computerized Adaptive Testing: A Primer*. Routledge, second edition, 2000.
- [Zeu10] N. Zeuch. *Rule-based item construction: Analysis with and comparison of linear logistic test models and cognitive diagnostic models with two item types*. PhD thesis, WWU Münster, 2010. Retrieved from: miami.uni-muenster.de.