LSAC RESEARCH REPORT SERIES

- **Unraveling the Relationship Between Testlet Features and Item Parameters: An Empirical Example**

Muirne C. S. Paap
Bernard P. Veldkamp
University of Twente, Enschede, The Netherlands

- **Law School Admission Council**
Research Report 12-06
October 2012

A Publication of the Law School Admission Council

The Law School Admission Council (LSAC) is a nonprofit corporation that provides unique, state-of-the-art admission products and services to ease the admission process for law schools and their applicants worldwide. More than 200 law schools in the United States, Canada, and Australia are members of the Council and benefit from LSAC's services.

LSAC fees, policies, and procedures relating to, but not limited to, test registration, test administration, test score reporting, misconduct and irregularities, Credential Assembly Service (CAS), and other matters may change without notice at any time. Up-to-date LSAC policies and procedures are available at LSAC.org.

Table of Contents

Executive Summary

A mathematical model called item response theory (IRT) is often applied to high-stakes tests to determine the characteristics of test questions (i.e., items), such as difficulty, discrimination, and susceptibility to guessing. Note that in this context, the term “discrimination” refers to how well an item distinguishes between higher- and lower-ability test takers. Often, these tests contain subsets of items grouped around a common stimulus (testlet). This grouping often leads to items within one group (testlet) being more strongly correlated among themselves than among items from other groups, which can result in moderate to strong testlet effects.

Recently, it was shown that stimulus features could be used to predict the size of the testlet effect. Furthermore, a strong relationship was found between average item difficulty and the magnitude of the testlet effect. This study explores the relationship between stimulus features and the IRT parameters of difficulty, discrimination, and guessing. It was found that stimuli associated with easy items consisted of many different (but commonly used) words as well as an intermediate proportion of negative words. Relatively short stimuli containing many different words were found to have a high information density, and thus are considered to be very useful for distinguishing between test takers of different ability levels. No useful predictions could be made with regard to susceptibility to guessing, since that parameter did not vary much from one item to the next. It was concluded that stimulus features can be used to manipulate passage texts in such a way that they will have “favorable” properties in terms of testlet effect and average item parameters.

Introduction

It has become common practice to use item response theory (IRT) models to analyze data obtained from high-stakes tests. Typically, testing agencies use standard IRT models, such as the Rasch model or the two- or three- parameter logistic model (2PLM, 3PLM). These models are popular because they are well known, are relatively straightforward to estimate if there is enough data, and can be implemented in standard software packages such as Bilog-MG (Zimowski, Muraki, Mislevy, & Bock, 1996).

These widely used IRT models are based on several assumptions, including local independence (LI) of item responses. This assumption holds if responses to the items in the test are dependent solely on the ability level (θ) of the candidate. However, many high-stakes tests include subsets of items that share a common stimulus (e.g., text passage or video fragment), commonly referred to as a *testlet*. This may lead to a breach of LI among the items belonging to the same testlet. If the local dependence (LD) is large within testlets, using standard IRT models may have serious consequences for model estimation; as DeMars (2012) eloquently summarizes, it could lead to errors in scaling and equating, biased item parameters, and item misfit (Glas, Wainer, & Bradlow, 2000; Li, Bolt, & Fu, 2005; Marais & Andrich, 2008). Therefore, many test developers aim to design stimuli that will not produce large testlet effects.

Several ways to deal with testlet effects have been proposed over the years. One of the most popular methods is the testlet model proposed by Bradlow, Wainer, and Wang (1999). They introduced a new parameter that accounts for the random effect of a person on items that belong to the same testlet, in order to adjust for the nested

structure. This parameter, $\gamma_{nt} \sim N(0, \sigma_{1t}^2)$, is referred to as the *testlet parameter* for person n on testlet t. It represents a random effect that exerts its influence through its variance: The larger the variance σ_{1t}^2, the larger the amount of local dependence (LD) between the items *j* within the testlet t (Wainer & Wang, 2000). Therefore, the variance or standard deviation of γ_{nt} can be seen as a measure of the testlet effect. Before this model was introduced, item scores within a testlet were commonly summed up, producing one "polytomous" score per testlet, which could then be analyzed with straightforward IRT models (e.g., Thissen, Steinberg, & Mooney, 1989). However, in calculating sum scores, the exchangeability of items is assumed, which may not be realistic in practice. Moreover, when a polytomous IRT model is used to model the summarized score per testlet, a guessing parameter at the item level cannot be taken into account. For these reasons, many researchers prefer the testlet model proposed by Bradlow, Wainer, and Wang.

A question left unaddressed until relatively recently is whether the testlet effect is related to features of stimuli. Although the testlet model performs well in recovering model parameters (Tao & Su, 2012, April), using a more parsimonious IRT model may still be preferred if testlet effects are small. Therefore, it would be helpful for test designers if they knew which testlet features contribute to larger testlet effects. We recently proposed a framework for investigating this issue (Paap, Glas, He, & Veldkamp, 2012). A new IRT model, the explanatory testlet response model (ETRM), was developed to decompose the testlet effect into an unexplained part and a part that is predicted by testlet features. We showed that tree-based regression (TBR) is highly useful for selecting relevant stimulus features that can be plugged into the ETRM to assess their impact on the testlet parameter. We were able to use these features to divide the testlets into groups that had a high testlet effect versus a medium testlet effect, based on the scores on the stimulus features. A next step might be to investigate whether there is a causal relationship between features and testlet effects by manipulating the passage texts. If a relationship is found, test designers could take these features into account when they are writing new testlets. The resulting testlets would be expected to have a low testlet effect, and from that point onward, regular IRT models could be used to calibrate the data.

In a subsequent study (Paap & Veldkamp, 2012), we showed that certain stimulus features had predictive value for the size of the testlet effect, even when the effect of the estimated item parameters (discrimination and difficulty parameter) was accounted for. Moreover, our results indicated that testlets that contained less difficult items showed large testlet effects (Figure 1). A possible explanation could be that item difficulty may have a moderating effect on the relationship between testlet features and the testlet effect.

In our previous work, the main focus was on finding variables that could be linked to the testlet effect. Besides testlet parameters, testlet response theory models also distinguish among item parameters such as difficulty, discrimination, and guessing. In this study, we focus on other important IRT parameters: the item parameters. We will explore the relationship between testlet features (as independent variables) and average item parameters per testlet (dependent variables) by using TBR—both univariate and multivariate. We chose to use TBR instead of the better-known linear regression, because TBR is a more flexible method for dealing with highly complex interactions, and it poses fewer statistical assumptions (e.g., regarding the distribution of the variables).

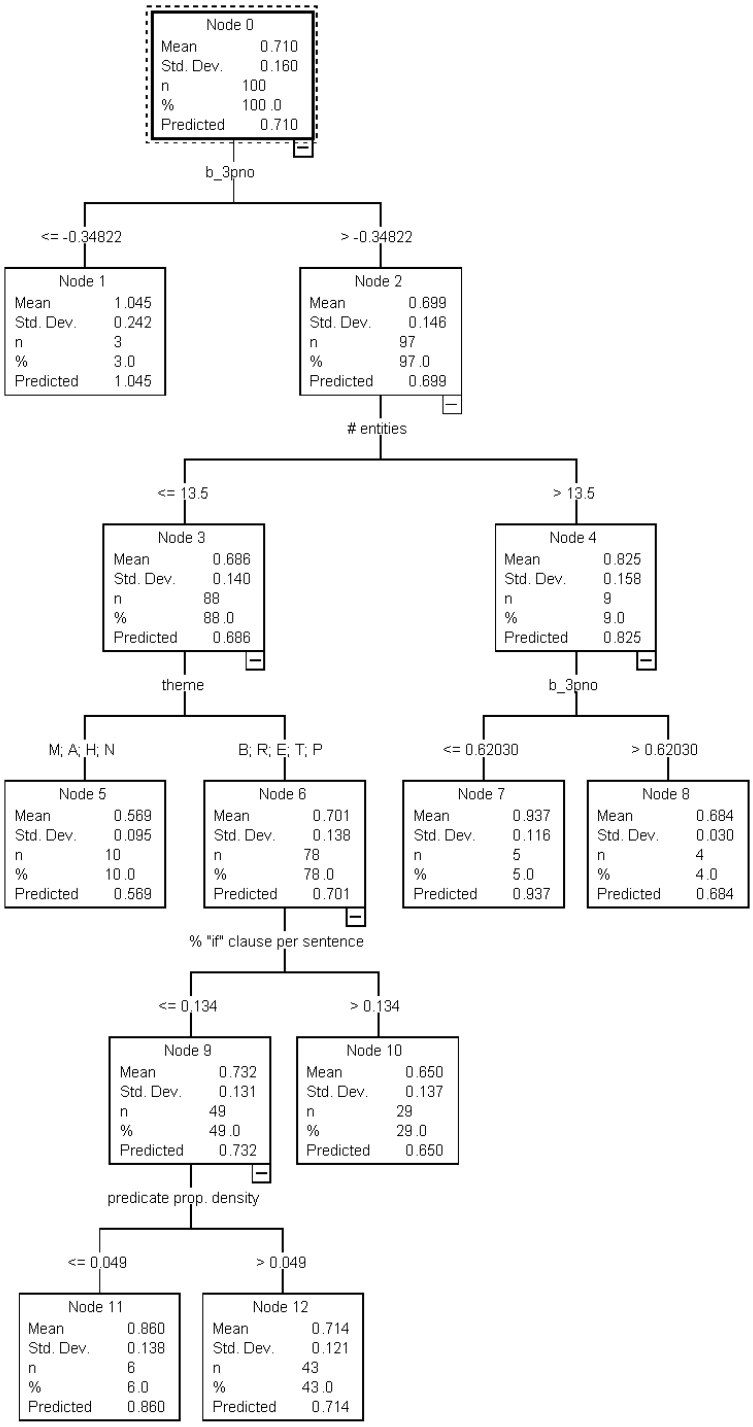

FIGURE 1. *Regression tree with the 3PNO-based testlet effect as a dependent variable and stimulus features and average item difficulty as independent variables.* Adapted from Paap, M. C. S., & Veldkamp, B. P. (2012). Minimizing the testlet effect: Identifying critical testlet features by means of tree-based regression. In T. J. H. M. Eggen & B. P. Veldkamp (Eds.), *Psychometrics in Practice at RCEC*, p. 69. Enschede, the Netherlands: RCEC, Cito/University of Twente. Copyright 2012 by RCEC, Cito/University of Twente, Enschede, Netherlands. Adapted with permission.

Method

Testlet Properties

Items in the Analytical Reasoning (AR) section of the Law School Admission Test (LSAT) are designed to test the ability of the examinee to reason within a given set of circumstances described in the stimulus (testlet-specific text passage). The stimulus contains information about a number of elements (people, places, objects, tasks, and so on) along with a set of conditions imposing a structure on the elements (e.g., ordering them, assigning elements of one set to elements of another set, and so on). AR stimuli always permit more than one acceptable outcome satisfying all of the requirements in the stimulus text. For detailed information about the AR section of the LSAT, the reader is referred to *The Official LSAT Handbook* (Law School Admission Council, Inc., 2010).

The testlet feature variables used in this study fall into three categories: (1) variables describing the logical structure of the stimuli (*structural variables*), (2) variables describing the themes contained in the stimuli (*theme variables*), and (3) surface *linguistic variables*. Two raters (the authors of this paper) independently coded the variables in Categories 1 and 2. In the case of incongruent scorings, a consensus was reached through a thorough discussion. Surface linguistic features were identified using the specialized text-mining software Python (Python Software Foundation, 2009). An overview of the structural, theme, and surface linguistic variables can be found in Table 1. We used the same dataset that we used in our previous studies (Paap et al., 2012; Paap & Veldkamp, 2012). For a detailed description of the extraction of testlet features, we refer to Paap et al. (2012).

TABLE 1

Overview of independent variables used in the regression analyses, divided into three categories

Independent Variables	Description/Remarks
(1) Structural Variables	
Number of Features	Takes values 2, 3, or 4; 2 if it only contains information on one type of entity variable ("x") and one position variable ("y"), 3 if it contains information about two entity variables and one position variable or one entity variable and two position variables, and 4 if it contains information about two entity variables and two position variables.
Stimulus Type	Only scored if the number of features is 3 or 4; it specifies whether the stimulus is of type 1 (two or more x's were assigned to one y), 2 (one x was assigned to two or more y's) or 3 (x was assigned to y, which was assigned to a higher-order position variable "z").
Number of Entities	The number of entities, summed over all entity variables present in the stimulus; entities are defined as the units in the stimulus that had to be assigned to positions.
Number of Positions	The number of positions, summed over all position variables present in the stimulus.
Cardinality of Entities	Takes values "1" or "multiple." The cardinality of entities is "1" if they can only be assigned to a position once and multiple if they can be assigned more than once.
Cardinality of Positions	Takes values "1" or "multiple." The cardinality of positions is "1" if only one entity can be assigned to a position and "multiple" if more than one entity can be assigned to a position.
Number of Entities Smaller/Larger Than Number of Positions	
Ordered Positions (Yes/No/Partially)	
(2) Theme Variable	
Theme/topic	Variable used to describe the main theme of the stimulus. The following categories are used: B (Business), E (Education), R (Recreation), M (Media), A (Animals), T (Transport/Vehicle), N (Nature), P (Intrapersonal Relationships/Family), and H (Health).
(3) Surface Linguistic Variables	
Word Token*	Length of the stimulus text, total number of words excluding punctuation.
Word Type*	Vocabulary size, total number of words excluding word repetition and punctuation.
Word Diversity	Word Type divided by Word Token.
Average Characters*	Average number of letters used per word in the stimulus text.
Percentage of Negative Words	Percentage of "negative" words such as "no," "not," "neither," and so on. May increase the difficulty of a text.
Brown News Popularity	The popularity of verbs, nouns, adjectives, adverbs, and names in the Brown news corpus. Note that the Brown corpus was the first million-word electronic corpus of English, created in 1961 at Brown University (Francis & Kučera, 1964). The 500 text sources contained in the corpus have been categorized by genre, such as *news, editorial*, etc. To calculate the Brown News Popularity variable, the Porter Stemmer algorithm was used to standardize each word.
Percentage of Content Words*	The number of verbs, nouns, adjectives, adverbs, and names divided by Word Token.
Modifier Propositional Density*	Number of adjectives divided by Word Token.
Predicate Propositional Density*	Number of verbs divided by Word Token.
Number of Sentences*	Number of sentences used in stimulus text.
Average Sentence Length*	Word Token divided by Number of Sentences.
Percentage of "If" Clauses	In the AR stimuli, "if" clauses are regularly used and could be expected to increase the difficulty of a text (both with respect to logical reasoning and sentence complexity).

*Based on work by Drum, Calfee, & Cook (1981), Embretson & Wetzel (1987), and Gorin & Embretson (2006).

Statistical Analysis

Parameter Estimation

The responses of 49,256 students to 594 items nested within a total of 100 testlets (stimuli) administered in the AR section of the Law School Admission Test (LSAT) were used for analysis. A fully Bayesian approach using a Markov chain Monte Carlo (MCMC) computation method (Glas, 2012a) was applied. The three-parameter normal ogive (3PNO) item response model was used to model the data. For the testlet model version, the probability of a correct response is given by

$$P(Y_{ni} = 1) = c_i + (1 - c_i)\Phi(\tau_{ni}), \quad (1)$$

where Φ is the cumulative normal distribution, that is

$$\Phi(s) = (2\pi)^{-1/2} \int_{-\infty}^{s} \exp\left(-\frac{t^2}{2}\right) dt, \quad (2)$$

and

$$\tau_{ni} = a_i\theta_n - b_i + \gamma_{nt(i)}. \quad (3)$$

The discrimination parameter of item i is denoted by a_i, its difficulty parameter by b_i, c_i denotes the guessing parameter of item i, and $\gamma_{nt(i)}$ is the random testlet parameter for person n on testlet t containing item i. The model was identified by restricting the mean and the variance of the distribution of the estimated ability θ_n to zero and one, respectively. The software package MIRT (Glas, 2010) was used to calibrate the model.

Model Building: Tree-Based Regression

In Step 1, univariate models were estimated in SPSS (SPSS, 2007), using the average a, b, and c parameters per testlet as dependent variables, respectively. Testlet features served as independent variables. In Step 2, the testlet features that showed a significant effect in Step 1 were used as independent variables, and a multivariate tree was estimated using the R (R, 2004) package mvpart (Therneau & Atkinson, 2009).

Step 1: Univariate Trees

The C&RT module in SPSS closely follows the algorithm called classification and regression trees (CART) described by Breiman, Friedman, Olshen, and Stone (1984). A short description follows. Using TBR, clusters of testlets with similar values on the predictor variable were formed by successively splitting the testlets into increasingly homogeneous subsets (“nodes”). The testlet feature that maximized the homogeneity with respect to the dependent variable in the two nodes was identified and selected at each stage of the algorithm. The split that maximizes the difference in deviance between the parent node (original set of items) and the sum of the child nodes (subsets of items created by the independent variable) results in a low value

for the impurity measure. The impurity measure $R(t)$ is measured by the prediction error in node t:

$$R(t) = \frac{1}{N}\sum_{i\in t}\left(y_i - \bar{y}(t)\right)^2, \quad (4)$$

where y_i are the observed values of the dependent variable and $\bar{y}(t)$ is the mean value of the dependent variable in the node t. The impurity of the tree is given by the sum of the impurity measures of all terminal nodes (nodes that are not split further). In SPSS, this value is reported as the "risk estimate." To indicate the fit of a tree, the relative error (RE) can be used, which is calculated by dividing $R(t)$ by the total variance of the dependent variable. The proportion variance explained by the model is then given by:

$$1 - \frac{R(t)}{Var(Y)} \quad (5)$$

Initially, a large tree that overfits the data was grown, to avoid missing important structures. Pruning the large tree yielded a nested sequence of subtrees; subsequently, a subtree of optimal size was selected from the sequence. Pruning entails collapsing pairs of child nodes with common parents by removing a split at the bottom of the tree.

Model building in Step 1 was done in a similar fashion to that in our previous work (Paap et al., 2012; Paap & Veldkamp, 2012). First, models were evaluated separately for each category of testlet features. In other words, a separate model was estimated containing all structure testlet features first, then a separate model was estimated containing the theme variables, and finally a separate model was estimated containing the linguistic testlet features as independent variables. The independent variables that were selected by the algorithm in each of these three models were then entered as independent variables in one joint model; only the ones that had a significant effect in this model were retained. In the case of competing models, the final model would be the one with the greatest number of splits resulting in a large difference in the mean of the dependent variable for the resulting nodes.

Following Matteucci, Mignani, and Veldkamp (2012), the rule of one standard error (*SE*) was adopted to choose the best tree size. Let $R(t)^*$ denote the $\mathrm{R(t)}$ value of the large, overfitted tree. Then, according to the 1−SE rule, the large tree was automatically trimmed to the smallest subtree that had an $\mathrm{R(t)}$ value that did not exceed $R(t)^* + 1SE$. A minimum change of improvement smaller than 0.000001 was used as a stopping rule. The change of improvement equals the decrease in impurity required to split a node (SPSS Inc., 2007). The maximum tree depth was set to 10 levels, and the minimum number of cases was set to 5 for parent nodes and 3 for child nodes.

Step 2: A Multivariate Tree

In the second step, we fitted a multivariate tree in R, using the package mvpart. Using multivariate analyses when there are several dependent variables has an added value over just using univariate analyses. Most importantly, in a multivariate analysis, the relationship between the dependent variables is taken into account. This is especially favorable if the correlation between the dependent variables is moderate: If it is very low, not much is gained; if it is very high, the two variables may

in fact measure the same construct. Also, a multivariate analysis can distinguish among groups of testlets based on a combination of scores on the dependent variables. Two general approaches are available in this package: building the tree in an "exploratory" way based on a predefined algorithm, and building the tree in a "user driven" way, where the user can interactively choose the preferred tree. The selection of the final model in the exploratory algorithm in this package differs somewhat from the C&RT procedure available in SPSS (i.e., a different impurity measure is used). To ensure comparability, we used the user-driven option and interactively chose a tree similar in size to the unidimensional trees. The independent variables entered were the ones selected in the univariate trees.

Results

Tree-Based Regression Step 1: Univariate Trees

Average a-Parameter as Dependent Variable

Visual inspection of the distribution of the *average a-parameter per testlet* indicated that it was normally distributed (mean = 0.74, *SD* = 0.21). The final tree had 10 (Figure 2) nodes and contained the following independent variables: Word Diversity, Average Characters, Ordered Positions, and Number of Entities. Word Diversity and Average Characters were normally distributed with a mean of 0.51 (*SD* = 0.06) and 4.47 (*SD* = 0.51), respectively. Sixty percent of the stimuli contained Ordered Positions, 35% did not, and the remaining 5% contained only Partially Ordered Positions. The Number of Entities was slightly positively skewed with a mean of 7.33 (*SD* = 3.28).

The variable Word Diversity was common to all stimuli and was used for the first split: stimuli containing 42.9% or lower Word Diversity were placed in the left branch (smaller average a-parameter) while stimuli with a Word Diversity score of at least 42.9% were placed in the right branch (larger average a-parameter). The left node was not split further. It can be seen from Figure 2 that the variable Average Characters was used to create three subgroups for the stimuli with 43% or higher Word Diversity: Stimuli with an Average Characters score of 3.982 or lower (Group 1, Node 3), stimuli with a score between 3.982 and 5.053 (Group 2, Node 5), and stimuli with a score higher than 5.053 (Group 3, Node 6). The first group (Node 3) had the highest average a-parameter, followed by Group 2 (Node 5). The fifth node was split further using the variable Ordered Positions: Stimuli that did not contain ordered positions were placed in the left node (smaller average a-parameter), while stimuli with (partially) ordered positions were placed in the right one (larger average a-parameter). The sixth node was split using the variable Number of Entities: stimuli with seven entities or fewer were placed in the left node (larger average a-parameter) and those with more than seven entities in the right node (smaller average a-parameter). Note that the largest average a-parameter was found for Node 3 and the smallest for Node 10. The proportion explained variance for this model was 0.35.

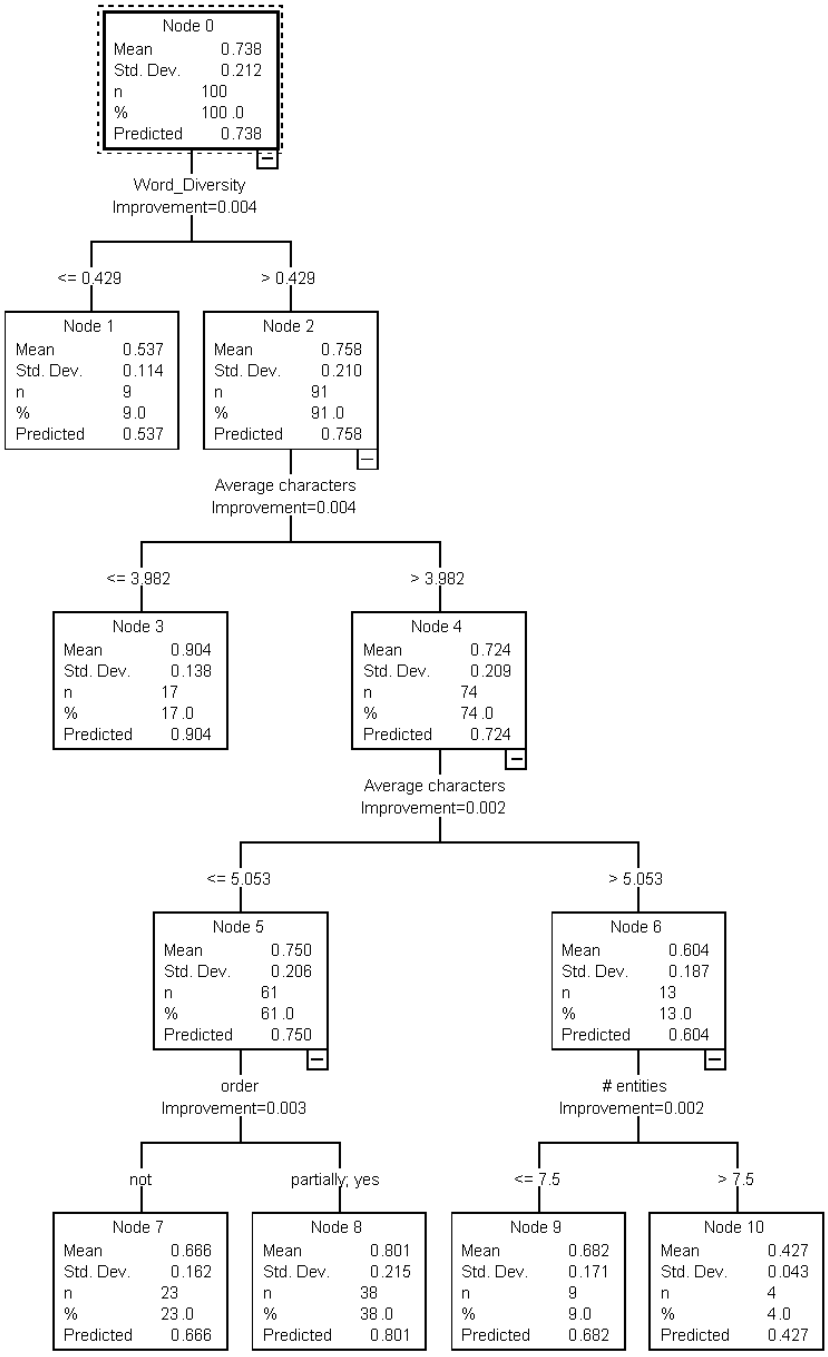

FIGURE 2. *Univariate regression tree for the average α-parameter per stimulus/testlet contained in the AR section of the LSAT*

In the following paragraphs, the trees are presented and explained in detail. For each split, the relevant independent variable and its associated cut-score will be presented. Furthermore, it is indicated per child-node pair whether the dependent variable has a larger value compared to the other child node, or a smaller value.

Average b-Parameter as Dependent Variable

The *average b-parameter per testlet* had a bimodal distribution (Mode 1 at 0.25, Mode 2 at 1.1) with a mean of 0.59 (*SD* = 0.56). The final tree (Figure 3) had 16 nodes and contained the following independent variables: Brown News Popularity, Word Diversity, Percentage of Negative Words, and Percentage of Content Words. The Brown News Popularity was slightly positively skewed with a mean of 26.55 (*SD* = 11.52). Percentage of Negative Words was highly positively skewed with a mean of 2.13% (*SD* = 1.87%). Percentage of Content Words was normally distributed with a mean of 32.21% (*SD* = 5%).

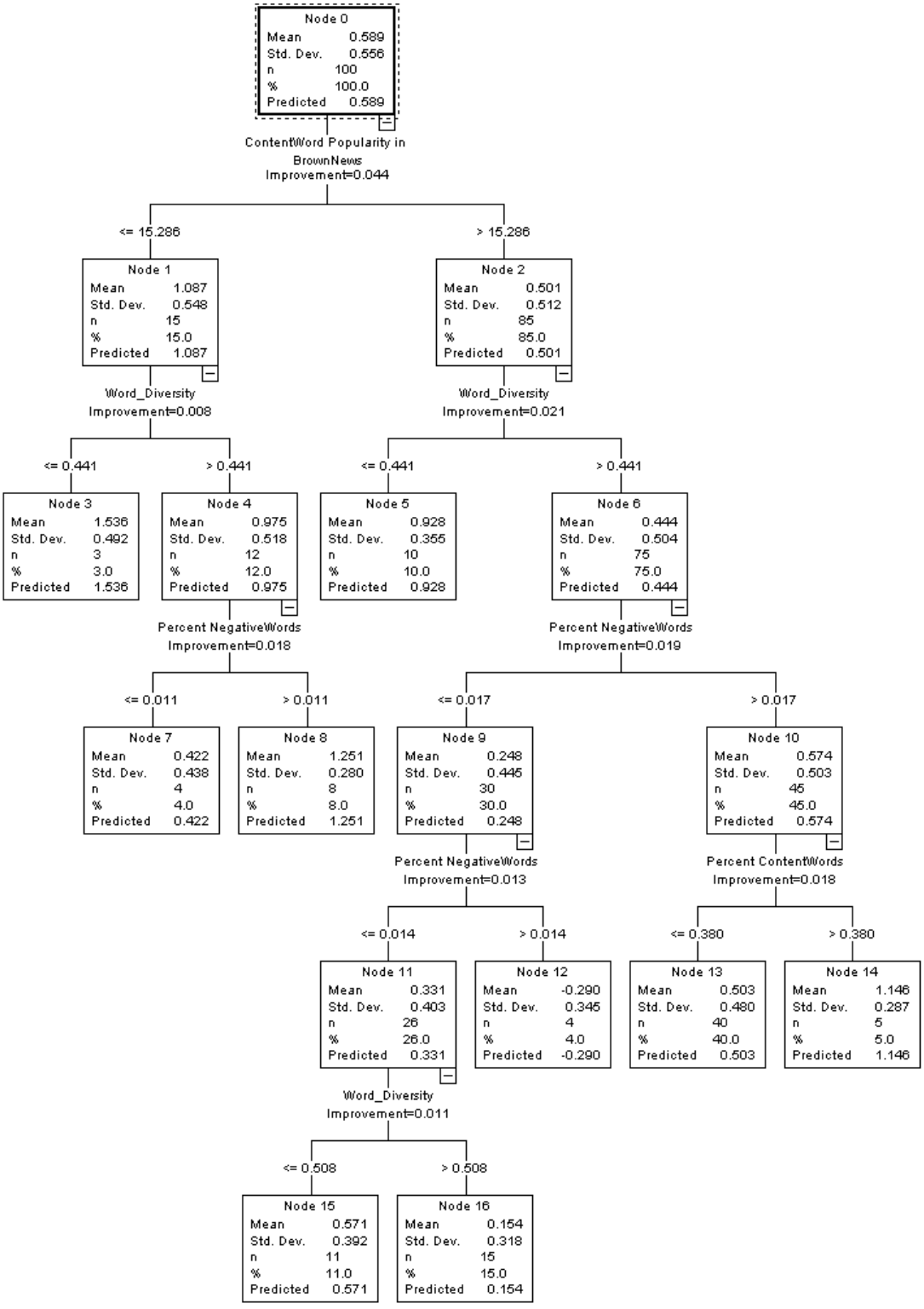

FIGURE 3. *Univariate regression tree for the average b-parameter per stimulus/testlet contained in the AR section of the LSAT*

The Brown News Popularity was common to all stimuli and was used for the first split: Stimuli with a score of 15.286 or lower were placed in the left branch (larger average b-parameter) while stimuli with a score over 15.286 were placed in the right branch (smaller average b-parameter). For both resulting branches, the same value of Word Diversity was used for the subsequent split: Stimuli with 44.1% or lower Word Diversity were associated with a larger b-parameter than stimuli with a higher percentage Word Diversity. For both branches, a terminal node (see Figure 3, Nodes 3 and 5) was reached for the group of stimuli with 44.1% or lower Word Diversity, whereas the other nodes were split further using the variable Percentage of Negative Words. In the left branch, stimuli with at most 1.1% negative words were placed in the left terminal node (smaller average b-parameter), and stimuli with more than 1.1% negative words were placed in the right terminal node (larger average b-parameter). In the right branch, three subgroups were created using Percentage of Negative words: Stimuli with 1.4% (Group 1, Node 11) or less, stimuli with more than 1.4% but less than or equal to 1.7% (Group 2, Node 12), and stimuli with more than 1.7% negative words (Group 3, Node 10). The second group ended up in a terminal node (Node 12). Node 11 was split further using the variable Word Diversity: Stimuli with a score of up to 50.8% were placed in the left terminal node (larger average b-parameter), and those with a score of more than 50.8% were placed in the right terminal node (smaller average b-parameter). Node 10 was split using the variable Percentage of Content Words: Stimuli with 38% or less were placed in the left terminal node (smaller average b-parameter) and those with more than 38% content words were placed in the right terminal node (larger average b-parameter). Note that the largest b-parameter was found for Node 3 and the smallest for Node 12. The proportion explained variance for this model was 0.56.

Average c-Parameter as Dependent Variable

The average c-parameter had a mean of 0.22 (*SD* = 0.06). Since the spread was so small, there was not much to be gained by creating subgroups (nodes) using TBR: The largest difference in node means was 0.08. Therefore, the model is not shown here.

Tree-Based Regression Step 2: Multivariate Tree

In this model, both the a- and b-parameters served as dependent variables.[1] The correlation between the variables was −.26. The final model is depicted in Figure 4. The means are not printed in numbers in this figure; instead bar plots are used. Note that the height of the light blue bars represents the mean average a-parameter in each of the nodes, and the height of the dark blue bars represents the mean average b-parameter. The final tree had 16 nodes and contained the following independent variables: Brown News Popularity, Word Diversity, Percentage of Negative Words, Percentage of Content Words, Ordered Positions, and Number of Entities.

[1] Since the univariate analysis of the average *c*-parameter showed the node means differed little from each other, the *c*-parameter was not incorporated into the multivariate tree. Note that the average *c*-parameter had a correlation of .52 and −.05 with the average *a*- and *b*-parameters, respectively.

The first two splits were based on the variables that were used for the first split in the two univariate trees: The Brown News Popularity was used for the first split (cut-score 15.29, identical to the cut-score for the univariate tree based on the b-parameter), followed by Word Diversity (cut-score 44.1%, versus 42.9% for the univariate tree based on the a-parameter and 44.1% for the b-parameter tree). As was true for the univariate tree, a relatively large mean score on The Brown News Popularity was associated with a lower average b-parameter. It can be seen from the bar plots in Figure 4 that this variable does not have a strong association with the average a-parameter. This can be seen as a confirmation of the results of the univariate a-parameter model which did not include The Brown News Popularity. Word Diversity was included in both univariate trees, and its effect is similar to that found in those models: A relatively large Word Diversity score is associated with a somewhat higher average a-parameter and a low average b-parameter. Three subgroups were created using Percentage of Negative words using the same cut-scores as in the b-parameter tree: stimuli with 1.4% (group 1, node 9, small average b-parameter) or less, stimuli with 1.5–1.7% (Group 2, Node 10, very small average b-parameter), and stimuli with more than 1.7% negative words (Group 3, Node 5, larger average b-parameter). The average a-parameter was not much different for these groups. Node 10 was a terminal node. Node 9 was split once more, using the variable Word Diversity: Stimuli with a score of 50.76% or higher were placed in terminal Node 16 (larger average a-parameter, smaller average b-parameter), and those with a score lower than 50.76% were placed in terminal Node 15 (smaller average a-parameter, larger average b-parameter). Node 5 was split using the variable Number of Entities: Stimuli with five entities or fewer were placed in Node 7 (larger average a- and b-parameters) and those with more than five entities in Node 8 (smaller average a- and b-parameters). Node 7 was split once more, using the variable Ordered Positions: Stimuli that contained ordered positions were placed in Node 11 (larger average a- and b-parameters), while stimuli that contained partially or no ordered positions were placed in Node 12 (smaller average a- and b-parameters). Node 8 was split using the variable Percentage of Content Words: Stimuli with 39.3% or less were placed in the left terminal node (larger average a-parameter, smaller average b-parameter) and those with more than 39.3% content words were placed in the right terminal node (smaller average a-parameter, larger average b-parameter). The proportion explained variance for this model was 0.47.

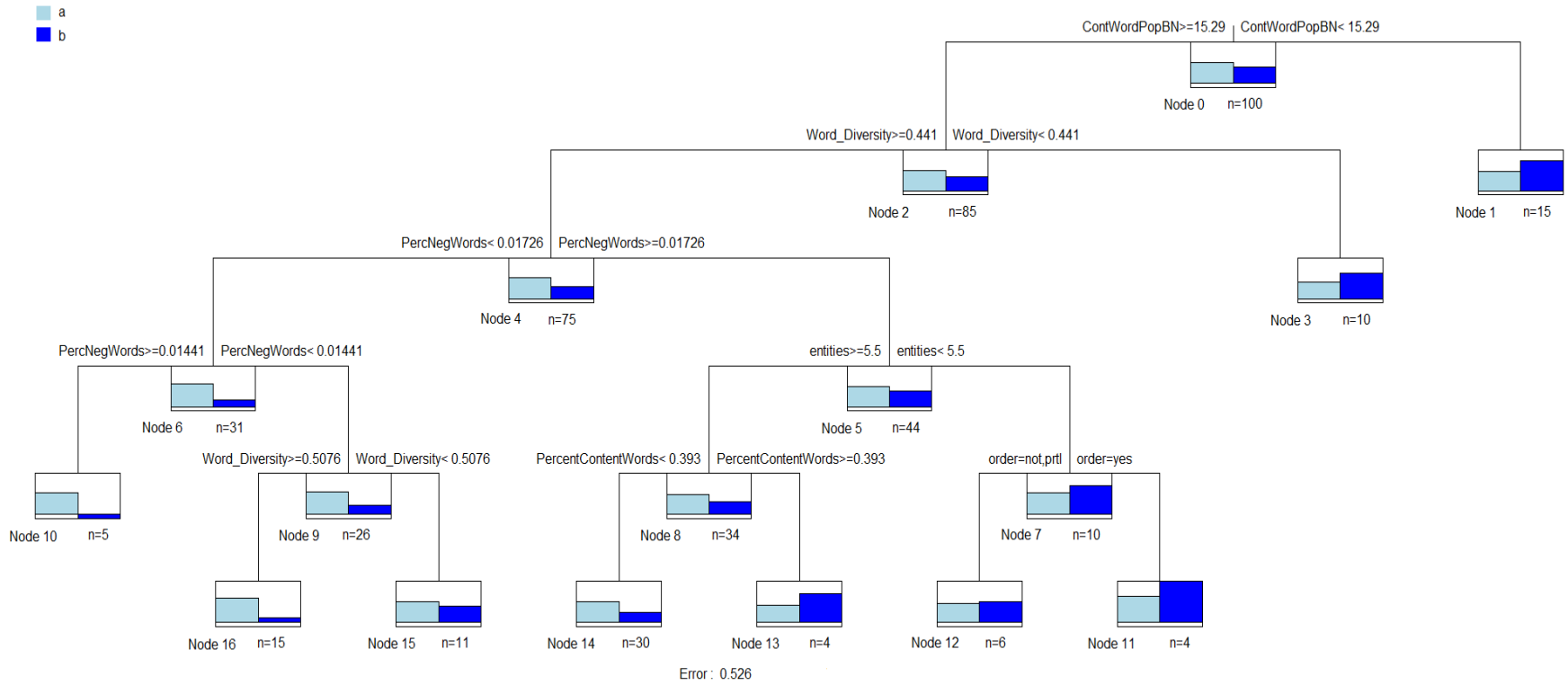

FIGURE 4. *Multivariate regression tree for the average a- and b-parameters per stimulus/testlet contained in the AR section of the LSAT*

Discussion

In recent studies, we showed that stimulus features were related to the size of the testlet effect (Paap et al., 2012; Paap & Veldkamp, 2012). In this study, we showed that the average item difficulty and discrimination parameter can be predicted using stimulus features.

The part of the LSAT that we analyzed in this study, the AR section, assesses the ability to reason within a given set of circumstances. These circumstances are described in a stimulus. Our results showed that stimuli associated with easy items consisted of many different but commonly used words (relatively high score on Word Diversity and Brown News Popularity) and an intermediate proportion of negative words. The most difficult stimuli were those with infrequently used words (low score on Brown News Popularity) and a low Word Diversity. It may be inferred that stimuli that are rich in information content that is easy to process generally have items that are rather straightforward to solve, whereas stimuli that have rather specific information that is not that easy to process have items that on average are more difficult.

Our results also showed that the average discrimination parameter per testlet was associated with both structural features and linguistic features. The average discrimination was highest for stimuli with a relatively high Word Diversity and a relatively low number of Average Characters. Thus, stimuli that contain many different words that are relatively short, and thus have a high information density, are very useful for distinguishing between respondents with different ability levels. The average discrimination was lowest for relatively lengthy stimuli containing many different words and a complicated set of circumstances (a relatively high Word Diversity, number of Average Characters, and Number of Entities). A possible explanation could be that other skills (i.e., skills other than analytical reasoning, such as reading comprehension) are needed to solve these kinds of problems.

The multivariate analyses, where the a- and b-parameters were predicted jointly, contained three variables that were only present in the univariate b-parameter tree, two that were only present in the a-parameter tree, and two that were present in both. An important advantage of the multivariate analysis is that one can see at a glance the effect of a predictor on both the a-parameter and the b-parameter, as well as the relationship between the two.

In previous studies, we identified testlet features that distinguished groups of testlets according to the size of the testlet effect, and we showed that the average item difficulty per testlet has a strong influence on testlet effect size. In the current study, we showed that a set of testlet features could distinguish between groups of testlets with varying average item difficulty. A future study could unravel the relationship among testlet features, a- and b-parameters, and the testlet effect even further—for example, by investigating whether average item difficulty does in fact moderate the relationship between testlet features and the testlet effect. Future research may also be directed at assessing the generalizability of our finding. A future study could investigate whether our results can be replicated for another set of stimuli—stimuli that measure both the same construct (analytical reasoning) and other constructs.

Both our current findings and the findings of our earlier reports (Paap et al., 2012, Paap & Veldkamp, 2012) provide further insight into the relationship between testlet features and their psychometric properties. It was promising to see that the results of our statistical analysis were validated by content experts, who were able to translate

our results to cognitive processes that are presupposed to underlie the response behavior. Even though Glas (2012b) revealed that the incremental validity of our ETRM is still relatively small, the model provides some information about how stimuli can be selected or manipulated in the test development phase to produce more favorable psychometric properties in terms of testlet effects and average item parameters.

References

Bradlow, E. T., Wainer, H., & Wang, X. H. (1999). A Bayesian random effects model for testlets. *Psychometrika, 64*(2), 153–168.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth International Group.

DeMars, C. E. (2012). Confirming testlet effects. *Applied Psychological Measurement, 36*(2), 104–121. doi: 10.1177/0146621612437403

Drum, P. A., Calfee, R. C., & Cook, L. K. (1981). The effects of surface structure variables on performance in reading comprehension tests. *Reading Research Quarterly, 16*(4),
486–514.

Embretson, S. E., & Wetzel, C. D. (1987). Component latent trait models for paragraph comprehension tests. *Applied Psychological Measurement, 11*(2), 175–193. doi: 10.1177/014662168701100207

Francis, W. N., & Kučera, H. (1964, 1971, 1979). *A standard corpus of present-day edited American English, for use with digital computers (Brown).* Providence, RI: Brown University.

Glas, C. A. W. (2010). *Preliminary manual of the software program Multidimensional Item Response Theory (MIRT).* Enschede, the Netherlands: Department of Research Methodology, Measurement and Data-Analysis, University of Twente.

Glas, C. A. W. (2012a). *Estimating and testing the extended testlet model* (LSAC Research Report, RR 12-03). Newtown, PA: Law School Admission Council.

Glas, C. A. W. (2012b). *Fit to testlet models and differential testlet functioning*. Manuscript submitted for publication.

Glas, C. A. W., Wainer, H., & Bradlow, E. T. (2000). MML and EAP estimation in testlet-based adaptive testing. In W. J. van der Linden & C. A. W. Glas (Eds.), *Computer Adaptive Testing: Theory and Practice* (pp. 271–288). Dordrecht, the Netherlands: Kluwer.

Gorin, J. S., & Embretson, S. E. (2006). Item diffficulty modeling of paragraph comprehension items. *Applied Psychological Measurement, 30*(5), 394–411. doi: 10.1177/0146621606288554

Law School Admission Council Inc. (2010). *The Official LSAT® Handbook*. Newtown, PA: Law School Admission Council, Inc.

Li, Y., Bolt, D. M., & Fu, J. (2005). A test characteristic curve linking method for the testlet model. *Applied Psychological Measurement, 29*(5), 340–356. doi: 10.1177/0146621605276678

Marais, I. D., & Andrich, D. (2008). Effects of varying magnitude and patterns of local dependence in the unidimensional Rasch model. *Journal of Applied Measurement, 9*, 105–124.

Matteucci, M., Mignani, S., & Veldkamp, B. P. (2012). Prior distributions for item parameters in IRT models. *Communications in Statistics—Theory and Methods*, *41*(16–17), 2944–2958.

Paap, M. C. S., Glas, C. A. W., He, Q., & Veldkamp, B. P. (2012). *Using testlet features to predict response behavior on testlets: The explanatory testlet response model.* Manuscript submitted for publication.

Paap, M. C. S., & Veldkamp, B. P. (2012). Minimizing the testlet effect: Identifying critical testlet features by means of tree-based regression. In T. J. H. M. Eggen & B. P. Veldkamp (Eds.), *Psychometrics in Practice at RCEC*. Enschede, the Netherlands: RCEC, Cito/University of Twente.

R. (2004). *R reference manual: Base package*, Vol. 1. Bristol: Network Theory Ltd.

SPSS. (2007). *SPSS for Windows*, Rel. 16.0.1. Chicago: SPSS Inc.

Tao, W., & Su, Y.-L. (2012, April). *Setting up critical values for identifying local dependence using the testlet model*. Paper presented at the the Annual Meeting of National Council on Measurement in Education, Vancouver, British Columbia, Canada.

Therneau, T. M., & Atkinson, B. (2009). mvpart: Multivariate regression trees. R package version 3.1-41, R port by Brian Ripley, extensions and adaptations of rpart to mvpart by G. De'ath. http://CRAN.R-project.org/package=mvpart

Thissen, D., Steinberg, L., & Mooney, J. A. (1989). Trace lines for testlets: A use of multiple-categorical-response models. *Journal of Educational Measurement, 26*(3), 247–260. doi: 10.1111/j.1745-3984.1989.tb00331.x

Wainer, H., & Wang, X. (2000). Using a new statistical model for testlets to score TOEFL. *Journal of Educational Measurement, 37*(3), 203–220.

Zimowski, M. F., Muraki, E., Mislevy, R. J., & Bock, R. D. (1996). *BILOG-MG: Multiple-group IRT analysis and test maintenance for binary items* [Computer software]. Chicago: Scientific Software International.