Adaptive Testing Without IRT in the Presence of Multidimensionality

Duanli Yan
Charles Lewis
Martha Stocking
Adaptive Testing Without IRT in the Presence of Multidimensionality

Duanli Yan, Charles Lewis, and Martha Stocking

Educational Testing Service, Princeton, NJ

April 2002
Research Reports provide preliminary and limited dissemination of ETS research prior to publication. They are available without charge from:

Research Publications Office
Mail Stop 10-R
Educational Testing Service
Princeton, NJ 08541
Abstract

It is unrealistic to suppose that standard item response theory (IRT) models will be appropriate for all of the new and currently considered computer-based tests. In addition to developing new models, we also need to give some attention to the possibility of constructing and analyzing new tests without the aid of strong models. Computerized adaptive testing currently relies heavily on IRT. Alternative, empirically based, nonparametric adaptive testing algorithms exist, but their properties are little known. This paper introduces a nonparametric, tree-based algorithm for adaptive testing and shows that it may be superior to conventional, IRT-based adaptive testing in cases where the IRT assumptions are not satisfied. In particular, it shows that the tree-based approach clearly outperformed (one-dimensional) IRT when the pool was strongly two-dimensional.

Key words: Computerized adaptive testing, item response theory (IRT), regression tree
Introduction

Wainer, Lewis, Kaplan, and Braswell (1991) and Wainer, Kaplan, and Lewis (1992) introduced a testlet-based algebra exam and then compared a hierarchically constructed (adaptive) four-item testlet with a linear (fixed format) testlet under various conditions. Through cross-validation, they compared an adaptive test using an optimal four-item tree to a best fixed four-item test (both defined in terms of maximum differentiation) and found overall that the adaptive testlet dominates the best fixed testlet, but the superiority (at a considerable cost for adaptive testlet over fixed testlet) is modest. They also found that the adaptive test outperforms the fixed test for the groups with extreme observed scores. They concluded that, for circumstances similar to their cases, “a fixed format testlet that uses the best items in the pool can do almost as well as the optimal adaptive testlet of equal length from that same pool.”

Schnipke and Green (1995) compared an item selection algorithm based on maximum differentiation among test takers with one using item response theory and based on maximum information. Overall, adaptive tests based on maximum information provided the most information over the widest range of ability values and, in general, differentiated among test takers slightly better than the other tests. Although the maximum differentiation technique may be adequate in some circumstances, adaptive tests based on maximum information were clearly superior in their study.

This paper introduces an adaptive testing algorithm that balances maximum differentiation among test takers with stable estimation at each stage of testing and compares this algorithm with a traditional one using IRT and maximum information. In particular, we simulate one- and two-dimensional item pools to see how dimensionality affects the relative performance of these two approaches to adaptive testing. This is an extension and revision of our paper presented at the 1998 annual meeting of National Council on Measurement in Education, San Diego, CA.

Method

In this paper, we consider adaptive testing as a prediction system. Specifically, we use adaptive testing to predict the observed scores that test takers would have received if they had taken every item in a reference test or a pool. (We restrict our attention to binary items, scored correct or incorrect.) This is a nonparametric approach in the sense that we do not introduce
latent traits or true scores. We are considering only the observed number-correct scores test
akers would have received if they had taken every item we could have given. In other words,
our criterion is the total observed score for a pool or reference test. The adaptive testing
algorithm we introduce in this paper is based on the classification and regression tree approach

In order to construct an adaptive test as a prediction system, we need to have a calibration
sample. Specifically, we need a sample of test takers who take every item in the pool that will be
used for adaptive testing. (For operational use, incomplete calibration designs would obviously
be necessary.) We can then compute the criterion (total observed score) for these test takers.
This is analogous to the calibration sample one needs when using IRT to do adaptive testing.
However, the purpose of the IRT calibration sample is to calibrate items. Our purpose is not to
calibrate items individually but to generate a regression tree.

Figure 1 is an example of such a regression tree. The vertical axis represents the stage of
testing and the horizontal axis identifies the prediction of the total score at each stage. In this
example, there are nine stages (i.e., each test taker would be administered nine items). The
nodes of the tree are plotted as octagons with item numbers inside. The branches represent the
paths test takers could follow in the test, taking the right branch after answering the item in the
octagon correctly and the left branch after answering it incorrectly. At the end, the locations of
the terminal nodes, or leaf nodes plotted as circles, give the final predictions of test takers’ total
scores.
Figure 1. Regression tree structure.
Once the regression tree has been constructed (and validated—see below), it may be used to administer an adaptive test. Thus, based on Figure 1, all test takers would be administered item 31 first. Test takers answering correctly would receive item 27. Those answering incorrectly would get item 28. Test takers continue through the tree to the terminal nodes and receive the corresponding final predicted total score as their score on the test. For instance, test takers who receive item 5 as the last item and answer it correctly would have a predicted total score of 32.5.

Returning to the construction of the tree, suppose we have a calibration sample of test takers who answered every item in a pool of items. The total number of correct responses for each test taker is the criterion we will use. Our regression tree begins with the item (in Figure 1, item 31) that best predicts the observed score in a least squares sense for these test takers. It splits the calibration sample into two subsamples: those test takers who answered the item incorrectly and those who answered it correctly. They are represented as the nodes for items 28 and 27 in Figure 1. These two subsamples have maximum differentiation between them (i.e., maximum sum of squares between subsamples). The horizontal locations of the nodes are the mean total scores for the subsamples.

We continue the construction of the tree by finding the best predicting item for those test takers responding correctly to the first item (in Figure 1, item 27), as well as the best item for those with an incorrect response to the first item (in Figure 1, item 28). At each stage, the total calibration sample is split into subsamples, and an optimal item is chosen for each subsample. At each stage, subsamples with similar average criterion scores are combined as the tree progresses to keep the total number of such subsamples within reasonable limits. In Figure 1, the nodes for test takers who correctly answered item 28 and for test takers who incorrectly answered item 27 are combined, and the combined subsamples are administered item 16. At the end of the process, the adaptive test score given to each test taker is the average criterion score for the final subsample in which the test taker has been classified (in Figure 1, the combined leaf nodes).

A portion of the prediction for the calibration sample capitalizes on chance. To evaluate the procedure, we construct the regression tree in a calibration sample and apply the predictions from the calibration sample to compare to the observed scores in an application sample. This application sample has the same structure as the calibration sample. In other words, every test
taker answers every item, so a criterion-observed score can be computed. The precision of estimation using the regression tree as an adaptive test may be measured using the mean of the squared discrepancies (or residuals) between predicted and observed scores in the application sample. For purposes of interpretation, this quantity may be compared to the variance of the observed scores in the application sample. In particular, we will consider the proportion of variance accounted for by the tree-based predictions.

Results

We wanted to see how our approach worked when the item pool was multidimensional. So we carried out a unidimensional simulation as our baseline, followed by a two-dimensional simulation. We compared results from the regression tree approach with a traditional approach to CAT using 3PL IRT and maximum information for these two simulations.

One-Dimensional Simulations

For our first set of simulations, we constructed our calibration sample using the 3PL IRT model to generate item responses for a sample of 500 simulated test takers with 494 items in an actual item pool for an operational computer adaptive test assessing quantitative reasoning. (Specifically, we used the 3PL IRT model with item parameters set equal to the estimates from the operational pool.)

We constructed a regression tree as described in the method section for a 19-item adaptive test for this calibration sample. The mean of the squared residuals between predictions and total observed scores for the calibration sample is 11.7. This quantity may be compared to the variance of the total observed scores for the sample, or 6,586.0. Thus 99.8% of the total observed score variance is accounted for in the calibration sample using the predictions from the regression tree.

Next, we used the regression tree predictions based on the calibration sample to compare with the total (IRT-based) true scores (rather than total observed scores) in an application sample of size 10,000, constructed in the same way as the calibration sample. The mean squared residual in the application sample, based on the calibration predictions at the end of the 19-item test, is 1223.7 (with original true score variance of 6,457.5), which means the predictions
account for 81.0% of the total true score variance. From this result, we see that the calibration sample had a substantial capitalization on chance.

From the same calibration sample we used to construct the regression tree, we also obtained 3PL item parameter estimates using PARSCALE (Muraki & Bock, 1993). We then carried out an IRT-based (maximum information) adaptive testing simulation on the application sample using these estimated item parameters. As estimates of total true score, we used maximum likelihood estimates of the latent trait, transformed using the test characteristic curve for the entire pool. The mean squared residual between these estimates and the total true scores in the application sample is 514.5. Comparing this with the total true score variance for the application sample, we see that the IRT-based estimates account for 92.0% of that variance, substantially more than when using the tree-based predictions. Figure 2 provides a more detailed comparison of the regression tree and IRT-based CATs as a function of test length.

![Figure 2](image)

Note. I=IRT; T=Tree.

**Figure 2.** Comparison of tree-based and IRT CATs in one-dimensional application sample (referring to true scores).
To compare the performance of the two approaches further, we restricted our attention to the full length (19 item) tests and looked at the characteristics of the true score estimates as a function of the true score. Figure 3 shows the bias in these estimates for the IRT and the tree-based approaches. As can be seen, the IRT estimates have virtually no bias, but the tree-based estimates have substantial bias, at least for the extreme scores. The nature of the bias—positive for low scores and negative for high scores—is a typical regression phenomenon: The estimates are regressed toward the overall mean score.

In Figure 4, the variances of the estimates are plotted as a function of true score. The IRT-based estimates show substantially less variance at all true score levels than do the tree-based estimates. The lower part of Figure 4 shows the result of combining squared biases and variances for the two approaches, yielding mean squared differences between the estimates and the true scores. The IRT-based estimates have substantially smaller mean squared differences than do the tree-based estimates. This is especially true for extreme true scores. As we have seen, this is the result of the biases in these estimates.

Two-Dimensional Simulations

For our second set of simulations, we considered what would happen if the items in the pool were multidimensional. We used the same pool, but we split it into two equal parts such that half of the items were considered to measure one latent trait and the other half measured a second, uncorrelated latent trait. The parameters for all items were left unchanged. A calibration sample consisting of the responses to all items in the pool for a sample of 500 simulated test takers was generated, based on the two-dimensional latent trait model just described. Specifically, for each simulated test taker, two latent trait values were sampled. (Figure 5 shows the bivariate frequency distribution—for the application sample—of these two latent traits.) One of these was used as the basis for response generation for items in the first half of the pool, while the other was used to generate responses to items in the second half of the pool. We used the resulting data as our calibration sample for both the tree-based approach and the one-dimensional IRT model, as before. (Note that the 3PL model was fit simultaneously to all items in the pool, ignoring which half they were in.)
Note. I=IRT; T=Tree; other=Observed.

Figure 3. Mean score biases in the one-dimensional case.
Note. I=IRT; T=Tree; other=Observed.

Figure 4. Variances and MSE in the one-dimensional case.
Figure 5. Distribution of the two-dimensional thetas.
The result of the IRT calibration was uncritically adopted for the adaptive testing simulation with the application sample. In particular, no items were excluded from the pool due to lack of fit. However, it is worth briefly noting one important aspect of the results of the calibration. Figure 6 plots the estimated slope parameters (A) against the true values, using two different plotting symbols for items associated with the two dimensions (open and closed circles). It is clear from the plot that the slopes associated with the first dimension were recovered reasonably well (given that the calibration sample size is only 500), while those for items measuring the second latent trait were all estimated at a value close to zero. In other words, the calibration essentially focused on the first dimension and ignored the second dimension. Such a result has been described and discussed previously by (for example) Reckase (1979).

We also generated responses to all items in the pool for a new sample of 10,000 simulated test takers in the manner just described for use as our application sample. We used the tree-based predictions from the calibration sample in the application sample for the regression tree approach. We also carried out an (one-dimensional) IRT adaptive testing simulation for this application sample, using the item parameters obtained from the calibration sample. The two-dimensional, IRT-based total true scores for the pool served as the evaluation criterion for both procedures. Specifically, we compared the mean squared residuals obtained for the two methods.

As shown in Table 1, fitting a regression tree to the data for the calibration sample produced a mean squared residual of 20.8, compared with a total observed score variance of 3,025.9. In other words, 99.3% of the observed score variance can be accounted for in this calibration sample using the predictions from the regression tree. (Note that the total observed score variance in this sample is much smaller than that obtained in the calibration sample based on the one-dimensional IRT model: 3,025.9 compared to 6,586.0. This is a result of the fact that the between-set item correlations in our second design are all zero.)

In the application sample, using the tree-based predictions from the calibration sample to predict the total true scores gave a mean squared residual of 1,187.0 (see Table 2). The total true score variance in the application sample is 3,180.0, so 62.7% of this variance is accounted for by the tree-based predictions, as shown in Table 3. (Here we see an even more substantial capitalization on chance than in the one-dimensional case.) True score estimates based on a 3PL
IRT CAT produced a mean squared residual of 1,960.0 in the application sample, so that only 38.4% of the total true score variance is accounted for by these estimates. Figure 7 provides a more detailed comparison of the regression tree and IRT-based CATs as a function of test length.

Note. ?=Items associated with Theta 1; ?=Items associated with Theta 2.

Figure 6. Comparison of A parameters in the two-dimensional case.
### Table 1
*Mean Squared Residual After 19-item Test in Calibration Sample*

<table>
<thead>
<tr>
<th></th>
<th>1-dim</th>
<th>2-dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>11.7</td>
<td>20.8</td>
</tr>
<tr>
<td>Total</td>
<td>6,586.0</td>
<td>3,025.9</td>
</tr>
<tr>
<td>Proportion of Variance Accounted for</td>
<td>99.8</td>
<td>99.3</td>
</tr>
</tbody>
</table>

### Table 2
*Mean Squared Residual After 19-item Test in Application Sample*

<table>
<thead>
<tr>
<th></th>
<th>1-dim</th>
<th>2-dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRT</td>
<td>514.5</td>
<td>1,960.0</td>
</tr>
<tr>
<td>Tree</td>
<td>1,223.7</td>
<td>1,187.0</td>
</tr>
<tr>
<td>Total</td>
<td>6,457.4</td>
<td>3,180.0</td>
</tr>
</tbody>
</table>

### Table 3
*Proportion of Variance Accounted for After 19-item Test in Application Sample*

<table>
<thead>
<tr>
<th></th>
<th>1-dim</th>
<th>2-dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRT</td>
<td>.920</td>
<td>.384</td>
</tr>
<tr>
<td>Tree</td>
<td>.810</td>
<td>.627</td>
</tr>
</tbody>
</table>
A comparison of the performances of the two approaches is shown in Figure 8. The biases in the estimates for the IRT-based approach are much larger than those for the tree-based approach. Figure 9 shows the variances of the estimates as a function of the true score. The variances for the tree-based approach are much larger than those for the IRT-based approach. Figure 10 shows the mean squared differences between the estimates and true scores as a result of combining squared biases and the variances for the two approaches (see Figure 11). The IRT-based estimates have substantially larger mean squared differences than do the tree-based estimates, primarily as a result of the large biases.

Note. I=IRT; T=Tree.

Figure 7. Comparison of tree-based and IRT CATs in two-dimensional application sample (referring to true scores).
Figure 8. Mean score biases in the two-dimensional case.
Note. ?=IRT; t=Tree.

Figure 9. Variances in the two-dimensional case.
Figure 10. Mean square differences (vs. true) in the two-dimensional case.

Note. ? = IRT; t = Tree.
Figure 11. Squared mean score biases in the two-dimensional case.

Note. ?=IRT; t=Tree.
Returning to Figure 8, we noticed that the pattern of the biases for the tree-based approach, namely positive bias for low true score and negative bias for high ones, is similar to what was observed in the one-dimensional case and may be understood as a regression effect. The pattern of biases for the IRT-based estimates is more complex. It appears to have the form of a distorted diamond.

To better understand this pattern, these biases have been plotted in Figure 12 as a function of the two latent traits that formed the basis for the true scores being estimated. It can be seen that the bias surface is essentially a tilted plane. For a given value of the first latent trait, the bias is essentially a linear function of the second trait: As the second trait increases (and, hence, the total true score), the bias becomes more negative. This is a result of the fact that the IRT calibration ignored this second dimension. There is a smaller effect for the first latent trait as well, indicating that it, too, was not completely identified by the calibration.

Discussion

For our one-dimensional example, Figure 2 shows that, once the adaptive test is long enough, the IRT-based CAT produces consistently better estimates of true scores than does the tree-based approach. It is worth noting, however, that in the early stages of testing, the maximum likelihood estimates from the IRT-based CAT are very poor compared to those from the regression tree. This suggests a possible hybrid algorithm, using a regression tree to select the first few items on an adaptive test and then switching to a maximum information, IRT-based algorithm. This leaves open the question of how best to make the transition from regression tree to maximum likelihood estimates.

In our two-dimensional example, the regression tree clearly provides better prediction than the IRT-based CAT at all test lengths, as shown in Figure 7. The average numbers of items used by the tree-based approach are 9.5 and 9.5 for two dimensions, while those used by the IRT-based approach are 19 and 0 for two dimensions. This result is consistent with our earlier observations regarding the IRT calibration of the two-dimensional pool. It also shows that the tree-based approach functioned appropriately in the presence of multidimensionality.
It should be noted, however, that our example is based on an extreme version of a two-dimensional model in which every item measures either one or the other dimension (but not both), and the two uncorrelated dimensions are taken to be equally important. There might be ways to make IRT work better, but our main point is that the tree-based approach can deal with multidimensionality.

One of the limitations of the tree-based approach described in this paper is that there is no control of item exposure rates. (For instance, our algorithm now has everyone take the same first item.) Another limitation is that no attempt is made to control the content of the adaptive tests. A third limitation is that all test takers in the calibration and application samples were assumed to have answered all items in the pool. (All these limitations also apply to the IRT-based algorithm we used for comparison purposes in this study. It should be noted, however, that operational IRT CATs have none of these limitations.) Future research will address these and related issues.
Conclusions

We have developed a nonparametric, tree-based approach to adaptive testing and shown that it may be superior to conventional, IRT-based adaptive testing in cases where the IRT assumptions are not satisfied. In particular, we showed that the tree-based approach clearly outperformed (one-dimensional) IRT when the pool was strongly two-dimensional.
References


Appendix

Description of the Algorithm

Our regression trees are constructed as follows: For each node, we select an unused item that gives the maximum differentiation (in a least squares sense) on the criterion score for splitting the current node into two nodes. For each stage, we compare all the nodes at that stage by computing the pair-wise t-statistics and effect size measures using the criterion score. If, for some pair of nodes, the absolute value of the t-statistic is less than some preset critical value or the absolute value of the effect size measure is less than some preset critical value, then we combine the two nodes. If more than one pair of nodes meet either of these criteria, we start by combining the pair with the smallest t-statistic (or smallest effect size if no t-statistic is less than the critical value). We then compute all t-statistics and effect sizes for this new node with the others and repeat the process until all pairs of nodes are distinct in terms of their t-statistics and effect sizes.

We continue constructing the regression tree stage by stage in this manner until a specified fixed test length is reached. At the final stage, each test taker in a sample is classified by leaf node after matching his or her response pattern to the regression tree structure. The prediction of that individual’s criterion score is the average score of the leaf node in which the individual has been classified.

Exhibit A1 reproduces an edited version of a portion of the output from the computer program we use to construct regression trees. Specifically, it is the output describing the construction of the tree illustrated in Figure 1. The information given in line 007 describes the complete calibration sample (node 0) as having 250 (simulated) test takers, a mean criterion score of 34.7360, and a sum of squared deviations of individual scores around this mean (Deviance) of 28608.5760. Line 012 repeats some of this information and notes that item 31 has been selected as the first item in the tree. The output in lines 015–023 will be of more interest at later stages.

Lines 027 and 028 describe nodes 1 and 2, which are defined as those test takers who answer item 31 incorrectly or correctly, respectively. Specifically, there are 71 of the former and 179 of the latter, with mean criterion scores of 25.3521 and 38.4581, respectively. Lines 029 and 030 give the t-statistic and effect size measure used to compare nodes 1 and 2. Both values exceed their respective criteria, so no combining of nodes occurs at this stage. Lines 035 and
036 indicate that items 28 and 27 have been chosen for nodes 1 and 2, respectively. Line 039 gives the total within-node sum of squares at stage 1 as 19876.6329. Note that this is the sum of the sums of squares for each of the two nodes at this stage. Line 041 gives the proportion of variance accounted for at this stage, obtained by subtracting the ratio of the deviance at this stage to the deviance at stage 0 from unity. Lines 047 and 048 report the standard deviations for nodes 1 and 2.

Lines 052–055 describe the four nodes at stage 2 defined by incorrect and correct answers to items 28 and 27. Lines 056–062 give all pair-wise comparisons for the nodes at this stage, as well as the comparison with the smallest t-statistic (obtained for nodes 4 and 5). Since this value (-1.2691) is less in absolute value than our critical value of 2.0, these two nodes are combined. The new nodes are described in lines 068–070, and the comparisons are given in lines 073–076. No further combination is indicated, so items are chosen for each of these nodes (2, 16, and 33, respectively), and the final description is given in lines 078–095. The actual output continues in this fashion until the specified number of stages (test length) has been reached.
Exhibit A1

Sample Output From the Program to Construct Regression Trees

<table>
<thead>
<tr>
<th>Stage</th>
<th>Node</th>
<th>Item</th>
<th>Parents</th>
<th>N</th>
<th>Deviance</th>
<th>Mean</th>
<th>T</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>31</td>
<td>-9,-9,-9,-9,-9</td>
<td>250</td>
<td>28608.5760</td>
<td>34.7360</td>
<td>999.0000</td>
<td>999.0000</td>
</tr>
</tbody>
</table>

Total Deviance at Stage 0: 28608.5760

Proportion of Variance Accounted For: 0.0000

Standard Deviation at root: 10.6974

Standard Deviation for Each Node at This Stage:

<table>
<thead>
<tr>
<th>Node</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.6974</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage</th>
<th>Node</th>
<th>Item</th>
<th>Parents</th>
<th>N</th>
<th>Deviance</th>
<th>Mean</th>
<th>T</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>28</td>
<td>0,-9,-9,-9,-9</td>
<td>71</td>
<td>3716.1972</td>
<td>25.3521</td>
<td>-11.6993</td>
<td>-1.5452</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>27</td>
<td>0,-9,-9,-9,-9</td>
<td>179</td>
<td>16160.4358</td>
<td>38.4581</td>
<td>999.0000</td>
<td>999.0000</td>
</tr>
</tbody>
</table>

(Exhibit continues)
Exhibit A1 (continued)

039  Total Deviance at Stage 1: 19876.6329
040
041  Proportion of Variance Accounted For: 0.3052
042
043  Standard Deviation at root: 10.6974
044
045  Standard Deviation for Each Node at This Stage:
046
047  Node = 1 SD = 7.2347
048  Node = 2 SD = 9.5017
049
050  ---------------------- Stage = 2 ----------------------
051
052  NodeInfo: istage= 2 inode= 3 Nsubj= 44 yval= 22.8409 Dev=1069.8864
053  NodeInfo: istage= 2 inode= 4 Nsubj= 27 yval= 29.4444 Dev=1916.6667
054  NodeInfo: istage= 2 inode= 5 Nsubj= 70 yval= 31.7857 Dev=3251.7857
055  NodeInfo: istage= 2 inode= 6 Nsubj= 109 yval= 42.7431 Dev=7790.8073
056  In Combine: inode, jnode, t, e = 3 4 -3.6175 -0.9405
057  In Combine: inode, jnode, t, e = 3 5 -8.0368 -1.4907
058  In Combine: inode, jnode, t, e = 3 6 -17.9650 -2.8575
059  In Combine: inode, jnode, t, e = 4 5 -1.2691 -0.3012
060  In Combine: inode, jnode, t, e = 4 6 -7.2206 -1.5573
061  In Combine: inode, jnode, t, e = 5 6 -9.4833 -1.4190
062  In Node, jnode, tmin, emin = 4 5 -1.2691 -0.3012
063
064  Estimations during combination:
065
066  Stage Node Item Parents N Deviance Mean T E
067
068  2 3 1,-9,-9,-9,-9 44 1069.8864 22.8409 999.0000 999.0000
069  2 4 1, 2,-9,-9,-9 97 5275.2577 31.1340 -1.2691 -0.3012
070  2 5 1,-9,-9,-9,-9 109 7790.8073 42.7431 999.0000 999.0000
071
072
073  In Combine: inode, jnode, t, e = 3 4 -7.7947 -1.3126
074  In Combine: inode, jnode, t, e = 3 5 -17.9650 -2.8575
075  In Combine: inode, jnode, t, e = 4 5 -10.4748 -1.4563
076  In Node, jnode, tmin, emin = 3 4 -7.7947 -1.3126
077

(Exhibit continues)
Exhibit A1 (continued)

<table>
<thead>
<tr>
<th>Stage</th>
<th>Node</th>
<th>Item</th>
<th>Parents</th>
<th>N</th>
<th>Deviance</th>
<th>Mean</th>
<th>T</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>079</td>
<td>2</td>
<td>3</td>
<td>2 1,3,4,6,9</td>
<td>44</td>
<td>1069.8864</td>
<td>22.8409</td>
<td>-7.7947</td>
<td>-1.3126</td>
</tr>
<tr>
<td>081</td>
<td>2</td>
<td>4</td>
<td>1, 2, 4, 6, 9</td>
<td>97</td>
<td>5275.2577</td>
<td>11.1340</td>
<td>999.0000</td>
<td>999.0000</td>
</tr>
<tr>
<td>082</td>
<td>2</td>
<td>5</td>
<td>2, 4, 33</td>
<td>109</td>
<td>7790.8073</td>
<td>42.7431</td>
<td>999.0000</td>
<td>999.0000</td>
</tr>
</tbody>
</table>

Total Deviance at Stage 2: 14135.5514

Proportion of Variance Accounted For: 0.5059

Standard Deviation at root: 10.6974

Standard Deviation for Each Node at This Stage:

Node = 3 SD = 4.9311
Node = 4 SD = 7.3746
Node = 5 SD = 8.4543