Data Dictionary for Item Selection Adaptive Algorithm

Smarter Balanced Assessment Consortium
Test Delivery System

Components: Test Authoring

Smarter Balanced Task Order 02

Revision History

Revision Description		Date
0.1 - First Draft	Alexander Kulakov	Jan. 26, 2016

Table of Contents

1 Adaptive Algorithm 3

1.1 Overview 3

1.2 Parameter Properties 3

1.3 Parameter Groups 3

1.4 Parameter Descriptions 4

1.4.1 “Weights” parameters 4

1.4.2 “Ability” parameters 5

1.4.3 “Sets” parameters 7

1.4.4 “Precision” parameters 7

1.4.5 “Termination” parameters 8

1.4.6 “Min/Max” algorithm parameters 8

1.4.7 “Common” parameters 9

1.4.8 “OffGrade” parameters 9

2 Additional Parameter Description Spreadsheet 10

1 Adaptive Algorithm

1.1 Overview

1. For best results, the Adaptive Algorithm (the "algorithm") uses a pool of items that is much larger than the number of items on a test.
2. When the algorithm needs to select the next item, it uses all eligible items in the pool.
3. Initially, the algorithm selects a subset of items from the entire pool (set1, usually 20 items) using blueprint information and constraints.
4. Next, the algorithm selects from set1 the best subset (set2, usually 5 items) of items using item statistics (item "complexity", item score dimensions and so on) and student statistics (student "ability", student information and so on). At this point the algorithm calculates metrics to define which items are the best fit.
5. Finally, the algorithm selects one item at random from set2, deliveries this item to student, marks it as selected and returns to step 2.

1.2 Parameter Properties

Item Selection Adaptive Algorithm parameters can be divided based on the roles they play in the algorithm (see below). Each parameter also has a "purpose" property with two possible values: "scalar" and "blueprint". If this parameter is "scalar" it means that it will be used for global calculations. The value "blueprint" means that this parameter can be used for Blueprint, RC (reporting category) and Affinity Group calculations.

1.3 Parameter Groups

1. "Weights" parameters
   - 1.1. bpweight
   - 1.2. rcabilityweight
   - 1.3. abilityweight (blueprint)
   - 1.4. abilityweight (scalar)
   - 1.5. itemweight
2. "Ability" parameters
   - 2.1. startability
   - 2.2. startinfo
   - 2.3. abilityoffset
   - 2.4. slope, intercept
   - 2.5. computeabilityestimates
3. "Sets" parameters
   - 3.1. cset1size
   - 3.2. cset1order
   - 3.3. cset2random

3.4. cset2initialrandom

4. “Precision” parameters

4.1. precisiontarget

4.2. precisiontargetmetweight

4.3. precisiontargetnotmetweight

4.4. adaptivecut

4.5. toocloseses

5. “Termination” parameters

5.1. terminationoverallinfo

5.2. terminationrcinfo

5.3. terminationmincount

5.4. terminationtooclose

5.5. terminationflagsand

6. “Min/Max” algorithm parameters

6.1. minitems

6.2. maxitems

6.3. ftminitems

6.4. ftmaxitems

6.5. ftstartpos

6.6. ftendpos

7. “OffGrade” parameters

7.1. offGradeProbAffectProficiency

7.2. offGradeMinItemsAdministered

7.3. proficientPLevel

8. “Common” parameters

8.1. bpmetricfunction

8.2. selectionalgorithm

8.3. isstrictmax

1.4 Parameter Descriptions

1.4.1 “Weights” parameters

Main formula:

$$f_{ijt} = w_2 \sum_{r=1}^{R} s_{rit} p_r d_{rj} + w_1 \sum_{k=1}^{K} q_k h_{1k}(v_{kijt}, V_{kit}, t_k) + w_0 h_0(u_{ijt}, U_{it}, t_0)$$

1.1. bpweight

1.1.1. Overall weight for the BP-match (w_2 in main formula; from itembank.tblsetofadminsubjects table)

   1.1.2. Weight of blueprint satisfaction metric relative to ability match metric
   1.1.3. blueprint metric weight relative to ability metric weight of 1.0
   1.1.4. (for purpose = 'blueprint') blueprint weight relative to other elements

1.2. rcabilityweight –
   1.2.1. Info weight with regards to reporting category (w_1 in main formula)
   1.2.2. rc (Reporting Category) ability weight

1.3. abilityweight (purpose = "blueprint") –
   1.3.1. Blueprint info weight (from tblAdminStrand and AffinityGroup tables, can be used for q_k in main formula)
   1.3.2. Called "scalar" in legacy system – relative ability strand weight

1.4. abilityweight (purpose = "scalar")
   1.4.1. Overall info weight (w_0 in main formula; from itembank.tblsetofadminsubjects table)
   1.4.2. Explicit ability weight for future monkey business
   1.4.3. ability weight

1.5. itemweight
   1.5.1.Relative weight of the BP-match metric with regards to the ability metric for the purposes of pruning items from the selected item group
   1.5.2. Weight of blueprint satisfaction relative to ability when pruning items from final itemgroup selected
   1.5.3. blueprint weight of items in chosen group relative to ability match for final pruning step

1.4.2 "Ability" parameters

1.1. startability –
   1.1.1. Starting theta if no student-related ability data is available
   1.1.2. start ability

1.2. startinfo –
   1.2.1. Starting info value. Used in ability match calculation.
   1.2.2. start information

1.3. abilityoffset –
   1.3.1. A spurious value to offset the ability estimate by
   1.3.2. offset for ability estimate

1.4. slope, intercept –
   1.4.1. Linear parameters which we have to use to transform the ability value(s) regardless of whether they came from the current or prior year.
   1.4.2. slope
   1.4.3. intercept

1.5. computeabilityestimates – compute ability estimates

1.4.2.1 Functionality

When starting a new test opportunity, the start ability value is currently pulled from one of the following sources, in order, until a value is found.

1) Simulation ONLY: A testee attribute of TDS_ID = 'INITIALABILITY' can be added for the testee.
2) The overall theta score from a previous opportunity on the same test (latest scored)
3) The overall theta score from an opportunity for the same subject but different test (latest scored)
4) The initial ability value from a test in the previous year, same subject. The max initial ability value will be taken. The value will be multiplied by the slope and added to the intercept, both values defined in configs.client_testproperties table for the current testid.
5) The initial ability value in the item bank for the segment.

Note:

- The first case is only used by the Simulator
- #3-4 will only be executed if configs.client_testproperties.initialabilitybysubject = 1.
- #2 is handled by setting useforability = 0|1 for those testid, measures in the TDS configs database.
- The current code need to check for a test within the same subject.
- With regards to #4, slope and intercept values are currently only applied if the test came from the prior year.
- #5 is the startinfo from the item bank.
- #3. There is a field to configs.client_testproperties called initialabilitytestid (varchar(100) null).
  - If that field's populated, we won't look for other tests in the same subject; instead, we'll look only for other tests of this testid (current year and prior year if necessary). We'll disregard the configs.client_testproperties.initialabilitybysubject in this case, even if it's set to 1. The abilityslope and abilityintercept used to transform the ability value(s) regardless of whether they came from the current or prior year.
  - If the initialabilitytestid is not populated, there will be no changes from the current approach. configs.client_testproperties.initialabilitybysubject will dictate whether or not we look by subject, same as now.

See also `ItemSelectionDLL.FN_GetInitialAbility_FN()` method.

1.4.2.2 How these parameters are used

1) Download parameters from test_specification.xml file to Blueprint class members.

2) Ignore the *computeabilityestimates* parameter.
3) Upload Student History and load initialbility from session.testopportunity table.
4) If initialbility value is not -9999, populate Blueprint.startability and Blueprint.theta (start theta for this test) to this value. Otherwise Blueprint.startability and Blueprint.theta will have item selection algorithm values.
5) For each Reporting Category if rc.getStartAbility() == rc.theta we populate Reporting Category start ability to this value.

1.4.3 "Sets" parameters

1.1. cset1size –

1.1.1. Size of Set1

1.1.2. number of top-ranked itemgroups by blueprint satisfaction to send on to ability match stage

1.1.3. cset 1 size

1.2. cset1order – the abilityweight control parameter.

1.2.1. When selection metrics are normalized and if cset1order = "DISTRIBUTION".

1.2.1.1. The ability metric is given a weight of 0.

1.2.1.2. The rcabilityweight = 0.

1.2.1.3. Note that the normalization routine has also been modified to factor in the rcabilityweight.

1.2.2. Otherwise, the weight is explicitly set to 1 (default cset1order value is "ABILITY"). Has to use the value in tblsetofadminsubjects.abilityweight instead of a static 1 if cset1order != "DISTRIBUTION

1.2.3. cset 1 tie breaker

1.3. cset2random –

1.3.1. Size of Set2 (Its fortune: cset2random => randomizer => randomizerindex => "cSet2Size = (randomizerIndex <= cSet1Size)? randomizerIndex : cSet1Size;")

1.3.2. randomizer for item selection

1.4. cset2initialrandom –

1.4.1. Random number generator initial number (Its fortune: cset2initialrandom => initialrandom => randomizerInitialIndex)

1.4.2. randomizer for first item selection

1.4.4 "Precision" parameters

1.1. precisiontarget –

1.1.1. Target overall information

1.1.2. precision target

1.2. precisiontargetmetweight –

1.2.1. Information weight when the overall precision target has been hit.

1.2.2. precisionTargetMetWeight

1.3. precisiontargetnotmetweight –
    1.3.1. Information weight when the precision target has not yet been hit
    1.3.2. precisionTargetNotMetWeight
1.4. adaptivecut –
    1.4.1. The overall score cutscore, usually proficiency, used in consideration of terminationTooClose (see below)
    1.4.2. adaptive cut point
    1.4.3. NOTE(!): REQUIRED PARAMETER
1.5. toocloseses –
    1.5.1. The value of standard errors below which the difference is considered “too close” to the adaptiveCut to proceed. In general, this will signal proceeding to a final segment that contains off- grade items. Used with terminationTooClose (see below)
    1.5.2. too Close SEs

1.4.5 “Termination” parameters

1.6. terminationoverallinfo – whether to use the overall information target as a termination criterion
1.7. terminationrcinfo – whether to use reporting category information target as a termination criterion
1.8. terminationmincount – whether to use minimum test size as a termination condition
1.9. terminationtooclose – terminate if you are not sufficiently distant from the specified adaptive cut
1.10. terminationflagsand – whether the other termination conditions are to be taken separately or conjunctively

1.4.6 “Min/Max” algorithm parameters

1.1. minitems –
    1.1.1. Minimum number of the operational items in the test as default
    1.1.2. minimum number of operational items for a test to be complete
1.2. maxitems –
    1.2.1.Maximum number of the operational items in the test as default
    1.2.2. maximum number of operational items allowed on a test
1.3. ftminitems – Minimum number of the field test items in the test as default
1.4. ftmaxitems – Maximum number of the field test items in the test as default
1.5. ftstartpos –
    1.5.1. Start position for the field test items if they exist in the test
    1.5.2. first field test item position
1.6. ftendpos –
    1.6.1. End position for the field test items if they exist in the test
    1.6.2. last field test item position

1.4.7 “Common” parameters

1.1. bpmetricfunction – TDS default value is ‘bp1’. Don’t change value.

1.2. selectionalgorithm – TDS default value is ‘adaptive2’. Don’t change value.

1.3. isstrictmax –

1.3.1. Flag for Item Selection Adaptive Algorithm try to satisfy blueprint maximum conditions exactly

1.3.2. strict maximum flag

1.4.8 “OffGrade” parameters

1.1. offGradeProbAffectProficiency – 'Probability that introducing off-grade items will influence the student proficiency must be less than this value'

1.2. offGradeMinItemsAdministered – 'Minimum number of on-grade operational items administered before considering for off-grade items'

1.3. proficientPLevel – 'Level at which student is considered proficient for the test'.

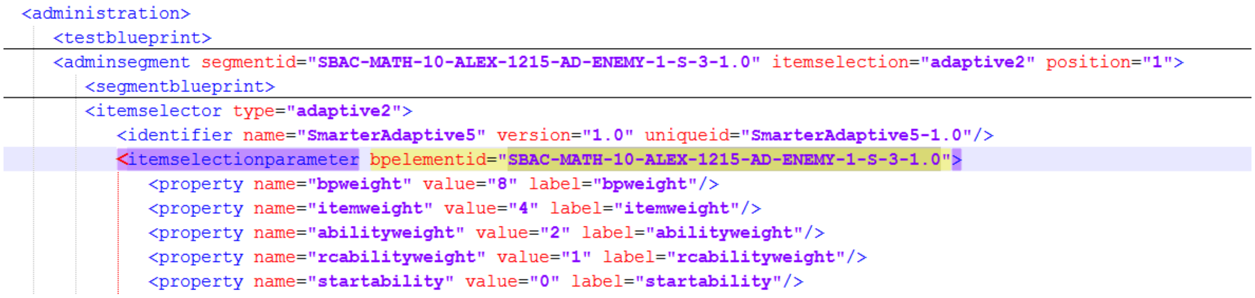
2 Additional Parameter Description Spreadsheet

Refer to the companion spreadsheet called TestAuthoring_AA_parameters.xlsx, located in the Test Authoring documentation folder. The first spreadsheet tab contains all parameters, names and default values. The second tab “Tracks”, one can see the path which every parameter has from test_specification.xml file to JAVA class member. The tracking begins from Test Authoring >> Settings >> Item Selection Algorithm parameters set.

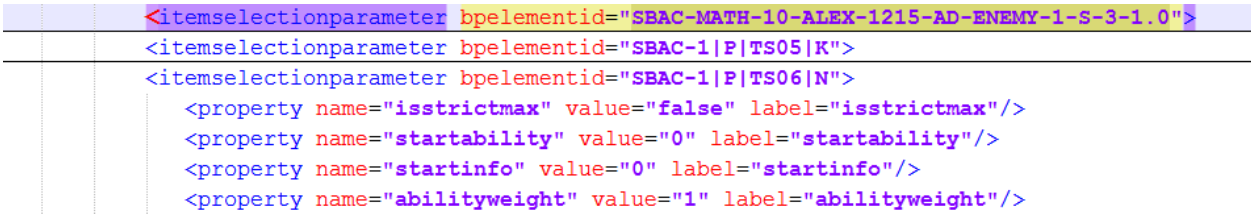
Test Authoring code writes these parameters in test_specification.xml file in the elements *testspecification.administration.adminsegment.itemselector.itemselectionparameter* with

- *bpelementid* = segmentid for ‘scalar’ purpose parameters
    - Stored procedure ‘loader_segmentitemselectionproperties’ loads name, value and label from xml file to loader_segmentitemselectionproperties table then stored procedure ‘load_adminsubjects’ loads parameters in tblsetofadminsubject table, and
- *bpelementid* = strandid, rcid or affinitygroupid for ‘blueprint’ purpose parameters
    - stored procedure ‘loader_segmentitemselectionproperties’ loads name, value and label from xml file to loader_segmentitemselectionproperties table then stored procedure ‘load_adminstrands’ loads parameters in tbladminstrand table and stored procedure ‘load_affinitygroups’ loads parameters in affinitygroup table

      Below is an example of these elements



```
<administration>
   <testblueprint>
   <adminsegment segmentid="SBAC-MATH-10-ALEX-1215-AD-ENEMY-1-S-3-1.0" itemselection="adaptive2" position="1">
      <segmentblueprint>
      <itemselector type="adaptive2">
         <identifier name="SmarterAdaptive5" version="1.0" uniqueid="SmarterAdaptive5-1.0"/>
         <itemselectionparameter bpelementid="SBAC-MATH-10-ALEX-1215-AD-ENEMY-1-S-3-1.0">
            <property name="bpweight" value="8" label="bpweight"/>
            <property name="itemweight" value="4" label="itemweight"/>
            <property name="abilityweight" value="2" label="abilityweight"/>
            <property name="rcabilityweight" value="1" label="rcabilityweight"/>
            <property name="startability" value="0" label="startability"/>
```



```
<itemselectionparameter bpelementid="SBAC-MATH-10-ALEX-1215-AD-ENEMY-1-S-3-1.0">
<itemselectionparameter bpelementid="SBAC-1|P|TS05|K">
<itemselectionparameter bpelementid="SBAC-1|P|TS06|N">
   <property name="isstrictmax" value="false" label="isstrictmax"/>
   <property name="startability" value="0" label="startability"/>
   <property name="startinfo" value="0" label="startinfo"/>
   <property name="abilityweight" value="1" label="abilityweight"/>
```