SGP Calculation Using Existing Coefficient Matrices

12 June 2013

This report presents and explains the R code used to (re)produce Student Growth Percentiles (SGPs) using the coefficient matrices that were constructed in the original analyses conducted from 2010 to 2012. These methods can be useful for producing SGPs for students whose data were not included in the original analyses. It can also be used to investigate the impacts of certain model and data specifications, such as the number of priors (data panels) included in the model. In a similar vein, the concept and calculation of baseline referenced SGPs are also introduced using beta-versions of baseline coefficient matrices.

1 Introduction

This report provides details on the procedure used to produce and/or reproduce student growth percentiles (SGPs) using previously computed coefficient matrices. This process differs from the typical production of SGPs in that only a small subset of the data (down to a single student's test score history) is required. This is possible only after the *entire* cohort data set has been used to produce the annual norm referenced values and related coefficient matrices. The 2011 - 2012 academic year data provided by the College Ready Promises (TCRP) consortium members and the Los Angeles Unified School District (LAUSD) was first analyzed in November 2012. After that initial analysis, several of the consortium members have requested and received training on how to reproduce their SGPs or use the existing coefficient matrices to produce norm referenced SGPs for students whose data was not available at the time of the initial data submission. A desire to investigate the impact that the number of priors used might have on both individual student SGPs and teacher median SGPs has also been expressed.

Furthermore, other members have requested training on using these annual cohort referenced matrices to produce baseline referenced SGPs rather than wait for the entire cohort data to be submitted by other members and LAUSD. Although a single year baseline is a possibility, it would be advisable to produce baseline coefficient matrices that use multiple years' data. With the 2013 data submission, there will be enough data to produce matrices that incorporate three annual data panels with two prior years into single *super-cohorts* (three panels of at least three years' data). To this end, a *beta-version* of the TCRP/LAUSD baseline referenced coefficient matrices are introduced and further discussion of the concept of the baseline referenced growth is provided below.

2 Data

Current data for the California Standards Tests (CST) program were supplied by the TCRP members and LAUSD to the National Center for Improvement of Educational Assessment (NCIEA) for analysis in the fall of 2012. The current dataset includes academic years 2008-2009 through 2011-2012.

2.1 Longitudinal Data

Growth analyses on assessment data require data to be available for individual students over time. Student growth percentile analyses require, at a minimum two, and preferably three years of assessment data for analysis of student progress. To this end it is necessary that a unique student identifier be available so that student data records across years can be merged with one another and subsequently examined. Because some records in the assessment data set contain students with more than one test score in a content area in a given year, a process to create unique student records in each content area by year combination was required in order to carry out subsequent growth analyses. The elimination of duplicate records can be accomplished by selecting one of the multiple records based upon a decision rule. Following consultation with the TCRP members, the following data cleaning rules were implemented:

1. When multiple scores exist in the same year and content area (e.g., ELA) the record with the highest score is retained.
2. Records with a *GRADE* value outside of tested grades were invalidated.
3. Records from assessment programs other than the CST were invalidated (e.g. CMA, STS, etc.).
4. Records with missing scores or scores outside the range of valid scores were invalidated.

The data used in this tutorial is an anonymized, random subset of LAUSD student data that has been cleaned in the manner described above. The data is not available for use by the consortium members, but it is a good representation of what a CMO-level dataset should look like for use in this tutorial. Furthermore, a second R script is available in the tutorial directory that mirrors the one with the TCRP specific code, but uses a subset of the toy data included in the SGP package. The required data objects are also available for that tutorial as well (with “DEMO” rather than “TCRP” in the file names).

3 Analysis and R Code

The following sections document the R statistical software code used to produce cohort and baseline referenced SGPs. Along with the *.R code scripts, there are several *.Rdata files in the tutorial directory that are required. Access to longitudinal student data is also required, although a second set of scripts is available that use the DEMO toy data set from the SGP package. Familiarity with both the R programming language and the SGP package are assumed based of previous trainings and data analytic experience.

3.1 Required R Packages and Data Objects

Set the working directory (may be different than the one provided in the code), load the required R packages, pre-constructed data objects (coefficient matrices, etc.).

```
require(SGP)
require(data.table)

load("Coefficient_Matrices/TCRP_Cohort_Matrices.Rdata")
load("Coefficient_Matrices/TCRP_Baseline_Matrices.Rdata")

load("SGP_CONFIG/TCRP_SGP_Norm_Group_Preference.Rdata")
```

Read in the raw longitudinal data text files (student test data and teacher-student links). The teacher linking data is not required to produce SGPs. It is used in producing summary tables.

```
MY_CMO_Data_LONG <- data.table(read.table(
        file='Data/Base_Files/TCRP_Training_Data.txt', sep='|', header=TRUE),
        key=c("VALID_CASE", "CONTENT_AREA", "YEAR", "ID"))
MY_TEACHER_STUDENT_LINKS <- data.table(read.table(
        file='Data/Base_Files/MY_TEACHER_STUDENT_LINKS.txt', sep='|', header=TRUE),
        key=c("ID", "CONTENT_AREA", "YEAR"))
```

In the past several months we have changed how we deal with ‘end of course tests’ (EOCTs). We have incorporated features into the source code that were previously done in an ad hoc fashion outside of the package and its functions. Many of these workarounds were included in the original TCRP analysis source code and the code provided to CMOs that requested individual training and consultation on (re)producing their SGPs.

In the current version of the SGP package, the GRADE values for student records associated with these courses now need to be set to ‘EOCT’. The chunk of code below makes this change to the appropriate subject records only. However, we want to retain the original reported grade information for summarization, so we create a new variable named GRADE_REPORTED before changing GRADE. Inspect the cross tab of grades and content area after changing it with the table(...) function.

```
MY_CMO_Data_LONG$GRADE_REPORTED <- MY_CMO_Data_LONG$GRADE

MY_CMO_Data_LONG$GRADE[MY_CMO_Data_LONG$CONTENT_AREA %in% c("WORLD_HISTORY",
    "US_HISTORY", "GENERAL_MATHEMATICS", "ALGEBRA_I", "ALGEBRA_II", "GEOMETRY",
    "SUMMATIVE_HS_MATHEMATICS", "EARTH_SCIENCE", "INTEGRATED_SCIENCE_1", "LIFE_SCIENCE",
    "BIOLOGY", "CHEMISTRY", "PHYSICS")] <- "EOCT"

table(MY_CMO_Data_LONG$CONTENT_AREA, MY_CMO_Data_LONG$GRADE)[, c(3:10, 1, 2,
    11)]
```

```

                          2    3    4    5    6    7     8     9   10
  ALGEBRA_I               0    0    0    0    0    0     0     0    0
  ALGEBRA_II              0    0    0    0    0    0     0     0    0
  BIOLOGY                 0    0    0    0    0    0     0     0    0
  CHEMISTRY               0    0    0    0    0    0     0     0    0
  ELA                  1479 1460 1508 3736 4930 7733 11517 10857 7798
  GENERAL_MATHEMATICS     0    0    0    0    0    0     0     0    0
  GEOMETRY                0    0    0    0    0    0     0     0    0
  HISTORY                 0    0    0    0    0    0 11757     0    0
  INTEGRATED_SCIENCE_1    0    0    0    0    0    0     0     0    0
  LIFE_SCIENCE                0    0    0    0    0    0     0     0    0
  MATHEMATICS              1479 1462 1510 3747 4929 7005     0     0    0
  PHYSICS                     0    0    0    0    0    0     0     0    0
  SCIENCE                     0    0    0 3754    0    0 11525     0    0
  SUMMATIVE_HS_MATHEMATICS    0    0    0    0    0    0     0     0    0
  US_HISTORY                  0    0    0    0    0    0     0     0    0
  WORLD_HISTORY               0    0    0    0    0    0     0     0    0

                             11  EOCT
  ALGEBRA_I                   0 14271
  ALGEBRA_II                  0  4954
  BIOLOGY                     0  9472
  CHEMISTRY                   0  4710
  ELA                      3781     0
  GENERAL_MATHEMATICS         0  4755
  GEOMETRY                    0  7760
  HISTORY                     0     0
  INTEGRATED_SCIENCE_1        0  3833
  LIFE_SCIENCE                0  1424
  MATHEMATICS                 0     0
  PHYSICS                     0  1644
  SCIENCE                     0     0
  SUMMATIVE_HS_MATHEMATICS    0  2290
  US_HISTORY                  0  3832
  WORLD_HISTORY               0  8236
```

3.2 SGP Object Preparation

Following initial data inspection and cleaning the prepareSGP function is used to create the initial SGP object.

```
MY_CMO_SGP <- prepareSGP(MY_CMO_Data_LONG, state='CA',
        data_supplementary = list(INSTRUCTOR_NUMBER = MY_TEACHER_STUDENT_LINKS))
```

```
Started prepareSGP Tue Jun 11 07:34:07 2013

NOTE: ID in @Data converted from class numeric to character to accommodate
      data.table >= 1.8.0 changes.
NOTE: VALID_CASE in @Data converted from class factor to character to
      accommodate data.table >= 1.8.0 changes.
NOTE: CONTENT_AREA in @Data converted from class factor to character to
      accommodate data.table >= 1.8.0 changes.
NOTE: YEAR in @Data converted from class integer to character to
      accommodate data.table >= 1.8.0 changes.
NOTE: GRADE in @Data converted from class integer to character to
      accommodate data.table >= 1.8.0 changes.
NOTE: Added variable HIGH_NEED_STATUS to @Data.
NOTE: Added variable STATE_ENROLLMENT_STATUS to @Data.
NOTE: Added variable DISTRICT_ENROLLMENT_STATUS to @Data.

Finished prepareSGP Tue Jun 11 07:34:14 2013 in 6.747sec
```

This creates a SGP class object named MY_CMO_SGP and populates its @Data, @Data_Supplementary, @Names, and @Version slots. Typically this is the point at which the data analysis truly begins with a call to the analyzeSGP function. However, for our purposes we will first need to make two sets of coefficient matrices available to the function. The first is the previously calculated cohort referenced coefficient matrices. These will be embedded within the SGP object itself in the (currently empty) @SGP slot. The second is the *beta-version* baseline referenced coefficient matrices, which will be embedded in the SGPstateData object. This is a meta-data object that is a part of the package. It contains various pieces of assessment program information and data components that are needed to compute SGPs. In the future the "official" baseline matrices may be included in the SGPstateData and ***this step will no longer be required***.

```
MY_CMO_SGP@SGP[["Coefficient_Matrices"]] <-
        TCRP_Cohort_Matrices
SGPstateData[["CA"]][["Baseline_splineMatrix"]][["Coefficient_Matrices"]] <-
        TCRP_Baseline_Matrices
```

3.3 Cohort and Baseline SGP Computation

3.3.1 *Grade-Level Analyses*

Now we are ready to run the `analyzeSGP` function. We are using the SGP object named `MY_CMO_SGP` and have indicated that the function should reference the California ("CA") entry in the `SGPstateData` object. In this first step we are only analyzing grade-level ELA and Math. These are the only two subjects with "typical" analyses. That is, they have grade specific tests and the priors used have the *exact* same content area designation in the longitudinal data. Next we specify that we will reproduce SGPs for the 2012 data only in this tutorial and give the range of `grades` to be used.

The next six arguments to the function are the crux of this tutorial. The `sgp.*.*` arguments specify which of the student growth percentile and projection analyses to run. They all default to TRUE, so if you do not want them they must be set to FALSE here. We will be producing both cohort and baseline SGPs as well as "lagged" projections, which are used to determine ***adequate*** growth (the SGP projected to bring a student to the proficiency cut-point or maintain proficiency within a certain time frame). Note that if the baseline referenced coefficient matrices are not embedded in the `SGPstateData` object but one requests the baseline SGPs be produced, the function will attempt to build the matrices itself with the data that is available. That could be a **bad** idea here because the only data we have is a small subset of the full cohort data and generally we want as big of a "super-cohort" as possible to have a meaningful baseline with which to compare.

The last argument, `sgp.use.my.coefficient.matrices = TRUE`, is where we tell the function that we already have cohort referenced coefficient matrices available that we want it to use rather than going through the time and computationally intensive process of computing them (or in this case, the *impossible* task since the full dataset is not available). If we didn't specify this argument the function would produce its own coefficient matrices using the subset of the data as a norm group. Although this may be of interest in other contexts, it is not what we want here.

```
MY_CMO_SGP <- analyzeSGP(MY_CMO_SGP, state='CA',
                         content_areas=c("ELA", "MATHEMATICS"),
                         years='2012',
                         grades = 2:11,
                         sgp.percentiles = TRUE,
                         sgp.projections = FALSE,
                         sgp.projections.lagged = TRUE,
                         sgp.percentiles.baseline = TRUE,
                         sgp.projections.baseline = FALSE,
                         sgp.projections.lagged.baseline = FALSE,
                         simulate.sgps = FALSE,
                         sgp.use.my.coefficient.matrices=TRUE,
                         parallel.config=list(
                            BACKEND="PARALLEL",
                            WORKERS=list(
                               PERCENTILES=4,
                               BASELINE_PERCENTILES=4,
                               LAGGED_PROJECTIONS=4)))
```

3.3.2 *Heterogeneous grade-level and Middle School EOCT progression analyses*

Now we move on to the more complex analyses that use various heterogeneous content area course progressions. These include grade-level Science and History, as well as the subject specific "End-of-Course" tests (EOCTs) that exclusively use grade level ELA and Math tests as priors. Science and History courses use grade-level Math and ELA respectively as priors. For example, if a student took either Math or Algebra I in 7th grade then we will use that as a prior for 8th grade Science. Along with these two subjects, many middle school students take Algebra I or General Math in middle school, and we use grade level Math as priors for them.

Because of this heterogeneous `CONTENT_AREA` progression, we must explicitly specify the `sgp.config` argument in `analyzeSGP`. In the section above (homogeneous course progressions), the `analyzeSGP` function can produce these lists of analyses internally. The `sgp.config` argument is a compound list object. At the ultimate level of the list(s) are the individual subject/year/course progression analyses configurations. These lists have three **mandatory** components. Basically, these configurations are used internally to select the appropriate cases that will be passed on to the lower level package functions like `studentGrowthPercentiles`.

1. **`sgp.content.areas`** - A character string giving the specific course progression. The progression is given in the convention `c('OLDEST_PRIOR_COURSE', 'SECOND_OLDEST_PRIOR_COURSE', ..., 'CURRENT_COURSE')`.
2. **`sgp.panel.years`** - A string giving the academic year (time) progressions. For example, `c('2009', ..., '2012')`.
3. **`sgp.grade.sequences`** - A character string *nested within a list* that gives the grade sequence(s) to be used. In the case of the grade level courses in this section, this element may look something like this: `sgp.grade.sequences = list(c('6', '7', 'EOCT'))` were grades 6 and 7 Math are used as priors for `ALGEBRA_I` or `SCIENCE` (or likely both).

Other elements of these lists are used here, but are not necessary in all circumstances:

- *`sgp.exact.grade.progression`* - Boolean, `FALSE` by default if left out. If `TRUE`, `analyzeSGP` will only produce SGPs for students that have *at least* the grade and subject sequence provided. If `FALSE` all possible subsets of the grade progression would also be analyzed.

- *sgp.norm.group.preference* - This is used in the construction of the `SGP_Norm_Group_Preference` object. This establishes an order (lowest being the **MOST** preferred norm group). Generally the norm group with the most priors available is the preferred one. However, in the case of multiple course progressions (e.g. a student repeats Algebra I in 2012, but also falls into a second Algebra I progression because they have a 2011 Grade 7 Math score as well) a business rule decision is required. The preference in this case is generally given to the repeat course progression.

The lists we will use here are created by sourcing in the scripts located in the SGP_CONFIG/ directory of these training documents. We now load ALL 2012 configuration lists and create the grade-level specific `sgp.config` list by concatenating the sub-lists together into a single object called `TCRP_Grade_Level.config`.

```
source("SGP_CONFIG/EOCT/2012/MATHEMATICS.R")
source("SGP_CONFIG/EOCT/2012/SCIENCE.R")
source("SGP_CONFIG/EOCT/2012/SOCIAL_STUDIES.R")

TCRP_Grade_Level.config <- c(
        HISTORY_2012.config,
        SCIENCE_2012.config,
        ALGEBRA_I_MS_2012.config,
        GENERAL_MATHEMATICS_MS_2012.config
)
```

Now we are ready to run `analyzeSGP` using our custom config list. Note that now we no longer need the `years` `content_areas` or `grades` arguments. We are still specifying `sgp.use.my.coefficient.matrices = TRUE` so the function will need to find a coefficient matrix in the @SGP slot whose meta-data matches the sgp.config elements ***EXACTLY***. The matrix meta data *can* be changed to make things work with your config if need be (that is what we have done in preparation for this training - updating old matrices' meta data to work with the new conventions incorporated in the package).

```
MY_CMO_SGP <- analyzeSGP(MY_CMO_SGP, state='CA',
                         sgp.config = TCRP_Grade_Level.config,
                         sgp.percentiles = TRUE,
                         sgp.projections = FALSE,
                         sgp.projections.lagged = FALSE,
                         sgp.percentiles.baseline = TRUE,
                         sgp.projections.baseline = FALSE,
                         sgp.projections.lagged.baseline = FALSE,
                         simulate.sgps = FALSE,
                         sgp.use.my.coefficient.matrices = TRUE,
                         parallel.config=list(
                            BACKEND="PARALLEL",
                            WORKERS=list(
                               PERCENTILES=4,
                               BASELINE_PERCENTILES=4)))
```

3.3.3 *High School EOCT course progression analyses*

As students move into and through high school, the permutations of possible course progressions expands greatly! Concurrently there is an expansion is the number of configurations needed to produce SGPs for them. In order to maximize the number of students that would receive an SGP (and form the largest cohort possible) it was decided that the recorded grade levels for these students would be ignored. That is, in these analyses a student's specific course taking sequence is used as the progression of interest rather than grade to grade matriculation. In doing so we are comparing ALL students that had a similar course progression (e.g. ALL students that took Geometry (2011 prior) and then Chemistry (2012 current), rather than separating 9th and 10th graders from the 10th and 11th graders that have the same course sequence). Although ELA and Math are not analyzed in this fashion, we convert high school level ELA and middle school Math records' `GRADE` values to 'EOCT'. This is consistent with how US History and World History analyses were conducted in 2010-12. This may be something that TCRP considers changing in 2013 and going forward.

Change all `GRADE` values of the ELA and Math priors to 'EOCT'

```
MY_CMO_SGP@Data$GRADE[MY_CMO_SGP@Data$CONTENT_AREA %in% c('ELA', 'MATHEMATICS')] <- 'EOCT'
```

and establish the second custom configuration list.

```
TCRP_EOCT.config <- c(
        ALGEBRA_I_2012.config,
        ALGEBRA_II_2012.config,
        BIOLOGY_2012.config,
        CHEMISTRY_2012.config,
        GENERAL_MATHEMATICS_2012.config,
        GEOMETRY_2012.config,
        INTEGRATED_SCIENCE_1.2012.config,
        LIFE_SCIENCE_2012.config,
        PHYSICS_2012.config,
        SUMMATIVE_HS_MATHEMATICS_2012.config,
        US_HISTORY_2012.config,
        WORLD_HISTORY_2012.config
)
```

Now we are again ready to run `analyzeSGP` using our new config list. Note that the named list object provided to the `sgp.config` argument is the only thing that has changed from the code above.

```
MY_CMO_SGP <- analyzeSGP(MY_CMO_SGP, state='CA',
                        sgp.config = TCRP_EOCT.config,
                        sgp.percentiles = TRUE,
                        sgp.projections = FALSE,
                        sgp.projections.lagged = FALSE,
                        sgp.percentiles.baseline = TRUE,
                        sgp.projections.baseline = FALSE,
                        sgp.projections.lagged.baseline = FALSE,
                        simulate.sgps = FALSE,
                        sgp.use.my.coefficient.matrices = TRUE,
                        parallel.config=list(
                           BACKEND="PARALLEL",
                           WORKERS=list(
                              PERCENTILES=4,
                              BASELINE_PERCENTILES=4)))
```

For reporting/summarization purposes, we will return the `GRADE` variable to its original form:

```
MY_CMO_SGP@Data[["GRADE"]] <- MY_CMO_SGP@Data[["GRADE_REPORTED"]]
MY_CMO_SGP@Data[["GRADE_REPORTED"]] <- NULL
```

3.4 Merge SGPs Into the Longitudinal Data

```
SGPstateData[["CA"]][["SGP_Norm_Group_Preference"]] <- TCRP_SGP_Norm_Group_Preference

MY_CMO_SGP <- combineSGP(MY_CMO_SGP, state = "CA")
```

```
Started combineSGP Wed Jun 12 09:07:41 2013
```

```
NOTE: Multiple SGPs exist for individual students. Unique SGPs will be
created using SGP Norm Group Preference Table for CA.
```

```
NOTE: Multiple Baseline SGPs exist for individual students. Unique
Baseline SGPs will be created using SGP Norm Group Preference Table for
CA.
```

```
NOTE: No SGP lagged baseline projections available in SGP slot. No
baseline referenced student growth projection targets will be produced.
Finished combineSGP Wed Jun 12 09:07:44 2013 in 2.025sec
```

4 Concepts and Advanced Topics

4.1 Cohort Referenced Student Growth Percentiles

Growth percentiles, being quantities associated with each individual student, can be easily summarized across numerous grouping indicators to provide summary results regarding growth. Being state (or in this case large urban/regional) normed percentiles, across all students, with perfect data fit the median of all student growth percentiles in any grade and subject is 50. The median of a collection of growth percentiles is used as the measure of central tendency to summarize the distribution as a single number. Median growth percentiles well below 50 represent growth less than the state "average" and median growth percentiles well above 50 represent growth in excess of the institutional (or teacher) "average".

To demonstrate the normative nature of the growth percentiles viewed at the CMO level, Table 1 presents growth percentile medians by grade level and subject.

Table 1: Median *Cohort* Student Growth Percentile by Grade, Content Area and Year

	Grades								
Content Area	3	4	5	6	7	8	9	10	11
2012									
~~Algebra I~~					~~48~~	~~52~~	~~48~~	~~37~~	~~47~~
Algebra II						77	57.5	50	48
Biology							48	58	44
Chemistry							24	66	38
ELA	49	48	52	46	54	50	49	48	51
General Mathematics						50	53		
Geometry						59.5	48.5	46	44
History						49.5			
Integrated Science 1							51	40	34.5
Life Science								38	
Mathematics	50	54	50	48	46.5				
Physics							53	72	47
Science			51			50			
Summative HS Mathematics							53	50.5	49
US History									51
World History							47	49	52

Based upon perfect model fit to the data, the median of **all** state growth percentiles in each grade by year by subject combination should be 50. That is, in the conditional distributions, 50 percent of growth percentiles should be less than 50 and 50 percent should be greater than 50. At the state/system level, deviations from 50 indicate imperfect model fit to the data. Imperfect model fit can occur for a number of reasons, some due to issues with the data (e.g., floor and ceiling effects leading to a "bunching" up of the data) as well as issues due to the way that the SGP function fits the data. The results in Table 1 will not necessarily be perfect, however, because the data has been restricted to a random subset of a single district.

It is important to note how, at the state/system level, the *normative* growth information returns very little information. What the results indicate is that a typical (or average) student in the district demonstrates growth near the 50th percentile. That is, a "typical students" demonstrate "typical growth". The power of the normative results follows when subgroups are examined (e.g., schools, district, ethnic groups, etc.). Examining subgroups in terms of the median of their student growth percentiles, it is possible to investigate why some subgroups display lower/higher student growth than others. Moreover, because the subgroup summary statistic (i.e., the median) is composed of many individual student growth percentiles, one can break out the result and further examine the distribution of individual results.

4.2 Baseline Referenced Student Growth Percentiles

Baseline SGPs provide us with a way to look at normative growth through another lens. Rather than considering a single year's cohort, baseline SGPs are referenced against a "super-cohort" of several years of students linked by common course/grade progressions. This allows us to examine whether or not the system as a whole might be improving (or declining) over time relative to the established baseline. That is, if the system is improving over time, we would see that improvement the form of median SGPs that are greater than 50 (what *was* typical growth in the past would now be lower growth). A major assumption required here is that the scale scores are well anchored. If this assumption does not hold, then any deviation from "typical" growth may be purely an artifact of the test scaling procedure.

Table 2 provides the median Baseline referenced results from this test subset of data.

Table 2: Median *Baseline* Student Growth Percentile by Grade, Content Area and Year

	Grades								
Content Area	3	4	5	6	7	8	9	10	11
2012									
Algebra I					53	53	50	40	51
Algebra II						74	56.5	50	50
Biology							48	58	45
Chemistry							25	68	39
ELA	52	50	52.5	48.5	55	50	50	47	51

General Mathematics						50	49.5		
Geometry						66	44	45	42
History						49			
Integrated Science 1							52	42	37
Life Science								41	
Mathematics	51	52.5	48	45	47.5				
Physics							51	72	48
Science			51			52			
Summative HS Mathematics							55	49	48
US History									51
World History							45	49	52

4.3 Model and Data Specifications

The interpretation of a student growth percentile is always dependent on the data used to produce it. A student's change in test scores from one year to the next may seem impressive if taken in a simple pre-test post-test context and they would have a high SGP. However, if more data were available we might see that the immediate prior year was an anomalously low score and that the current year is merely typical. If the *entire, true score history* were available for that student we could obtain a very different normative description of their growth. This is a data (availability) issue, as are issues with accuracy and precision of SGP estimates due to measurement error. Model parameters and specifications, such as the choice of knots and boundary locations for the cubic polynomial B-spline basis functions, the estimation method used in the quantile regression calculations, isotonizing the regression curves to prevent quantile crossing (or not) etc., can also impact the interpretation and estimated value of an individual students growth.

These factors can, in varying degrees, telegraph into changes in aggregate measures such as teacher and school median SGPs. The extent of these changes is, however, greatly reduced by the aggregation of the individual measures. The effects usually tend to even out (mean effect of 0). Furthermore, the median is a robust measure of central tendency so a few extreme deviations that can occur when some feature of the model or data is changed will not cause a concomitant extreme deviation in the median.

4.3.1 *Number of priors used to produce SGPs*

In many cases, questions about the impact of model and data specifications are best answered through empirical investigation. One TCRP consortium member has asked whether the increasing availability of prior data has caused some of the variability in they are witnessing in teacher medians from one year to the next. The following code shows how one can produce SGPs for students using 1, 2 and 3 priors. The results can be analyzed in any number of ways. Those inquiries and investigations are presently left to the interested parties.

The process proceeds as follows. Add/change the `SGP_Configuration` element of the `SGPstateData` object to control the `max.order.for.percentile` argument in `studentGrowthPercentiles` and add the previously calculated TCRP Coefficient Matrices for 2012 to the SGP object as before to take advantage of the argument `sgp.use.my.coefficient.matrices=TRUE.`

```
TEST_SGP <- prepareSGP(MY_CMO_Data_LONG, state='CA', create.additional.variables = FALSE,
        data_supplementary = list(INSTRUCTOR_NUMBER = MY_TEACHER_STUDENT_LINKS))

TEST_SGP@SGP$Coefficient_Matrices <- TCRP_Cohort_Matrices

SGPstateData[["CA"]][["SGP_Configuration"]] <- list(max.order.for.percentile=1)

TEST_SGP <- analyzeSGP(TEST_SGP, state='CA',
                       content_areas=c("ELA", "MATHEMATICS"),
                       grades = 2:11,
                       years='2012',
                       sgp.projections = FALSE,
                       sgp.projections.lagged = FALSE,
                       sgp.percentiles.baseline = FALSE,
                       sgp.projections.baseline = FALSE,
                       sgp.projections.lagged.baseline = FALSE,
                       simulate.sgps = FALSE,
                       sgp.use.my.coefficient.matrices = TRUE)
```

Verify that all Norm Groups have only **ONE** prior, and then merge the results back in to the longitudinal `@Data` slot. Then rename the SGP related variables of interest so that we can compare them to others produced in subsequent runs below with different data specifications.

```
TEST_SGP <- combineSGP(TEST_SGP, state='CA')
setnames(TEST_SGP@Data, c('SGP', 'SGP_NORM_GROUP'), c('SGP_1_PRIOR', 'SGP_NORM_GROUP_1_PRIOR'))
```

We will now repeat the above step twice, setting the `max.order.for.percentile` argument to 2 and then 3. However, before we do that, we need to get rid of the first round of results located in the `@SGP` slot. We might not want to completely eliminate them, so here we first add them to a new element in the `@SGP` slot first and then NULL the 'official' `@SGP` SGPercentiles element.

```
TEST_SGP@SGP$SGPercentiles_1_Prior <- TEST_SGP@SGP$SGPercentiles
TEST_SGP@SGP$SGPercentiles <- NULL

SGPstateData[["CA"]][["SGP_Configuration"]] <- list(max.order.for.percentile=2)

TEST_SGP <- analyzeSGP(TEST_SGP, state='CA',
                       content_areas=c("ELA", "MATHEMATICS"),
                       grades = 2:11,
                       years='2012',
                       sgp.projections = FALSE,
                       sgp.projections.lagged = FALSE,
                       sgp.percentiles.baseline = FALSE,
                       sgp.projections.baseline = FALSE,
                       sgp.projections.lagged.baseline = FALSE,
                       simulate.sgps = FALSE,
                       sgp.use.my.coefficient.matrices = TRUE)

TEST_SGP <- combineSGP(TEST_SGP, state='CA')

setnames(TEST_SGP@Data, c('SGP', 'SGP_NORM_GROUP'), c('SGP_2_PRIOR', 'SGP_NORM_GROUP_2_PRIOR'))

TEST_SGP@SGP$SGPercentiles_2_Prior <- TEST_SGP@SGP$SGPercentiles
TEST_SGP@SGP$SGPercentiles <- NULL

SGPstateData[["CA"]][["SGP_Configuration"]] <- list(max.order.for.percentile=3)

TEST_SGP <- analyzeSGP(TEST_SGP, state='CA',
                       content_areas=c("ELA", "MATHEMATICS"),
                       grades = 2:11,
                       years='2012',
                       sgp.projections = FALSE,
                       sgp.projections.lagged = FALSE,
                       sgp.percentiles.baseline = FALSE,
                       sgp.projections.baseline = FALSE,
                       sgp.projections.lagged.baseline = FALSE,
                       simulate.sgps = FALSE,
                       sgp.use.my.coefficient.matrices = TRUE)

TEST_SGP <- combineSGP(TEST_SGP, state='CA')

setnames(TEST_SGP@Data, c('SGP', 'SGP_NORM_GROUP'), c('SGP_3_PRIOR', 'SGP_NORM_GROUP_3_PRIOR'))
```

There is now a dataset in the `@Data` slot with 3 different versions of the student SGPs using 1, 2 and 3 priors. There are any number of comparisons one can make about the differences in both student SGPs and teacher or school aggregations (Median SGPs). We leave those to your imagination, but the code includes a few descriptives for the student level to get you started.

5 References

Betebenner, D. VanIwaarden, A., Domingue B. and Shang, Y. (2013). SGP: An R Package for the Calculation and Visualization of Student Growth Percentiles & Percentile Growth Trajectories. (R package version 1.1-0.0. URL http://schoolview.github.com/SGP/

Betebenner, D. (2009). Norm- and criterion-referenced student growth. *Educational Measurement: Issues and Practice*, 28 (4), 42–51.

Betebenner, D. (2008). Toward a Normative Understanding of Student Growth. *In The Future of Test-based Educational Accountability*, in K. E. Ryan & L. A. Shepard (eds). Mahwah, NJ: Lawrence Erlbaum Associates.

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

6 R Session Information

This document was compiled with the following system specifications:

```
sessionInfo()
```

```
R version 3.0.1 (2013-05-16)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods  
[8] base     

other attached packages:
[1] plyr_1.8         stringr_0.6.2    data.table_1.8.8 SGP_1.1-7.0     
[5] SGPdata_7.0-0.0  knitr_1.2       

loaded via a namespace (and not attached):
 [1] Cairo_1.5-2      codetools_0.2-8  colorspace_1.2-2 DBI_0.2-7       
 [5] digest_0.6.3     doParallel_1.0.3 evaluate_0.4.3   foreach_1.4.1   
 [9] formatR_0.7      grid_3.0.1       gridBase_0.4-6   iterators_1.0.6 
[13] quantreg_4.98    reshape2_1.2.2   RJSONIO_1.0-3    RSQLite_0.11.4  
[17] sn_0.4-18        SparseM_0.99     splines_3.0.1    tools_3.0.1     
```

All contents Copyright 2013. All rights reserved. CSS adapted from Substance.io and Scott.Fortmann-Roe.com and modified.