Managing group random assignment in UpGrade

Steve Ritter
Carnegie Learning
sritter@carnegielearning.com

April Murphy
Carnegie Learning
amurphy@carnegielearning.com

Stephen Fancsali
Carnegie Learning
sfancsali@carnegielearning.com

ABSTRACT

Running randomized trials has become standard practice for many software developers [5], but it is much less common within educational software. One of the barriers to running experiments in educational contexts is that instruction often takes place in groups, and it is undesirable to have different students within a group be assigned to different conditions within the learning experiment. Thus, experiment assignment must take into account the student's membership in a group (e.g., a class, the set of students taught by a particular teacher, a school or a school district). The assignment of individuals to condition as a group is called "group random assignment" (sometimes called "cluster random assignment") [2]. Consistently managing group random assignment, particularly at scale, is a difficult task. We have developed the free and open source UpGrade platform to help manage this type of randomization.

Author Keywords

Field experimentation; A/B testing, group-randomization

INTRODUCTION

Statisticians have long recognized the need to analyze school-based interventions in a way that accounts for instruction in group settings [4]. The statistical argument is that the performances of individuals learning within a class are not independent; students are influenced by a common teacher, classmates and school environment. Group effects like these may exist even when educational technologies personalize learning for students.

Our concern in this paper is less about statistical analysis than about conditions that encourage us to provide common educational experiences to students within a group and which require experimental designs that assign all students in the group to the same treatment. We consider three main factors that encourage group assignment. The first is consistency for teachers. With many educational technologies, teachers assist students with their work and should reasonably expect to be prepared to provide assistance that is consistent with the approach given in the software. It would impose a burden on teachers if, for example, different students in their class received substantially different software experiences. This concern is especially prominent in cases where the manipulation is fundamental to instruction. If we were to test, for example, two different approaches to understanding how to divide fractions, we probably would want to present a consistent approach within a classroom, so that the teacher can provide similar guidance to all students. If the required teacher preparation is substantial (e.g., if teachers need to spend time learning the fraction division approaches), then we might want to ensure that all students of a particular teacher receive a similar approach, even if the students are in different classrooms (this would be assigning by teacher, not classroom). In contrast, if the intervention involved, for example, differing wording for motivational messages (e.g. [3]), it might be acceptable to randomly assign different conditions to each student within a class. A second factor to consider in group assignment is fairness. Students are often aware of what other students in their class are doing. If an experiment were to offer some students a game-based "brain break," students who were not assigned to the "fun" condition might consider the assignment to be unfair. Fairness might also be a consideration if the activity which is the focus of the experiment constitutes a significant portion of the student's grade or if the intervention could reasonably be expected to provide large and long-lasting differences between students in the different conditions. Finally, researchers should consider "spillover effects," the ways in which assignment to an experimental condition may affect students who are not assigned to that condition. An experiment asking some students within a class to self-explain might have the unintended effect of those students explaining their reasoning to friends in the classroom who are not in that condition. If both self-explanation and receiving such an explanation from a friend are effective educational interventions, the experiment might be judged to produce a smaller effect (since students in the control group receive some of the benefit of the intervention), even though the intervention, in fact, had a large effect.

Group random assignment will ordinarily reduce the statistical power of an experiment; more students are required to detect an effect of the same size if you assign by group than if you assign by student. The decision to assign by group (and, if so, by which grouping) can be complex. Our intention is not for UpGrade to make these decisions for the experimenter. But when group assignment is appropriate for an experiment, UpGrade will ensure that such assignment is carried out consistently.

GROUP RANDOM ASSIGNMENT AT SCALE

An experimental design that desires to assign students by group cannot always be carried out. To manage large scale

L@S '20, August 12–14, 2020, Virtual Event, USA © 2020 Copyright is held by the owner/author(s).

experiments, we need to define rules for how to deal with anomalies that prevent ideal group assignment. For example, many software programs are self-paced. Suppose we wish to determine the relative efficacy of two methods for teaching students to factor quadratics, and we wish to assign all students in a class to use a consistent method (see Figure 1). At the time the experiment starts, Student 1 has already completed the instructional module related to factoring quadratic equations, using the current method. If we want to ensure consistency in the classroom, we must then allow all students in the class to receive the current method. Since none of the students in the class would be randomly assigned, they cannot be included in the experiment. Strict adherence to consistency may thus reduce the size of the experiment, sometimes quite significantly: in an experiment assigning by district, a single student could force the entire district to be excluded from the experiment.

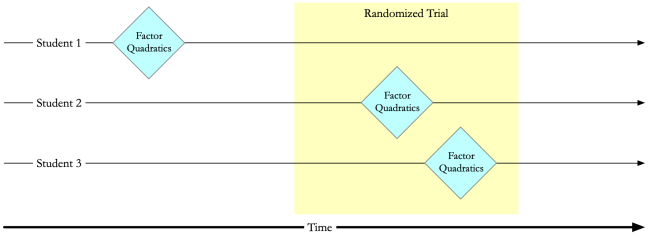

Figure 1: Three students in the same class reach the "Factor Quadratics" lesson at different times. Student 1 receives the current instructional approach, since the trial has not yet begun. If we require all students in the class to receive consistent instruction, we must also give other students in the class the current approach (and exclude them from the experiment). Alternatively, we might allow a violation of consistency and randomly assign Students 2 and 3 to a condition in the experiment.

Alternatively, a researcher might decide that violating group consistency is preferable to excluding the whole class from the experiment. We might assume, for example, that Student 1, who reached the quadratic factoring experiment early, is less likely to need the teacher's assistance, and so it is acceptable to allow that student's instruction to differ from the rest of the class. If that is the case, when Student 2 reaches "Factor Quadratics", we randomly assign the rest of the class to a condition. The decision about which approach to take involves a compromise between statistical power and the importance of class consistency (which might include considering whether Student 1 is likely to return to the "Factor Quadratics" lesson at a later time). Researchers might also need to consider whether either approach is likely to introduce bias as to which groups (and students) are assigned within the experiment. In a self-paced course, more proficient students reach more advanced topics earlier, so excluding classes containing such students could affect the generality of the results. Such decisions are complex and UpGrade cannot make these decisions for researchers, but it does allow them to choose an approach and implement the rule consistently.

Finally, researchers may need to consider situations where it is valid for students to be in multiple classes, either simultaneously or sequentially. If a student is in both class A and class B and we want to strictly enforce within-class consistency, then we must treat classes A and B as a single class for the purposes of assignment, even if only one student is a member of both. UpGrade currently has a simple mechanism for handling cases where a student may be a member of multiple groups. Its assumption is that only one group can be active for a student at any time, and assignment decisions are made with respect to that group. We are in the process of understanding use cases and desired assignment behavior for more complex multiple-group scenarios.

GROUP ASSIGNMENT RULES IN UPGRADE

What is a group?

The standard UpGrade configuration defines four types of groups relevant in K-12 education: district, school, teacher and class. However, researchers can define additional types of groups. For example, a researcher could assign all students who reside in the same dorm at a university to the same condition. Any grouping of students could be used, with the main requirement being that some external source must provide UpGrade with information about student membership within the groups.

Not all experiments will assign students by group; some might assign by individual. When experiments assign by group, they use a single type of group. For example, an experiment may assign students by class or by school but not both. Students may be in multiple experiments, each with different assignment rules.

Currently, UpGrade has minimal semantics for these groups. It simply knows that there are group types (class, school, etc.) and that students may be in one or more of these groups. UpGrade assumes that the source of group information is accurate and timely and that any new information about student group membership replaces previously-transmitted information. These are simplifying assumptions that handle many simple cases. A future version of UpGrade may require advanced group semantics in order to handle membership in multiple groups and changes in group membership.

We contemplate two kinds of sources for group membership information. When the student logs on to the application that is the subject of the experiment, data can be transmitted directly from that application to UpGrade indicating student group membership. This approach is simple, particularly since the educational application needs to interact with UpGrade anyway to find out the student's condition(s) for any experiments. However, in cases where group membership changes more frequently than use of the educational application, it is possible that UpGrade may make assignments based on out-of-date information. Consider Figure 2. Student 1 completes the lesson before the experiment begins. If we are strict about class consistency, Class A will be excluded from the experiment. If Student 1

then transfers from Class A to Class B, there is no reason to exclude Class A from the experiment. However, if group membership is being transmitted by the educational application (blue diamonds) and Student 1 does not log in again before the experiment starts, then UpGrade will not know about the student's current group membership and so will improperly exclude Class A (and not Class B) from the experiment.

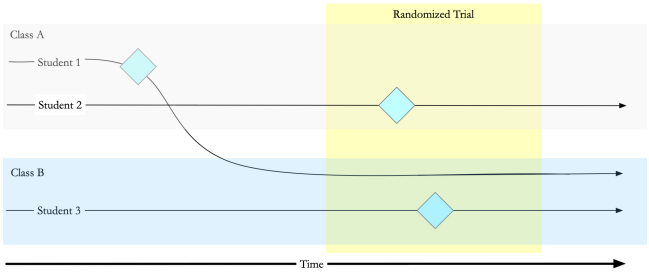

Figure 2: Student 1 completes the lesson in Class A before the experiment begins and then transfers to Class B.

If this kind of scenario is likely, UpGrade can be configured to receive group membership updates on a schedule that is not tied to usage of the educational application. For example, a rostering service (or intermediary, like Clever) can be used to provide nightly (or more frequent push) updates to UpGrade.

One of our requirements for UpGrade is that it not store personally-identifiable information; student IDs and group IDs are unique identifiers used only by UpGrade. This architecture increases security, but it means that data providers (both educational applications and sources of group membership information) map internal identifiers to anonymous UpGrade identifiers. For this reason, group membership updates are likely to be passed through an anonymizer as an intermediary before such updates are passed to UpGrade.

If a student is a member of more than one group of the same type (e.g. more than one class), experimental assignment is made consistent with the student's "working group." The idea of a "working group" is that, when the student uses the educational application, the student is acting as a member of one of the groups. In most cases, the educational software itself will specify the working group, not a rostering service (though it is also possible for the rostering service to provide this information).

Controlling consistency

UpGrade is designed to manage the kinds of complex decisions described above in a way that is consistent with the intent of the researcher. Our approach in designing UpGrade is to allow researchers to express their general intent with respect to the relative importance of keeping group experiences consistent versus keeping individual experiences consistent. We believe that expressing this desire as part of the experimental design will allow UpGrade to make appropriate choices in anomalous cases without the researcher having to think through all of the details of these cases. For each experiment, researchers define a "consistencyRule" parameter, which controls how to manage these decisions. If the consistencyRule is *group*, then consistency within a group is primary, and UpGrade will prefer to exclude groups from the experiment than to vary student experience within a group. In the situation shown in Figure 1, for example, we would exclude the class from the experiment (and provide all students with the current method). If, in contrast, the "consistencyRule" is *individual* then we value individual consistency over group consistency. Under this rule, Student 1 would continue to receive the current method of instruction, but the rest of the class would be randomly assigned. This means that Students 2 and 3 might be assigned to a condition that uses a different instructional method than the one used for Student 1. Under this rule, however, Student 1 would be excluded from the analysis of the experiment (even if random assignment resulted in the rest of the class receiving the current method), since Student 1 was not randomly assigned.

Individual consistency also controls UpGrade's response to situations where students change group membership. If the experiment is using *individual* consistency, then a student transferring from class A to class B would continue to see condition A. If the experiment is using *group* consistency, then the student would be switched to condition B upon transferring to class B. This student would be randomly assigned and so would be included in the experiment.

Anticipating experiments

As Figures 1 and 2 make clear, decisions regarding assignment to condition in group-randomized experiments may depend on what students do prior to the start of the experiment, not just during the period when the experiment is active. In fact, since an arbitrary amount of time may pass between the time Student 1 reaches the Factor Quadratics lesson and the time the experiment starts, it is perfectly plausible that the researcher has not even decided to run an experiment involving the Factor Quadratics lesson at the time that Student 1 completes it. So how is it possible for UpGrade to be able to keep track of student behavior relevant to an experiment that doesn't exist at that point?

The answer rests in the way that educational applications interact with UpGrade. In order to provide different experiences to students in different conditions, the educational application must call out to UpGrade in order to determine the student's condition and then conditionalize the student's experience based on the response from UpGrade. We call the point(s) at which the experimental application must react to the student's condition *experiment points*. Experiment points can be anywhere in the application and are typically inserted in the application code specifically to implement an experimental condition. For example, an experiment presenting various motivational messages to students when they make errors might be inserted at the point (or points) where such errors are determined. Other

experiment points represent common places for experimentation and can be placed in anticipation of experiments utilizing them. We term these *permanent experiment points*. In educational applications, selecting or presenting an activity to a student would likely be a permanent experiment point, on the assumption that many experiments will vary something about the activities presented to students. When a student reaches an experiment point in the application, UpGrade keeps a record. For active experiments, this is a way of tracking which students have experienced one of the experimental conditions, but such marking takes place whether or not an experiment is running. Permanent experiment points add data that helps mark the particular reason that point was reached. For example, an experiment point representing starting an activity would be marked along with the ID of the activity. In the case in Figure 1, Student 1 is marked as having started the "Factor Quadratics" activity because there is a permanent experiment point representing starting an activity. When an experiment relating to that activity becomes active, UpGrade knows that this student has already seen that activity and can appropriately assign the student to condition (or exclude the student from the experiment). Similarly, when Student 2 in Figure 1 starts Factoring Quadratics, that experiment point is marked for Student 2, and UpGrade determines that student's condition (or exclusion status).

Permanent experiment points are an imperfect mechanism, because they require application developers to anticipate key decision points in their applications that may be subject to experiments. But when these points can be anticipated, UpGrade allows researchers to design and launch experiments at any time during the school year, and UpGrade will act as if it always knew that the experiment would take place.

Determining group membership

Group random assignment requires that the assignment system understand student group membership. Among UpGrade's APIs, is an API for specifying student membership in groups and a separate API for specifying the current "working group." The API specifying general group membership might be used by the educational software that is the subject of the experiment (particularly when such software acts as an LMS) or it could be engaged through a rostering service (for example, such a service could send anonymized group membership based on a nightly Clever update). A "working group", for some applications, may indicate that the student is rostered in multiple groups (e.g. multiple classes), but the software is used in the context of one of those classes at a time (the "working group"). For this reason, we rely on the educational software itself to set the working group.

[1] https://github.com/CarnegieLearningWeb/educational-experiment-client

FUTURE DIRECTIONS

Although UpGrade is a new system, it provides a great deal of support for running group-random assignment experiments (in addition to individual-assignment experiments). We have prioritized features that we believe to be relevant to experiments conducted in schools, where group assignment is particularly important, but we intend the system to be applicable in non-school contexts (indeed, UpGrade has already been used for experiments on games used outside of school).

We are actively seeking collaborators to help us prioritize and expand the functionality of UpGrade. With respect to group assignment, we recognize the potential to support more efficient group-randomized experimental designs (e.g. allowing stratification, [1]). In addition, since UpGrade is aware of the experimental design, UpGrade could be extended to output information that guides the statistical analysis, including recommending nested models and identifying possible anomalies. Source code for UpGrade is available[1] and additional information about the platform will be available at upgradeplatform.org. If you are interested in using or contributing to UpGrade, you can email upgradeplatform@carnegielearning.com.

ACKNOWLEDGEMENTS

This work was supported by grants from the Bill & Melinda Gates Foundation and Schmidt Futures. The findings and conclusions contained within are those of the authors and do not necessarily reflect positions or policies of the foundation.

REFERENCES

[1] Douglas C. Montgomery. 2004. *Design and Analysis of Experiments, 10th edition*. John Wiley and Sons.

[2] David M. Murray. 1998. *Design and Analysis of Group-Randomized Trials*. Oxford University Press.

[3] David Paunesku. 2013. *Scaled-up social psychology: Intervening wisely and broadly in education*. Unpublished doctoral dissertation, Stanford University.

[4] Stephen W. Raudenbush & Anthony S. Bryk. 2002. *Hierarchical Linear Models: Applications and data analysis methods. (2nd ed.)*. Thousand Oaks, CA: Sage Publications.

[5] Stefan Thomke. 2020. *Experimentation works: The surprising power of business experiments.* Harvard Business Review Press.