Sequence Matters, But How Do I Discover How? Towards a Workflow for Evaluating Activity Sequences from Data

Shayan Doroudi[1], Kenneth Holstein[2], Vincent Aleven[2], Emma Brunskill[1]
[1]Computer Science Department, [2]Human-Computer Interaction Institute
Carnegie Mellon University
{shayand, kjholste, aleven, ebrun}@cs.cmu.edu

ABSTRACT

How should a wide variety of educational activities be sequenced in order to maximize student learning? We recently proposed the Sequencing Constraint Violation Analysis (SCOVA) method to help address this question. In this paper, we propose how SCOVA could be transformed into a workflow in LearnSphere so that other researchers and practitioners can find answers to the aforementioned question in their own datasets. We hope that such a workflow will lead to more and better research into this important question, as well as interesting new findings for both the educational data mining and learning sciences communities.

Keywords

sequencing, ordering, Intelligent Tutoring Systems, LearnSphere, DataShop, workflow.

1. INTRODUCTION

How to sequence educational activities is an important pedagogical question [12]. Much of the existing work on sequencing activities consists of theoretical analyses [2, 4, 7] and empirical studies [1, 13, 5, 11]. While empirical studies can help address questions that compare two or three different ways to sequence a curriculum (e.g., whether topics should be blocked or interleaved), it cannot effectively scale to analyzing the myriad of potential sequences that could be considered. However, educational data mining (EDM) techniques can enable one to simultaneously study different types of sequences based on past data. We recently proposed one such method—Sequencing Constraint Violation Analysis (SCOVA)—for comparing the efficacy of different sequencing constraints given a dataset that is rich in the variety of sequences it explores [3]. SCOVA can be used to analyze a wide variety of sequencing constraints, such as prerequisite relationships, constraints on when different learning mechanisms should be introduced, blocking, interleaving, and spiraling. SCOVA can both be used to better understand how problems should be sequenced in specific learning environments, including intelligent tutoring systems (ITSs), as well as to find some generalizable trends that may inform the learning sciences literature (e.g., on whether blocking or interleaving is more effective or in what order learning mechanisms should be supported). SCOVA can also be used to inform the creation of adaptive policies for ITSs. However, SCOVA will most likely *not* be used for any of these purposes if it just remains in a paper that a few researchers might, at best, read and cite. Rather, its benefit will likely only outlive the confines of a one-off EDM paper if it is released as a workflow on a platform like LearnSphere that is used by researchers and practitioners. If released as such a workflow, SCOVA can also introduce researchers who may not have otherwise considered the question of how activities should be sequenced in their learning environments to find a newfound interest in this area, which we believe is becoming increasingly important to both the learning sciences and educational data mining communities.

2. WORKFLOW METHOD

2.1 Data Inputs

SCOVA is applicable to datasets with substantial variability in the types of activity sequences that students complete. This variability is typical of many datasets, including ones that include randomness in how problems were presented to students (e.g., [9]), ones where adaptive policies were used for problem selection resulting in sequences that vary from student to student (e.g., [10]), and ones where students are able to do choose which problems to work on themselves (e.g., [8]). The workflow can work with datasets in the PSLC DataShop format. Given that SCOVA is a very general-purpose method, which can be used to analyze how a wide variety of sequencing constraints impact potentially different measures of student performance (e.g., within-tutor performance, posttest scores, learning gains, time on task, etc.), it may potentially need to utilize a variety of the columns in a DataShop dataset. However, for simplicity we will describe a version of SCOVA that is limited to analyzing sequencing constraints that may only depend on within-tutor correctness and properties of the activities presented to students and can only measure the impact with respect to within-tutor performance and functions of pretest and posttest scores (such as learning gains).

In full, SCOVA needs three input files:

1. The DataShop transaction-level file. For every step in a transaction-level dataset, SCOVA needs to know the problem name and whether the step was answered correctly or not.
2. A mapping of every problem name to categories to which the problem belongs. For example, when using SCOVA on our fractions ITS [3], we labeled each problem with one of three topic labels (making and naming fractions, fraction equivalence and ordering, and fraction addition) as well as one of three activity types corresponding to learning mechanisms from the Knowledge-Learning-Instruction (KLI) framework (sense-making, induction and refinement, and fluency-building) [6]. These category labels will then be used as

the building blocks of sequencing constraints, as explained in Section 2.2.

3. A file that gives the pretest and posttest score for each student.

2.2 Workflow Model

The workflow begins with the researcher selecting different sets of sequencing constraints that they want to analyze. Each sequencing constraint can be selected by first choosing a category (e.g., topics or activity type) and then selecting a pattern that corresponds to the sequencing constraint. The pattern can take on one of three forms:

1. Specifying a particular sequence (e.g., ABCABCABC, which may correspond to interleaving different activity types or topics).
2. Specifying that a student should be exposed to a problem with label A before a problem of label B (e.g., a student should be shown a number line problem before being shown a fraction equivalence problem)
3. Specifying that a student should have reached some performance threshold on a problem with label A before a problem with label B (e.g., a student should have 95% accuracy on fraction equivalence problems before being exposed to fraction addition)

The researcher can select as many sequencing constraints of the three forms above. Then for each possible permutation of category labels (e.g., A = fraction equivalence, B = fraction addition, C = naming fractions), SCOVA computes a score for how well each student's sequence in the dataset matches the given sequencing constraints. The score is the proportion of problems in the trajectory where a sequencing constraint was violated. SCOVA then learns a linear regression model that uses the degree to which a student violates a particular set of sequencing constraints to predict some chosen outcome variable (i.e., some measure of within-tutor performance or some function of the posttest and pretest scores). Notice that if the model has a negative correlation then that implies the more a student obeys a particular sequencing constraint, the better that student learns/performs in the tutoring system, i.e. negative correlations are indicative of beneficial sequencing constraints. The final step of SCOVA is to compare the model fits for different sets of sequencing constraints to guide the practitioner/researcher to which sequencing constraints have the largest positive impact on student learning. For more details on the method and particular instantiations of sequencing constraints, refer to [3].

2.3 Workflow Outputs

The primary output is a table of BIC values of models for every set of sequencing constraints evaluated. The practitioner can choose from a set of options how they want the table organized. For example, if we were evaluating the impact of constraints of the form topic A should come before topic B, which should come before topic C in tandem with constraints of the form activity type X should come before activity type Y, which should come before activity type Z, this could be represented in a 6-by-6 table where the rows correspond to the different permutations over topics and the columns correspond to the different permutations over activity types. (If there was a third category of interest with three different labels, such as say whether the difficulty level of the problem was easy, medium, or hard, then the workflow could display six different tables, one for each permutation of difficulty levels.) For an example of such a table, see Table 3 in [3].

In addition to showing BIC values, the table will highlight those cells where the violation of sequencing constraints correlates negatively with performance/learning (again an indicator that the sequencing constraint is beneficial for students rather than harmful), and will designate the model with the lowest BIC (i.e., the best-fitting model).

There will also be a toggle to display other quantities of importance in place of BIC, such as the coefficients of the predictors in the models. In the case of evaluating sequencing constraints over a single category (e.g., only how activity types should be sequenced), the user can choose to display the scatter plots used to fit each model and the best-fit lines themselves. The user can also choose to color-code each point of the scatter plots with the value of some feature (e.g., how many problems that student received). This color-coding of the plots can help identify potential confounds (e.g., students who do more problems might tend to violate fewer of a sequencing constraint and also do better simply because they did more problems).

Finally, the workflow will allow doing exploratory analyses to detect other potential confounds. For example, if the sequences in the data were generated according to adaptive policies, one potential confound is that a student's performance affects the degree to which sequencing constraints are violated in addition to the intended causal direction of the degree to which a sequencing constraint is violated influencing the student's performance. To analyze the presence of such a confound, models can be learned where the outcome variable is the student's pretest score (rather than say posttest score); since the pretest score comes before the students' use of the tutor, we know that the only reason it would correlate with violations of certain sequencing constraints is if the adaptive policies discriminated between students with different amounts of prior knowledge. In using SCOVA on our fractions tutor, we found that while this reverse causal direction did exist, it was seemingly negligible and actually biasing against the conclusions that our results support [3]. Such a workflow should allow users the ability to do exploratory analyses before making firm conclusions using SCOVA.

3. DISCUSSION

Having a workflow for analyzing the impact of different sequencing constraints can have a number of benefits for both the EDM and learning science communities. SCOVA can both be used to better understand how problems should be sequenced in specific learning environments, as well as to find some generalizable trends that may inform the learning sciences literature (e.g., on whether blocking or interleaving is more effective or how learning mechanisms should be sequenced). SCOVA can also be used to inform the creation of adaptive policies for ITSs. However, for SCOVA to be used in such a fashion, it will likely have to be readily available as a workflow on a platform like LearnSphere that is used by researchers and practitioners. Additionally, by having such a workflow on LearnSphere, more researchers may be attracted to the question of how to sequence problems in their learning environment of interest.

Furthermore, if LearnSphere also includes workflows for other methods of analyzing sequencing constraints such as [9], more research can be done in comparing these methods. Currently when such a method is published it is not widely adopted either in practice or by other researchers, and it is not compared to methods that succeed it. By putting all methods that do similar styles of analyses on one platform, LearnSphere can lead to more productive research, including hopefully better ways of understanding how we should sequence educational activities in different learning environments.

4. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grants R305A130215 and R305B150008 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Dept. of Education.

5. REFERENCES

[1] W. Battig. Intratask interference as a source of facilitation in transfer and retention. *Topics in learning and performance*, pages 131–159, 1972.

[2] R. E. Clark, D. Feldon, J. J. van Merri¨enboer, K. Yates, and S. Early. Cognitive task analysis. *Handbook of research on educational communications and technology*, 3:577–593, 2008.

[3] S. Doroudi, K. Holstein, V. Aleven, and E. Brunskill. Sequence Matters, But How Exactly? A Method for Evaluating Activity Sequences from Data. In *EDM*, 2016.

[4] J.-C. Falmagne, M. Koppen, M. Villano, J.-P. Doignon, and L. Johannesen. Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review*, 97(2):201, 1990.

[5] S. Kalyuga. Expertise reversal effect and its implications for learner-tailored instruction. *Educational Psychology Review*, 19(4):509–539, 2007.

[6] K. Koedinger, A. Corbett, and C. Perfetti. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757-798, 2012.

[7] K. Korossy. Modeling knowledge as competence and performance. *Knowledge spaces: Theories, empirical research, and applications*, pages 103–132, 1999.

[8] Y. Long and V. Aleven. Supporting students' self-regulated learning with an open learner model in a linear equation tutor. In *AIED*, 2013.

[9] Z. A. Pardos and N. T. Heffernan. Determining the significance of item order in randomized problem sets. 2009.

[10] M. A. Rau, V. Aleven, and N. Rummel. Complementary effects of sense-making and fluency-building support for connection making: A matter of sequence? In *AIED*, 2013.

[11] A. Renkl and R. K. Atkinson. Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational psychologist*, 38(1):15–22, 2003.

[12] F. E. Ritter, J. Nerb, E. Lehtinen, and T. M. O'Shea, editors. In order to learn: how the sequence of topics influences learning. Oxford University Press, 2007.

[13] D. Rohrer and K. Taylor. The shuffling of mathematics problems improves learning. *Instructional Science*, 35(6):481–498, 2007.