Teaching Recommender Systems at Large Scale: Evaluation and Lessons Learned from a Hybrid MOOC

JOSEPH A. KONSTAN and J. D. WALKER, University of Minnesota
D. CHRISTOPHER BROOKS, EDUCAUSE
KEITH BROWN, University of Minnesota
MICHAEL D. EKSTRAND, Texas State University

In the fall of 2013, we offered an open online Introduction to Recommender Systems through Coursera, while simultaneously offering a for-credit version of the course on-campus using the Coursera platform and a flipped classroom instruction model. As the goal of offering this course was to experiment with this type of instruction, we performed extensive evaluation including surveys of demographics, self-assessed skills, and learning intent; we also designed a knowledge-assessment tool specifically for the subject matter in this course, administering it before and after the course to measure learning, and again 5 months later to measure retention. We also tracked students through the course, including separating out students enrolled for credit from those enrolled only for the free, open course.

Students had significant knowledge gains across all levels of prior knowledge and across all demographic categories. The main predictor of knowledge gain was effort expended in the course. Students also had significant knowledge retention after the course. Both of these results are limited to the sample of students who chose to complete our knowledge tests. Student completion of the course was hard to predict, with few factors contributing predictive power; the main predictor of completion was intent to complete. Students who chose a concepts-only track with hand exercises achieved the same level of knowledge of recommender systems concepts as those who chose a programming track and its added assignments, though the programming students gained additional programming knowledge. Based on the limited data we were able to gather, face-to-face students performed as well as the online-only students or better; they preferred this format to traditional lecture for reasons ranging from pure convenience to the desire to watch videos at a different pace (slower for English language learners; faster for some native English speakers). This article also includes our qualitative observations, lessons learned, and future directions.

Categories and Subject Descriptors: K.3.1 [**Computers and Education**]: Computer Uses in Education—*Distance learning*; K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education*

General Terms: Measurement, Performance

Additional Key Words and Phrases: Massively Online Open Course (MOOC), learning assessment

Authors' addresses: J. A. Konstan, Department of Computer Science & Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN 55455; email: konstan@umn.edu; J. D. Walker, Sr VP Academic Affairs/Provost, University of Minnesota, 100 Church St SE, Minneapolis, MN 55455; email: jdwalker@umn.edu; D. C. Brooks, EDUCAUSE, 282 Century Place, Suite 5000, Louisville, CO 80027; email: cbrooks@educause.edu; K. P. Brown, Office of Information Technology, University of Minnesota, 1985 Buford Ave, St Paul, MN 55108; email: brown299@umn.edu; M. D. Ekstrand, Dept. of Computer Science, Texas State University, 601 University Drive, San Marcos, TX 78666; email: ekstrand@txstate.edu.

DOI: http://dx.doi.org/10.1145/2728171

ACM Reference Format:
Joseph A. Konstan, J. D. Walker, D. Christopher Brooks, Keith Brown, and Michael D. Ekstrand. 2015. Teaching recommender systems at large scale: Evaluation and lessons learned from a hybrid MOOC. ACM Trans. Comput.-Hum. Interact. 22, 2, Article 10 (April 2015), 23 pages.
DOI: http://dx.doi.org/10.1145/2728171

1. INTRODUCTION

In Fall 2013, we offered *An Introduction to Recommender Systems* as a full-semester hybrid online course. We offered this course simultaneously in two formats: as an online course through Coursera and as a 3-credit graduate-level course at the University of Minnesota, using a modified flipped-classroom model where on-campus students enrolled in and completed all of the course activities on Coursera while also having live sessions with faculty and a teaching assistant to provide extra support on understanding course material and completing course assignments. This course was offered as part of the University of Minnesota's exploration of MOOCs (Massive Open Online Courses) and was launched after a strategic decision by the Department of Computer Science and Engineering to explore the medium and its implications. As part of this exploration, we focused extensively on gathering data for research; this course is the first such class we are aware of that supplements student process and outcome data with both a survey of student background/intent and a subject matter mastery pretest/posttest to assess student learning outcomes.

Much of this course design derives from its original intent—to explore the medium of massive-scale online education as a vehicle for delivering in-depth advanced technical content at a level normally associated with a graduate-credit course. As we will discuss in detail, this led to both a longer format and a more intensive level of assignments. In our early design efforts, however, we recognized that a substantial number of potential students would lack the technical programming skills—or the inclination to invest programming time—to complete such a course. Accordingly, we adapted the design into a two-track course: students can complete a "concepts" track with a basic mathematics background (and without programming assignments) or may complete a comprehensive "programming" track that includes the concepts track plus programming lectures and six programming assignments. Both groups are able to earn statements of accomplishment. Much of our analysis looks at the differences in student intent and performance between those enrolled in the concepts and programming tracks.

This article is an extended version of a paper published at the ACM Learning@Scale conference [Konstan et al. 2014]. Among the changes and additions in this article are (1) a revised/improved analysis of student grade outcomes that more correctly adjusts for the different grade bases associated with different course tracks; (2) improved analysis of student learning and performance by track, using behavioral metrics as a more accurate replacement for declared intent; (3) a new 5-month postcourse learning retention study; (4) a new 5-month postcourse student evaluation and feedback, including data on how course learnings were or were not used after the course; (5) a new analysis of the effect of the MOOC on departmental reputation; and (6) more extensive discussion of and citations to related work.

The rest of this article is organized into six sections. First, we provide a narrative and statistical overview of the course, followed by a review of our research goals and methods. The following section presents our quantitative results. Finally, we conclude with a discussion of these results, some qualitative results and anecdotes, and lessons learned.

2. OVERVIEW OF THE COURSE

The title of the course, "Introduction to Recommender Systems," reflects the course's origin and goals. Approximately 2 decades after the initial development of automated collaborative filtering—a technology for predicting user preferences based on the

preference ratings of like-minded individuals—the field of recommender systems had blossomed. We sought to introduce students to the core algorithmic approaches to recommendation (nonpersonalized approaches, content-based filtering, user- and item-based collaborative filtering, dimensionality-reduction/matrix factorization methods, and brief introductions to case-based reasoning and social network/trust-based recommendation), to evaluation and metrics, to issues related to recommender systems data (implicit and explicit ratings, ratings scales, acquiring data, data validity), and to user interface and recommender design issues. We also decided we wanted to expose students to a broad set of perspectives and, as a result, invited a large set of experts to join us for interviews on areas where they had special expertise, had built novel systems, or had conducted interesting research (mostly recorded over Skype or Google Hangouts, though in some cases recorded onsite).

The course grew naturally out of the context of the field. Our research group had founded the ACM Recommender Systems conference in 2007 (which has grown from about 120 attendees to over 300, and regularly moves among the United States and Europe, and for the first time in 2013, Asia). Several books had been published in the field, including an introductory text and a research handbook. And we, and others, had regularly been invited to give tutorials and short courses on the topic. But to our knowledge, there were few if any regular courses covering the field.

This need juxtaposed nicely with the interests of both our university and department in exploring the MOOC space. The department interest emerged from a strategic planning exercise in 2012 where the faculty determined that we should experiment with MOOC education for at least three reasons: (1) to explore how it may affect future University education, (2) to explore how MOOC instruction affects department visibility and reputation worldwide, and (3) to better understand the effort involved, technology, and teaching methods. We volunteered and were accepted into our university's set of trial MOOCs.

From the beginning, we were unique in three ways: First, we did not want to choose between offering this new material to the world and offering it to our students. We decided to design what we thought would be a good 14-week, 3-credit graduate course, adjust that design to reflect MOOC delivery, and then deliver it both to the world and to our own students (through the flipped classroom model). We also recognized that the face-to-face sessions could be useful to MOOC students, so we recorded about half of these to make them available online.

Second, we were developing the software platform through which students could carry out many of the activities in the course. LensKit [Ekstrand et al. 2001] is an open-source recommender toolkit developed specifically to support experimentation. It includes a set of core recommender algorithms and metrics, and it provides both a scripting interface and an API to allow users to experiment with these and build their own.

Third, we wanted this course to be accessible to nonprogrammers as well as programmers. From our work in the field, we knew that many of the people interested in recommender systems focused on product marketing, business analysis, and other areas. In addition, we were concerned that there might not be enough students with the skills and willingness to install a new software toolkit and engage in some fairly complex programming.

Given our goals, we promoted the course as widely as possible. We reached out to colleagues through mailing lists related to recommender systems and related topics. ACM agreed to announce the course to the roughly 1,000 students who had watched Konstan's webinar in ACM's webinar series. And, of course, Coursera itself provides listings and promotion through their website and course recommender tools.

In the end, the course comprised eight modules (1-week introduction and wrap-up, and six 2-week core modules organized around algorithms: five on different families

of algorithms and one focused specifically on evaluation and metrics). Together, these include:

- 42 recorded lectures (ranging from 10 to 45 minutes, most with pop-up comprehension questions);
- 14 recorded interviews with 12 experts in the field;
- 12 recorded face-to-face class sessions, mostly focused on Q&A, though some contained other enrichment topics;
- 7 nonprogramming "written assignments," most with video introductions;
- 6 programming assignments, most with video introductions;
- 2 multiple choice exams (36 questions each) on aspects not including programming; and
- a collection of online readings and references (some required, others for reference for interested students).

Face-to-face students experienced the course much as the online students did—all of their readings, lectures, assignments, and exams were conducted through the online Coursera course—but with five specific enhancements:

- Face-to-face students had three required classroom sessions (these sessions were not videotaped). The first session was a start-of-class overview to explain how the course would proceed; the second and third sessions were midcourse and end-of-course candid feedback sessions.
- Face-to-face students had a weekly TA session where they could go for help with any aspects of the course; in general, attendance at these sessions was light (10% to 20% of the class) except for a couple of sessions where students received help setting up tools for programming assignments.
- Face-to-face students had a weekly session with faculty (mostly recorded for online students to watch if they chose) that focused on extended Q&A and further elaboration on or clarification of points from the classroom.
- Face-to-face students were able to contact the course faculty and TA for help (via email, through office hours, or just stopping by).
- Finally, given concerns about peer-grading, face-to-face students were offered the option to have any assignment regraded by the course TA (few availed themselves of that opportunity; only one had a case of clear misgrading that was corrected).

The final course design had two tracks. The programming track included all content and assignments. The concepts track excluded the programming assignments and a few video lectures specific to programming details. Face-to-face students were required to complete the programming track.

Total enrollment for the course reached 28,389 students, of whom, 21,357 were active in some way (watching any of a lecture, submitting any assignment, posting an item to a discussion forum) during the course. The number of people participating on a substantial and regular basis is much smaller. As Clow [2013] found to be common in MOOCs, there was a high attrition rate. For Module 1 lecture videos, the number of student views per video was nearly 9,000, as compared to only 2,195 in Module 8. Likewise, the number of submissions for each of the seven written assignments (to be completed by both concept and programming track) dropped from 5,420 for the first assignment to 530 for the final assignment. A total of 5,643 students earned a nonzero final grade in the course.

Much of the interest of MOOCs lies in the breadth of their enrollment—in the fact that they recruit students from around the world. For this reason, understanding who MOOC students are, why they are taking the MOOC, and what they regard as success is critical to the task of assessing the impact of a MOOC.

From a preclass survey, it appears that the students taking this course were a youngish, heavily male group largely residing outside the United States who were experienced and confident with respect to the course's subject matter:

- 70% of students reported having a degree in computer science or a related field, while 57% said they had taken more than five college-level computer science courses
- 80% said they were very or moderately confident in their programming skills
- 71% planned to complete the entire course, including assignments and assessments
- 87% were male
- 68% were under 35
- 67% reported being nonnative English speakers, but 76% reported proficient or advanced proficient English skills.
- 68% were residents of a country other than the United States
- the group was heavily weighted toward working professionals along with graduate students studying computer science or a related field.

The high level of prior education and heavy weighting toward working professionals and graduate students are not surprising, both in light of the advanced subject matter (appropriate for a graduate in course) and in light of (Emanuel 2013) which reports that many MOOC students are educated professionals seeking practical skills. We were somewhat surprised at the high level of respondents who indicated that they planned to complete the course (particularly in view of Emanuel's results), but we recognize this may be a selection effect.

3. RESEARCH GOALS

The empirical investigation of this course attempts to address the following research questions:

1. Do students learn in this MOOC? If so, how much, and which ones? Which variables predict normalized subject matter learning gains among the MOOC students?
2. Do students in a face-to-face recommender systems course, who have access to MOOC resources, learn more than a comparable group of MOOC students who have access to recorded face-to-face instructional sessions?
3. What demographic, background, and behavioral factors predict course completion, normalized subject matter learning gains, and course grades in this MOOC?
4. What is the interaction between learning (and practicing) concept subject matter on recommender systems and learning (and practicing) the programming of that subject matter in the context of a MOOC?
5. What are different types of reasons for taking this MOOC? Do these correlate with demographics or with learning gains?
6. Do the MOOC students retain what they learned, when subject matter knowledge is measured 5 months after the end of the course? Do the face-to-face students retain more of what they learned than the MOOC students do?

4. RESEARCH METHODS

4.1. Design

This study used a single-group cross-sectional research design to address questions having to do with the online MOOC student population. It also used a pretest-posttest nonequivalent groups design to address questions comparing the online student population with students in the face-to-face recommender systems class. Because students self-enrolled in the MOOC and in the face-to-face class, random assignment to treatment conditions was not possible. However, the instructor, course content and objectives, and main course assessments were held constant, and data on demographic

characteristics and precourse subject matter knowledge were used to ensure that the online and face-to-face groups being compared were similar in the relevant respects.

4.2. Participants

The participants in this study included 39 students in the face-to-face section of CSci 5980: Recommender Systems as well as approximately 4,844 students who completed a precourse survey and a precourse recommender systems knowledge test. Students reporting an age of less than 18 years were removed from the study to comply with IRB regulations.

4.3. Measures

This study used pre- and postclass surveys designed to measure students' background, intentions with respect to the MOOC they are taking, and reactions to the MOOC experience. It also employed a 20-item instructor-generated recommender systems knowledge test designed to measure gains in students' subject-matter knowledge over the semester. The knowledge test was administered three times to MOOC participants and face-to-face students: once at the beginning of the MOOC/semester to all enrolled participants, once at the conclusion of the MOOC/semester to all of those who had completed the baseline exam, and once approximately 5 months after the end of the course to only those who had completed the second exam. The last of these was designed to measure retention of subject matter knowledge, an important but frequently overlooked component of this type of research.

All knowledge test questions were multiple choice, with four content choices and a fifth "I have no idea" choice to permit students to admit they do not know. Examples of questions include:

1. Which of these best describes a case-based recommender?
   a. A recommender that provides recommendations for large sets of products sold together.
   b. A recommender the uses correlations among users to predict which items each user would enjoy.
   c. A recommender that uses product ratings to build a profile of attribute interests.
   d. A recommender that uses a database of examples and forms queries from user requests to explore items that meet user criteria.
   e. I have no idea.
2. What is the core idea behind dimensionality reduction recommenders?
   a. To reduce the computation from polynomial to linear.
   b. To strip off any product attributes so products appear simpler.
   c. To reduce the computation time from O(n3) to O(n2)
   d. To transform a ratings matrix into a pair of smaller taste-space matrices.
   e. I have no idea.

The preclass test (N = 4,844) showed an acceptable level of difficulty and discriminated well among students with high and low levels of subject matter knowledge, with a mean score of only 18.85% and a standardized deviation of 15.13; only three students scored above 80% on the test and no student scored 90% or more.

4.4. Preliminary Analyses

We began our data analysis by examining a set of questions that asked students to rate the strength of 15 different reasons for enrolling in a MOOC. An exploratory principal components analysis was conducted, and we extracted four factors that had

Table I. Summary of Participants and Response Rates (in parentheses)

	MOOC (online-only students)	CSCI 5980 (hybrid f2f students)
N	28,389 (100.0%)	39 (100.0%)
Active participants	21,357 (75.2%)	39 (100.0%)
Nonzero grade (of active participants)	5,643 (26.4%)	39 (100.0%)
Precourse knowledge test (of active participants)	4,844 (22.7%)	22 (56.4%)
Postcourse knowledge test (of precourse knowledge test)	304 (6.3%)	10 (25.6%)
Retention test (of postcourse knowledge test)	91 (29.9%)	6 (60.0%)

Eigenvalues >1 along with acceptable factor loadings for all of the 15 individual items. The underlying constructs for these factors were:

- University/instructor-related reasons (e.g., "because this course is offered by a prestigious university");
- Pragmatic/access reasons (e.g., "I am not geographically close to educational institutions");
- Professional reasons (e.g., "To obtain a badge or certification that will be useful to me professionally");
- Interest/enjoyment-related reasons (e.g., "I think taking this course will be fun and enjoyable").

These constructs were used to help us segment and define the student population in this course—for instance, by serving as predictors in the regression analyses described in the following text. We should note that we were not trying to develop a reusable taxonomy or classification of goals for enrollment such as the work of Kizilcec et al. [2013] or of Wilkowski et al. [2014]. We performed our analysis as a data-driven empirical exploration of motivation solely for purposes of identifying factors to use in subsequent analysis. We expect that future work will want to build on the emerging taxonomies, particularly as the field converges on taxonomies that show significant utility.

5. RESULTS

5.1. Completion and Retention

While much early MOOC evaluation focused on very low rates of full course completion, as we improve our understanding of the reasons for which students take MOOCs, it becomes imperative to attend to the notion that "success" might be relative to the individual. To explore this idea, we collected data on the amount of the MOOC that students intended to complete and on how much they completed relative to those expectations.

However, not unlike other researchers in this nascent field, we struggle with how to define completion rates. Moreover, since we also collected data longitudinally and efficiently (e.g., only requesting postcourse responses from those who completed precourse materials), our denominator for calculating response rates shifted necessarily. Table I consolidates our data on numbers of respondents and participant/response rates for our study. Subsequent regression models may report lower N values when individual subjects' missing values cause them to be excluded from analysis.

We found that the vast majority of students (72.5%) who completed the precourse survey intended to complete the entire course, rather than only certain parts, or the course material but not the assignments. Furthermore, a large majority of students (72.4%) reported that they completed as much or more of the MOOC than they had intended to. Finally, nearly all of the students (95.75%) who said they completed less

of the MOOC than they intended also reported that they found the experience useful nonetheless.

We also examined two different nonrelative measures of course completion: completing the sixth writing assignment in the class and completing the third part of exam 2. We constructed a multivariate logistic regression model for each of these measures, based on students' demographic characteristics, motivations, baseline knowledge, and activity during the semester, in an effort to understand how the characteristics of students influence completion.

(A methodological note: For each dependent variable we analyze, we report only the model with the greatest explanatory power, or in other words, the model that accounts for the greatest amount of variation in the dependent variable. Predictor variables are reported only if they contribute to the best-fitting model.)

Both of the models were significant, but the amount of variance in the dependent variables explained by the models was extremely small (pseudo-$R^2 = 0.059$ and 0.066). Nonetheless, as the data in Table II show, the models offer several insights.

First, it is worth noting how many factors did not affect course completion, defined either by the writing-based or the exam-based variables. Age, sex, English proficiency, and United States residency all had no impact on completion, nor did status as a professional, graduate or undergraduate student, or most of the reasons students expressed for taking this MOOC.

However, the models show that knowledge, experience, and strong intentions influenced both measures of completion. The higher a student's score on the knowledge pretest, the greater the number of MOOCs she had taken in the past, and the stronger her intention to complete the course, the more likely was it that she would complete both the writing assignment and exam in question.

The two models are also similar in that completion in both senses is negatively predicted by a student's reporting greater introversion and by taking a larger number of concurrent courses. The latter conclusion may be related to the common finding that time pressures are the factor most often reported by MOOC students as a reason for not completing as much of a course as they had intended to. The two models diverge when it comes to a student's taking the class for reasons of interest or enjoyment, which only predicts significantly her completing the writing assignment, and with respect to a student's level of programming confidence, which only predicts significantly her completing the exam.

Result: Intention predicts completion; little else does.

5.2. Recommender Systems Knowledge

It is generally difficult to determine how much or how well students learn in a MOOC because the student population is diverse and is not progressing through a sequence of prerequisite courses. As a result, even with end-of-course assessments, one typically does not know how much understanding or knowledge students began with. Some studies (e.g., Buerk et al. [2013]) avoid this question by focusing solely on final outcomes (e.g., final grades), but we feel this does not adequately assess learning, particularly in the MOOC context where we know many enrolled students already enter the course with substantial expertise.

In this study, as has been mentioned previously, the instructors produced a 20-item knowledge of recommender systems exam that was administered to students at the beginning and at the end of the course. The precourse knowledge test ($N = 4{,}844$, response rate 32.1%) was intended to establish the level of recommender systems knowledge with which students began the course so that learning gains over and above that baseline could be determined. It was designed to focus specifically on the type

Table II. Determinants of Completion as Measured by Written Homework 6 and Exam 2: Part III, Course Participants, Logistic Regression

		Written Homework 6	Exam 2: Part III
Student Reasons	Professional	1.069 (0.115)	1.097 (0.107)
	University/Instructor	0.9874 (0.1020)	0.891 (0.083)
	Interest/Enjoyment	1.364* (0.202)	1.235 (0.161)
	Pragmatic/Access	0 .902 (0.088)	0.960 (0.084)
Programming Skills Confidence		1.150 (0.115)	1.257* (0.116)
Track	Programming	1.058 (0.256)	0.985 (0.211)
	Concepts	1.330 (0.366)	1.218 (0.302)
Experience	Professional	0.799 (0.224)	0.610 (0.157)
	Graduate Student	1.061 (0.277)	0.834 (0.203)
	Undergraduate Student	1.057 (0.412)	0.697 (0.253)
Aptitude (Baseline Knowledge Test)		1.107*** (0.024)	1.113*** (0.022)
Intention to Complete Course		1.985*** (0.314)	2.168*** (0.316)
Number of Courses Taken Concurrently		0.859** (0.040)	0.881** (0.035)
Number of MOOCs Taken Previously		1.025* (0.013)	1.029** (0.011)
Hours/Week Available		1.013 (0.009)	1.008 (0.008)
Introversion/Extroversion		0.880* (0.044)	0.874** (0.039)
Sex		1.223 (0.306)	1.095 (0.240)
English Proficiency		0.733 (0.131)	0.803 (0.129)
Location: United States		1.104 (0.192)	0.913 (0.146)
Age		1.078 (0.046)	1.062 (0.042)
Constant		0.002*** (0.002)	0.004*** (0.003)
N		3326	3326
Chi-Square		102.18****	133.59****
Pseudo-R^2		0.059	0.066
Log likelihood		−810.965	−950.188

Note: Reporting odd ratios (standard errors)
$^{*}p < .05$; $^{**}p < .01$; $^{***}p < .001$; $^{****}p < .0001$.

Table III. Comparison of Recommender System Course Participants Respondents to Precourse Survey to Aggregated Coursera Data

	US residents	Gender	Age	Students
Coursera data	35%	88% male	76.8% under 40	30%
Precourse survey data	32%	87% male	80% under 40	30%

Table IV. Knowledge Test Gains: Paired t-test Results, All Course Participants

N	Pretest	Posttest	Normalized gain	p-value
262	24.71 (15.85)	69.90 (17.80)	60.02%	<.001

Note: Cell entries are mean test scores, standard deviations (in parentheses), and N.

of content students would learn in the Concepts track but was developed before the course material was fully designed.

To ensure that precourse survey respondents were representative of the larger population of enrollees, precourse survey data were compared with the data available through the Coursera reporting interface, which was derived from standard questions delivered by Coursera to every student. The nature of the Coursera data limited the possible comparisons, but from the available information, precourse survey respondents seemed to be reasonably similar to the larger group (see Table III).

The postcourse knowledge test was taken by far fewer students (N = 304, response rate 6.4%) due to the MOOC student attrition that is well known in the field. However, appropriate statistical tests revealed that students who took the postknowledge test were reasonably representative of the larger group in terms of demographic characteristics, reasons for taking the class, and so on. The main difference of note was in the baseline knowledge test, with a large difference (almost 0.5 of a standard deviation) favoring posttest takers. So the posttest students were an elite group with respect to their incoming knowledge of recommender systems but were otherwise similar to their fellow students.

For each student who took both the pre- and postcourse knowledge test, normalized learning gains were calculated [McConnell et al. 2005], defined as a student's knowledge posttest score minus her pretest score, divided by the magnitude of her possible knowledge gains (i.e., posttest – pretest/100% – pretest). This variable, which is the main learning outcome of interest in this study, takes a student's starting point into consideration and tries to account for the fact that it is more difficult to make gains when one begins near the top of the testing scale. We used this measure because, for the purposes of this investigation, we were not primarily interested in whether students attained some threshold level of knowledge. We wanted to know instead how well students at all levels of incoming knowledge could learn in the MOOC environment.

Finally, as a preliminary test of validity, we determined that knowledge posttest scores correlated well, and significantly, with the exam portion of students' final grades in the course ($r = .621$, $p < .001$). Normalized gains also correlated moderately well with students' exam scores ($r = .509$, $p < .001$).

Across all students who took either the pre- or postcourse knowledge test, the course appeared to result in large learning gains, as shown by the difference in mean scores for the pre- (18.85%, N = 4,844) and postknowledge tests (69.05%, N = 304). Much of this effect could, however, be due to student self-selection, if only the stronger students remained in the class at the end of term and took the posttest.

To examine this possibility, we used a paired-samples t-test which revealed that the 262 students who took both the pre- and postknowledge tests also showed large gains in recommender systems knowledge, indicating that the apparent improvement in student knowledge is not spurious (see Table IV).

Result: Student knowledge increased.

Table V. Knowledge Test Gains: Face-to-Face and MOOC Students

	N	Pretest	Posttest	Normalized gain
Face-to-face students	10	25.00% (18.41)	75.50% (10.91)	66.71%
MOOC students	252	24.80% (15.74)	69.74% (18.01)	58.31%
Independent-samples t-test p-value		.969	.317	.314

Note: Cell entries for face-to-face and MOOC students are mean test scores (std. deviations).

It is usually difficult to study systematically the differences in learning between face-to-face and MOOC students, due to the lack of a measure of baseline understanding or knowledge.

In this study, our precourse knowledge test provided that baseline measure. If we compare face-to-face students and online students who took both the pre- and post-knowledge tests, we find that these two groups of students were statistically equivalent in terms of the recommender systems knowledge they began the course with (Table V). Our design provides a useful contrast against Colvin et al. [2014], which reports comparable learning among MOOC and on-campus students, but which is not comparing students engaging in the same learning activities, but rather students in a traditional face-to-face course. Also Colvin et al. [2014] look at homework performance (on assignments with substantial overlap) rather than exam-based assessments of knowledge gain.

We can then compare the two groups in terms of the normalized gains in knowledge they achieved over the semester. We find a nominal difference of 8.4% favoring the face-to-face students. This difference is moderate in size (about one third of a standard deviation), although the very small N in the face-to-face group prevents this effect from attaining statistical significance.

The low N in the face-to-face group limits the statistical analyses that can be performed, and it may also limit the external validity of our findings. Regardless, this finding parallels that of studies that suggest that blended learning environments produce greater learning gains than online environments alone [Means et al. 2010].

Result: Limited data suggests that face-to-face students learned at least as much as online-only students.

One worry about online education has traditionally been that support and assistance for struggling students are limited, so students with weaker backgrounds in a subject may drop out or fail to benefit as much as they might in a face-to-face course.

To determine whether this was true in the Recommender Systems course, we divided the Recommender Systems students into quartiles based on their scores on the baseline knowledge test. We then used a one-way ANOVA test to compare the normalized knowledge gains of the four groups.

Broadly speaking, students with different incoming levels of recommender system knowledge benefited to very similar degrees from the course. So it is not the case that the course was beneficial to students who already knew a good deal about recommender systems, but left-behind students who were relative neophytes.

An ANOVA analysis did reveal that the difference between the two highest quartiles 3 and 4 is statistically significant ($p < .05$), possibly reflecting ceiling effects limiting the potential to measure learning gains for top-quartile pretest scorers.

Result: Students at all incoming knowledge levels benefited similarly from the course.

One analytic dimension of interest was the difference between the programming and concepts tracks in the course. Grades in the course were a combination of exam scores (24%), written assignment grades (40%) and programming assignment grades (36%). Programming students were expected to complete all assignments in the course and were thus graded out of 100, with a score of 80 required for completion "with

Table VI. Average Raw Knowledge Gains, by Track, All Students

Track	N	(post - pre) mean raw gain	Significance
Concepts	54	47.22	$p < .001$
Programming	165	47.88	$p < .001$

Table VII. Average Normalized Knowledge Gains, by Tracks, All Students

Track	N	Mean normalized gain	Std. Dev.	Significance
Concepts	54	61.56%	19.65	$p = .768$
Programming	165	62.60 %	23.13	

Table VIII. Mean Final Course Grades, by Track, All Students

Track	N	Mean final grade	Std. Dev.	Significance
Concepts	625	38.22%	13.19	$p < .001$
Programming	617	77.76%	12.83	

distinction"; concepts students were not expected to complete the programming assignments and were required to obtain a score of 50 (out of 64) for a certificate of completion. Indeed, any student earning 50 points would receive such a certificate. Distinction was labeled as specifically including mastery of programming recommender systems.

Although one precourse survey question asked students which track they intended to enroll in, we found that actual performance did not consistently match intent. Accordingly, we used a behavior-based method to define and compare the two groups, treating any student who completed more than half of the programming assignments as a programming student.

Examining the pre-course and postcourse knowledge tests shows that students in the concepts track began the course with somewhat weaker recommender systems knowledge but made gains that were similar to those of students in the programming track. We should note the caveat that the pretest, posttest, and course exams were designed to test only recommender systems concepts knowledge, and specifically did not test programming knowledge.

Among students who took both the pre- and postknowledge tests, programming and concepts track students showed very similar large gains in recommender systems knowledge (see Table VI). An independent-samples t-test shows no significant difference between the normalized gains of the concepts and programming track students (see Table VII).

Mindful of our caveat, we recognize that programming track students should have gained a set of knowledge related to programming recommender systems not learned by the concepts track students. It appears that they did, as shown by the substantially higher final grade earned by programming students which reflects successful completion of the graded programming assignments (see Table VIII). Recall that students could achieve above 64% only by completing programming assignments (which some, but few, concepts track students chose to do).

Result: Students in the programming and concepts tracks had similar gains in concepts knowledge, but programming students gained further knowledge.

5.3. Factors Predicting Learning and Student Success

To help us understand what factors contribute to learning in a course with such a breadth of student characteristics, we constructed an OLS regression model that attempts to predict normalized knowledge gains. The predictors in this model were variables that were plausibly causal—not student perceptions or opinions, but students'

Table IX. Determinants of Normalized Knowledge Gains, OLS Regression, All Students

Hours per Week on Course	.015/.056 (.019)
Forum Posts	–.001/–.041 (.001)
# of Written Assignments	.052/4.230 (.012)****
# of Progr. Assignments	.004/.061 (.007)
Progr. Skills Confidence	.006/019 (.024)
Progr. vs. Concepts Track	–.105/–.163 (.053)*
Progr. Courses Taken	.010/.083 (.009)
Precourse Knowl. Test	–.004/–.281 (.001)****
Native English Speakers	–.022/–.041 (.046)
Sex	.014/.018 (.052)
English Proficiency	.044/.148 (.023)
Location: United States	.021/.038 (.044)
Age	.007/.049 (.010)
Constant	.134 (.142)
N	207
F	5.030****
Adjusted R^2	0.202

Note: Cell entries are unstandardized/standardized OLS coefficients with standard errors (in parentheses). Only factors that contribute to the highest adjusted R^2 model are listed.
*$p < .05$; ****$p < .0001$.

demographic characteristics, motivations, baseline knowledge, and activity during the semester.

While the model fits the data rather well ($F = 5.030$; $p < .0001$), the amount of variance in the dependent variable explained by the model is relatively small (adjusted $R^2 = 0.202$). Nonetheless, as the data in Table IX show, the model yields several conclusions.

First, the Recommender Systems course treated students equally across many dimensions. The following variables made no difference to the normalized gains a student achieved:

- sex
- age
- US residence
- native English speaker
- programming confidence
- number of programming courses taken
- number of concurrent courses
- being a professional
- being a graduate student
- being an undergraduate student
- reasons for taking the course

Second, the following variables did predict normalized gains:

- baseline recommender systems knowledge (negatively)
- being in the concepts track
- number of written assignments completed (strongest)
- English proficiency (marginally significant; $p = .059$).

Interestingly, despite the large effect size associated with the number of written assignments a student completed (over one third of a standard deviation per standard deviation increase in the predictor), the other variables in the model that measured student academic effort and activity did not approach statistical significance.

Predictor variables having to do with programming also did not predict significantly students' normalized knowledge gains—except for being in the programming track as opposed to the concepts track, which was associated with a slight *decrease* in normalized gains.

If we apply this model to the students in the concepts and programming tracks as separate groups, we find that the model predicts just about as well for each track separately as for both together (concepts track adjusted $R^2 = .189$, $p = .011$, programming track adjusted $R^2 = .200$, $p < .001$.

While it is a success for a course to not reward certain subpopulations of students more richly than others, the failure of the model to predict much of the variability in normalized gain scores indicates that the model is mis- or underspecified. In all likelihood, the right set of predictor variables was not available, and this may reflect the diversity of the student population who enrolled in this MOOC. Champaign et al. [2014] also find surprising associations between student characteristics and outcomes, indicating that the predictors of MOOC student success call for further research. DeBoer et al. [2014] agree that traditional educational variables require reconceptualization to support understanding and prediction in a MOOC environment.

Result: Normalized knowledge gains are very difficult to predict; measures of relevant effort were strongest.

To further understand the determinants of student success in this course, we constructed OLS regression models that attempted to predict final grades in the course, on the basis of students' demographic characteristics, motivations, baseline knowledge, and activity during the semester. Given that concepts and programming students had differential chances to earn points in the course, we constructed separate models for the two tracks, in an effort to predict end-of-term grades.

To begin with, as was the case with normalized learning gains, a number of factors were not predictive of final grades in the Recommender Systems course in either track. The following variables made no difference to final grades:

- sex
- age
- English proficiency
- native English speaker
- programming confidence
- being a professional
- being a graduate student
- being an undergraduate student

We found it difficult to predict final grades for concepts students, with the best model predicting only 6.9% of the variation in the dependent variable (see Table X).

We had greater success predicting final grades for programming students, achieving an adjusted R^2 of 0.319 (see Table XI).

As a last step, we tried to predict the concepts component of final grades for the programming students, and we located a model with $R^2 = 0.222$ ($F = 5.404$, $p < .001$, other data not shown).

When the predictive power of a model is as low as it is in the case of our model for concepts students' grades, one should not interpret the significance of individual coefficients with much confidence. In the case of our model for programming students' grades, however, we should note the significance of two indicators of effort exerted—hours

Table X. Determinants of Final Grade for Concept Students Only, OLS Regression

Hours per Week on Course	–.047/–.036 (.071)
Forum Posts	.817/.190 (.232)***
Plans for Completion	–2.947/–.110 (.1.476)*
Progr. Skills Confidence	–1.738/–.128 (.774)*
University-Related Reasons	–.2.081/–.127 (.1.038)*
Professional Reasons	1.983/.123 (1.050)
Precourse Knowl. Test	.014/.017 (.043)
Native English Speaker	.877/.032 (1.739)
Sex	–1.474/–.038 (2.185)
Location: United States	–1.136/–.042 (1.760)
Age	.577/.086 (.379)
Constant	44.454 (5.695)***
N	329
F	3.215***
Adjusted R^2	0.069

Note: Cell entries are unstandardized/standardized OLS coefficients with standard errors (in parentheses). Only factors that contribute to the highest adjusted R^2 model are listed.
*$p < .05$; ***$p < .001$.

Table XI. Determinants of Final Grade for Programming Students Only, OLS Regression

Hours per Week on Course	2.666/.248 (.745)***
Self-Reported Amt. Learned	2.088/.171 (.848)*
Amt. of Course Completed	6.383/.331 (1.359)***
Progr. Courses Taken	1.698/.320 (.367)***
Pragmatic Reasons	–3.564/–.264 (.939)***
Precourse Knowl. Test	.073/.113 (.045)
Native English Speaker	3.573/.168 (1.825)
Sex	.783/.021 (2.574)
Location: United States	–.190/–.009 (1.787)
Age	–.363/–.062 (.413)
Constant	57.460 (6.348)***
N	154
F	8.220***
Adjusted R^2	0.319

Note: Cell entries are unstandardized/standardized OLS coefficients with standard errors (in parentheses). Only factors that contribute to the highest adjusted R^2 model are listed.
*$p < .05$; ***$p < .001$.

spent on the course, and amount of the course completed. Finally, it is probably no surprise that number of programming courses taken predicts grades for these students.

Our conclusions are that we can predict programming *grades* moderately well, but *concepts* grades not well at all. This may be because some variables available to us are well aligned with programming performance, such as programming courses taken, programming confidence, and so forth. Our set of predictor variables does not include anything comparable for concepts knowledge.

Further, we can predict grades for programming *students* moderately well, but can not predict grades for *concepts* students well at all. This may be a sign of demographic

Table XII. Knowledge Retention (Postcourse vs. Follow-up): Face-to-Face Students and MOOC Participants (Combined)

Mean postcourse score	Mean follow-up score	Pooled standard deviation	N	Significance
75.83	70.21	13.64	97	$p < .001$

or other differences between the programming and concepts students, and indeed statistical tests on available variables show many differences in demography, academic history, and so on (data not shown).

Result: Predicting student end-of-term performance is difficult; appropriate predictor variables may be lacking.

5.4. Knowledge Retention over Time

Finally, we wanted to test not only the immediate impact of the course on student subject matter knowledge but also student retention of that knowledge over time. Longer-term knowledge retention is studied less frequently than short-term gains due in part to associated practical difficulties (e.g., locating students several months after a class has ended, incentivizing students to exert effort on a follow-up knowledge test). In their compendious review of empirical literature on postsecondary education, Pascarella and Terenzini [2005] cite evidence that "students can retain somewhere between 70% and 85% of the subject matter content usually introduced in postsecondary settings" (see also Semb et al. [1993]).

We recruited students by email approximately 5 months after the end of the course and asked them to complete a very short follow-up survey along with the same recommender systems knowledge test that had been administered preclass and postclass. The only incentive we offered was a copy of the results of our study. Out of 261 students, 125 responded in some way, for a response rate of 47.9%. Of these, 119 completed the follow-up knowledge test. While these response rates are quite high (in our experience) for uncompensated follow-up studies, we were not able to conduct a nonrespondent study and can only make claims about representativeness based on the response profile of the follow-up survey takers.

Appropriate bivariate tests revealed that follow-up survey takers were quite similar to the larger body of recommender systems students, with no significant differences between the groups in terms of sex, US residence, professional status, English proficiency, motivations for taking the course, and so on. Significant differences did emerge in two areas: follow-up survey takers reported taking fewer courses concurrently with the recommender systems course, and they reported a slightly higher mean age. Furthermore, students who took the follow-up survey appeared to be an elite group in terms of their pre-course recommender systems knowledge (almost 7 percentage points higher than the larger group) and their postcourse knowledge (also about 7 percentage points higher).

We used a paired-samples t-test to examine student knowledge retention as measured by the postcourse and follow-up tests. Table XII shows we found a significant difference in mean scores (about one third of a standard deviation) favoring the posttest, indicating that while students had lost some ground, they still retained about 92.6% of the recommender systems knowledge they ended the course with. This result is strong, in that it shows knowledge retention that well exceeds the average retention figures cited by Pascarella and Terenzini [2005], but we must acknowledge that a selection effect may be in play, and that students with lower postcourse knowledge scores might have lower retention.

Table XIII compares follow-up test scores for face-to-face and MOOC students, and we found a nominal mean difference of 2.34 percentage points favoring the face-to-face

Table XIII. Knowledge Retention (Postcourse vs. Follow-up): Face-to-Face Students and MOOC Participants

Group	Mean follow-up score	Standard deviation	N	Significance
Face-to-face	72.50	12.94	6	$p = .696$
MOOC	70.16	14.19	91	

Table XIV. Knowledge Retention (Postcourse vs. Follow-up) of Face-to-Face Students and MOOC Participants (Combined), by Concepts and Programming Tracks

Group	Mean follow-up score	Standard deviation	N	Significance
Concepts	65.22	14.58	23	$p = .046$
Programming	71.89	13.62	74	

students. As was the case with normalized knowledge gains, the very low N in the face-to-face group limits the statistical tests that can validly be performed.

Table XIV compares follow-up test scores for concepts and programming students, and we discovered a statistically significant mean difference of 6.67 percentage points favoring the programming students. This difference was statistically significant and represented an effect size of nearly half of a standard deviation.

Finally, we wondered if retention was related to whether the students reported using what they learned in the course in the time between the end of the course and the follow-up survey. We found no significant difference in retention scores based on this response.

Result: Among students who responded to a 5-month follow-up, most student learning gains were retained after 5 months. Programming students retained more than concepts students did; very limited data on face-to-face students suggests their retention is at least as high as that of online-only students.

6. STUDENT EVALUATION AND SURVEY RESULTS

6.1. Department and Program Reputation

One common motivation often cited by academic leaders for offering courses in MOOC format is the hope that doing so will enhance the reputation of the department or program offering those courses. We tried to measure changes in the reputation of the University of Minnesota's Department of Computer Science and Engineering by asking students the following question on both the precourse and postcourse surveys:

With respect to overall academic quality, what is your impression of the Computer Science department at the University of Minnesota?

1. One of the top 2 or 3 in the world
2. One of the top 10 in the world
3. One of the top 25 in the world
4. One of the top 50 in the world
5. One of the top 100 in the world
6. Don't know

We found that taking the Recommender Systems course appears to give students an impression of the quality of the department. From precourse survey to postcourse survey, the percentage of "don't know" answers dropped from 74.7% to 32.9%. We also note that this means most students responding enrolled in the course despite not knowing the quality of the department. Given that this course was offered only by one program, this may not be surprising. We think it would be interesting to look at the role perceptions of school, department, and instructor quality have in course selection among MOOC students.

We also found that the reputation of the department improved significantly in the eyes of students who answered the reputation question both precourse and postcourse. (A paired t-test showed a mean difference of .377 on a 5-point scale, $p = .002$, for a moderate effect size of slightly over one third of a standard deviation.) This improvement is welcome, but we should note once more that a confounding selection effect is quite possible here. In other words, it may be that the students who did not answer the reputation question on the postcourse survey would not have rated the department highly.

Interestingly, there was no significant correlation between reputation ratings and any of the other survey- or performance-based variables we tested, including student ratings of the course or instructor; course grades; normalized gains; student self-ratings how much they learned in the class, or how difficult the class was. This leaves us with something of a mystery: What drives students' perceptions of the quality of the department?

6.2. Other Quantitative Follow-Up Questions

As we designed our follow-up study, we asked a set of questions linked primarily to our own curiosity about the value of the course and how to improve it going forward. We recognize the limitations of this method (n = 129, self-selected) but found the results interesting. Specific results of this survey included:

- 95% of respondents responded "agree" (23%) or "strongly agree" (72%) to the statement that the course was effective in helping them learn about recommender systems;
- 60% of students felt the longer, 14-week format was most effective for the course material; 19% would have preferred smaller, shorter courses; 21% had no preference;
- 84% of students felt they took the right track; only 4% felt they completed the wrong track; and
- 54% of students have had the opportunity to use what they learned since the end of the course.

7. QUALITATIVE RESULTS AND ANECDOTES

As with any educational research, many of the lessons we learned come from the rich set of interactions with the course students. Too often, the dominant message from MOOCs is about the large number of students who sign up but do not complete (and in many cases do not even start). This may be an inevitable, and not undesirable, consequence of free registration—it encourages students to enroll in more courses than they intend to finish and to make choices over time. What impresses us, however, is the deep commitment from many of the students who take the course seriously, and the potential for impact in ways that exceed that of the traditional classroom.

Two anecdotal stories stand out. Early in our course (during module 2, the module on nonpersonalized recommenders), we saw a post from an enrolled student linking to the website he operated (a marketplace for rare coins) showing that he had incorporated the product association recommender technique we had taught the week before into the site. A few weeks later, we heard from a set of students in Russia who were incorporating their new understanding of recommender systems into a web-business consulting service. These are not isolated examples; the class forums and private conversations show extensive interest in "use-it-now" learning.

These indications of impact and the separate pieces of positive feedback (it was wonderful meeting some of the students from China at the Recommender Systems conference in Hong Kong) warm the hearts of faculty; sometimes this warmth is quite needed. We learned early that the free nature of such courses does not prevent vehement requests from students who want the course customized to their needs and

goals (we stated up front the need for Java in the programming assignments, and the use of LensKit, but a vocal contingent still protested regularly, explaining why we could and should instead build the course around their preferred set of tools). Overall, however, we found students to be quite reasonable. We appreciated the humor when another student later replied with a humorous post explaining why we should teach the course in Fortran-77, his preferred language. "After all," the student said, "the whole point of free education is to have instructors create exactly what you want."

These effects are also long-lasting. Both enrolled students and prospective future students have continued to approach both co-instructors over the 14 months since the end of the course. A scan of emailed requests shows more than 170 specific contacts from prospective students (most of whom found the course online and viewed "preview" videos) and more than 25 contacts from past students. There is no question that offering a MOOC creates a level of public visibility not associated with traditional classroom teaching.

We had some experiences that cause us to wonder about the as-yet unsettled culture of these MOOCs. Numerous students complained about assignments that we were "forcing them to do," indicating that they felt these assignments should be optional. Our reply that everything is optional met with responses that these students felt their grade, and statements of accomplishment, were a critical point of pride. They were clearly very grade-concerned. Other students' protests that the point of open education is to be able to pick and choose did not resonate with them. We valued greatly the pickers and choosers, the students who did the assignments they found valuable and skipped what they did not. But the question of whether MOOCs should be designed as rigorous with students opting out or minimal with students opting in for more rigor is still an unsettled one.

Moreover, our experience with grade-conscious enrollees and with pickers and choosers leads us to believe that, as educators, we lack effective metrics for the success of MOOCs. High dropout rates seem artificial (especially in a context where few students commit more than a click when enrolling). And even measures such as dropouts after completing a certain quantum of material (e.g., dropouts after 20% of assignments) do not reflect the reality that some students get what they need without completing the course. We reflect more on this in the conclusions.

In our experience, students really hated peer grading (not uniformly, but a large vocal number, with almost no students voicing support for it). We recognize this differs from the experience of some others and suspect it has a lot to do with the nature of what is graded. Our biggest challenge was in getting students to grade work where the point of peer grading was not qualitative evaluation of creative and divergent work, but rather almost mechanical grading of work too complex to automate (e.g., some written analysis exercises where the results could not be reduced to numbers or parseable strings, and one programming assignment where we graded the results automatically but had students peer-grade the code as a sanity check that the submitted work was not simply calculated results but was actually programmed). Our preference would be to avoid grading these types of exercises entirely, but many students value getting grades. Automation is good as far as it goes, but it puts significant limits on the assignment design. Perhaps at some point the answer will be small fees and paid crowdsourcing of grading. For now, we do not have a solution to this challenge, but we think it is a factor worth serious consideration when designing course assessments and assignments.

At the end of the course, we held a lengthy debriefing session with the on-campus students. For many of them, this debriefing was one of the few times they had come to a face-to-face session. We were somewhat surprised that the overwhelming number of students preferred this format to a traditional lecture-based course. Students cited the benefits of being able to review the lectures at their own pace and at their own

convenience. Some nonnative English speakers cited the benefits of being able to pause and replay things that were hard to understand. In general, the reactions of on-campus students were similar to those of online students, though the on-campus students were particularly unhappy with peer-grading of their work by the online students (though we did offer the option of a TA review to correct serious misgradings).

In our follow-up study, we asked people how they used what they learned in the course—64 students answered this question. The most common response was use at work, followed by use in schooling. A number of students responded that they were incorporating recommender systems into entrepreneurial efforts (start-up companies and the like). We were also delighted to see that some students found the content of the course useful outside recommender systems (applying the mathematics and algorithms to other problems).

One last small lesson: We came into this course skeptical about the ability to offer meaningful examinations at scale. We have rarely used multiple-choice questions, particularly when dealing with advanced topics such as recommender system design and detailed algorithmics. It was, therefore, a pleasant surprise that the exams we did use proved to be consistent and effective. We had two exams, each covering 7 weeks of the course. Each exam had 36 multiple-choice questions divided into three 12-question, 30-minute timed parts. The separate parts were designed to keep the time needed for each chunk small, to limit the potential harm of a student being unable to complete a part, and to help students assess whether they needed to study further before taking the second part. Exam scores correlated well between the parts (correlations between adjacent parts ranged from 0.7 to 0.71; correlations between parts 1 and 3 of each exam were 0.68 (exam 1) and 0.64 (exam 2); and the exams were one of the areas that generated the fewest complaints from students. The time needed to write good multiple-choice questions is fairly high, but the time saved in grading is substantial. We are experimenting with more multiple-choice exams in future face-to-face classes.

8. DISCUSSION AND LESSONS

Based on the results presented earlier, it is clear that this massive, online, open course attracted a significant and diverse group of students, that they come to the course with different goals and intentions, that those who persist leave with substantially more knowledge of the subject matter than they arrived with, and that those who agreed to follow-up testing showed significant retention of those knowledge gains 5 months after completing the course.

Predicting course completion is hard—about the only precourse factor that we found to be highly predictive is intention, and even this predictive power is probably dominated by the fact that those who do not intend to complete usually do not. Predicting knowledge gains is even harder. The good news is that we found knowledge gains did not correlate significantly with age, sex, student level, or motivation for taking the course. The factors that were predictive all relate to effort, prior courses, and baseline knowledge (in what appears to be only a negative effect resulting from ceiling effects).

This case study—both the underlying course and its evaluation—differs from prior MOOC studies in several ways.

First, the course was simultaneously offered as a typical online-only MOOC and as a flipped-classroom face-to-face course. Survey numbers from the face-to-face class were too low to permit statistically significant conclusions, but what data we have shows comparable-or-better knowledge gains and retention, and our qualitative experiences are consistent with that finding. Face-to-face students liked this format, though their reasons varied from simple convenience to the benefits of being able to experience lectures at their own pace. Supplemental face-to-face activities were poorly attended (though valued by some students) and are an area for possible improvement.

Second, the course mixed programming and nonprogramming students together into a two-track course model. Pedagogically, this approach worked well, but there are two practical reasons we feel it may not be worth repeating. The tools supporting grade and completion-reporting do not support this model. We recommend it may be useful to explore ways to formalize such multitrack learning, whether through tracks or through separate courses that share online infrastructure (including discussion forums). Also, in this particular case, it would be useful to enable other versions of the programming components (primarily those using other tools), which is difficult to incorporate in this model.

Third, we have performed extensive evaluation of student learning outcomes, measuring baseline knowledge, knowledge at the end of the course, and knowledge 5 months thereafter. We were gratified to find significant knowledge gains and retention, and to find that this MOOC was successful in reaching across age, sex, and other demographic categories.

We also wanted to share a few useful lessons from this course with others who may teach similar courses in the future:

- We found that a successful and motivating activity was the generation of a class-specific dataset used for the assignments. We had students contribute movie ratings (over 5,000 students contributed ratings to up to 100 movies in the first week) and then distributed that dataset to the course through the assignments. Students could attach an identifier to their data line to see how each recommender performed for them.
- We also assigned personal test data to each student for many of the programming assignments. We wrote scripts that assigned test cases to users (usually five of them). This increased the burden on grading software (which had to verify the correct test cases), and on pretesting (we had to ensure no student received degenerate cases), but it provided both interest and a barrier to cheating by passing around correct datasets.
- We found the use of open-source infrastructure for distributing course software was a big success. We used Maven, a tool that made it possible for distribution to be close to automatic. While some students had little experience with installing software, we found most were able to complete this task quickly without need for help.
- We struggled with the tension between course semantics and the Coursera-tool concepts for our assignments (this is not specific to Coursera, we expect it would be true with any similar tool). We had concepts we wanted to communicate: written assignments, programming assignments, exams, and so forth. The problem is that the tool has its own concepts (e.g., quizzes, exams, homework, programming assignments) and each have different types of grading options. This led to confusion. We had "written assignments" that needed to be "programming assignments" to support grading scripts, and "programming assignments" that were implemented as "quizzes." This is a challenge that could be helped by better hiding the tool implementation or having a complete mapping between concepts and grading options.

Finally, we have been asked what our plans are for the course in the future. After reflection, we have come to recognize that the effort involved in offering this course as we have done so is quite high, mostly due to individual effort associated with personalized grading. We also recognize that the fixed schedule of the course is an impediment to many learners. As an interim step we made the lecture material open shortly after closing the course. This has helped address immediate student demand (with between a dozen and a few hundred video views each week), but we receive regular requests to "open the assignments" that require substantial infrastructure.

We then undertook an effort to redesign the assignments to remove peer-grading (focusing on objectively-gradable problems) and to provide personalized test cases from a static data set (rather than a class-generated one). We are in the process of releasing the course in two components as an "on-demand" course with Coursera. The first component (launched in January 2015) corresponds to the concepts track of the course, with updated assignments that are automatically graded and that include more hand-and-spreadsheet computation. The second component is a lab course (intended as an add-on to provide the programming parts) that we hope to launch in Summer 2015. Creating a separate lab course for programming will allow alternative versions of the lab course to be developed to support students who prefer to use other tools for programming the assignments.

We will be exploring both the effects of those changes (through comparative assessment) and better ways of integrating a live class with the MOOC (including separate qualitative live exercises as a supplement) over the coming year (we have two separate live classes scheduled). We also plan an experimental assessment of providing past course discussion threads to students in the context of self-paced study.

ACKNOWLEDGMENTS

A massive course such as this one is the product of a team. We would like to thank our entire course team, with special thanks to our video team, led by James Ondrey and David Lindeman, our online assignment consultant Ken Reily, and especially our teaching assistant Michael Ludwig. We also thank our many colleagues who taught massive-scale courses before us and generously shared their advice. With this article, we remember our colleague and mentor John Riedl, who jointly conceived this course but passed away before it could be brought to fruition. And we want to especially thank our students for their patience as we have explored this space with them, for their participation in the surveys and knowledge tests we have used for this research, and for their incredibly valuable feedback throughout the course. The tools used for this course were developed with the support of NSF Award #1017697, and the development of this course was supported by the University of Minnesota.

REFERENCES

John P. Buerck, Srikanth P. Mudigonda, Stephanie E. Mooshegian, Kyle Collins, Nicholas Grimm, Kristen Bonney, and Hadley Kombrink. 2013. Predicting non-traditional student learning outcomes using data analytics–a pilot research study. *Journal of the Computer Science College* 28, 5 (May 2013), 260–265.

J. Champaign, K. F. Colvin, A. Liu, C. Fredericks, D. Seaton, and D. E. Pritchard. 2014. Correlating skill and improvement in 2 MOOCs with a student's time on tasks. *Proceedings of the 1st ACM Conference on Learning @ Scale*. ACM, New York, 11–20.

Doug Clow. 2013. MOOCs and the funnel of participation. In *Proceedings of the 3rd International Conference on Learning Analytics and Knowledge (LAK'13)*. ACM, New York, NY, 185–189.

K. Colvin, J. Champaign, A. Liu, Q. Zhou, C. Fredericks, and D. Pritchard. 2014. Learning in an introductory physics MOOC: All cohorts learn equally, including an on-campus class. *International Review of Research in Open and Distance Learning* 15, 4.

J. DeBoer, A. D. Ho, G. S. Stump, and L. Breslow. 2014. Changing "course": Reconceptualizing educational variables for massive open online courses. *Educational Researcher* 43, 2, 74–84.

M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl. 2011. Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*. ACM, New York, NY, 133–140.

E. J. Emanuel. 2013. Online education: MOOCs taken by educated few. *Nature* 503, 7476, 342–342.

René F. Kizilcec, Chris Piech, and Emily Schneider. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the 3rd International Conference on Learning Analytics and Knowledge*.

J. A. Konstan, J. D. Walker, D. C. Brooks, K. Brown, and M. D. Ekstrand. 2014. Teaching recommender systems at large scale: Evaluation and lessons learned from a hybrid MOOC. In *Proceedings of the 1st ACM Conference on Learning@Scale.* 61–70.

D. A. McConnell, D. N. Steer, K. D. Owens, C. Knight. 2005. How students think: Implications for learning in introductory geoscience courses. *Journal of Geoscience Education*, 53, 4, 462–470.

B. Means, Y. Toyama, R. Murphy, M. Bakia, and K. Jones. 2010. *Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies*. Center for Technology in Learning, U.S. Department of Education, Washington, DC.

E. T. Pascarella and P. T. Terenzini. 2005. *How College Affects Students: Volume 2, A Third Decade of Research*. Jossey-Bass, San Francisco, CA.

G. B. Semb, J. A. Ellis, and J. Araujo. 1993. Long-term memory for knowledge learned in school. *Journal of Educational Psychology* 85, 2, 305–316.

Julia Wilkowski, Amit Deutsch, and Daniel M. Russell. 2014. Student skill and goal achievement in the mapping with google MOOC. In *Proceedings of the 1st ACM conference on Learning @ Scale Conference* (*L@S'14).* ACM, New York, NY, 3–10.

Received June 2014; revised January 2015; accepted January 2015