Crowdsourcing Education:
A Game-Theoretic Analysis of Contribution and Learning in Peer to Peer Education Platforms

A thesis presented

by

Spencer de Mars

To

Applied Mathematics

in partial fulfillment of the honors requirements

for the degree of

Bachelor of Arts

Harvard College

Cambridge, Massachusetts

March 29th, 2013

Abstract

The Internet has provided a unique opportunity for many alternatives to the traditional classroom education, and one emerging example is the advent of online peer to peer or crowdsourced education platforms. On such websites any user is free to contribute courses in his or her area of expertise, and the platform aggregates these crowd submissions. In this paper we model the contribution and learning behavior in such a system, with the goal of determining the effect content presentation can have on contribution incentives, the distribution of contributed material with respect to difficulty level, and ultimately learning outcomes.

Our primary contribution in accomplishing this goal is the construction of a rich game-theoretic model capturing the learning and contribution mechanics. The model includes two populations, learners and potential contributors, in which each agent from either population has a discrete skill level. The system evolves over a series of discrete time steps which alternate between rounds of learning and contribution. We represent contribution in each time step as a complete information simultaneous game under a free-entry Nash Equilibrium, such that contributors may elect whether to participate and conditional upon participating create a course and an associated fee. Potential contributors are strategic in an impatient sense in that they seek to maximize profit in the next round of learning.

To model the effect of content organization we introduce the notion of a *mechanism* as the platform's method of matching a learner to a course, which the learner can enroll in or decline. As a baseline mechanism and simplification of unorganized content we consider random course delivery. We show this mechanism can lead to inefficiencies, including many redundant courses targeted at novices and a lack of advanced content. Next we introduce an alternative mechanism which breaks down the

size of each contributed course and enforces a clear difficulty ordering among aggregated content. We prove that under this alternative mechanism there is a desirable contribution pattern in that more skilled contributors are guaranteed to create more difficult courses. Additionally through simulation we show that this alternative mechanism can lead to better learning outcomes than random course delivery, facilitating more than twice the increase in aggregate learning utility in randomized simulations. These results suggest that enforcing a more transparent ordering of contributed content can lead to more complete contribution across difficulty levels and thus better learning outcomes.

Acknowledgments

I want to thank Yiling Chen for her help as a thoughtful and supportive advisor. Her constant feedback, insight, and encouragement were essential to the formulation of this thesis. I am also grateful to Andrew Mao for his valuable guidance and feedback throughout the process of conceptualizing and constructing the model.

Contents

1 Introduction

The Internet is already disrupting the traditional classroom educational model, and all signs indicate its role will only grow more important. The web allows educational content to scale far beyond the traditional classroom, eroding barriers and potentially allowing multitudes of students unprecedented access to teachers and educational content.

One manifestation of the online education movement is the appearance of crowd-sourced education platforms, which seek to in a sense democratize education through peer to peer teaching. The idea behind these platforms is to allow anyone to become a teacher and create educational courses. The website then aggregates this content and allows visitors to browse the courses and learn at their leisure. This emerging educational model is referenced by many names including crowdsourcing, user-generated content, and peer to peer learning. The key to this model is relying not on a dedicated staff of teachers for educational content but instead submissions from the user base or crowd.

Many real-world companies and websites have been built around this idea of peer to peer education. Some of the most popular examples include Udemy, Skillshare, Peer to Peer University, and WizIq. For example Udemy, launched in 2010, now offers more than 1500 courses in categories ranging from Music to Technology[1], and reportedly has enrolled over 600,000 students [12]. Teachers include famous successful businesspeople such as Yahoo! CEO Marissa Mayer and Facebook CEO Mark Zucherberg. Furthermore, some courses have grossed millions of dollars in revenue[2]. The strong traction Udemy has gained in such a short time, combined with the many other

[1] 1510 courses as of March 3, 2013

[2] For example, the course "Microsoft Excel 2010 Course Beginners/Intermediate Training" costs $99 and had 28022 students as for March 4, 2013, for a total revenue of over $2.7 million

companies competing in this space, indicates a strong potential and real investment in this type of learning platform.

The popularity and empirical demand for this platform indicates the practical importance of understanding the behavior of these peer to peer education systems. Furthermore it is clear that learning outcomes depend on the website and its content structure in two important ways: Firstly the organization of content impacts the learning experience directly, in that recommending or emphasizing certain courses over others will drastically affect which courses students take and therefore how much they learn. Secondly there is the indirect effect of considering what kind of content contributors are incentivized to create, which shapes the distribution of the content students can pick from. Broadly speaking the goal of this paper is to model and better understand the effect of the content aggregation and presentation on contribution and learning outcomes.

1.1 Contribution and Outline

We believe this paper is the first to employ the tools of game theory to model the incentives and contribution behavior in the unique environment of a crowdsourced education platform. To this end our primary contribution is a rich and complete model capturing both learning and contribution mechanics within such a platform.

We focus on the unique factors which differentiate the education setting from other applications of user-generated content. Learning in many subjects requires a depth of understanding, and so in most education settings there is a natural progression of interrelated topics of escalating difficulty. From our perspective the goal of such an online platform is to facilitate this increase in the skill level of learners as they master subjects. This creates a unique and complex audience for content, both because

the learners are asymmetric with respect to skill and because those skill levels are changing over time as learning takes place.

To represent this concept of learning we consider learners to each have a discrete skill level. Learning takes place through a series of discrete time steps in which learners attempt *lessons* and probabilistically increase their respective skill levels. Learners are motivated by the utility gains from learning but also face an effort cost of attempting lessons.

For contribution we assume a smaller pool of potential contributors, each of whom also has an associated skill level. In each time step these contributors play a simultaneous complete information game, in which each contributor seeks to maximize the profit generated in the next round of learning. These games are free-entry, such that potential contributors choose whether to contribute and then conditional upon contributing choose a *course*, or bundle of lessons, as well as a fee to charge for that course. Contributors earn revenue from students paying their fee to enroll in their courses, but also face an effort cost to creating a course, which depends on the contributor's skill level.

This Nash Equilibrium depends on the expected revenue of contributed courses, and this is where the organizational structure of the platform plays its role. We isolate the effect of content aggregation and presentation by modeling this content structure as a *mechanism* by which the platform selects which course to present to a given learner.

When presented with a course learners will myopically decide whether to enroll by comparing the expected utility gain from the course to the cost of the course's fee. The behavior of the mechanism combined with the learners' decision process defines the expected revenue for each course in the contributors' strategy space. This allows us to calculate the Nash Equilibrium contribution in each round, yielding a dynamic

system consisting of a series of simultaneous games alternating with enrollment and learning on the part of the learners. Through this model we can study how the choice of mechanism affects the evolution of the system in terms of the contribution and learning process.

As a baseline mechanism we consider the *independent course* mechanism, in which each learner is presented a random course. We prove that under this mechanism there will initially be many courses targeted at novices, and demonstrate that the mechanism can also lead to a host of inefficient learning outcomes including content redundancy and a lack of advanced content for more skilled learners.

We next present an alternative mechanism, termed the *ordered lesson* mechanism, which limits contributed courses to a single lesson and enforces an ordered structure of escalating course difficulty when pairing learners with courses. We prove that this mechanism guarantees the desirable property that under all circumstances more skilled contributors will create more difficult lessons. Using visualizations and simulations under both mechanisms we further argue that this alternative mechanism tends to lead to a more complete distribution of content over various difficulties and thus better learning outcomes, facilitating more than twice the aggregate learning utility gain over the baseline mechanism in 100 randomized simulations.

The remainder of the paper is outlined as follows: Section 2 more closely examines real-world examples of peer to peer education websites and motivates this paper by providing concrete examples of the contrast in content organization between crowdsourced sites and other educational settings. Section 3 introduces the dynamic learning and contribution model which is used for both mechanisms. Section 4 analyzes the independent course mechanism and provides the theoretical and simulated results under this mechanism. Section 5 introduces and analyzes the ordered lesson mechanism, proving the positive theoretical results of this mechanism. Section 6 con-

trasts the two mechanisms, providing side by side simulation visualizations as well as as aggregate learning metrics from a series of randomized simulations. Section 7 includes discussion and directions for future research.

1.2 Related Work

While we believe this is the first paper to take a game-theoretical approach to studying crowdsourced education platforms, there has been some recent qualitative work in this area. Weld et. al. [10] make the case for a variety of ways in which crowdsourcing could improve education, and highlight the potential of using the crowd to both create and organize educational content. John Seeley Brown and Richard P. Adler [2] argue that Web 2.0 applications will allow education to become a more participatory medium, allowing new collaboration and multiple modes of learning.

In terms of structure and methodology our paper is more closely related to work employing game-theoretic analysis to study crowdsourced or user-generated content in other settings.

There is a comparatively older and richer body of work analyzing crowdsourced contests [1][5] [3]. In these settings a pool of agents exert effort to create competing contributions, and rewards are distributed to those contributions judged to be best. These papers seek to find a distribution of rewards to the top contributors which encourages the the most effort and best submissions to be put forward by the crowd.

Our paper is more closely related to recent work employing game-theoretic analysis to study crowdsourcing from a human computation standpoint. Jain and Parkes [9] argue for a large role of game theory in studying and designing human computation systems such as games with a purpose. Additionally in [8] they employ game theory to demonstrate positive equilibrium results in the image-labeling game ESP. In this

same vein Jain et. al. [7] analyze the effectiveness of the virtual point reward system on Yahoo! answers, the popular question and answer forum. They find that while the point system encourages optimal equilibrium behavior for some types of questions it fails for certain question types, and further design point systems which encourage optimal behavior for these question types. Our paper builds on these ideas in that it also focuses on designing incentive structures to encourage the crowd to contribute useful content, in our case in an educational context. One notable difference is that in our model contributing agents are motivated by monetary profit, unlike in games with a purpose or question and answer forums.

There has also been related work studying user-generated content in the context of online communities. Ghosh et al.[6] [4] build a model of a generalized platform consisting of potential contributors and content viewers, and use game theory to evaluate the Nash Equilibrium outcomes for content contribution. This work assumes viewers rate the content, and then defines the website's mechanism as the way the website uses these ratings to direct viewer attention to each piece of content. Gosh et al. show that if the website uses the right mechanism to display higher-rated content in a certain way, the contributors will be induced to create optimal quality content as the number of viewers diverges.

Our work is analogous in that we study a website's content delivery mechanism and analyzing its effects on contribution. However instead of focusing on the quality of contributions we focus on the *difficulty distribution* of the contributed material, a feature unique to the education setting. Furthermore our model becomes very different and much more complex both because contributors and learners have associated skill levels, which breaks much of the symmetry, and because our model evolves over time, transforming the model into an evolutionary one and turning a single simultaneous game into a series of multiple simultaneous games with changing payoffs.

2 Empirical Motivation

We derive our motivation for this paper from a qualitative comparison of the content structures in existing crowdsourced and non-crowdsourced education settings. While in non-crowdsourced settings learning is usually presented as a tree or track of escalating difficulty, in current peer to peer platforms content is generally unorganized with respect to difficulty. This clear distinction begs the question of how these changes in organizational structure affect both the contribution incentives and learning outcomes.

In many traditional education settings there is an emphasis on building up expertise in a skill through incremental steps. Unless a single learning experience can guide a student from the beginner level to mastery, it is natural to think of different concepts building on one another. For example figure 1 demonstrates a handout detailing computer science classes at Harvard, in which classes are grouped by similar difficulty level with some classes acting as pre-requisites for others.

This concept of learning in a given subject as navigating a tree of escalating difficulty can be found in online settings as well. Consider Duolingo, a recently launched website allowing users to learn a foreign language online. It breaks down learning a language into many small lessons on vocabulary and grammar, allowing users to progress through a tree of skills towards gaining fluency (figure 2).

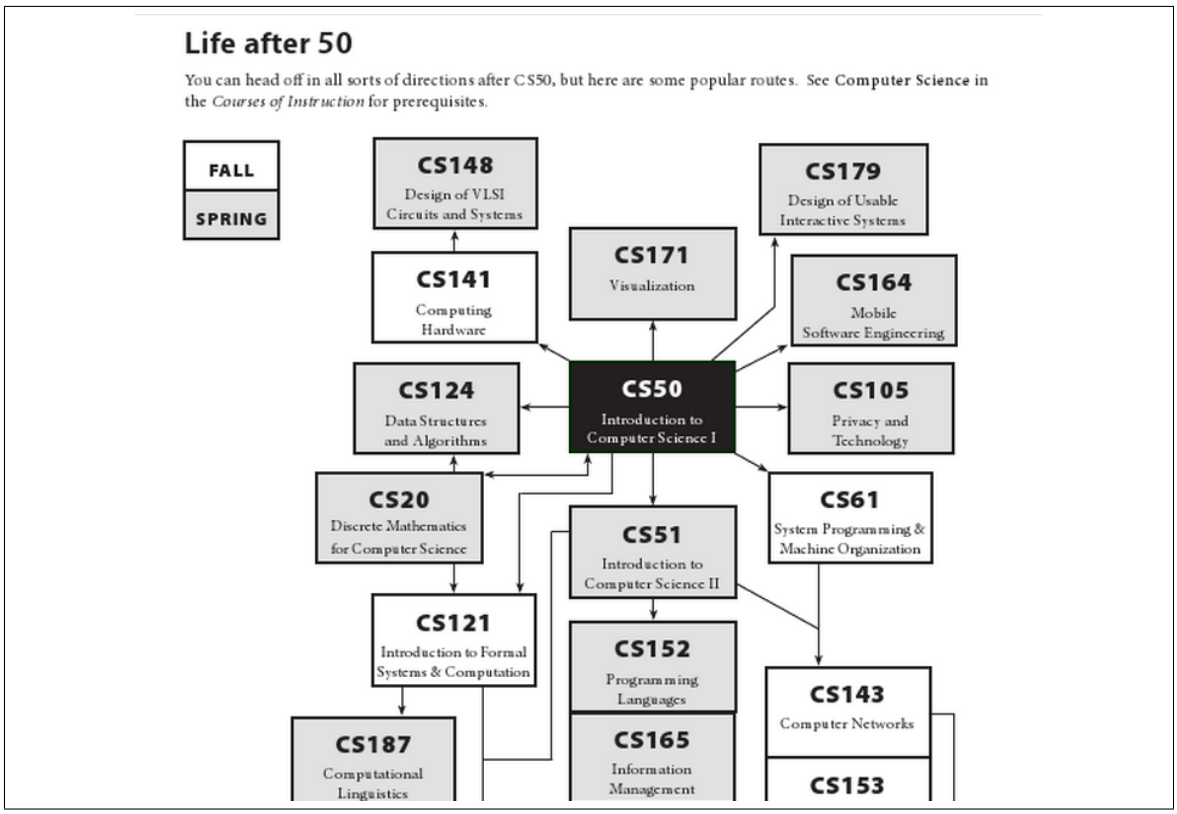


Figure 1: Guide to computer science classes at Harvard

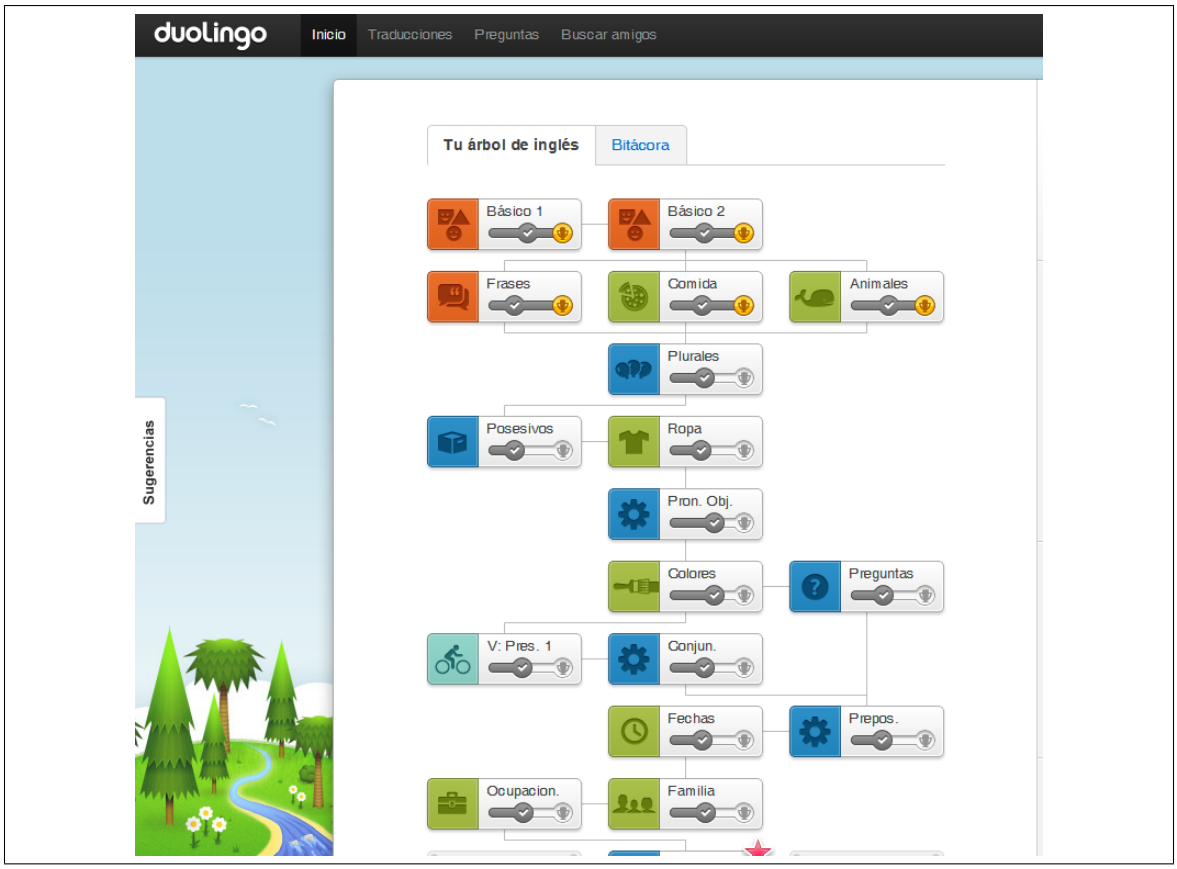


Figure 2: Duolingo language skill tree

Codecademy is another example of structured ordered learning online. It has a series of curated tracks on topics such as Javascript, Python, and APIs. Within each track there is a clear ordering and linear progression through topics, implying escalating difficulty and the layering of concepts as the student gains mastery (figure 3)

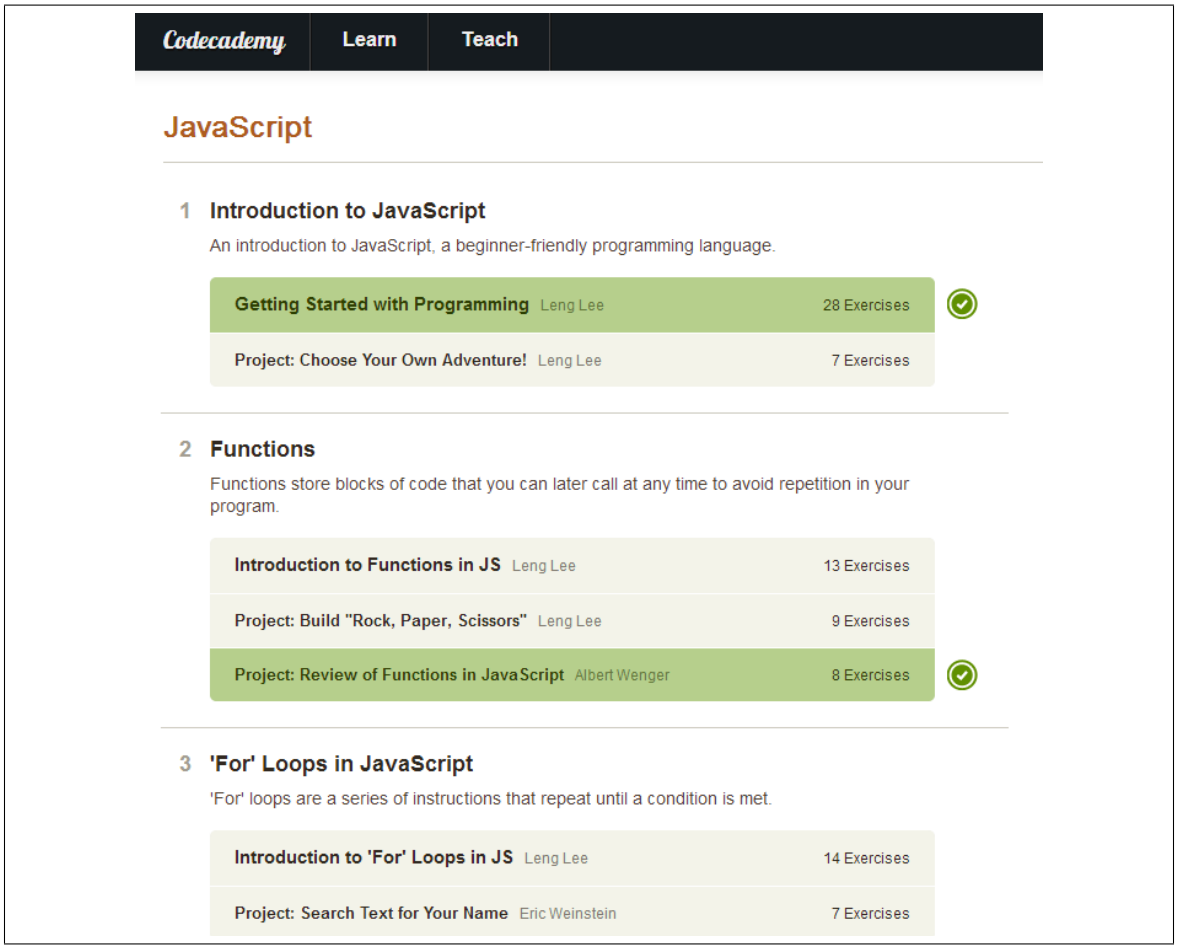


Figure 3: Codecademy Javascript track

Existing crowdsourced education platforms however almost all tend to use a less structured approach for their content. Generally each course offered is listed independently of the others, and so content creators are encouraged to make freestanding

courses not directly related to each other. This more open structure features courses that are organized by subject and searchable by title or teacher but do not include information about course difficulty (figures 4-6).

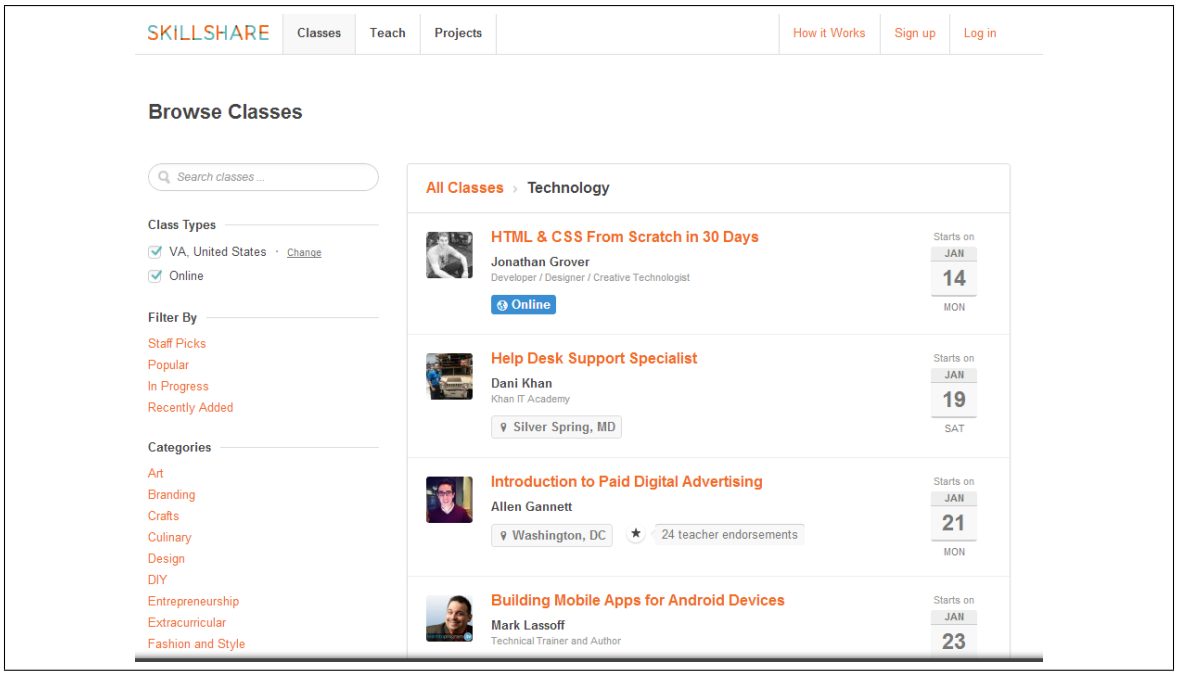


Figure 4: Skillshare Technology courses

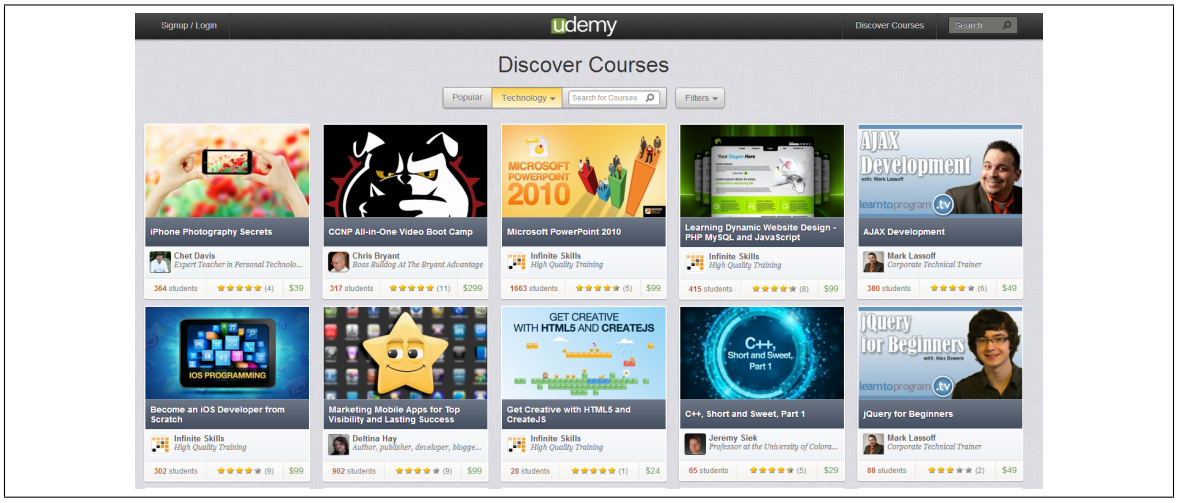


Figure 5: Udemy Technology courses

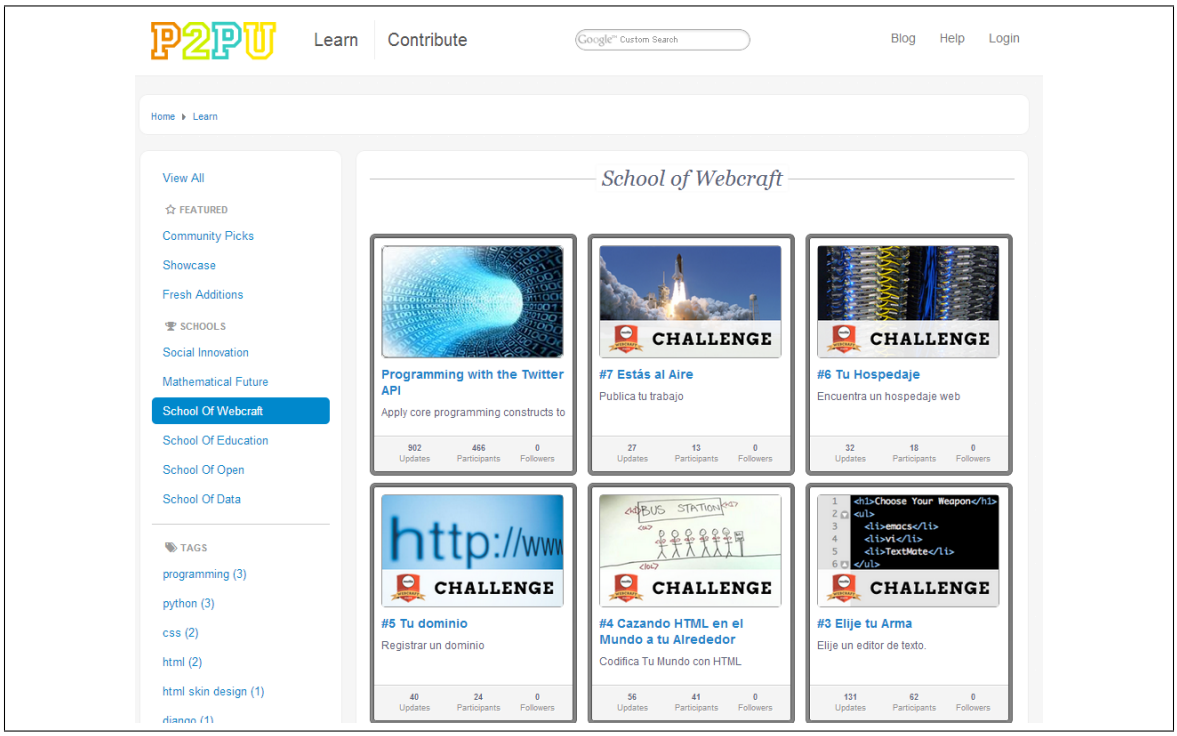


Figure 6: P2PU school of webcraft courses

Therefore none of these websites attempts to enforce ordering through a crowdsourced educational track, even though having a track or ordering is the dominant structure outside of these peer-to-peer sites. We acknowledge there may be good practical explanations for why crowdsourced platforms choose this presentation format. However this contrast justifies a closer examination of the ramifications of moving from ordered content to unordered content, in terms of the teaching and learning outcomes of the system. These two organizational schemes, ordered and unordered, provide the intuition and motivation for our two content delivery mechanisms.

3 Learning and Contribution Model

3.1 Introduction and Terminology

Our agents will consist of two separate pools, a large group of *learners* L and a smaller population of *potential contributors* C. Although any user of the website is free to become a teacher and contribute educational material, in practice only a small fraction of the community will have the interest and time to contribute. This separation of learners and contributors may seem counter to the idea of peer to peer education, but it has been shown in almost all peer production communities that a small fraction of users create the majority of the content [11]. This separation allows us to focus on the contributing agents in their role as contributors, as opposed to the majority of users who elect to only consume content.

This model represents teaching and learning within a specific subject area. Therefore we will represent skill or expertise as a discrete ordered skill level between 0 (novice) and n (mastery). Each agent, whether a learner or contributor, will have some skill level in $\{0, 1, \ldots, n\}$.

We will use two terms to refer to educational content on the platform: A *lesson* will refer to the smallest unit of material, one which we can consider too small to meaningfully break up among multiple contributors and small enough that it is realistic to assume a learner will either take the entire lesson or none of it. Each lesson will have an associated difficulty level $d \in \{1, \ldots, n\}$. A *course* is a bundle of one or more lessons along a contiguous block of difficulty levels. In our notation the course $[j, k]$ implies a set of lessons at difficulties $[j, j+1, \ldots, k]$. A course may consist of a single lesson, denoted $[j, j]$.

In our model the system will evolve over discrete time steps. In each time step there will be a round of content creation followed by a round of learning. We first focus on explaining the learning process, and then move on to introduce contribution.

3.2 Learning Mechanics

3.2.1 High Level Overview

A learner $i \in L$ is associated with three important properties: *skill level*, *state*, and *utility pools*. We use the notation $s_{i,t} \in \{0, 1, \ldots n\}$ to denote the skill of learner i at time t, which represents how much mastery learner i has over the subject matter. We use the notation $\delta_{i,t} \in \{$*browsing, enrolled, discouraged*$\}$ to denote the state of learner i at time t, which represents whether the learner is currently looking for a course, engaged in a course, or has left the website altogether. Each learner will have a course utility pool $B_{i,t} \in \mathcal{R}$ as well as a platform utility pool $V_{i,t} \in \mathcal{R}$, which represent how learner i feels about the current course he or she is enrolled in and the platform as a whole respectively. These will be explained in greater detail in section 3.2.3.

At a high level the learning process will take place in three phases:

1. Browsing and Enrollment Decision: All browsing learners will be presented

courses and must choose whether to enroll or decline.

2. Learning: All enrolled learners will attempt the next lesson in their current course, and update their skill level according to the result of this attempt.

3. Update: All browsing and enrolled learners will update the appropriate utility pools according to the utility change experienced in this time step, and update their state for the next time step as needed.

The decision process for whether a learner enrolls in a course requires an understanding of the learning mechanics, and therefore the steps are presented in the following order: The learning step is presented first, followed by the update step, then the browsing and enrollment specifics, and finally a detailed summary of the entire process.

3.2.2 Learning Phase

Learners are motivated by the utility they derive from reaching higher skill levels. We consider a population in which each learner has the identical concave utility function $u(s)$ over the learner's skill level s. To improve in skill level a learner must attempt and successfully complete a lesson at some higher difficulty level. If learner i with skill $s_{i,t}$ completes a lesson at level $k > s_{i,t}$, the learner will improve to level $s_{i,t+1} = k$. Note that the skill levels are ordered in that $0 < 1 < 2 \cdots < n$ but it is possible for $k - s_{i,t} > 1$, meaning that a learner can skip over intermediate levels up to a higher skill level upon successful completion of the higher difficulty lesson.

Agents successfully complete lessons probabilistically upon attempting them, and specifically learner i attempting a lesson at level k will succeed with probability $p(k - s_{i,t})$. This probability function by assumption only depends on $d = k - s_{i,t}$, the

difference between the lesson difficulty level and the learner skill level. Furthermore $p(d)$ is decreasing in d, in accordance with the intuition that it is less likely for a learner to successfully jump up many skill levels in one step. Therefore upon attempting a lesson at k, with probability $p(k - s_{i,t})$ the learner will pass and improve his or her skill level to $s_{i,t+1} = k$, and with probability $1 - p(k - s_{i,t})$ the learner will fail and remain at skill level $s_{i,t+1} = s_{i,t}$.

To represent the time and effort required in the learner's attempt of the lesson, there is an effort cost $e(k - s_{i,t})$ learner i exerts when taking the lesson. This effort cost is subtracted from the learner's utility regardless of whether the attempt results in a success or a failure. Like the probability function, the effort function depends only on the difference between k and $s_{i,t}$. The effort function $e(d)$ is an increasing function of d such that it takes more and more effort to try and increase multiple skill levels at once.

The table below summarizes these mechanics of learning, for learner i attempting a lesson at k:

Outcome	Probability	New Skill ($s_{i,t+1}$)	Utility Change
Success	$p(k - s_{i,t})$	k	$u(k) - u(s_{i,t}) - e(k - s_{i,t})$
Failure	$1 - p(k - s_{i,t})$	s	$-e(k - s_{i,t})$

Enrolled Learning Through a Course

Next we can consider courses and the process of learning through course enrollment. Recall that the course $[j, k]$ denotes a package of lessons at difficulties $[j, j + 1, \ldots k]$.

A state of $\delta_{i,t} = \textit{enrolled}$ means that learner i is in in the process of progressing through a course. In the learning phase of each time step all enrolled learners will attempt a single lesson from their respective courses. We will use the notation $[j, k]_{i,t}$ to refer to the difficulty range of whichever course learner i is enrolled in at time t.

If learner i has just enrolled in $[j,k]_{i,t}$ in the earlier browsing and enrollment phase of this time step, the learner will begin the course by attempting the appropriate lesson:

- If $s_{i,t} < j$ learner i will begin by attempting lesson j, which may involve attempting to jump up several skill levels if $j - s_{i,t} > 1$
- If $k > s_{i,t} \geq j$, learner i will begin by attempting lesson $s_{i,t}+1$ to avoid repeating material.

If learner i was already enrolled in the course, the learner will continue to take the course's lessons in order. That is upon successfully completing the lesson at j' at time t the learner will automatically attempt the next lesson at $j'+1$ at time $t+1$. Upon failing lesson j' at time t the learner will retake lesson j' at time $t+1$.

This process will continue until one of two things happen: Either the learner will successfully complete the course, meaning that the learner passes the most difficult lesson k, or the learner *quits* the course, meaning the learner stops taking lessons from this course. The concept of quitting reflects the reality that users get frustrated, especially after they have attempted a lesson and failed, possibly multiple times. This frustration can cause them to quit the course entirely instead of continuing to attempt lessons.

3.2.3 Update Phase

Course Pool Update and Quitting

To capture the mechanics of quitting there must be a way for the negative utility experienced when failing lessons to accumulate and cause the learner to stop taking lessons from that course. For this we introduce the idea of a course utility pool for

each enrolled learner. The utility pool $B_{i,t}$ represents the goodwill or satisfaction learner i feels towards the particular course i is enrolled in at time t.

When a learner first enrolls in a course he or she will begin with an identical initial pool of B_0, such that if i enrolls in $[j,k]$ at time s, $B_{i,s} = B_0$. This constant is symmetric across learners and can be thought of as the patience or tolerance for failure with which the learners approach each course. Intuitively this means that learners approach each new course with the same initial goodwill, regardless of past experience with other courses. Next any utility change resulting from an attempt of a lesson, whether positive from passing a lesson or negative from failing one, is added to the utility pool.

This update is shown in the table below, for learner i at $s_{i,t} = j'$ attempting lesson $j'+1$ in course $[j,k]_{i,t}$:

Outcome	$s_{i,t+1}$	$B_{i,t+1}$
Success	$j'+1$	$B_{i,t} + u(j'+1) - u(j') - e(1)$
Failure	j'	$B_{i,t} - e(1)$

If after the update $B_{i,t+1} < 0$, then learner i will quit the course out of frustration, meaning the learner is no longer enrolled and will stop taking lessons from this course. The fact that $B_{i,t}$ can both rise and fall captures the intuition that positive experiences with a course give a learner more goodwill towards that course and thus greater tolerance for future frustration before quitting.

Platform Utility and Discouragement

Just as a learner may get frustrated with a particular course, learners may also eventually get frustrated with the entire platform. Quitting courses takes a psychological toll on the learner and involves wasted time with no gain in skill, and with enough of

these bad experiences a learner may leave. If a learner leaves the platform altogether we term the learner's state as $\delta_{i,t} = \textit{discouraged}$. A discouraged learner is permanently frozen in terms of skill level and will not browse or enroll in any new courses.

To model this process of discouragement we will use the same utility pool concept as we did for courses, but using a separate pool to represent goodwill towards the entire platform. The pool $V_{i,t}$ represents the goodwill of i towards the entire platform at time t. At time zero all learners will start with an identical utility pool V_0 for the website, that is $V_{i,0} = V_0 \; \forall i \in L$. V_0 represents the learners' initial patience for the platform as a whole, or how many bad courses learners will tolerate before neglecting to try any new courses and leaving the platform altogether.

The course utility pool $B_{i,t}$ is updated according to the utility change of an attempted lesson. Analogously, the platform utility pool $V_{i,t}$ is updated after an experience with a course. This means when a learner completes or quits a course, the utility change associated with the entire course experience will be added to that learner's platform pool $V_{i,t}$. Given that learner i quits or passes a course at time t, this utility change is simply $B_0 - B_{i,t+1}$ because $B_{i,t+1}$ has been updated each time step according to the learner's experience with each lesson in the course. If after completing or quitting a course the updated utility pool $V_{i,t+1}$ is less than 0, then the learner will become *discouraged*. If after completing or quitting a course $V_{i,t+1} \geq 0$, the learner will return to a *browsing* state because the learner is willing to take a new course.

The relationship between quitting and becoming discouraged is outlined in the table below:

$\delta_{i,t}$, Recent Action	$B_{i,t+1} < 0$?	ΔV	$V_{i,t+1} < 0$?	New State ($\delta_{i,t+1}$)
Enrolled, Failed Lesson	No	n/a	n/a	Enrolled
	Yes, Quit	$B_{i,t+1} - B_0$	No	Browsing
			Yes	Discouraged

Separating the learner's utility into the separate pools $B_{i,t}$ and $V_{i,t}$ has two major advantages, one theoretical and one practical. In a theoretical sense it enriches the model by separating how the learner feels about their current course from how willing the learner is to try new courses. In an open platform courses can vary greatly according to the style of the instructor and so separating the goodwill towards a particular course from willingness to try new courses makes sense and allows for a more sophisticated specification of the learning psychology.

Practically this representation is immensely helpful because it means that the utility a learner expects to receive upon taking a course does not depend on $V_{i,t}$, which restores some symmetry across learners. This symmetry simplifies the contributor's decision making in determining which course to create and how to price that course, making the model more tractable.

3.2.4 Browsing and Course Enrollment Phase

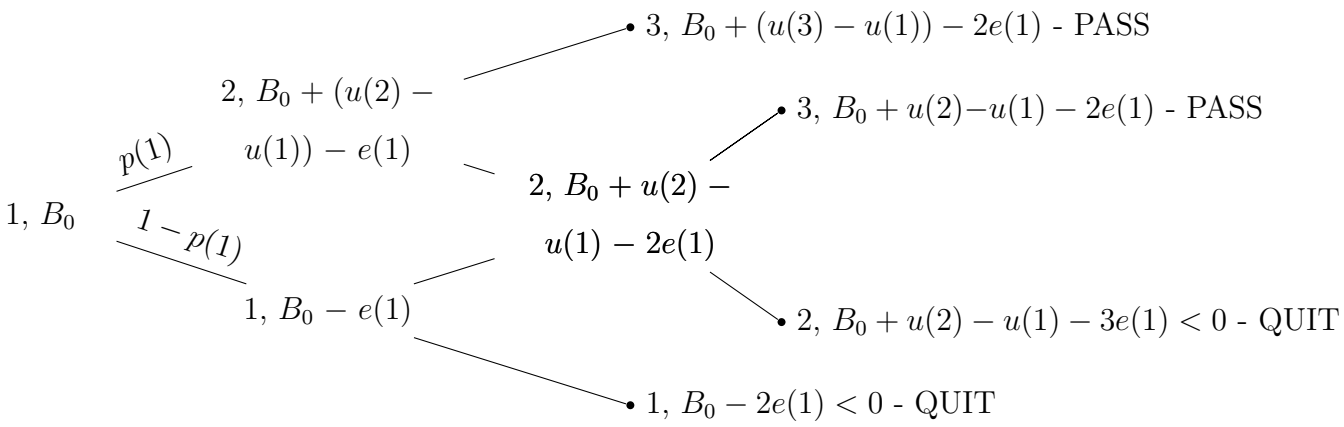
Course Valuation

We now turn to the initial browsing and enrollment phase of a learning time step. A *browsing* learner will be presented by the website with a course and must decide whether or not to enroll. In order to make that decision the browsing learner must determine how much he or she values a given course. For this we assume that learners are aware of the learning mechanics above, and so value a course according to the utility they expect to receive from taking that course. Therefore given learner skill s

and a course $[j, k]$ we can define $EU(s, [j, k])$ to be the utility the learner expects to receive upon enrolling in that course.

We can compute $EU(s, [j, k])$ by simulating the process of enrollment and learning. The learner begins at level s and upon attempting each lesson will either fail or pass probabilistically. This creates a probability tree, in which each branch will terminate with either the learner quitting or completing the course. If we sum from each endpoint of this tree the utility change of the endpoint multiplied by the probability of traveling along that branch, the aggregate sum will give us the expected utility of the learner enrolling in the course.

For illustration a possible example tree for $EU(1, [2, 3])$ is given below. At each node we list the skill level along with $B_{i,t}$. Each upward sloping branch is associated with passing the next lesson (with probability $p(1)$) and each downward sloping branch is associated with failing the next lesson (with probability $1 - p(1)$). Along any path we eventually pass the course or quit.



In this case we can sum over the four endpoints to compute the expected ending

utility, and subtract the initial pool value B_0 to compute the expected utility gain:

$$\begin{aligned} EU(1,[2,3]) = &(1-p(1))^2(B_0 - 2e(1)) \\ &+ 2p(1)(1-p(1))^2(B_0 - 2e(1)) \\ &+ 2p(1)^2(1-p(1))(B_0 + u(2) - u(1) - 2e(1)) \\ &+ p(1)^2(B_0 + (u(3) - u(1)) - 2e(1)) \\ &- B_0 \end{aligned}$$

Note that $EU(1,[2,3])$ will not necessarily take this specific form because the number of failures until $B_{i,t}$ becomes negative depends on the relative proportions of B_0 and the utility and effort functions.

Because all learners start a course with patience B_0, the value of course $[j,k]$ to a learner only depends on the learner skill level s, allowing us to define $EU(s,[j,k])$ as the value of $[j,k]$ to all learners $i \in L$ such that $s_{i,t} = s$. Also note that a course below the level of the agent provides no utility, that is $EU(s,[j,k]) = 0$ for $k \leq s$.

3.2.5 Presentation Mechanism and Enrollment Decision

Browsing users then myopically decide whether to enroll in courses according to whether the expected utility gain from taking the course outweighs the cost of the associated fee.

The organization of the educational content will be represented as the *mechanism* by which learners are (generally probabilistically) directed to courses. In our model a mechanism presents each browsing learner a single course in each time step. The learner then chooses to either enroll or decline and remain browsing for the next time step. While most real websites will typically present users with a list of courses

instead of a single course, we can consider the mechanism of presenting a single course as a simplified representation of the influence of displaying certain courses more prominently. Considering content presentation as a mechanism of matching users to a single item of content is also consistent with the modeling of user-generated content in other settings [6] [4].

Therefore formally in the first phase of each learning time step a browsing learner will be presented with some course $[j, k]$ and an associated fee F (set by the contributor who created that course). The learner at skill $s_{i,t} = s$ will then enroll in the course if and only if:

$$EU(s, [j, k]) - F \geq 0$$

One could argue that this myopic decision making is not truly strategic, as learners could pass on a course with a small utility gain in hopes of browsing a course with a larger utility gain in the future. However we believe our decision reflects a genuine uncertainty on the part of the learners about the other courses on the website, which would make it difficult for them to make accurate predictions of expected utility gains from browsing future courses. Additionally the model already captures some strategic decision making on the part of the learners, and to extend this concept further would add a lot of complexity for a relatively small gain in realism, especially when the greater focus is the strategic decision-making of the contributors, not the learners.

If the learner elects to pass and continue browsing, then that learner will pay a small browsing cost b for the effort of deciding on that course. This cost will be deducted from the platform utility pool such that $V_{i,t+1} = V_{i,t} - b$.

3.2.6 Learning Summary

In summation, each learner begins the learning time step in one of three states: discouraged, enrolled, or browsing. In the first phase browsing learners are each presented a course and elect either to enroll or to decline. Next all enrolled learners, including those who just became enrolled, will attempt the next lesson in whichever course they are currently enrolled.

Finally for all enrolled learners, upon successfully completing or failing a lesson, the learner's skill will be updated and the net change in utility resulting from the lesson outcome will be added to the course utility pool. Only if the user has passed the last lesson in the course or the updated utility pool is less than zero will the user leave the course upon completion or quitting respectively and the net change in the course utility pool will be added to the platform utility pool. Upon an update to the platform utility pool only if this pool is below zero will the learner become discouraged and leave the entire website, otherwise the learner will return to a browsing state for the next time period.

Note that a discouraged learner will do nothing during this process as they are no longer active in the system.

The entire learning process is presented formally and concisely according to these three steps in the following tables. In a slight abuse of notation we introduce the state $\delta'_{i,t}$ to denote the updated state after the browsing and enrollment phase but before the learning phase. This allows us to capture the fact that browsing learners can both enroll and then attempt a lesson in the different phases of the same time step. Also note that ΔB is shorthand for the update rule $B_{i,t+1} = B_{i,t} + \Delta B$ and similarly for ΔV.

1. First, each browsing learner will be presented by the website mechanism with

a course the learner may enroll in:

$\delta_{i,t}$	Option	$EU(s_{i,t},[j,k]) \geq F$?	ΔV	$V_{i,t+1} < 0$?	$\delta'_{i,t}$
Browsing	Presented Course	Yes: Enroll	0	n/a	Enrolled
		No: Decline	-b	No	Browsing
				Yes	Discouraged
Enrolled	n/a	n/a	n/a	n/a	Enrolled
Discouraged	n/a	n/a	n/a	n/a	Discouraged

Note that if a browsing learner i does enroll then that will set $B_{i,t} = B_0$ and $[j,k]_{i,t} = [j,k]$.

2. Next, any enrolled learners will attempt the next lesson in their course:

$\delta'_{i,t}$	Action	Outcome	$s_{i,t+1}$	ΔB
Enrolled	Attempt next Lesson at j'	Success	j'	$u(j') - u(s_{i,t}) - e(j' - s_{i,t})$
		Failure	$s_{i,t}$	$-e(j' - s_{i,t})$
Browsing	n/a	n/a	n/a	n/a
Discouraged	n/a	n/a	n/a	n/a

3. Finally learners will quit or become discouraged if their respective utility pools have gone below 0:

$\delta'_{i,t}$, Result of (2)	$B_{i,t+1} < 0$?	ΔV	$V_{i,t+1} < 0$?	$\delta_{i,t+1}$
Enrolled, Failed Lesson	No	n/a	n/a	Enrolled
	Yes, Quit	$B_{i,t+1} - B_0$	No	Browsing
			Yes	Discouraged

$\delta'_{i,t}$, Result of (2)	Passed course? ($s_{i,t+1} = k$?)	ΔV	$V_{i,t+1} < 0$?	$\delta_{i,t+1}$
Enrolled, Passed Lesson	No	n/a	n/a	Enrolled
	Yes	$B_{i,t+1} - B_0$	No	Browsing
			Yes	Discouraged
Browsing, n/a	n/a	n/a	n/a	Browsing
Discouraged, n/a	n/a	n/a	n/a	Discouraged

3.3 Contribution

In addition to learning, each time step importantly consists of a round of contribution. In the contribution stage of each time step the following simultaneous game is played:

The agents consist of the constant pool of potential contributors C. Let $s_c \in \{1, 2, \ldots n\}$ be the skill of contributor $c \in C$. This reflects the fact that not all teachers are necessarily experts on the subject, and teachers who are not experts are still capable of creating course at lower difficulties. We assume the contributors' skill levels are fixed and therefore that contributors do not seek to learn. Also we assume that there are no potential contributors at skill 0, who would be unable to contribute anything because of their novice skill level.

Each potential contributor has the same constant set of actions: the contributor decides whether to participate, and conditional upon participating chooses a course $[j, k]$ to create along with an associated fee F.

Contributors are taken to be profit-maximizing, and specifically seek to maximize profit from the very next round of learning. Here we assume bounded rationality in that the contributors myopically seek the optimal strategy for the payoff in the next round but do not anticipate or influence future rounds. This intuitively reflects

some combination of two assumptions about the contributors: Firstly that they are generally impatient and so value profit in the next round very highly, and secondly that solving for future revenue requires anticipating the behavior of other agents in all future rounds, which is too complex for the contributors to realistically account for.

The cost component of a contributor's profit represents the effort of course creation. The cost for a contributor to create a lesson is determined by a cost function $c(d)$. A contributor at skill level s_c who makes a lesson at k will undergo a cost of $c(s_c - k)$, such that the cost depends only on the difference between the contributor's skill level and the lesson difficulty level. There two restrictions on the cost function:

1. $c(s_c - k) = \infty$ for all $k > s_c$. This ensures no contributor can make a lesson to teach concepts above his or her own skill level, which would be nonsensical.

2. $c(d)$ is strictly decreasing in d, meaning that creation of a given lesson requires less effort from a more skilled contributor.

The cost to a contributor to create a course is simply the sum of the cost of creating each lesson in that course.

The benefit to a contributor is the expected revenue generated by the course in the next time step. This is the product of the fee F and the expected number of students who enroll in the next round of learning, which will depend on the platform mechanism. The objective profit is thus this expected revenue minus the cost of course creation. We will formalize the revenue, cost, and profit functions in section 4 and 5 in the context of each mechanism.

In determining equilibrium we consider free-entry Nash Equilibria, such that each contributor decides to create a course if and only if there is a non-negative expected

profit (at an expected profit of zero a contributor is indifferent between creating a course and not participating).

We also assume this simultaneous game takes place under complete information. This means in each round all potential contributors know the platform mechanism, the previously contributed courses, the distribution of learners across each skill level (how many learners at each level s are discouraged, browsing, and enrolled), all parameters of the learning model (u, p, e, B_0, V_0), and the number of other potential contributors at each skill level.

There are different ways this complete information assumption can be interpreted in a real setting. We can assume that the potential contributors can infer the platform mechanism by exploring the website and its organizational scheme. Knowledge of previously submitted courses is also straightforward in that potential contributors can browse the existing courses at any time and so will notice which new courses are published on the platform each time step. Knowledge of the learning mechanics is consistent with learners knowing their own learning mechanics.

As far as knowing the distribution of current learners, Udemy already makes public the number of students in every course and the courses taken by every student, and other online education sites such as Codecademy encourage the posting and sharing of course completions or milestones. One could therefore imagine potential contributors directly viewing or calculating aggregate statistics about enrollment in other courses along with traffic, therefore gathering at least a reasonable estimate of the distribution of browsing and enrolled students. Or, given knowledge or an estimate of the initial time zero population distribution, contributors at time t can analyze the courses submitted at $0, 1, \ldots t-1$ and calculate in expectation the time t learner distribution. Also note we do not assume that contributors know the exact utilities $B_{i,t}$ and $V_{i,t}$ of each learner, only how many learners at each level are browsing,

enrolled, or discouraged, which we believe is a little less restrictive.

Knowing the number and skill distribution of fellow potential contributors can similarly be interpreted as having an estimate of this distribution, at least for the first round. For $t > 1$ the difficulty distribution of published courses in the first $t-1$ rounds could then allow potential contributors to refine this estimate of the skill distribution of other contributors.

This conception of the contribution system as multiple rounds of simultaneous decision-making corresponds intuitively to the uncertainty faced by a real contributor: A given contributor can see courses that are already on the platform when the agent starts making his or her course, in our model represented by courses created in a previous time steps. However creating a course requires time investment and may involve a process of review and publishing on the part of the website. Therefore a contributor operates under considerable uncertainty regarding courses made by other contributors at about that same time, which is reflected in the simultaneous entry for contribution in a given time step.

4 Independent Course Mechanism

4.1 Overview

In order to solve for equilibrium contribution behavior we now turn to the mechanism for presenting courses to browsing learners. This captures the role of the platform's method of organizing content and represents the effect on the system we wish to isolate.

The first mechanism will simply be that each browsing learner is delivered a random course from the bank of previously submitted courses. This represents the in-

tuition that if courses are presented independently and no explicit information about course difficulty or ordering with respect to other courses is provided, it is up to the user to view courses individually and determine whether to enroll. Therefore this mechanism captures to a first approximation the system in place at most existing crowdsourced education platforms.

As a clarification we note that while these platforms do perform ordering with respect to estimates of quality and popularity they do not order with respect to difficulty, which is the specific factor we focus on in this paper. Furthermore recall that our model is situated within a given subject: A user is not presented a random course from the entire content base on the website but instead a random course in the subject of that user's interest, which one could imagine might be based on a category or search query.

We next formally solve for equilibrium under the independent course mechanism, and show that while this mechanism is sufficient to achieve moderate contribution and partial learning, it can be inefficient in a number of ways.

4.2 Fee, Revenue, and Profit

Consider a single time step of contribution under the independent course mechanism. Let C_t be the number of potential contributors who choose to contribute at time t under our Nash Equilibrium. Define K_t as the number of existing courses on the website before contribution at time t. This means that as the system evolves under equilibrium $K_t = \sum_{s=0}^{t-1} C_s$. Because each learner is presented a course at random, in expectation each course will be delivered to $\frac{1}{K_t+C_t}$ of the total browsing learners in the following learning time step.

Next consider a given contributor considering creating a specific course $[j, k]$. In

order to determine the revenue of this course the contributor must decide what to charge for the fee. To more easily analyze the fee and revenue for a course we will introduce the idea of a *support*, denoted $S([j,k],F)$. This represents the set of skill levels such that learners at a skill level $s \in S$ would be willing to pay F for the course $[j,k]$. Formally then:

$$S([j,k],F) = \bigcup s \text{ such that } EU(s,[j,k]) \geq F$$

This allows us to more easily define a revenue for a course $[j,k]$ and associated fee F in terms of this support.

For ease of notation let $L_t(s)$ denote the number of *browsing* learners with skill level s at time t:

$$L_t(s) = \sum_{i \in L} \mathbf{I}(s_{i,t} = s)\mathbf{I}(\delta_{i,t} = \text{browsing})$$

Then the revenue for a course $[j,k]$ and fee F is:

$$R_t([j,k],F) = \frac{1}{K_t + C_t} \sum_{s \in S([j,k],F)} L_t(s) * F$$

Next consider that for any course $[j,k]$ there is some optimal fee F_t^* which maximizes the revenue from that course. If the contributor charges a very low fee, near $F = 0$, the support will be maximized but the revenue, being proportional to the fee itself, will go to zero. Additionally if the fee grows large enough no user will be willing to pay the price, and so both the support and revenue will go to zero. Therefore there is some moderate F_t^* which brings in the maximal amount of revenue for course $[j,k]$:

$$F_t^*([j,k]) = \arg\max_F R_t([j,k],F)$$

Also note this optimal fee does not depend on the contributor skill level, since that affects the cost of course creation but not the revenue of a given course. Therefore any profit maximizing contributor who creates course $[j,k]$ will surely charge F^*, allowing us to define an optimal revenue of any course.

$$R_t([j,k]) = R_t([j,k], F_t^*([j,k]))$$

Next consider the decision of a particular potential contributor at skill s. We have already calculated the revenue for creating a particular course. The cost of a course $[j,k]$ is simply the sum of the cost of creating each individual lesson according to the cost function. We can therefore extend the cost function to the creation of courses in a straightforward way:

$$c(s,[j,k]) = \sum_{l \in [j,k]} c(s-l)$$

This allows us to define a profit function π denoting the profit a contributor with skill s expects from creating course $[j,k]$:

$$\pi_t(s,[j,k]) = R_t([j,k]) - c(s,[j,k])$$

Then any contributor will choose to create the course which yields the highest expected profit. This allows us to define an optimal profit function π^* in terms of contributor skill level:

$$\pi_t^*(s) = \max_{[j,k]} \pi_t(s,[j,k])$$

Additionally all contributors at skill level s will have some profit-maximizing course $[j,k]_t^s$:

$$[j,k]_t^s = \arg\max_{[j,k]} \pi_t(s,[j,k])$$

4.2.1 Equilibrium Condition

We will next formally define equilibrium under this mechanism. Let $C(s)$ represent the total number of potential contributors at each skill level s, as this does not change through time. Then define $C_t(s)$ to be the number of potential contributors at skill s who choose to participate and contribute a course in time t.

This means that C_t, which we earlier defined to be the total number of contributions at time t, is simply the sum of participating contributors from each level:

$$C_t = \sum_{s \in [1,n]} C_t(s)$$

The free entry condition yields the following characterization of a free-entry equilibrium:

Definition 4.1: Under the independent course mechanism, the contribution Nash Equilibrium is characterized by $C_t(s)$ such that:

- *If $\pi_t^*(s) < 0$, $C_t(s) = 0$. No agents will wish to contribute at negative profits.*
- *If $\pi_t^*(s) = 0$, $0 \leq C_t(s) \leq C(s)$. Agents are indifferent between contributing and not contributing at zero profit.*
- *If $\pi_t^*(s) > 0$, $C_t(s) = C(s)$. All agents wish to contribute at positive profits.*

4.3 Theoretical Results

Without any further assumptions we can prove that there will be some threshold skill level above which contributors elect to participate:

Theorem 4.1: Under the independent course mechanism, if there is any contribution

at time t, there will be some skill level s_t^* such that:

- *Agents with $s < s_t^*$ will not contribute: $\pi_t^*(s) < 0$, $C_t(s) = 0$.*
- *Agents with $s = s_t^*$, will contribute (probabilistically): $\pi_t^*(s_t^*) \geq 0$, $0 < C_t(s_t^*) \leq C(s_t^*)$.*
- *Agents with skill $s > s_t^*$ will all contribute: $\pi_t^*(s) > 0$, $C_t(s) = C(s)$.*
- *More skilled agents will generate higher profits: $\pi_t^*(s_t^*) < \pi_t^*(s_t^* + 1) < \cdots < \pi_t^*(n)$*

Proof: We assumed at least one contribution, so we know a skill s_t^* with a contributing agent exists. Next consider that for any $s < t$, $\pi^*(s) < \pi^*(t)$. To see this define $[j,k]^s$ as the optimal course for s, such that $\pi(s,[j,k]^s) = \pi^*(s)$. Then we know $\pi^*(t) \geq \pi(t,[j,k]^s) > \pi(s,[j,k]^s) = \pi^*(s)$ because $R([j,k])$ does not depend on the skill of the contributor and $c(t,[j,k]) < c(s,[j,k])$ because for each lesson built $c(t-i) < c(s-i)$. Therefore the agent at t can do strictly better in terms of profit just by building the same course $[j,k]^s$. Thus if we define s_t^* to be the smallest s such that $\pi^*(s_t^*) \geq 0$, we are guaranteed that for all $s > s_t^*$, $\pi^*(s) > 0$ and all agents at those skill levels will contribute at equilibrium.

This indicates that there is some threshold potential contributor skill level above which all agents contribute courses and below which none do. It also shows that among those who do contribute, agents with higher skill levels will earn higher profits, which stems from their lower lesson creation costs.

Next we turn to the form of the actual contribution. In general the distribution of contributed courses and relationship between contributor skill level s and the optimal

course $[j,k]_s^t$ are highly sensitive to the parameters of the model. However we can show more about the first round of contribution, at time zero. At time zero we assume that the initial population of learners (who are all browsing), $L_0(i)$, is a decreasing function of i, capturing the intuition that initially there are more novices who wish to learn than intermediate or advanced learners.

The initial round of contribution is important because there will likely be more contribution at this first time step than any future round. This is evident from two factors: Initially the entire population of learners is browsing, and none have enrolled or become discouraged. Therefore the browsing population will never be any larger than it is in the first round. In addition the number of existing courses on the site is initially zero, which means that the first round of course contributions will not have to compete with any existing courses, increasing profits substantially. The combination of these effects can lead to a very big influence of the first round of contribution on the contribution and learning outcomes for the system.

In this first round we show that all contributors will "target" complete novices with their courses. This means all contributors who contribute will create a course that a learner at skill level 0 would choose to take. We can formalize this in terms of the support of the course:

Theorem 4.2: Under the independent course mechanism, for any course $[j,k]$ *contributed at* $t = 0$, $0 \in S([j,k], F_0^*([j,k]))$

Proof: Let $S([j,k]) = S([j,k], F_0^*([j,k]))$, since we know any contributor will charge F^*. For contradiction assume that some contributor finds it optimal to make a course $[j,k]$ such that $s_{min} = \min_s S([j,k]) > 0$. The support of the course must be non-empty if the contributor maximizes profit at this course. Then consider the

course $[j - s_{min}, k - s_{min}]$ with the same fee $F = F^*([j,k])$. We are guaranteed that $s_{min} \leq j$ because it is impossible for $t > j$ to be in $S([j,k])$ but j not to be in S, given that $EU(j,[j,k]) > EU(t,[j,k])$. Therefore $[j - s_{min}, k - s_{min}]$ will be a valid course. Next we can see that for any $s \in S([j,k])$, $s - s_{min} \in S([j - s_{min}, k - s_{min}], F)$. This is true because $EU(s,[j,k]) \leq EU(s - s_{min}, [j - s_{min}, k - s_{min}])$ by the assumption that the utility function u is concave, which guarantees that the utility gain associated with completing course $[j - s_{min}, k - s_{min}]$ for a learner at $s - s_{min}$ is at least as large as the utility gain for completing course $[j,k]$ for a learner at s. Thus the total population within the support of $[j - s_{min}, k - s_{min}]$ must be at least as big as the total population of the support of $[j,k]$, because for every $s \in S([j,k])$ there is some $s - s_{min} \in S([j - s_{min}, k - s_{min}], F)$ and $L_0(s) \leq L_0(s - s_{min})$. Therefore $R([j - s_{min}, k - s_{min}]) \geq R([j - s_{min}, k - s_{min}], F) \geq R([j,k])$. Furthermore $c(t,[j - s_{min}, k - s_{min}]) < c(t,[j,k])$ for any contributor skill level t because it is strictly cheaper to make easier lessons. This means $\pi(t,[j - s_{min}, k - s_{min}]) > \pi(t,[j,k])$ for any contributor and creating course $[j,k]$ was not optimal.

Therefore initially there will be a flood of easy courses aimed at the beginning end of the skill spectrum. Note that while the support of the courses will include complete novices at level 0, this does not necessarily imply the courses themselves will start at level 1.

For future rounds of contribution, the model is flexible enough that there is no definite relationship between contributor skill level and the difficulty of the course that contributor makes. To demonstrate the various contribution outcomes that can occur under different parameters of the model we will make use of simulation.

4.4 Simulation Results

There is an enormous amount of potential variability in the functional forms of our parameters given the general assumptions of the model. The simulations presented below therefore are not intended as an exhaustive demonstration of contribution outcomes. Rather they are intended as examples of various contribution patterns, as well as to provide concrete examples and intuition for the inefficiencies that can result under the independent course mechanism.

4.4.1 Functional Forms

The functional forms used in the model are given below, along with the associated constant influencing the specific shape of each function. Recall that many learning functions depend on the distance between the learner skill level and the lesson level, for example the effort function e. In the table we denote d to represent this distance argument, such that a learner at s attempting a lesson at k would face a distance $d = k - s$ and therefore pay effort $e(d)$.

Description	Parameter	Functional form	Restrictions
Effort function	$e(d)$	d^{e_a}	$e_a > 0$, increasing
Learning Success Probability	$p(d)$	$(1 - \frac{d}{n})^{p_a}$	$p_a > 0$, decreasing
Learning Utility	$u(s)$	s^{u_a}	$u_a \leq 1$, concave
Initial Learner Distribution	$L_0(s)$	e^{-s*l_a}	$l_a \geq 0$, weakly decreasing
Building cost	$c(d)$	$(n - d)^{c_a}$	$c_a > 0$, decreasing

These functions were chosen to be as simple as possible while still allowing for

easy manipulation of the relative concavity or convexity of each function through a constant. The restrictions on the constants stem from earlier assumptions about the functions.

Constants

In order to isolate the effects of the function shapes listed above and somewhat restrict the degrees of freedom of the model we chose to set some of the constants to the same values for all simulations. The following parameters and their values were held constant for all simulations in the remainder of the paper:

Description	Parameter	Value
Number of skill levels	n	10
Learner Population	L_0	300
Contributor Population	C_0	50
Contributor Distribution	$C_t(s)$	C_0/n for $n \geq 1$
Course utility pool	B_0	0.5
Platform utility pool	V_0	1.0
Browsing effort	b	0.15

The constants for the total populations were chosen to induce a moderate amount of contribution for best illustrative purposes. The utility pool and browsing cost constants were chosen relative to a effort cost scaled such that $e(1) = 1$. This means the utility pool constants can be thought of as the number of small failures tolerated before any success, and the browsing cost can be thought of as the effort required in browsing a course relative to taking a single lesson. We chose a uniform distribution for contributor skill over $\{1, \ldots n\}$ for simplicity and to ensure at least some contributors at every skill level in order to best demonstrate relationships between

contributor skill level and courses produced.

4.4.2 System Evolution Visualization

We first show a visualization of a "typical" system evolution under the independent course mechanism. In this context typical means that this example has enough contribution to be illustrative and the overall patterns discussed hold true over a range of moderate values of our parameter constants. Therefore this example provides intuition for what the independent course mechanism does well and why it performs poorly in certain cases.

Below are the constants chosen for this simulation:

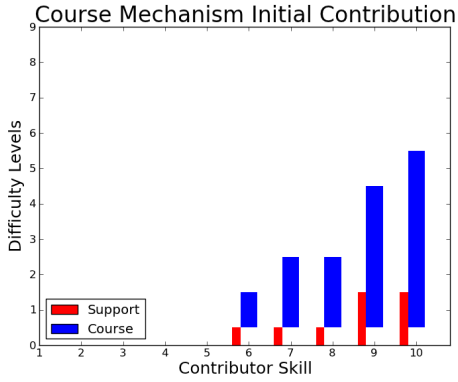
Description	Functional form	Simulation value
Effort function	$e(d) = d^{e_a}$	$e_a = 2.0$
Learning Success Probability	$p(d) = (1 - \frac{d}{n})^{p_a}$	$p_a = 2.5$
Learning Utility	$u(s) = s^{u_a}$	$u_a = .8$
Initial Learner Distribution	$L_0(s) = e^{-s*l_a}$	$l_a = .5$
Building cost	$c(d) = (n - d)^{c_a}$	$c_a = 2.0$

Also note that the effort, probability, and cost functions are convex. Although this assumption is not required for any of our proofs we believe this is more realistic in all three cases: As a learner tries to jump multiple skill levels it seems natural that the probability of success would fall sharply and the effort required should rise sharply, even with only a moderate gap in skill level. This reflects the intuition that, for example, moving from a gap of 1 to a gap of 2 involves a larger change in probability and effort than moving from a gap of 3 to a gap of 4. Similarly the

cost of creating easier lessons should fall more quickly at first, because there is a big difference between trying to teach something you barely know and something you are more comfortable with, and then more slowly, because there is not too much of a difference between teaching something relatively easy for you and something very easy for you. Therefore in future simulations we will continue to assume the convexity of these functions.

Contribution Visualization

The contribution in each time step is shown as a bar graph. The x axis represents the contributor skill level. The blue bar above contributor skill s represents visually the maximum profit course created by contributors at s, or $[j,k]_t^s$. A course $[j,k]$ is drawn from $[j-.5,k+.5]$ on the graph to make clear which levels it includes. The red line adjacent to the blue line represents the support of the given course, or $S([j,k]_t^s)$. This support represents the set of levels from which learners are willing to pay to enroll in course $[j,k]_t^s$. If there is no blue bar for a given contributor skill level this means all contributors at that skill level elect not to participate. As an example the contribution at time zero is given below:



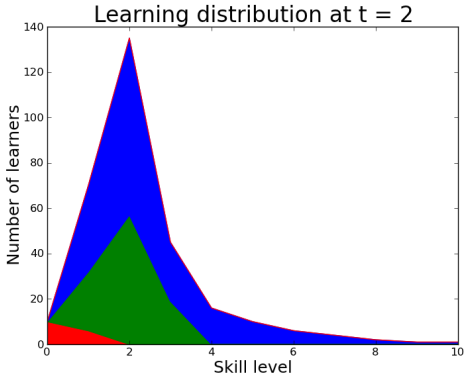
Thus in the above contribution graph the blue bar drawn over $s = 7$ from .5 to 2.5 indicates that in the first round of contribution, all contributors with $s_c = 7$ choose to submit the course $[j, k]_0^7 = [1, 2]$. This also shows that $s_0^* = 6$ because $s = 6$ is the lowest skill level from which potential contributors elect to participate.

Learning Visualization

Alternated with these contribution plots are plots displaying the distribution of learners at each time step. In each learning plot:

- Red area indicates discouraged learners
- Green area indicates enrolled learners
- Blue area indicates browsing learners

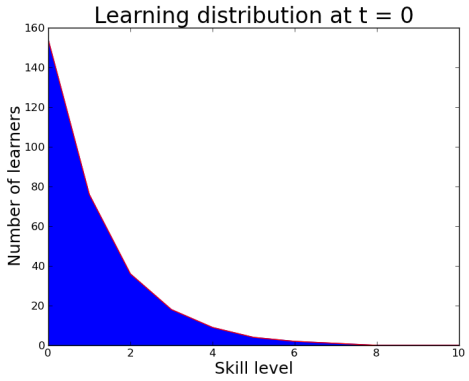
The learning distribution at $t = 2$ is provided as an example:

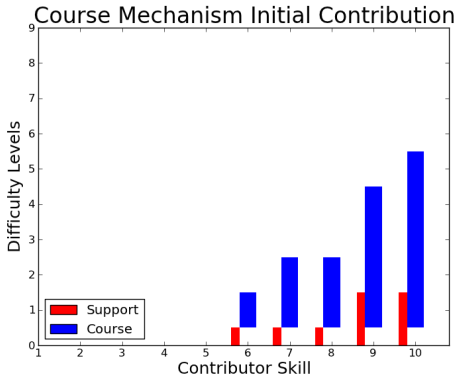


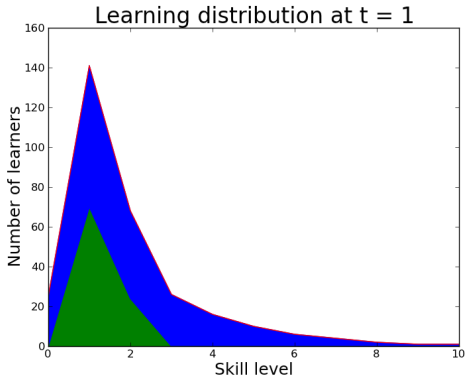
Here we can see that there are a small number of discouraged learners, around 10, at skill levels 0 and 1. There is a much larger number of enrolled learners, with around 45 enrolled learners at skill 2. And there are even more browsing learners, with $L_2(2) \approx 80$ browsing learners at skill level 2.

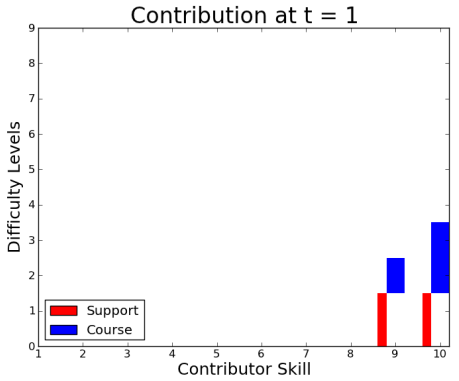
4.4.3 Visualization Results

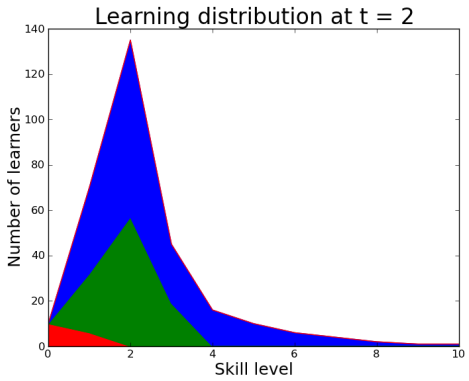
Alternating between contribution and learning graphs in each time step provides a visualization for the evolution of the system. The evolution for our illustrative example is shown below:

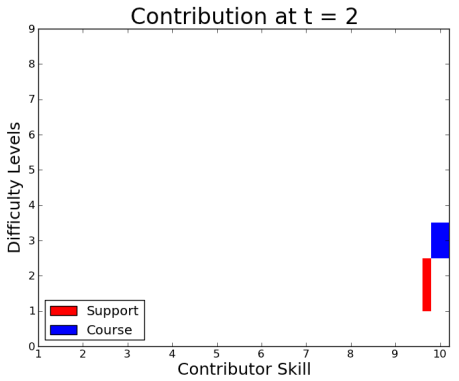




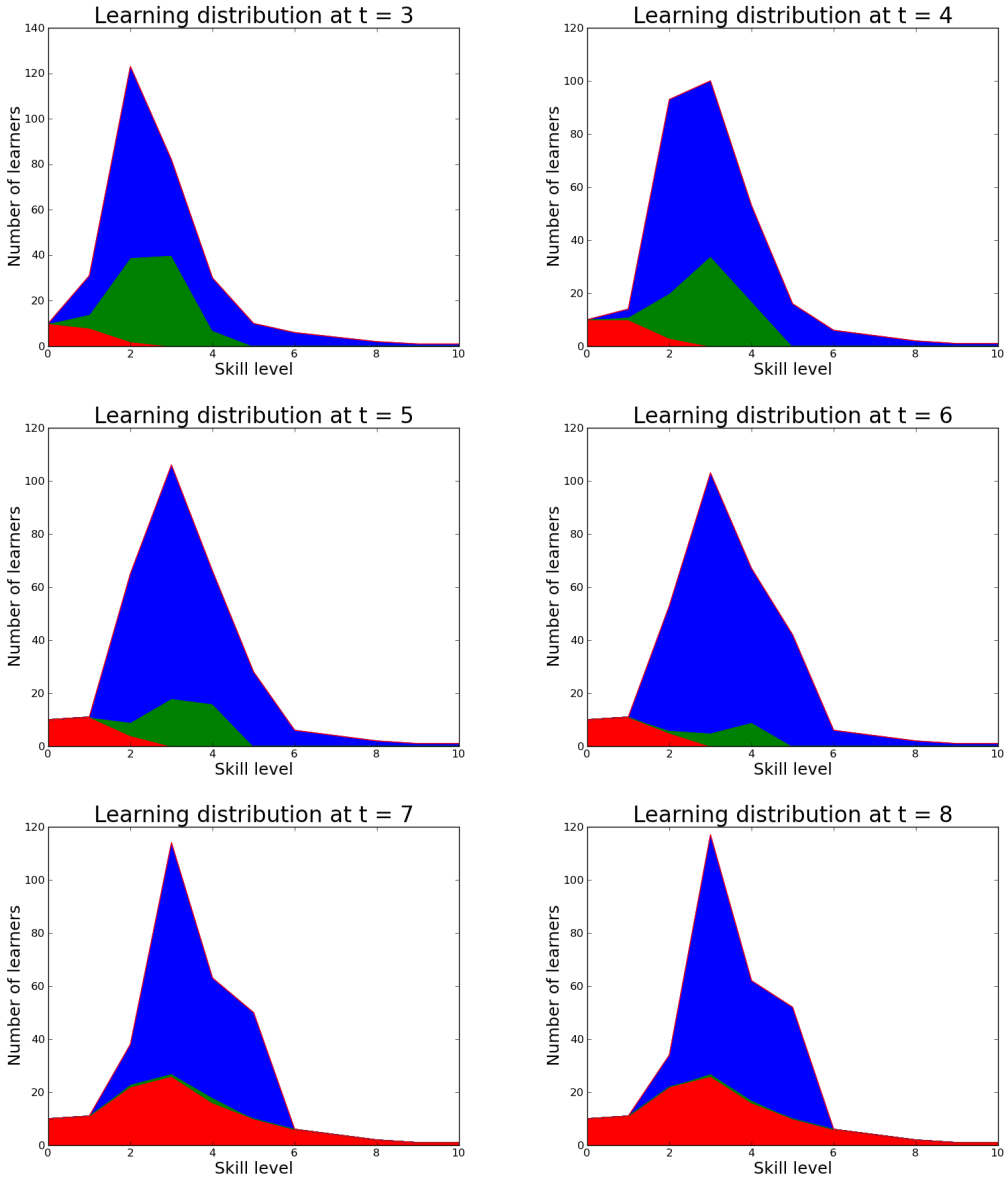




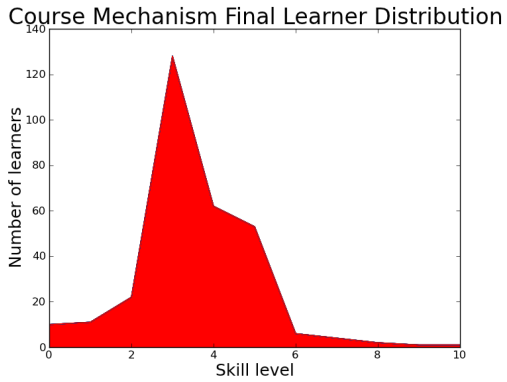




After $t = 3$ there is no longer any contribution, and so learning continues with the previously submitted courses.



Eventually all learners are either experts or discouraged, and the final distribution is given below:



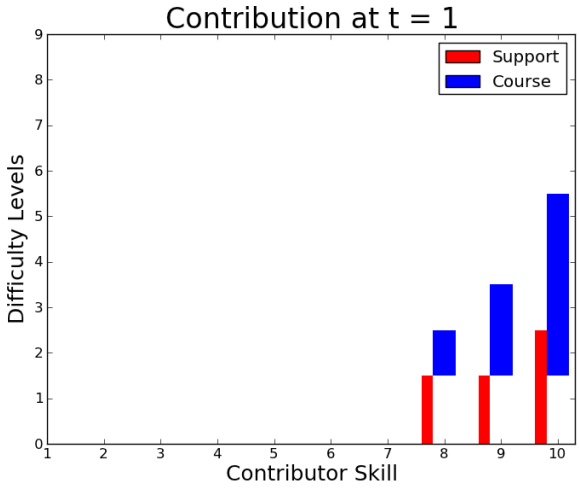
There are two key patterns which consistently result from the independent course mechanism:

- There is a lot of redundancy, especially in the initial round of contribution. Because most learners start out at the extreme novice end of the spectrum and the courses are delivered randomly, all contributors seek to make courses tailored for these novices in accordance with theorem 4.2. This is inefficient in the sense that redundant content does not directly enhance learning. The effect of this redundancy in the first round of contribution is made even more significant due to the fact that more than half of all courses submitted were submitted in the first time step.

- After many time steps there is a large population of learners at intermediate skill levels (here $s = 3, 4, 5$), yet contributors do not create more advanced courses. Because of the random course delivery, once there is already a large number of courses on the platform a newly created course will be delivered to only a small fraction of the learner population in each time step. This heavily reduces the expected revenue and thus incentive of contributors to create advanced courses, and leads to a lack of these more difficult courses.

While the distribution of course creation depends heavily on the choice of parameters, there are some cases where the contribution outcomes are even less desirable from a learning standpoint. While we have proven that all contributors target novices in the first round of contribution, these created courses do not necessarily start at level 1, the easiest level. For example we see below that contributors under these parameters choose to start their courses at level 2:

Parameter	Value
e_a	1.5
p_a	1.5
u_a	0.9
L_a	0.1
c_a	2.0



This leads to inefficient learning outcomes because the large population of learners at $s = 0$ must attempt the first lesson at $k = 2$, which will cause the learners to have a higher rate of failure and therefore a higher rate of quitting and discouragement.

Thus while this mechanism does achieve partial learning there are some major drawbacks. Firstly we are guaranteed that all contributors will create courses targeted towards novices in the first round of contribution, which initially floods the platform with novice courses. Secondly courses at $[j, k]$ may include in their support learners at $s < j - 1$. This causes those learners to have to jump a bigger gap when first enrolling, and so causes a higher failure rate. Thirdly and most prominently, because courses are delivered randomly and so contributors are unable to specifically target

advanced students, there can be a complete lack of advanced courses. This causes all learners to stall at an intermediate skill level and prevents them from reaching mastery.

5 Ordered Lesson Mechanism

5.1 Overview

We now present an alternative mechanism which we term the *ordered lesson* mechanism. This mechanism is inspired by the examples of online and offline non-crowdsourced educational settings outlined in section 2, in which content is presented in a clear order of difficulty. The salient features in this mechanism are a reduction in the size of each crowdsourced course and an ordering of those courses for better delivery to learners.

Formally this mechanism features two important differences from that of independent courses:

- Firstly, no contributor is permitted to create a course consisting of greater than one lesson. To enforce this we can imagine that the platform requires contributors to enter a single difficulty when submitting a piece of content and possibly even caps the maximum size or length of a submitted course.

- Secondly, a browsing learner will not get a random lesson from among the total set of freestanding lessons on the platform, but will instead be delivered a random lesson from among the set of lessons at the nearest difficulty above that browser's skill level. Specifically a browsing learner at level s will be delivered a random lesson at $s + 1$ if at least one lesson at $s + 1$ exists. If there are no

lessons at $s+1$ the learner will be delivered a lesson from $s+2$ instead, or if none exist there, $s+3$, etc. If there are no lessons at any level $t > s$ then the learner will pay the browsing cost b and view no lesson at all (which is the same outcome as if the learner were delivered a course below the learner's skill level).

In all other respects the model is the same. To clarify this means once a lesson is delivered to a learner the learner will behave as if it were delivered a course consisting of a single lesson. That is the learner can still choose to enroll or not enroll based on whether $EU[s, (s+1, s+1)] \geq F$, where F is the fee associated with that freestanding lesson set by the contributor of the lesson.

As with the independent course mechanism this represents a simplification of the effect the presentation and ordering of content has on the lessons browsed by a learner. There are two different intuitive explanations for how such a mechanism could be implemented on a real platform:

One explanation is that the the site owner really does know the skill level of each learner to deliver a lesson from the next highest difficulty. For example we can imagine the website requests a survey or quiz from learners initially to gain knowledge of their starting skill level. Then the website could track the progress of learners through lessons to maintain knowledge of each learner's current skill level, recommending the next lesson as the learner progresses.

A second explanation is that the operator of the website does not actually know the skill of each learner. Instead the platform aggregates the content by clearly laying out a spectrum of lessons along difficulty levels. In this case the learners, knowing their own skill level, would choose to browse a lesson at the next highest difficulty level because such lessons would be optimal to enroll in from the standpoint of expected utility. This requires the additional constraint that $EU(s, [s+1, s+1]) > EU(s, [k, k])$

for any $k > s+1$. This is reasonable however because intuitively people tend to learn most effectively when progressing through content of escalating difficulty gradually rather than "skipping" ahead to more difficult content. In this case the mechanism still makes sense even though the platform is not actually specifically presenting the next lesson.

We will show that this mechanism has a desirable theoretical property that allocates contribution in an efficient way, in that more skilled contributors make more advanced lessons in every round of contribution. Furthermore this mechanism is far better at delivering advanced lessons to learners, because these lessons can be targeted towards the relevant learners. This allows for much better learning outcomes than the independent course model, as we show through simulation.

5.2 Theoretical setup

We will reuse much of the same theoretical framework and notation from section 4 for analyzing contribution equilibrium under the ordered lesson mechanism. Instead of evaluating possible courses $[j, k]$ contributors will consider courses composed of a single freestanding lesson. For ease of notation we will denote a freestanding lesson as $[j]$, but it should be understood that the learners will evaluate and learn from this as if it were the course $[j, j]$.

In the independent course mechanism the revenue of a given course $[j, k]$ only depended on K_t, the number of existing courses, and C_t, the total number of contributed courses at time t. In the ordered lesson model however the revenue is dependent on the distribution of C_t and K_t across lessons.

Therefore let K_t^j and C_t^j represent the total existing and current equilibrium contribution of freestanding lessons at level $[j]$. As the system evolves under equilibrium

K_t^j will represent the total number of previously submitted lessons at $[j]$:

$$K_t^j = \sum_{s=0}^{t-1} C_s^j$$

The notion of support for a given lesson will also change under this mechanism. In the independent course mechanism any learner could potentially be delivered a given course, so any learner willing to pay for that course was considered part of the support. In this mechanism, however, a learner at s will not be delivered $[s+2]$ if there are lessons at $s+1$. This means that even though the learner at s may have been willing to pay for $[s+2]$, that learner is no longer part of the support in the sense that there is no expected revenue to be made by lesson $[s+2]$ from learners at s, who will never even be presented with the course.

We will then redefine support in the context, yielding:

$$S([j], F) = \bigcup s \text{ such that}$$
$$EU(s, [j, j]) \geq F \text{ and } C_t^i = K_t^i = 0 \text{ for } i \in [s+1, \ldots j-1]$$

This indicates that the support consists of learners who are both willing to pay for lesson $[j]$ and will actually be delivered $[j]$. The ordered lesson mechanism also means a slight modification to the revenue formula in terms of the support:

$$R_t([j], F) = \frac{1}{K_t^j + C_t^j} \sum_{s \in S([j,k],F)} L_t(s) * F$$

This is because all of the learners in the support for lesson $[j]$ will be matched to lessons at $[j]$, in which case the revenue will only be divided between lessons at that difficulty level.

It is still the case that there is some optimal fee F_t^* which maximizes the revenue from $[j]$.

$$F_t^*([j]) = \arg\max_F R_t([j], F)$$

This yields revenue directly in terms of a lesson assuming contributors will charge the optimal price

$$R_t([j]) = R_t([j], F_t^*([j]))$$

This yields a similar profit function:

$$\pi_t(s, [j]) = R_t([j]) - c(s, j)$$

However because the revenue from contributing at a given level is so dependent on the number of other contributors also creating at that level, the equilibrium for this mechanism takes a different form than than that of the independent course mechanism. Under the course mechanism, given all contributors at $C(s)$ contribute, there was some single course $[j, k]_t^s$ which maximized profit. Even if there was somehow an exact tie between two courses $[j, k]$ and $[j', k']$ the contributors would be indifferent between them and the profits of the contributors would not depend on how many chose $[j, k]$ versus $[j', k']$.

In the ordered lesson model, the profits from $[j]$ depend on how many other contributors are creating at $[j]$. This means that while $[j]$ might seem to yield the highest profit if there were only one contributor at s, once all contributors at s decide to create lessons at $[j]$ this may bring the profit below that of some other level $[j']$. In this case at equilibrium contributors of skill s must contribute across both $[j]$ and $[j']$, but only if the proportions of contribution to these levels are such that the expected profits are the same.

To capture these details let $C_t^j(s)$ be the number of contributors of skill s who contribute to difficulty level j at time t. Note that this is consistent with our other notation whether we sum over difficulty levels j or contributor skill level s:

$$\sum_j C_t^j(s) = C_t(s)$$

As introduced section 4 $C_t(s)$ represents all contributors at skill s who contribute at time t.

$$\sum_s C_t^j(s) = C_t^j$$

As stated earlier this represents all contributions to level $[j]$ at time t.

Next let J_t^s be the set of difficulty levels contributors at s contribute to, defined as:

$$J_t^s = \bigcup j \text{ such that } C_t^j(s) > 0$$

This J_t^s could consist of multiple levels, one level, or be empty, in the case that contributors from s do not contribute at all.

5.3 Equilibrium Results

We can characterize the free-entry Nash Equilibrium in terms of this contribution range J_t^s.

Definition 5.1: Under the ordered lesson mechanism, the contribution Nash Equilibrium is characterized by J_t^s such that:

- *If J_t^s is non-empty, there must exist some $\pi_t^*(s) \geq 0$ such that*

$$\forall j \in J_t^s,\ \pi_t(s,[j]) = \pi_t^*(s)\ ,\ and\ \forall j' \notin J_t^s,\ \pi_t(s,[j']) \leq \pi_t^*(s)$$

- *If J_t^s is empty then $\pi_t(s,[j]) \leq 0$ for all $j \in [1,n]$.*

This emphasizes that for any level in J_t^s contributors must be making non-negative profit, and that profit must be greater than the profit from contributing to some level outside of J_t^s. Furthermore if J_t^s contains more than one level, then the profit from contributing to any $j \in J_t^s$ must be the same, or the equilibrium would not be stable because contributors would prefer to switch from contributing to some $j \in J$ to another $j' \in J$ with higher profits.

We have defined our notation such that Theorem 4.1 holds identically for this mechanism as well under a virtually identical proof. This theorem is restated below for the ordered lesson mechanism:

Theorem 4.1b: Under the ordered lesson mechanism, if there is any contribution at all, there will be some threshold skill level s^* such that:

- *Agents with $s < s_t^*$ will not contribute: $\pi_t^*(s) < 0$, $C_t(s) = 0$, $J_s^t = \emptyset$*
- *Agents with $s = s_t^*$, will contribute (probabilistically): $\pi_t^*(s_t^*) \geq 0$, $0 < C_t(s_t^*) \leq C(s_t^*)$, $|J_t^{s_t^*}| \geq 1$*
- *Agents with skill $s > s_t^*$ will all contribute: $\pi_t^*(s) > 0$, $C_t(s) = C(s)$, $|J_s^t| \geq 1$*
- *More skilled agents will generate higher profits: $\pi_t^*(s_t^*) < \pi_t^*(s_t^* + 1) < \cdots < \pi_t^*(n)$*

Therefore just as in the independent course mechanism there will be some threshold skill level s^* below which no potential contributors contribute and above which all contributors contribute. Furthermore profits will increase with the skill level of the contributor.

5.3.1 Distribution of Content

One advantage of the ordered lesson model is that we are guaranteed a desirable relationship between s and J_t^s under the general assumptions of our model. Specifically the following theorem shows that more skilled contributors will always tend to create more difficult content:

Theorem 5.1: Let r,s be two differing skill levels such that $r < s$, $C_t(r) > 0$, *and thus* $C_t(s) > 0$. *Then letting* $j_{max}(r) = \max J_t^r$ *and* $j_{min}(s) = \min J_t^s$, $j_{max}(r) \leq j_{min}(s)$

Proof: Let $j_{min}(s) = \min J_t^s$ and $j_{max}(s) = \max J_t^s$. For contradiction, assume $j_{max}(r) > j_{min}(s)$. Then because $s > r$ we know the agent at skill s can make any lesson in J_t^r including $j_{max}(r)$. Because $j_{min}(s) \in J_t^s$:
$\pi(s, [j_{min}(s)]) \geq \pi(s, [j_{max}(r)])$
Thus $R([j_{max}(s)]) - c(s, [j_{max}(s)]) \geq R([j_{min}(r)]) - c(s, [j_{min}(r))])$ which implies:
$c(s, [j_{min}(r)]) - c(s, [j_{max}(s)]) \geq R([j_{min}(r)]) - R([j_{max}(s)])$
But the agent at r weakly prefers $j_{max}(r)$ to $j_{min}(s)$, implying
$R([j_{min}(r)]) - c(r, [j_{min}(r))]) \geq R([j_{max}(s)]) - c(r, [j_{max}(s)])$ and therefore:
$R([j_{min}(r)]) - R([j_{max}(s)]) \geq c(r, [j_{min}(r))]) - c(r, [j_{max}(s)])$
Combining these two statements yields:
$c(r, [j_{min}(r))]) - c(r, [j_{max}(s)]) \leq R([j_{min}(r)]) - R([j_{max}(s)]) \leq c(s, [j_{min}(r)]) - c(s, [j_{max}(s)])$
and therefore
$c(r, [j_{min}(r))]) - c(r, [j_{max}(s)]) \leq c(s, [j_{min}(r)]) - c(s, [j_{max}(s)])$
But contradicts the fact that:
$c(r, [j_{min}(r)]) - c(r, [j_{max}(s)]) > c(s, [j_{min}(r)]) - c(s, [j_{max}(s)])$
because the cost function $c(d)$ is strictly decreasing in d, meaning that the difference in cost between the two lessons must be less for s than for r, given that s is more skilled.

This theorem guarantees that there will never be an instance of a more skilled contributor making a less difficult course than a less skilled contributor under the ordered lesson mechanism. This is advantageous for two reasons: Although this paper does not study quality directly, it is intuitive that more skilled contributors are better suited to make more advanced content and less skilled contributors are better suited to make less difficult content, so this outcome seems more efficient in that sense and more likely to lead to high quality contributions. Secondly, as supported in the following section with simulation, this property is associated with contribution of content distributed over a wide range of difficulty levels, including advanced content.

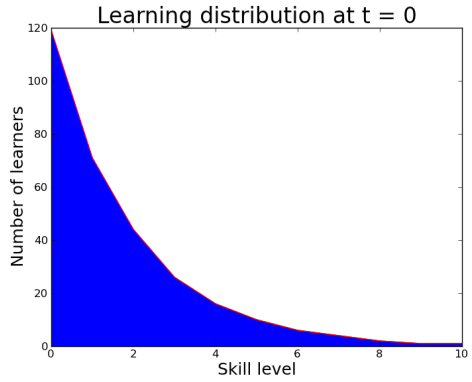
Therefore instead of all contributors targeting novices in the first round of contribution, there is evidence that under this mechanism contributions are more likely to be spread out over various skill levels in all rounds, with more skilled contributors creating more difficult lessons.

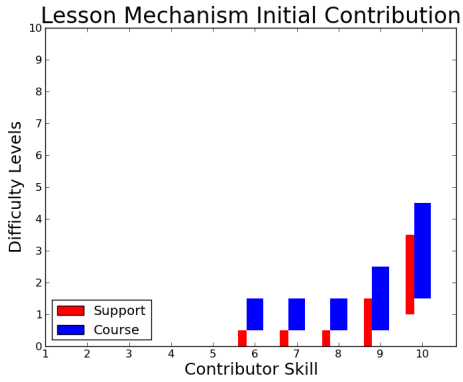
6 Mechanism Comparison

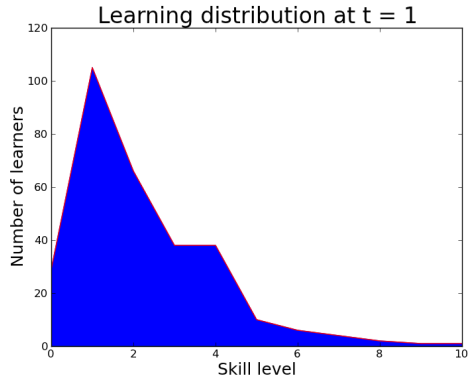
6.1 Ordered Lesson Mechanism Visualization

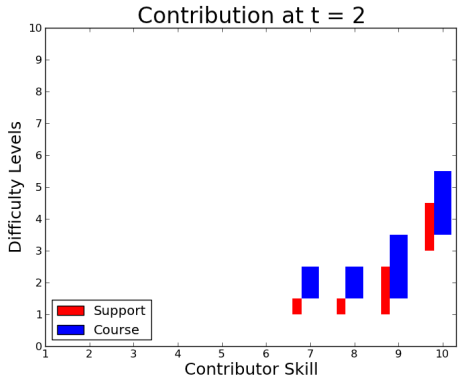
To illustrate the different patterns that are displayed in the ordered lesson mechanism and contrast it to the independent course mechanism, we simulate the ordered lesson mechanism with the same constants used in our visualization of the course mechanism. For the contributor plots under the ordered lesson mechanism we plot J_t^s over skill level s. This means, for example, that a bar from .5 to 2.5 over $s = 9$ implies that $C_t^1(9) > 0$ and $C_t^2(9) > 0$, or that some contributors at skill 9 create [1] while others create [2] under equilibrium.

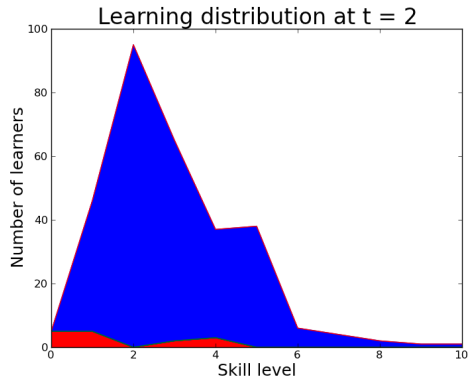
The visualization for the ordered lesson mechanism is given below. Note in this case there is never a body of enrolled learners because all courses consist of a single lesson, and B_0 is small enough that a learner will quit after a single failure of that lesson.

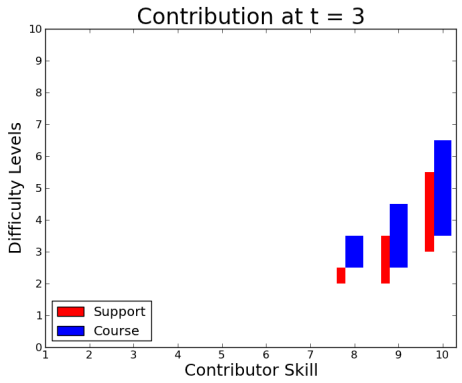


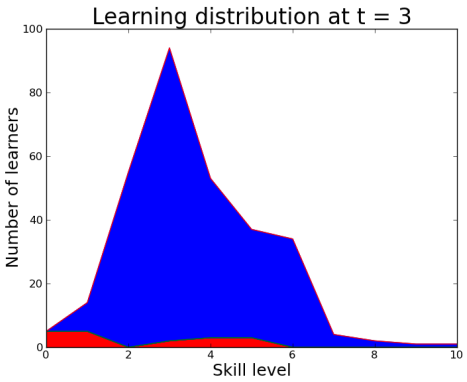


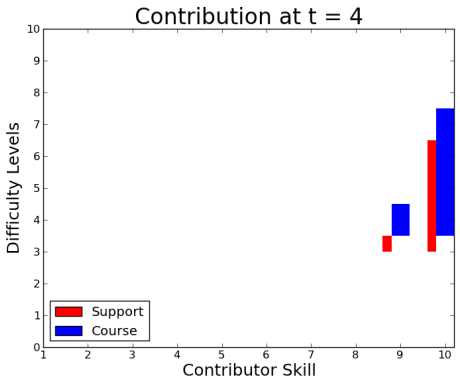


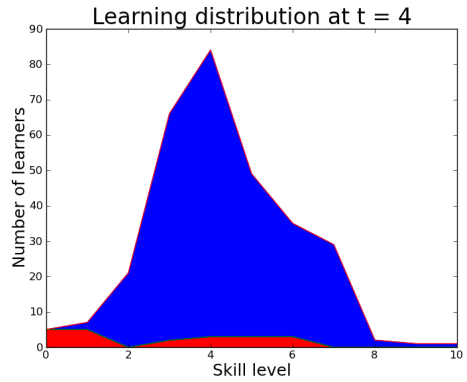


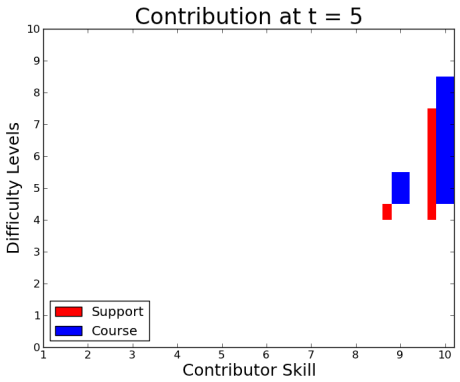




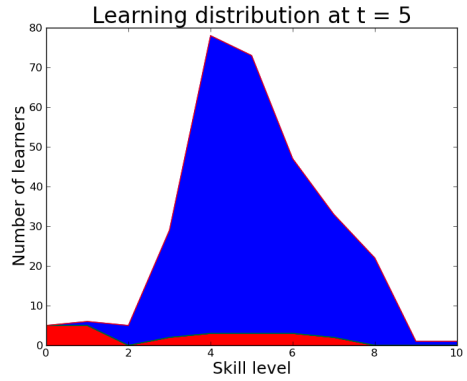


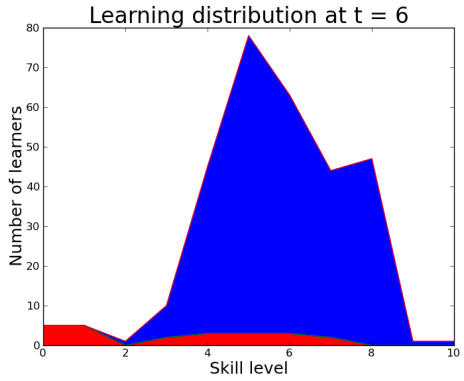


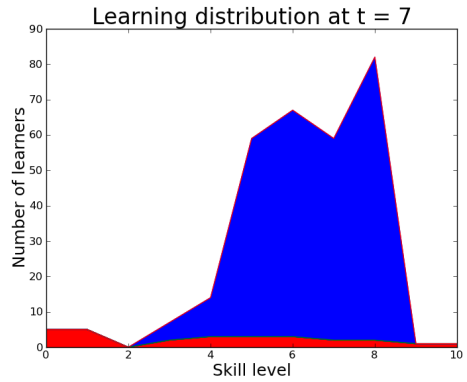


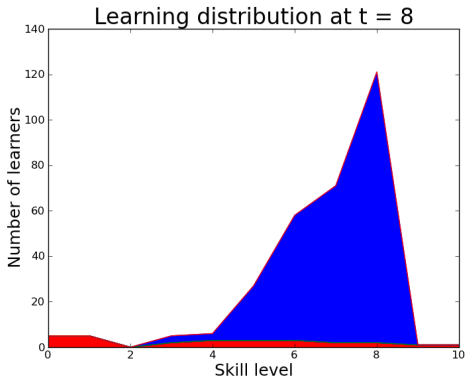


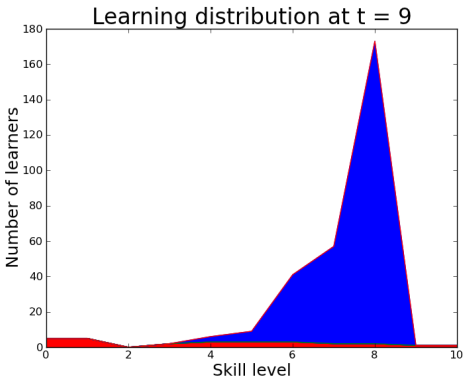
In time steps 6-10 there is no contribution, and learning continues with the existing lessons:

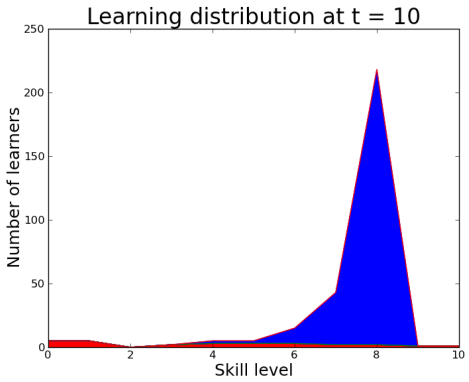




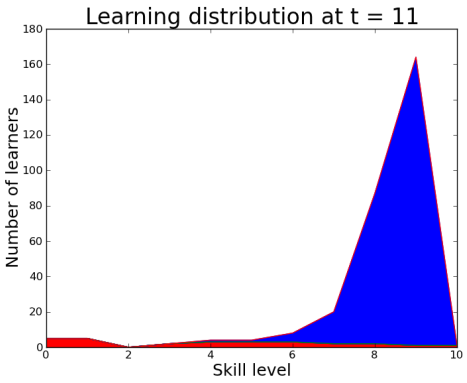


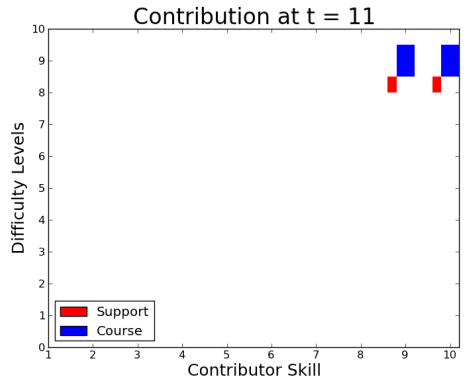


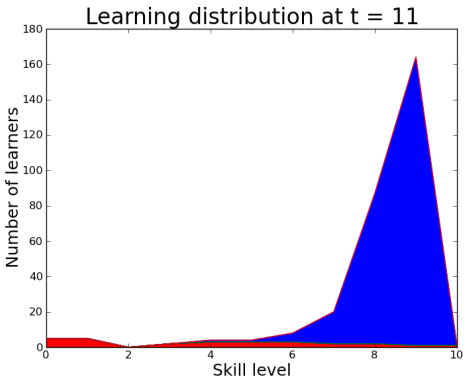


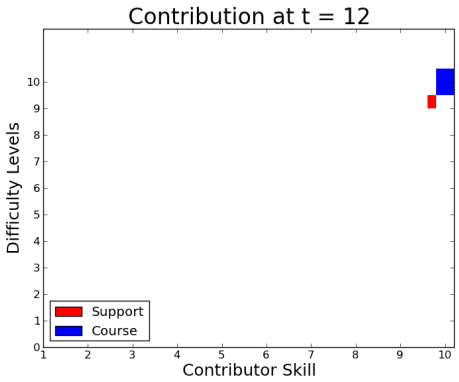


In time steps 11 and 12 there are two more rounds of non-zero contribution:

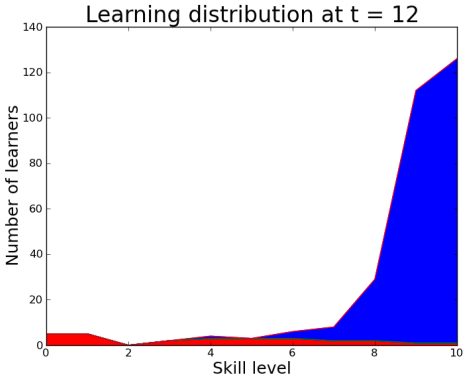


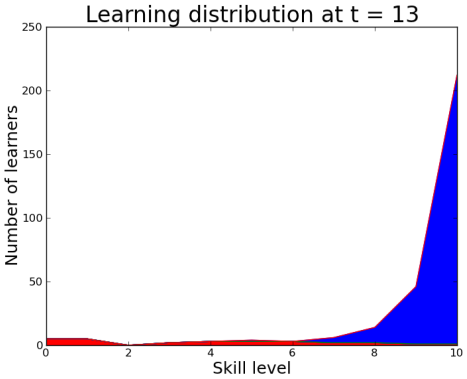




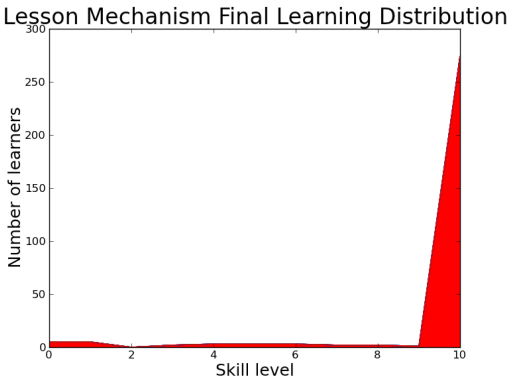


After the first 11 time steps there is no longer contribution, but learning continues with the existing courses:



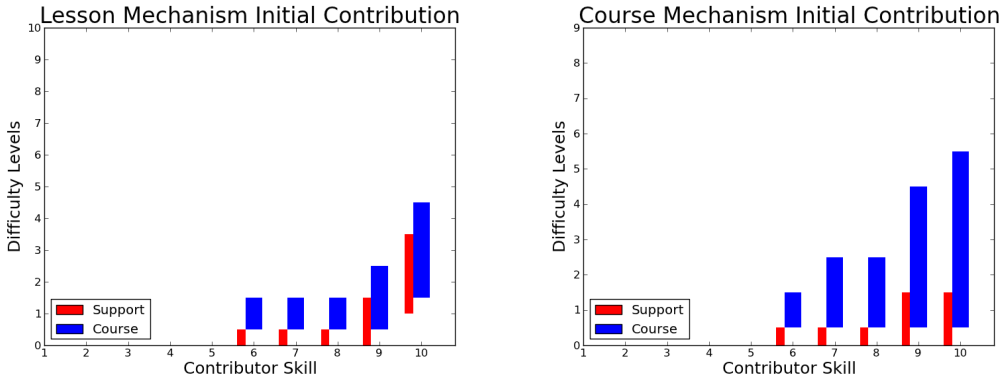


Eventually all learners will become experts or discouraged. The final distribution of learners under the ordered lesson mechanism is given below:



The process and final result of the learning process is vastly different between the two mechanisms, and we highlight the important differences in contribution patterns under the two mechanisms:

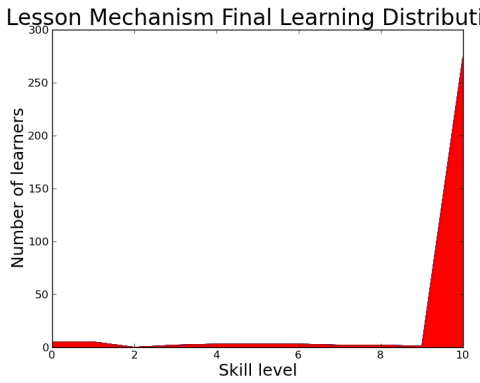
- There is a different relationship between contributor skill and the difficulty of the lessons created. This is best shown by comparing the first round of contribution in each mechanism:

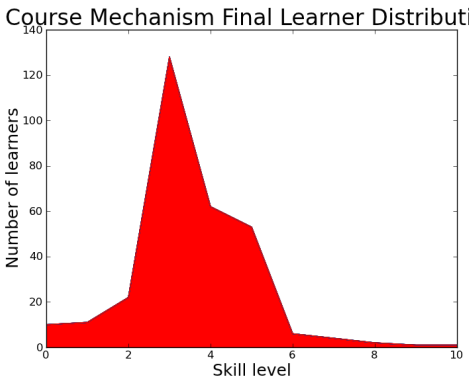


  In accordance with Theorem 5.1 the more skilled contributors tend to make more difficult lessons in the ordered lesson mechanism. Furthermore the ordered lesson mechanism immediately provides lessons for students as skilled as $s = 3$, represented by the highest red line in the first contribution graph, while the course mechanism only targets enrollment from students as skilled as $s = 1$.

This demonstrates that in the course mechanism all contributors rush to target novices initially, while in the ordered course mechanism skilled contributors are induced to target more skilled students from the very beginning.

- Most importantly for affecting the final learning distribution the ordered lesson mechanism incentivizes the contribution of many more advanced lessons than the independent course mechanism. The final distributions under the two mechanisms are presented side by side for comparison:

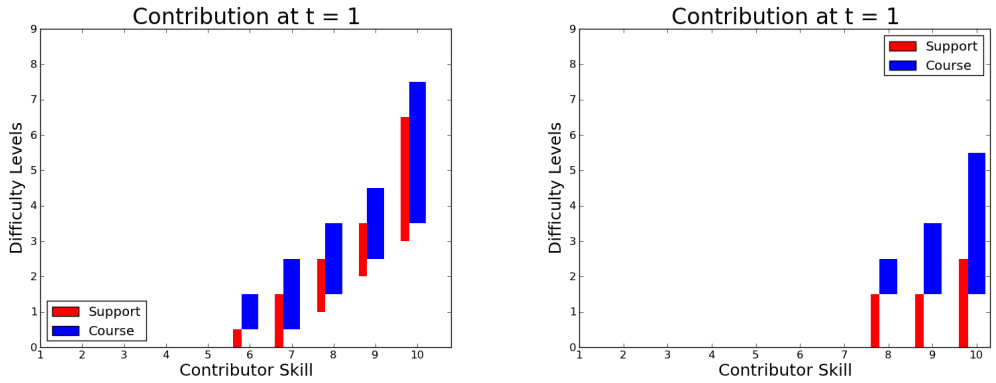




This advanced content allows learners to progress all the way to $s = 10$ or expert level, while those learners under the independent course mechanism cannot progress past $s = 5$. The ordered lesson mechanism facilitates the creation of advanced content because it allows contributors to create an advanced lesson with the knowledge that it will be delivered to all advanced students. Under the independent course mechanism contributors realize that only a small fraction of advanced students will be delivered an advanced course, because there are so many existing novice courses and courses are delivered randomly. This reduces the incentive to contribute advanced courses under the independent course mechanism.

We can also compare the first round contribution in the simulation case where the

course mechanism led to courses which targeted novices but started at level 2. Here the ordered lesson mechanism is shown on the left and the independent course mechanism on the right:



In this case the ordered lesson mechanism does deliver lessons starting at level 1, meaning the novice learners will not have to attempt to jump from level 0 to level 2. Furthermore it induces contribution of many more advanced lessons as well, allowing skilled learners to immediately begin learning.

6.2 Repeated Simulation

The simulations above demonstrate in a few cases that the ordered lesson mechanism can outperform the independent course mechanism and offer intuition for how that might generalize. To more quantitatively investigate the comparison of mechanisms under a wider range of circumstances we perform a series of random simulations under both mechanisms to compare learning outcomes.

6.2.1 Methodology

To gain a better sample within the parameter space we perform 100 trials picking parameters at random from certain ranges. The parameters were picked according to

uniform distributions in the following ranges:

Description	Functional form	Simulation value
Effort function	$e(d) = d^{e_a}$	$e_a = [1.0, 4.0]$
Learning Success Probability	$p(d) = (1 - \frac{d}{n})^{p_a}$	$p_a = [1.0, 4.0]$
Learning Utility	$u(s) = s^{u_a}$	$u_a = [.1, 1]$
Initial Learner Distribution	$L_0(s) = e^{-s*l_a}$	$l_a = [.05, 1.0]$
Building cost	$c(d) = (n - d)^{c_a}$	$c_a = [1.0, 4.0]$

These ranges were picked as moderate values which conformed to the desired convexity and concavity behavior of each function.

6.2.2 Aggregate Learning Utility

To evaluate a given learning outcome let the metric *aggregate learning utility* be the sum across the distribution of learners of the utility level of each learner as measured by our utility function u. Measured at the final time f, after all learners have either reached expert level or become discouraged, the final aggregate learning utility is then:

$$AU_f = \sum_{i \in L} u(s_{i,f})$$

One logical metric for determining how much learning took place in the system over the evolution of the platform is to compare this final aggregate utility with the initial aggregate utility and compute the difference:

$$\Delta AU = AU_f - AU_0 = \sum_{i \in L} u(s_{i,f}) - u(s_{i,0})$$

Note that this is not the sum of experienced utility of all learners. Rather it is a metric of the total learning that took place, in which we see how many skill levels each learner has gained and weight that increase in skill level according to the utility function u.

6.2.3 Random Simulation Results

Over the 100 trials, we calculated the following average aggregate utility changes (rounded to nearest tenth):

Mechanism	ΔAU	Standard Deviation
Independent Course	4478.5	2127.6
Ordered Lesson	10584.0	4478.5

While the absolute magnitudes are meaningless because the utility function along with all of the other parameters can be scaled arbitrarily, the relative magnitudes demonstrate a sharp contrast between the two mechanisms. On average the ordered lesson mechanism induced more than twice the gain in learning utility each trial when compared to the independent course mechanism.

Furthermore, the large standard deviations indicate that these parameters did generate wildly different system equilibriums. In some cases the change in aggregate learning utility was an order of magnitude more than in others. This gives us confidence that our parameters did explore an interesting and varied space, and we did not simply randomly pick from among a space of very similar outcomes.

Because of the flexibility of this model and the vastness of the parameter space it is difficult to sample over the entire space or generalize these results. However this quantitative comparison in combination with the intuition and visualization in the previous section gives us confidence that the ordered lesson mechanism does tend to

produce better learning outcomes, at least in the space of moderate parameters that we find most realistic.

7 Discussion

We view this paper as having an exploratory role in furthering our understanding of the recent phenomena of crowdsourced education systems. We believe it is the first to employ the tools of game-theoretic analysis to model the incentives and contribution behavior in these open education platforms.

To accomplish this we build a rich model which captures both the learning and contribution mechanics in an evolving system of discrete time steps in which learners gain mastery through discrete increases in skill level. We assume strategic but impatient decision-making on the part of the contributors, and thus each round of contribution is represented as a complete information simultaneous game with a free-entry Nash Equilibrium. This allows us to model contribution as a series of simultaneous games in which the payoffs evolve as the learners increase in skill.

To isolate the effect of content presentation we introduce two mechanisms by which a website can deliver submitted content to potential learners. In our baseline mechanism, representing a simplification of unorganized content, courses are delivered randomly to learners. We prove that under this mechanism there will initially be many courses targeted at novices, and further demonstrate other inefficiencies that can result. These inefficiencies include courses which target learners more than 1 level below the first lesson of the course, and most importantly a lack of advanced content, which can have a large adverse effect on learning outcomes.

We introduce an alternative mechanism in which contributors create courses consisting of a single lesson, and learners are presented the next most difficult lesson.

We prove that this mechanism guarantees that more skilled contributors will create more difficult lessons, which is desirable in the sense that more skilled contributors are likely better suited to create more difficult content. Through simulation we argue this mechanism tends to lead to a more complete distribution of content and therefore better learning outcomes, facilitating more than twice the gain in learning utility under randomized simulations.

These theoretical results suggest that existing platforms may be able to positively influence contributed content distribution and learning outcomes by encouraging contributors to include information about difficulty and making that information more transparent to browsing learners.

7.1 Future Work

We believe that these crowdsourced education platforms represent an exciting new kind of peer to peer community, with many avenues of potential study. Theoretically there are many ways to modify our model to study other phenomena associated with peer to peer education. One interesting modification would be to make contribution truly endogenous, such that any learner in the community could choose to be a contributor. A more extensive possibility is to revisit the notion of content quality and study the interplay between quality and difficulty.

Another realm of future work involves analyzing how more complex content organization and delivery mechanisms could be feasibly implemented. One challenge of the ordered lesson mechanism is determining the difficulty of lessons if these difficulties are not provided by the contributors or the contributors are not honest. It would be interesting to enlist the crowd in creating these orderings, by having people submit their own custom courses or tracks that they believe escalate in difficulty coherently.

Or one could use an artificial intelligence algorithm to determine ordering based on user behavior and failure rates.

As this new crowdsourced model of education continues to grow in popularity, research on these systems will be important in determining how best to structure platforms to incentivize desirable educational content and achieve optimal learning outcomes.

References

[1] Nikolay Archak and Arun Sundararajan. Optimal design of crowdsourcing contests. *ICIS 2009 Proceedings*, 200, 2009.

[2] John Seely Brown and Richard P Adler. Open education, the long tail, and learning 2.0. *Educause review*, 43(1):16–20, 2008.

[3] Shuchi Chawla, Jason D. Hartline, and Balasubramanian Sivan. Optimal crowdsourcing contests. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 856–868. SIAM, 2012.

[4] Arpita Ghosh and Patrick Hummel. A game-theoretic analysis of rank-order mechanisms for user-generated content. In *EC'11 Proc. 12th ACM Conference on Electronic Commerce*, pages 189–198. ACM, 2011.

[5] Arpita Ghosh and Preston McAfee. Crowdsourcing with endogenous entry. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 999–1008, New York, NY, USA, 2012. ACM.

[6] Arpita Ghosh and Aaron Roth. Incentivizing high-quality user-generated content. In *WWW '11 Proc. 20th ACM International World Wide Web Conference*, pages 137–146. ACM, 2011.

[7] Shaili Jain, Yiling Chen, and David C. Parkes. Designing incentives for online question and answer forums. In *EC '09: Proceedings of the tenth ACM conference on Electronic commerce*, pages 129–138, New York, NY, USA, 2009. ACM.

[8] Shaili Jain and David C Parkes. A game-theoretic analysis of games with a purpose. In *Internet and Network Economics*, pages 342–350. Springer, 2008.

[9] Shali Jain and David C. Parkes. The role of game theory in human computation systems. In *HCOMP '09 Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 58–61, New York, NY, USA, 2009. ACM.

[10] Daniel S Weld, Eytan Adar, Lydia Chilton, Raphael Hoffmann, and Eric Horvitz. Personalized online education-a crowdsourcing challenge. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[11] Dennis M Wilkinson. Strong regularities in online peer production. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 302–309. ACM, 2008.

[12] Ken Yeung. Udemy's teach2013 hits its stride as it signs up 15k experts, including dan rather and george lucas. *The Next Web*, March 2nd 2013.