Computation, Constructivism, and Curriculum Design

Thesis Proposal

Shayan Doroudi

November 16, 2017

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Emma Brunskill (Chair)
Vincent Aleven
Ken Koedinger
Chinmay Kulkarni
Eric Horvitz (Microsoft Research)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Copyright © 2017 Shayan Doroudi
Abstract

From the mechanical teaching machines of the early twentieth century to intelligent tutoring systems and the wave of massive open online courses in recent years, many have been motivated by the dream of personalized, adaptive instruction for all students. To achieve this goal, learning scientists and educational technology researchers have largely focused on rule-based systems that rely on extensive domain expertise and constrained machine learning algorithms combined with hand-selected rules for adaptive content selection. While this approach has led to the development of successful intelligent tutoring systems with high quality content, (1) developing such systems can be very costly and (2) they use a very limited form of adaptive content selection. In contrast, some researchers are now starting to apply black box statistical machine learning algorithms in attempting to achieve the dream of adaptive content selection. However, as I will show, these approaches have had relatively limited impact. Instead, I hope to demonstrate that combining insights from both approaches can help in building systems with cost-effective and impactful automated content selection. In particular, I address this using several methods that combine machine learning, human computation, and principles from the learning sciences. First, I will describe how reasoning about model mismatch (i.e., the fact that our statistical models of student learning do not accurately describe how students learn) can help point out limitations in existing approaches and help in creating more robust adaptive content selection policies. Second, I will show experiments that demonstrate how we can use students to create new content in a cost-effective way. In doing so, I will take motivation from the constructivist philosophy of education, whereby I view learner-generated solutions as being a projection of students’ constructions on the written plane, which can then be used to inform other students as they construct their own understandings. Third, I propose to demonstrate how using machine learning can help curate the best learner-generated content. Finally, I propose to show how we can use learning science principles to constrain the search for good activity selection policies. My dissertation aims to demonstrate both how insights from computer science and statistics can inform the learning sciences and how insights from the learning sciences can guide computational approaches with the goal of helping students learn.
Introduction

From Skinner’s teaching machines and programmed instruction in the 1960s to the wave of massive open online courses (MOOCs) in recent years, many attempts have been made to automate various aspects of education and curriculum design [Ferster, 2014]. Many recent developments in educational technology largely use automation to target two dimensions of educational impact: scalability and efficiency. Scalability is concerned with how many students are affected and efficiency is concerned with how much content is taught (in a fixed amount of time). For example, MOOCs try to scale course instruction to larger numbers of learners than we were previously capable of handling and use techniques like peer grading to scale assessment in a semi-automated way. Intelligent tutoring systems (ITSs) use cognitive mastery learning to help students more efficiently master a certain amount of material in a fixed amount of time. However, there is a third dimension of educational impact that I believe is not often used as motivation behind automated curriculum design, even though it is at the forefront of educational discussions in general: depth. The depth of learning is concerned with how well students understand the material, how well it transfers, how well it is retained etc. That is not to say that educational technologies that use automation never enable deeper learning, but that they are more often motivated in terms of their scalability and efficiency.

While many attempts have been made towards automating curriculum design to increase efficiency, for example by optimizing the sequence of activities for students to maximize learning, there are several limitations to these existing approaches that have limited their impact. By leveraging open-ended solutions generated by students and having other students use the solutions of their peers to construct their own understandings, I propose a new approach to automated curriculum design that aims to more deeply impact students. The idea is not to sacrifice impact in terms of scalability and efficiency, but rather to focus on depth in designing the approach to automated curriculum design.

Contributions

In my proposed dissertation, I aim to make four distinct but interrelated contributions:

1. I describe the limitations of existing approaches to automated curriculum design (namely mastery learning and reinforcement learning-based approaches) from three distinct perspectives: lack of empirical evidence, computational infeasibility, and incompatibility with constructivism. As part of this, I will reexamine the assumption that knowledge can be decomposed into independent components—an assumption that underlies existing approaches to automated curriculum design.
2. I propose a more constructivist approach to automated curriculum design that focuses on the creation of new educational content. I will describe results from a series of experiments in online educational settings that demonstrate how to effectively use learner-generated resources to help future students learn and how algorithms can be combined with learning theory to determine which resources should be used when. The vision of this approach is that if we start with no educational resources to teach a subject, we can use the crowd of learners and data-driven algorithms to create new resources—and if we do start with existing expert resources, using learner-generated resources can still enhance the existing curriculum. This approach is motivated by the constructivist philosophy of education, where I view learner-generated solutions as being a projection of students’ constructions on the written plane, which can then be used to inform other students as they construct their own understandings. Yet these experiments also contribute to existing research in the learning sciences by extending concepts such as the worked example effect, cognitive load theory, and self-explanation to the setting where learner-generated solutions are a resource in addition to expert solutions.

Much of the work in Chapter 1 is completed, while Chapter 2 consists of ongoing work and some work in early stages. At the end of this document, I provide a list of my relevant publications and a timeline of the steps I need to take in order to complete my dissertation.

**Computation**

My dissertation looks at how computational approaches can help automate curriculum design. I use the term computation in a broad sense, encompassing machine learning, artificial intelligence, and human computation. I discuss ways to use computation to automate curriculum design, but I also give insights into how computational and statistical principles (such as model robustness) can potentially help advance education research more broadly.

But at times, I use the term computation in an *even broader* sense to refer to the information processing that happens during (human) learning. While this is seemingly unrelated to the computational approaches used to automate curriculum design, there are actually a number of ways in which they are related. For example, a cognitive model of student learning based on an information processing psychology account of learning simultaneously tries to describe learning in computational terms, while also being able to help specify instructional policies that can be used to improve student learning (in an adaptive, automated fashion). I also use the term computation in this way to contrast it with constructivism, a theory of learning that is often seem to be at odds with information processing psychology.

**Constructivism**

It seems apt here to quote Duffy and Cunningham:

> An immediate difficulty confronts us...The term constructivism has come to serve as an umbrella term for a wide diversity of views. It is well beyond our purposes...to
detail these similarities and differences across the many theories claiming some kinship to constructivism. However, they do seem to be committed to the general view that (1) learning is an active process of constructing rather than acquiring knowledge, and (2) instruction is a process of supporting that construction rather than communicating knowledge. [Cunningham and Duffy, 1996]

For the purposes of this proposal, there are two important aspects of constructivism. First, constructivists challenge the notion that knowledge can be decomposed into independent parts and that instruction should teach each knowledge component independently until mastery [Jonassen, 1991; Resnick and Resnick, 1992; Shepard, 1991]. This has been an area of contention between constructivists and information processing psychologists [Anderson et al., 1999]; we will revisit this issue in the first part of my proposal. Second, as the quote above illustrates constructivist instruction should be focused on supporting students knowledge construction. By taking a constructivist stance as motivation, I look towards ways of using learner-generated resources to help students as they co-construct their own understandings and solutions.

Curriculum Design

In my dissertation, I consider three components of curriculum design: (1) the content, (2) the form of the activities that surround the content, and (3) and the (potentially personalized and adaptive) sequence of the content and activities. As a concrete example, Figure 1 shows an example of content and activities surrounding that content that we may use in a proof-based mathematics or number theory course. Experiments can shed light on what pieces of content are effective, what activities are effective, and how to effectively sequence these content and activities. The three components of curriculum design are not independent, but it is useful to study them independently as well as considering their interrelationships. For example, the efficacy of different forms of activities might depend on the content that gives them substance. Similarly, some activities that may seem ineffective in certain circumstances may be effective when provided at the right time in a sequence. For example, the worked example effect predicts that studying worked examples (e.g., Figure 1b) is more effective at teaching how to solve problems than actually doing problem solving activities (e.g., Figure 1d) [Sweller and Cooper, 1985]. However, the expertise-reversal effect suggests that while worked examples are more effective for novice students, students who have more expertise may benefit more from problem-solving activities [Kalyuga et al., 2003]. This effect not only sheds light on the relative efficacy of different activities, but also the way in which they should be sequenced. Both effects contribute to cognitive load theory, a rich theory which can help inform curriculum design. To determine principles of effective curriculum design, one can take a bottom-up approach where experiments validate particular principles (sometimes with limited generalizability) but can contribute to a larger theory. For example, a study that determines the efficacy of studying worked examples compared to problem solving in a physics course contributes to our understanding of the worked example effect as well as cognitive load theory. On the other, by taking a top-down approach, theories such as cognitive load theory can make predictions about how various aspects of curriculum design affect human cognition. These predictions can then be validated empirically to strengthen or modify the theory.
November 15, 2017
DRAFT

Question: Prove that $\sqrt{2}$ is irrational.

Solution:
1. Assume by way of contradiction that $\sqrt{2}$ is rational.
2. Let $\sqrt{2} = \frac{p}{q}$, where $p$ and $q$ are natural numbers that share no common factors.
3. Notice that $\sqrt{2}^2 = \frac{p^2}{q^2} \Rightarrow 2q^2 = p^2$ is even, so $p^2$ must be even.
4. Since $p^2$ is even, $p$ must be even, which means that $p^2$ must be divisible by 4.
5. But then $q^2 = \frac{p^2}{2}$ must be even, and so $q$ must be even.
6. This is a contradiction, since we assumed $p$ and $q$ share no common factors.

(a)

Review the following example:

Question: Prove that $\sqrt{2}$ is irrational.

Solution:
1. Assume by way of contradiction that $\sqrt{2}$ is rational.
2. Let $\sqrt{2} = \frac{p}{q}$, where $p$ and $q$ are natural numbers that share no common factors.
3. Notice that $\sqrt{2}^2 = \frac{p^2}{q^2} \Rightarrow 2q^2 = p^2$ is even, so $p^2$ must be even.
4. Since $p^2$ is even, $p$ must be even, which means that $p^2$ must be divisible by 4.
5. But then $q^2 = \frac{p^2}{2}$ must be even, and so $q$ must be even.
6. This is a contradiction, since we assumed $p$ and $q$ share no common factors.

(b)

Fill in the missing steps:

Question: Prove that $\sqrt{2}$ is irrational.

Solution:
1. Assume by way of contradiction that $\sqrt{2}$ is rational.
2. Let $\sqrt{2} = \frac{p}{q}$, where $p$ and $q$ are natural numbers that share no common factors.
3. Notice that $\sqrt{2}^2 = \frac{p^2}{q^2} \Rightarrow 2q^2 = p^2$ is even, so $p^2$ must be even.
4. Since $p^2$ is even, $p$ must be even, which means that $p^2$ must be divisible by 4.
5. But then $q^2 = \frac{p^2}{2}$ must be even, and so $q$ must be even.
6. This is a contradiction, since we assumed $p$ and $q$ share no common factors.

(c)

Answer the following question:

Question: Prove that $\sqrt{2}$ is irrational.

(d)

Figure 1: Content: (a) shows two examples of content (a question and a solution). Activities: (b), (c), and (d) show three examples of activities that can surround the content in (a). In the case of (b) and (c), the activities make different uses of the solution, whereas in (d) only the question is used (but perhaps the solution is presented to the student after they attempt to solve it).

However, another approach to curriculum design is to automate one or more of its components. In my dissertation, I focus on various approaches to automated curriculum design. I begin by examining existing approaches to automated curriculum design that have focused on the third component of curriculum design by looking at how to effectively adaptively sequence the curriculum. I show a number of existing limitations with this approach that have limited its success. I then look towards using human computation or crowdsourcing to provide a semi-automated approach to curriculum design to generate new content. Given the nature of this new content, we must also think about how students should engage with them in educational activities and how they should be sequenced with other existing educational resources. Rather than focusing on strictly computational techniques for doing curriculum design, I discuss how it can be advantageous to combine human computation, machine learning, and theories of learning to create curricula that can more deeply impact students in a semi-automated fashion.
Designing a curriculum, especially in the context of traditional education, can be much broader than the aspects of curriculum that I tackle in my dissertation. First of all, the approaches to automation that I consider here are only concerned with sequencing small scale pedagogical activities or generating solutions to problem solving tasks. By no means do I consider the automation of larger scale activities and content such as course projects or textbooks or aspects of curriculum that span across courses. Moreover, I do not consider automation with respect to other factors that significantly impact the curriculum, such as learning objectives, assessment, relationship to state standards, and the broader ecosystem in which the curriculum is positioned. However, I do look at automated curriculum design in the context of a MOOC. In this case, our results might suggest how impactful it is to automate various components of the curriculum in the broader context of a course where students have to engage with a lot content that is preexisting and not automatically generated.
Chapter 1

The Status Quo of Automated Curriculum Design

There have been several attempts to automate various aspects of curriculum design. One prominent approach, which I focus in on my dissertation, is choosing what educational activities to give students at any given time, typically in order to maximize the students’ learning. Cognitive mastery learning [Bloom, 1968, Corbett, 2000] is a standard approach to doing this that is used in many ITSs. Cognitive mastery learning typically assumes a model of student learning, such as Bayesian knowledge tracing (BKT), and provides students with practice on each knowledge component until the student is believed to have reached mastery for that knowledge component. An assumption made in cognitive mastery learning is the knowledge decomposition hypothesis—that knowledge can be decomposed into parts that can be learned independently once all prerequisite knowledge is learned [Corbett, 2000]. More recent approaches to automated curriculum design use reinforcement learning (RL) to try to find an instructional policy (a method of sequencing problems that is adaptive to some student state) to maximize some reward signal (such as learning gains from using the tutoring system), often assuming some model of student learning [Beck et al., 2000, Chi et al., 2011, Koedinger et al., 2013, Rafferty et al., 2015].

In this part of my dissertation, I will show limitations of both the cognitive mastery learning and reinforcement learning based approaches to automated curriculum design from three distinct perspectives. First, by doing an extensive review of the literature, I will show that there is a lack of empirical evidence that these approaches have been very successful. Then, I will describe two broad factors that could help explain this lack of empirical success: computational infeasibility and incompatibility with constructivism. However, I do not just seek to take a critical lens on these methods. Rather, I hope to find generalizable insights on what can factors can lead to more successful applications of these approaches to automated curriculum design. To that end, understanding why these methods have not been very successful could be helpful in trying to identify

1While the term cognitive mastery learning and mastery learning are often used interchangeably, I will try to use cognitive mastery learning to specifically refer to cases where mastery learning involves some cognitive model of student learning. We can contrast this with other approaches to mastery learning that were used in pen-and-paper settings [Bloom, 1968] as well as mastery learning approaches that use heuristics such as three-correct-in-a-row [Kelly et al., 2016].
cases where automated adaptive sequencing could be impactful. Before proceeding, I will give some necessary background on models of student learning.

1.1 Background: Student Models

Student models are often used in educational data mining and learning analytics research to make predictions about student learning given some input features. The most pervasive kind of student model and the kind we will be most concerned with are models that track whether or not a student answered questions correctly over time and make predictions as to whether or not a student will answer the next question correctly. In this section, we will briefly describe two of the most commonly used student models: the Bayesian knowledge tracing (BKT) model and the additive factors model (AFM). These models are often fit to data using standard machine learning techniques, but sometimes the models are used with parameters that are set arbitrarily or set by an expert. Student models can be used in a number of ways. One common usage is to make predictions about student learning in order to make inferences about student learning in some environment; to this end, AFM is often used to see how well students learn each skill taught by an ITS, which can then be used to modify our interpretation of which questions correspond to the same skill and in turn lead to improving the design of the ITS [Stamper and Koedinger, 2011].

Another use of student models, which is of more relevance to us, is using the model to derive an instructional policy. As such, the BKT model is typically used as the underlying model behind cognitive mastery learning in ITSs [Corbett, 2000]. Moreover, reinforcement learning methods use models such as Markov decision processes (MDPs) or partially observable Markov decision processes (POMDPs). Existing student models such as BKT can be augmented with a reward model so that they can be used to do reinforcement learning. Finally, student models can be used to simulate students answering questions, under the assumption that students actually learn according to the model. Such simulations can be used to evaluate instructional policies by having the instructional policy assign activities which are then “answered” by the simulated student. We now briefly describe BKT and AFM.

BKT is a two-state hidden Markov model that keeps track of the probability that a student has learned a particular skill and the probability that the student will be able to answer a question on that skill correctly over time. At each practice opportunity $i \geq 1$ (i.e., when a student has to answer a question corresponding to the skill), the student has a latent knowledge state $K_i \in \{0, 1\}$. If the knowledge state is 0, the student has not learned the skill, and if it is 1, then the student has learned it. The student’s answer can either be correct or incorrect: $C_i \in \{0, 1\}$ (where 0 corresponds to incorrect and 1 corresponds to correct). After each practice opportunity, the student is assumed to learn the skill with some probability. The BKT model is parametrized by the following four parameters:

2A reward model is a model that specifies a reward for the outcome of each action. For example, giving a student a problem that is answered correctly might have some positive reward. Alternatively, we may assume the only reward is given after the student is done interacting with an instructional policy, at which point the reward might be the student’s performance on some exam or posttest.
• $P(L_0) = P(K_1 = 1)$: the initial probability of knowing the skill (before the student is given any practice opportunities)
• $P(T) = P(K_{i+1} = 1|K_i = 0)$: the probability of learning a skill at each practice opportunity (if the student has not yet mastered the skill)
• $P(G) = P(C_i = 1|K_i = 0)$: the probability of guessing
• $P(S) = P(C_i = 0|K_i = 1)$: the probability of “slipping” (answering incorrectly despite having learned the skill)

BKT can be used online as students answer questions (for example in an intelligent tutoring system). For each skill, one can keep an updated belief about the student’s probability of having learned the skill ($P(K_i = 1)$). The standard approach to cognitive mastery learning is to assume that when $P(K_i = 1) > 0.95$ for a skill, then the student has mastered that skill (which means the student is very likely to have learned the skill). Cognitive mastery learning teaches each skill until we assume the student learns mastery, and then teaches skills that might be considered more complex or which have mastered skills as prerequisites. Notice that a key assumption of this approach is that every skill is independent; a student’s practice on skill $A$ has no implications for the student’s knowledge of skill $B$ and vice versa. The only potential dependency among skills is that some skills will be prerequisites for others, and cognitive mastery learning typically handles this by teaching prerequisite skills before postrequisites.

AFM is another popular model of student learning, which unlike BKT does not make any assumptions about whether the student knows the skill or not, but rather only tries to predict the probability that the student will answer a question correctly [Cen, 2009]. AFM is a logistic regression model that relates the probability of a student answering a question of a particular skill correctly (the dependent variable) to the number of times the student has had practice on that skill, the difficulty of the skill, and some general student ability parameter (i.e., an individualized parameter for each student that is the same regardless of the skill). AFM is slightly more general in that it allows for a question to target more than one skill. Concretely, the AFM model for some skill $k$ and student $i$ and question $j$ is governed by the following equation:

$$\log \left( \frac{p_{ij,T+1}}{1 - p_{ij,T+1}} \right) = \theta_i + \sum_k Q_{jk} \beta_k \gamma_k T_k$$

$p_{ij,T+1}$ is the probability that student $i$ will answer question $j$ correctly at time $T + 1$, $Q$ is a binary matrix that specifies which skills correspond to each questions, $T_k$ is the number of practice opportunities the student has had on skill $k$ up until time $T$, $\beta_k$ is the difficulty of skill $k$, $\theta_i$ is student $i$’s ability, and $\gamma_k$ is the learning rate at which practice on a skill leads to improved performance the skill. The parameters $\beta_k$, $\gamma_k$, and $\theta_i$ are typically simultaneously fit to data for all skills $k$ and students $i$ using standard algorithms that fit logistic regression models.

1.2 Lack of Empirical Evidence

I will begin by examining the empirical evidence for both cognitive mastery learning and RL-based approaches to sequencing educational activities and content. Cognitive mastery learning
has gained widespread use in ITSs, given preliminary results that it can successfully improve student learning [Corbett and Anderson, 1994]. This widespread use and the continued demonstration of the success of Cognitive Tutors and other ITSs that use it [Pane et al., 2014] has led to the general impression that cognitive mastery learning is beneficial for student learning. However, recent results have presented reason to question the importance of cognitive mastery learning. For example, a large-scale two year deployment study of the Cognitive Tutor Algebra I (CTAI) tutoring system across seven states found that in the second year of the study, high school students in classrooms that used the tutor had significantly higher posttest scores [Pane et al., 2014]. However, post-hoc analysis of this data using an analysis based on principal stratification (a causal inference method) found that the effect of the tutoring system was less (or at least not greater) for students who were more likely to adhere to mastery learning than for students who are less likely to master skills [Sales and Pane, 2017]. This result suggests that other factors beyond mastery learning may lead to the efficacy of the ITS, which could include for example the feedback mechanisms used by the ITS, the scaffolding it uses, or the way it was adopted in classrooms. But even if we were to assume that mastery learning is effective in some settings, how necessary is it to use cognitive mastery learning—that is, to use a cognitive model of student learning in order to determine when the student reaches mastery—as opposed to using simpler heuristics for mastery? Many systems such as Khan Academy and ASSISTments use mastery learning with a much more simpler method of detecting when the student has reached mastery: the student has mastered the skill when they answer the skill correctly $N$ times in a row (where $N$ is typically 3) [Hu, Kelly et al., 2016]. Researchers have recently compared cognitive mastery learning to this $N$-correct-in-a-row heuristic, and have found that the latter can be as good or even better than the former [Kelly et al., 2016, Pelánek and Rihák, 2017]. Furthermore, Pelánek and Rihák have shown that even if we assume BKT is the true student model (i.e., use it to simulate students), then using the true BKT model to detect mastery is only slightly better than using the $N$-correct-in-a-row heuristic [Pelánek et al., 2016].

However, my primary focus of this section will be on other approaches to adaptively sequencing content and activities, primarily using reinforcement learning-based approaches. I will conduct an extensive literature review of papers that have attempted to use reinforcement learning (or related methods, such as multi-armed bandits and other algorithmic ways of deriving instructional policies) to identify the degree to which such attempts have been successful (both in terms of finding a statistically significant result and more importantly, in terms of the effect size of the intervention). I will try to situate each result in the literature based on the educational domain the intervention took place in as well as the baselines that were compared to. The literature review will also include an in-depth analysis of two experiments that I helped conduct to test the efficacy of various adaptive instructional policies in an intelligent tutoring system that teaches fractions to fourth and fifth grade students (see e.g., [Doroudi et al., 2017a]). Both of these experiments showed no statistically significant difference between six different instructional policies, even though the instructional policies were very behaving very differently. Based on my under-

---

3 Students may not adhere to mastery learning for a particular skill even though the tutor uses cognitive mastery learning, either because they are likely to deplete all of the tutors’ questions on the skill without reaching mastery, or because for whatever reason their teacher progresses them to the next section before they are able to master the skill.
standing of the literature, my current understanding is that many approaches have not been very successful and effects have been meager at best; doing a thorough, objective literature review will help me develop a more informed opinion that I believe will be of value to the community of researchers engaged in and interested in automated curriculum design. However, it will also be useful to identify cases where RL can successfully be applied to find good instructional policies. My hope is that such a literature review can help find generalizable insights on how and when such approaches to automated sequencing may be impactful.

1.3 Computational Infeasibility

In trying to find factors that could help explain this lack of empirical success, I will first turn to the idea that it is computationally infeasible to find good instructional policies. By computational feasibility, I do not strictly refer to computational efficiency, but arguably more importantly, the statistical efficiency of the methods used to try to derive good instructional policies. I will give a variety of arguments from my own research on reinforcement learning and educational data mining to support this claim. For example, one reason is that if we assume a model of student learning that is too simple to capture the true complexity of student learning to be correct, then even with an infinite amount of data, we may learn parameters that are not only incorrect, but can have harmful implications for student learning [Doroudi and Brunskill 2017]. This is due to the bias of our choice of student models. I illustrate why using an overly-simplified model of student learning might lead to harmful implications for student learning in the case study below. On the other hand, if we resort to complex student models to model the inherent complexities of student learning, fitting accurate parameters to such a model would require prohibitively large amounts of data; this is due to the high variance of complex models. Furthermore, researchers, including myself, have shown that existing approaches to choosing a good instructional policy using existing data can lead to grossly inaccurate predictions [Doroudi et al., 2017a,b, Mandel et al., 2014]. In prior work, I do however provide one way to address this challenge by simulating instructional policies on various student models that make different assumptions about how students learn [Doroudi et al., 2017a]. If we find an instructional policy that helps simulated students learn, robust to the choice of the model that was used to simulate students, then we should have greater confidence that such an instructional policy will improve student learning in practice.

1.3.1 Case Study: Model Mismatch and Mastery Learning

Suppose student learning is actually governed by a 10-state HMM with ten consecutive states representing different levels of mastery. From each state, the student has some probability of transitioning to the next state (slightly increasing in mastery), and from each state, the student has a probability of answering questions correctly, and this probability strictly increases as the student’s level of mastery increases. Specifically consider the model presented in Table 1.1. Now suppose we try to use a standard BKT model to fit data generated from this alternative model of student learning. The first two columns of Table 1.2 show the parameters of BKT models fit to
<table>
<thead>
<tr>
<th>Parameter</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(K_0 = k)$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$P(C_i = 1</td>
<td>K_i = k)$</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>$P(K_i = k + 1</td>
<td>K_i = k)$</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 1.1: Alternative model of student learning where there are ten levels of mastery.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>10-State HMM</th>
<th>AFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(L_0)$</td>
<td>0.30</td>
<td>0.09</td>
</tr>
<tr>
<td>$P(T)$</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$P(G)$</td>
<td>0.27</td>
<td>0.14</td>
</tr>
<tr>
<td>$P(S)$</td>
<td>0.44</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Table 1.2: BKT models fit to data generated from the model described in Figure 1.1 and an additive factors model described in the text. The first column for each model is fit to 500 sequences of 20 practice opportunities, while the second column is fit to 100 sequences of 200 practice opportunities. The models were fit using brute-force grid search over the entire parameter space in 0.01 increments for the parameters using the BKT Brute Force model fitting code [Baker et al., 2010].
500 sequences of 20 practice opportunities or 100 sequences of 200 practice opportunities, both
generated from the model in Table 1.1. Notice that the model fits very different parameters
in the two cases. When we only have 20 observations per student, the model estimates a very
high slip parameter; this is because it has to somehow aggregate the different latent states which
correspond to different levels of mastery, and since not many students would have reached the
highest levels of mastery in 20 steps, it is going to predict that students who have “mastered”
the skill are often getting it wrong. However, what’s more interesting is that for the same model,
if we simply increase the number of observations per student from 20 to 200, we find that the
slip parameter is reasonably small, but now the guess probability is 0.49! This is because, by
this point most students have actually reached the highest level of mastery, so to compensate for
the varying levels of mastery that occurred earlier in student trajectories, the model will have to
estimate a high guess parameter. This is a counterintuitive phenomenon that we believe is not
the result of not having enough data (students) to fit the models well, but rather the result of the
mismatch between the true form of student learning and the model we are using the fit student
learning.

We find similar results if we fit a BKT model to data generated from an AFM model. In particular,
we used the model
\[
P(C_i = 1) = \frac{1}{1 + \exp(-\theta + 2 - 0.1i)}
\]
where \(\theta \sim \mathcal{N}(0, 1)\) is the student’s ability\(^4\). The second two columns of Table 1.2 show the
parameters of BKT models fit to data generated from this model. We again find that when using
only data with 20 practice opportunities, we fit a high slip parameter, but when we using data
with 200 practice opportunities, we fit a higher guess parameter and a very small slip parameter.
These observations have important implications for how learned models might be used in auto-
mated sequencing of content, such as cognitive mastery learning. Using such a BKT model to
predict student mastery can lead to problematic inferences. For example, for the first model in
Table 1.2 the BKT model assumes that when a student has reached mastery, they have a 56%
chance of answering a question correctly, whereas a student who has actually mastered the skill
will have a 90% chance of answering correctly (see Table 1.1). Thus, an intelligent tutoring
system that uses such a BKT model to determine when a student has had sufficient practice on
a problem, will likely give far fewer problems to the student than they actually need in order to
reach mastery. To illustrate this, Table 1.3 shows the expected number of practice opportunities
the first model in Table 1.2 will give, when students actually learn according to Table 1.1. In
contrast, the average number of practice opportunities needed to reach mastery according to the
true model is around 100. Thus, cognitive mastery learning could lead to a significant amount
of under-practice, even with a very high mastery threshold (0.9999). This case study provides
an example of how reasoning about model mismatch (i.e., lack of robustness to the choice of
student model) can be informative in terms of the instructional consequences of our models.

\(^4\)This model suggests that students who are two standard deviations above the mean initially will answer correctly
half the time, and after 20 practice opportunities the average student will answer correctly half the time.
### Table 1.3

<table>
<thead>
<tr>
<th>Mastery Threshold</th>
<th>Exp # Opp. to Mastery</th>
<th>% Students with Under-Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95</td>
<td>28.4</td>
<td>99.4%</td>
</tr>
<tr>
<td>0.99</td>
<td>38.3</td>
<td>99%</td>
</tr>
<tr>
<td>0.9999</td>
<td>53.4</td>
<td>95%</td>
</tr>
</tbody>
</table>

Table 1.3: The expected number of opportunities needed (averaged over 500 students) for the model given in the first column of Table 1.2 to reach mastery for various mastery thresholds, given that the true model is the model from Table 1.1. The third column shows the percentage of simulated students that received less practice than needed. In contrast, the average number of opportunities that it took simulated students to reach mastery was around 100.

### 1.4 Incompatibility with Constructivism

The second factor that I believe may contribute to the lack of empirical success of existing automated curriculum design methods is that these approaches are incompatible with constructivism (and related theories of learning). As mentioned above, a constructivist perspective would contest that knowledge can be decomposed into independent knowledge components [Jonassen, 1991, Resnick and Resnick, 1992, Shepard, 1991]. If students learn in a more constructivist fashion, then existing approaches that assume knowledge decomposition should not be expected to necessarily succeed. While constructivists have made arguments against the decomposition of knowledge, these arguments are often qualitative and have not been convincing to many cognitive psychologists [Anderson et al., 1999]. I am currently using a novel educational data mining approach to reexamine the knowledge decomposition hypothesis from a quantitative perspective. I do not mean to claim that knowledge decomposition is never a useful approximation, but rather, that, at least in some cases, knowledge of different skills may interact in complex ways that current approaches do not take into account.

Koedinger et al. have recently compared two models of student learning that correspond to different theories of knowledge transfer and make different claims about knowledge decomposition: one model (standard AFM) assumes that all skills are independent and a second model assumes that there is only one skill (a strict no-decomposition model, or AFM where all questions map onto the same skill) [Koedinger et al., 2016]. They show that the full-decomposition model has higher predictive accuracy on a number of datasets than the no-decomposition model. They use this to make the claim that a particular theory of transfer (the component theory) that assumes knowledge transfers only between questions that target the same skill but not from skill to skill is more plausible than another theory of transfer (the faculty theory) which assumes the more practice one gets on any skill, the better one gets on all skills. While I think that is a reasonable claim (to the extent that these models accurately capture the associated theories of transfer), I think there may be alternative explanations of transfer and knowledge decomposition that are not explained by looking at these models alone. Moreover, in several cases, the two models have

5As a technical note, the researchers actually compare multiple versions of these models. For the current purpose, I am only considering their Weak IRT + γ model and Weak AFM. Additionally, I am only considering the comparison of these models using student-stratified cross validation for reasons I will explain in my dissertation.
similar or equal predictive accuracy. Why might this be if the no-decomposition model is making such a different claim about the nature of knowledge?

I argue that there is an alternative possibility that knowledge cannot be decomposed into different knowledge components, but rather different knowledge “components” are actually intertwined and interrelated in complex ways. Furthermore, I hypothesize that if that is the case, then AFM with a single-skill (in other words, the no-decomposition model) will fit student performance data better than a model that assumes full knowledge decomposition, because the former captures the fact that skills might be interrelated, even if it does not model their relationships perfectly. To support this claim, in ongoing work, I am simulating students answering problems under the assumptions of various hypothetical models of complex learning (i.e., where practice on various skills reinforce other skills to different extents), and then fit both a no-decomposition model and a full-decomposition model to the simulated data. I show that the no-decomposition model often has higher predictive accuracy and is thus more robust to more complex, constructivist models of learning. I am now looking into when a no-decomposition model has higher predictive accuracy than a full-decomposition model on real datasets to gain insights on domains and settings where we have reason to believe that knowledge does not decompose into independent components. Notice that we could have alternatively compared more complex models of student learning that are better aligned with constructivism to see if they have a higher predictive accuracy than the two extremes of no-decomposition or full-decomposition. However, this would not actually work because (1) there are many possible models that could capture the potentially complex interrelationships between skills, and (2) even if we knew how to model student learning, we would need prohibitively large amounts of student data to fit such a complex model—as we argued above—and so we would actually expect such a model to have lower predictive accuracy than the others. Thus by analyzing the robustness of these two models in predicting student learning assuming learning happens in more complex ways, we now have a more concrete way to quantitatively reexamine the knowledge decomposition hypothesis. This also serves as a concrete example of how reasoning about model robustness can help improve our understanding of student learning.

Beyond the issue of knowledge decomposition, there are other ways in which constructivism may be incompatible with current methods towards automatically sequencing curricula for students. For example, constructivist theories typically promote giving students more agency [Jonassen, 1991]. Approaches to automated curriculum design that mandate a particular sequence of activities to a student severely limit the amount of agency the student has. This may have negative motivational effects as well as possibly negative cognitive effects to the extent that the student is able to determine in some instances what educational activity they need to pursue next to further their own learning. Recent results indicate that giving students agency in the activities they do can outperform reasonable ways of (non-adaptively) sequencing problems for students [Adjei et al., 2017; Segal et al., 2016]. In my dissertation, I will examine such arguments from a constructivist perspective against current approaches to automated curriculum design. There are ways to mitigate these issues and make automated curriculum design more amenable to constructivism. For example, one can use models or instructional policies that are more robust to constructivist notions of learning (as described above); even if the models used to derive instructional policies are incorrect, robustness to different models of learning may help give us confidence about their im-
pact. Furthermore, to address the issue of student agency, perhaps instructional policies can give students options (at least in certain circumstances); these options could be given when the policy has uncertainty over what action is beneficial. In this way, we might be able to simultaneously address the issue of student agency and the issue of lacking confidence in the performance of our policies. Further work in these areas could advance the impact of automated curriculum design while making these approaches more amenable to constructivism. In what follows, I will take a different approach by looking towards a more constructivist approach to automated curriculum design that focuses on the semi-automated generation of new educational content.
Chapter 2

A New Approach to Automated Curriculum Design

The core of my dissertation will focus on proposing a new approach to automated curriculum design that focuses on the creation of new educational content. My work builds on the recently developed concept of learnersourcing [Kim et al., 2015]—using the activity of learners to improve the learning experience of other learners—by expanding on a formative theory of how and when to present learner-generated solutions and how algorithms and learning theory can be used to determine which solutions are best and when they should be presented. The vision of this approach is that if we start with no educational resources to teach a subject, we can use the crowd of learners and data-driven algorithms to create new resources—and if we do start with existing expert resources, using learner-generated resources can still enhance the existing curriculum. Learnersourcing is a specific instantiation of crowdsourcing or human computation using people to do tasks that are difficult or impossible to do with a computer and leveraging the wisdom of crowds to get reliable solutions to those tasks. Recent work on learnersourcing has given some insights into how learner-generated explanations can be effectively presented to future learners. Of most relevance to my work, Williams et al. developed a system for improving the quality of explanations over time through learnersourcing and the use of multi-armed bandits (MABs). The researchers demonstrated the efficacy of the system by having crowdworkers write and rate explanations for a mathematical task, and using their MAB algorithm to try to discover the best explanations for teaching future learners, and showed that explanations that result from the system can be of comparable quality to one generated by an expert teacher [Williams et al., 2016]. Moreover, in recent years, researchers have written vision papers on how human computation can impact the future of education. In 2012, Weld et al. described how human computation can address new challenges in personalizing online education in the wake of Massive Open Online Courses (MOOCs) [Weld et al., 2012]. One of the challenges they discussed was content creation and curation in online courses, and how crowds of students could be used for that purpose. Their paper could be seen as a call to action for human computation researchers; this section of my dissertation can be seen as an answer to that call. Moreover, in 2016, Heffernan et al. predicted that “in many ways, the next 25 years of adaptive learning technologies will be driven by the crowd” and described their efforts to begin to use crowdsourcing for content creation in ASSISTments (a system that teachers use to teach mathematics in the classroom) [Heffernan et al., 2016].
The first question in determining how to use learner-generated content is how do we generate it? In many settings, students will naturally generate solutions to problems. In other settings, where generating solutions is not natural, students can be asked to give a self-explanation of a concept or of how they approached a problem. In my dissertation, I will primarily focus on learner-generated solutions to problems, but I believe the methods could be extended to generating explanations of concepts, hints, etc. Once we have a way of eliciting learner-generated content, we must answer several questions in order to discover how to best use this content. While our approach to automated curriculum design focuses on using human computation to have learners create new educational content, the creation of this new type of content necessitates us to asks how we should modify other aspects of curriculum design so that we make the most effective use of this content.

First, in what ways should students engage with content generated by their peers? In other words, what kind of activities should surround the content? Once we have identified effective ways of engaging with the content, we can return to the question of the automated design of content and ask: how can we curate the best content for students to engage with? Finally, in when generating this content in settings where there is already an existing curriculum (e.g., in K-12 education or an online course), we must ask what the place of this new content should be in this curriculum. That is, how should we integrate the learner-generated content with other educational resources such as expert solutions? Here we return to the form of automated curriculum design that we started with: sequencing. The idea is that engaging with learner-generated content may be useful beyond engaging with expert-generated content. If so, how can we effectively make use of both by sequencing them appropriately? We will now briefly discuss how we propose studying each of these questions that target three components of curriculum design: activities, content, and sequencing. But first, I will describe the domains and settings in which we are testing these ideas.

### 2.1 Domains

We are currently running experiments in two online educational settings. The first is a crowdsourcing setting where crowdworkers do typically small tasks for small wages. We are interested in exploring the ability to train crowdworkers to do more complex tasks. To this end, we have been running experiments to train crowdworkers to do complex web search tasks, where workers have to answer complicated questions by making a series of cleverly crafted search queries. Complex crowdsourcing tasks are interesting settings to test the efficacy of learner-generated content because in such settings there may not be existing curricula to train crowdworkers and as the nature of work can constantly change, it may not be feasible to develop such curricula using expert knowledge. Our second setting is MOOCs. In particular, we are currently planning to run some experiments on a MOOC entitled “Introduction to Mathematical Thinking” taught by Keith Devlin. We are collaborating with the instructor to test the efficacy of having students engage with different activities and sequences of activities that involve learnersourced mathematical proofs. This course is already using peer evaluation to help students obtain better
proof evaluation skills, so it is a natural setting in which to test the learning gains of peer eval-
uation. Any interventions run in this course will only modify certain assignments in the course;
by testing the efficacy of various interventions on long-term course outcomes (e.g., the quality
of students’ proofs at the end of the course) we get to see how effective the ideas developed in
my thesis can actually be when integrated into authentic educational settings. In order to test the
generalizability of my findings, I also hope to run experiments in at least one other MOOC or
crowdsourcing setting.

While these two domains are seemingly quite different, they share something in common. In
both settings engaging with peer solutions, and in particular evaluating peer solutions is an au-
thentic task that the learners will need to engage in if they continue in that field. For example,
in crowdsourcing settings, crowdworkers will often have to evaluate the work of their peers, as
task requesters need third-party confirmation that the work done by a worker was correct and ade-
quate. This is also certainly true in mathematics where peer evaluation of proofs is a necessary
part of developing mathematical knowledge. According to Cobb in his constructivist critique of
information-processing approaches to math education, “a mathematical truth is true because a
community of knowers makes it so...it is the dialectical interplay of many minds that determines
whether a theorem is both interesting and true” [Cobb 1990]. He then cites De Millo et al.:

> After enough internalization, enough transformation, enough generalization, enough
> use, and enough connection, the mathematical community eventually decides that
> the central concepts of the original theorem, now perhaps greatly changed, have an
> ultimate stability. If the various proofs feel right and the results are examined from
> enough angles, then the truth of the theorem is eventually established. The theorem
> is thought to be true in the classical sense—that is, in the sense that it could be
demonstrated by formal deductive logic, although for almost all theorems no such
deduction ever took place or ever will [De Millo et al. 1980].

From this perspective, having students validate mathematical proofs generated by their peers
simultaneously engages them in an authentic practice that mathematicians engage in while hope-
fully also improve their own proof writing techniques. Thus, in this framework, peer solutions
are not to be thought of as just a poor man’s substitute for expert examples, when we do not have
access to the latter. By having learners generate content, we are creating content that might be
useful in ways that other content is not.

### 2.2 Activities

Learnersourced content cannot, in general, be used in the same way as expert examples because
we do not know the quality of the content and whether it is even correct. Therefore, it is essen-
tial that we think of new ways that students can engage with this content. In my prior work, I
identified that having crowdworkers validate solutions of other crowdworkers to a complex web
search task can be an effective way of training those workers (i.e., increasing their accuracy on
future web search tasks), and can possibly be as effective as reading through expert solutions and
more effective than problem solving with self-explanation [Doroudi et al. 2016]. While this is
a promising result, many open questions remain that I hope to address with future experiments.
How does validating a peer solution compare to simply reading it (as one would read an expert example)? What if we instead present multiple peer-generated examples and have learners generate their own solutions after comparing and contrasting the peer solutions? I hypothesize the latter would be an effective way of getting learners to simultaneously learn from peer content but also providing enough flexibility to construct their own understandings. By prompting students to construct their own solutions after reviewing peer solutions, we hope to off-load the task of personalization to the students themselves. We present examples of student work to inspire the student, but it is up to the student to take the parts of different solutions that are most meaningful to them to construct their own solution. Good constructivist instruction must be robust to the variety of ways knowledge is organized in different students’ minds; I hope to achieve this robustness by having students compare and contrast solutions generated by their peers and construct their own understanding as a result. Notice that this is a qualitative re-interpretation of constructivist instruction as an instance of model robustness. However, I do not think presenting a random set of solutions to the student as inspiration would necessarily be effective. Instead, I propose using a heuristic way of constructing a set of solutions given a rubric for grading solutions. For example, we can construct a set of solutions such that no solution has a perfect score on every rubric item, yet for each rubric item, there exists at least one solution in the set that has a perfect score.

2.3 Content

Once we have an understanding of the cognitive and motivational benefits of different types of tasks involving peer examples, we can then try to use computational methods to curate these tasks. The second type of experiment that I propose running is aimed at identifying and curating the best learnersourced solutions to present to future learners. Of course, the quality of a solution will depend on the activity whereby students engage with the learner-generated content. For example, if students are to simply read a peer example, the best solutions would most likely be examples that are factually correct and resemble expert examples. However, if students are to evaluate solutions, we may want to present them with some wrong solutions that are lacking in many ways for pedagogical purposes. It would be useful to have an automated way of curating good solutions to present to learners regardless of the activity in which learners engage with the solutions.

I propose to do this by first automatically extracting features of learnersourced solutions (e.g., how long the solutions are, how many steps they contain, what kind of language is used, bag of words representations of the solution etc.) and learning a model that predicts accuracy on future tasks based on the solutions that are initially presented to the learners. We can then use this model in a multi-armed bandit algorithm to determine which solution to present learners at any given time; MAB algorithms try to balance exploration (trying to present new solutions to find good ones) and exploitation (presenting solutions that the algorithm has so far identified to be the most effective). The reward signal that we will use for our MAB will be how well learners perform on future problem solving tasks; but we could incorporate other signals such as learners’ perception of how effective the example was in teaching them. This relates to recent work by
Williams et al., which used a MAB algorithm to try to discover the best explanations for teaching future learners [Williams et al., 2016]; however, there, a standard MAB algorithm was used and not one that shares information across different solutions. My prior work has given preliminary evidence that features of solutions can be reasonable proxies for their efficacy in helping future learners; in particular, we found that having crowdworkers validate a peer-generated solution beyond a certain length could be as effective (or possibly more effective) than reading expert examples [Doroudi et al., 2016]. I seek to expand on this work by using features that are shared across solutions so that we can evaluate the efficacy of newly generated solutions without actually needing to test every new solution on students. This can be especially useful in settings such as MOOCs where we may continually generate more and more solutions over time. We can also use MAB algorithms to automatically find sets of good solutions in activities where we want students to engage with such sets, provided that we use a featurized representation of the set of solutions. Identifying what features are most salient for good solutions or sets of solutions could be an informative result for the learning sciences community in its own right. Additionally, we hope to make algorithmic advances on MAB algorithms that could be of interest to the broader machine learning community.

2.4 Sequencing

Finally, the third type of experiment that I propose is to discover how to best sequence tasks involving expert and learner-generated resources. This is especially useful in settings where we already have expert resources, but we want to use learner-generated resources to augment the existing curriculum. In hypothesizing effective ways of integrating expert and peer examples together, we turn to the learning sciences literature. The expertise-reversal effect claims that novice students benefit more from studying worked examples but expert students benefit more from problem solving [Kalyuga et al., 2003]. This effect has been justified in terms of cognitive load theory, which claims the cognitive load of problem solving is too high for novices, but is reduced when a novice obtains expertise. I propose to build on this literature by extending cognitive load theory to the setting where novices can also interact with the work of their peers. I hypothesize it would be advantageous to have students initially read expert examples, followed by validating or improving peer solutions, followed by problem solving, due to the perceived cognitive load of each task. I predict that validating or improving peer solutions has higher cognitive load than simply reading an expert example, as the validation process requires the student to engage in more effortful information processing. At the same time, the cognitive load should be less than that of problem solving, because the student does not need to solve the problem from scratch. I will attempt to verify this experimentally by comparing several ways of sequencing these different tasks.

If we experimentally identify the appropriate ordering of tasks, we can then try to adaptively personalize the sequence of tasks. There have been several successful attempts to adapt the sequence of tasks based on cognitive load theory. Researchers have identified various ways of identifying the cognitive efficiency of a student working on a particular task and using that cognitive efficiency to determine whether to provide a task with more or less scaffolding (i.e.,
worked example, partially worked example, or complete problem solving task) at the next time step [Kalyuga and Sweller, 2005, Najar et al., 2016]. Additionally, Salden et al. developed a BKT-based algorithm that determined what level of scaffolding to give to a student based on how the belief of them mastering the skill compared to various thresholds [Salden et al., 2010]. I propose taking a similar approach whereby we use a proxy for the cognitive load on each student to determine if they are ready to move on to the next task type. Notice that this form of adaptive sequencing is much more constrained than finding open-ended instructional policies. And even though it shares much in common with mastery learning, it does not assume knowledge decomposition and it is lower stakes—giving a problem with the wrong level of scaffolding is presumably less problematic than giving a student a problem for a skill they are not ready for.
Chapter 3

Relevant Publications and Timeline

3.1 Relevant Publications

Below are a list of some of my relevant publications that inform the completed parts of my dissertation.

  
  **Best Paper**

  
  **Nominated for Best Paper**


3.2 Timeline of Proposed Work

Below is a timeline for my proposed work towards completing my thesis.

- By January 2018:
• Complete content curation experiments for web search tasks and submit to International Joint Conference on Artificial Intelligence.

• By March 2018:
  • Complete review of reinforcement learning approaches to adaptive content selection and submit paper to Journal of Educational Data Mining.
  • Complete experiment testing different learnersourcing activities on MOOC and submit as Work-in-Progress to Learning @ Scale.

• By May 2018:
  • Complete initial activity sequencing experiment on MOOC.

• By August 2018:
  • Complete adaptive sequencing experiment on MOOC.
  • Start writing dissertation.

• By December 2018:
  • Attempt to replicate any interesting findings (for example, run bandit experiment performed on web search domain on the mathematical thinking domain) as time permits.
  • Complete dissertation.
Conclusion

In this dissertation I have taken a critical view towards existing approaches to automated curriculum design, and taking insights from the limitations of the status quo, I motivated a new form of automated curriculum design. But if curriculum design was the sole focus of my dissertation, I could have titled it: Computational Constructivist Curriculum Design. Rather, I believe the ideas in my dissertation extend beyond curriculum design and also speak to the broader relationship between computation and constructivism, which are often taken to be at ends. For example, Cobb identifies a “trade-off between experience and precision” in characterizing the differences between constructivism and information processing psychology respectively [Cobb 1990]. While I show there is some truth to this, I also show there are many ways in which computation (and precision) can inform constructivism and vice versa. I hope that my dissertation can serve as an exemplar of melding different approaches to education research, discovering which aspects are and are not compatible, and doing more impactful research as a result.

Acknowledgements

The research reported here was supported, in whole or in part, by the Institute of Education Sciences, U.S. Department of Education, through Grants R305A130215 and R305B150008 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Dept. of Education. I am fortunate to have worked on (and continue to work on) much of the research presented here with a number of collaborators. My advisor Emma Brunskill was a close collaborator on much of the work presented throughout the dissertation. The work in Chapter 1 was done and is being done in collaboration with Ken Holstein, Vincent Aleven, Phil Thomas, Seth Adjei, and Neil Heffernan. The work in Chapter 2 is being done in collaboration with Ece Kamar, Eric Horvitz, Minsuk Chang, Juho Kim, and Keith Devlin.
Bibliography


John R Anderson, Lynne M Reder, and Herbert A Simon. Applications and misapplications of cognitive psychology to mathematics education. 1999. (document) 1.4


Shayan Doroudi and Emma Brunskill. The misidentified identifiability problem of bayesian


