Enhancing Individualized Instruction through Hidden Markov Models

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Mathematical Science in the Graduate School of The Ohio State University

By

David Seaman Lindberg III, M.S.

Graduate Program in Mathematics

The Ohio State University

2014

Thesis Committee:
Rodica Costin, Advisor
Dennis Pearl
Bart Snapp
Abstract

Online education in mathematics has become an important research topic. With more assessment going online, we have to ask ourselves, “How do we measure student performance through a computer?” A second question we ask is, “How do we evaluate the individualized instruction platform itself?”

We first provide a summary of personalized education in mathematics. We discuss case studies on certain individualized instruction platforms with commentary on how students are learning mathematics. Analysis of these case studies inform us on possible challenges in their design and use.

We next present a hidden Markov model as a way to analyze student learning in an individualized instruction system. The Baum-Welch algorithm provides a means to determine the parameters of a hidden Markov model. It’s these parameters that give insight into exercises and how students are performing with each exercise.

We then consider the Viterbi algorithm, which is an algorithm used to uncover the hidden states in a hidden Markov model. A student’s sequence of observations is recorded by the system, which is the input to the Viterbi algorithm. The hidden performance states of the student (unknowing, emerging, and knowing) are generated as a time-series sequence representing student understanding on a particular exercise over time.
We finally provide recommendations for implementing the *hidden Markov model* in individualized instruction systems and further research questions are posed.
I acknowledge and thank Dr. Bart Snapp for his guidance, patience, and encouragement through the process of writing this thesis.

I also acknowledge and thank Dr. Rodica Costin for being my academic advisor through the MMS for Educators program. I also thank her for serving on my thesis committee.

I also acknowledge and thank Dr. Dennis Pearl for his service on my thesis committee.

I finally want to thank all of the professors, students, and staff that I came to know at The Ohio State University.
Vita

June 2003 ............................... Palm Harbor University High School

December 2006 .......................... B.S. Applied Mathematics, Georgia Institute of Technology

December 2006 .......................... B.S. Electrical Engineering, Georgia Institute of Technology

May 2008 ................................. M.S. Electrical and Computer Engineering, Georgia Institute of Technology

August 2012 to December 2013 .......... Graduate Teaching Associate, Department of Mathematics, The Ohio State University.

January 2014 to present ............... eLearning Systems Developer, Office of Distance Education and eLearning, The Ohio State University

Fields of Study

Major Field: Mathematics
# Table of Contents

Abstract ................................................................. ii  
Acknowledgments ......................................................... iv  
Vita ................................................................. v  
List of Tables ........................................................... viii  
List of Figures .......................................................... ix  

Chapter: ................................................................. Page  

1. Introduction .......................................................... 1  
2. Personalized education ................................................. 3  
   2.1 Introduction to individualized instruction ...................... 3  
   2.1.1 One-to-one tutoring ......................................... 4  
   2.1.2 Individually Prescribed Instruction Mathematics ........ 5  
   2.1.3 Khan Academy ............................................. 7  
   2.2 Challenges of individualized instruction platforms .......... 10  
   2.2.1 Challenges of one-to-one tutoring ......................... 10  
   2.2.2 Challenges of Khan Academy ............................... 10  
   2.2.3 Challenges of Individually Prescribed Instruction  
       Mathematics .................................................. 21  
   2.3 Design considerations of Individualized Instruction ........ 23  
3. Hidden Markov models and education ............................... 26  
   3.1 Introduction to hidden Markov models ......................... 26  
   3.1.1 Basics of probability ..................................... 27
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Number of total attempts for each number of switches.</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Additional experimental data on observations needed for memorization.</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of experimental data to proposed model for attempts to memorization.</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>Summary of example hidden Markov model parameters.</td>
<td>84</td>
</tr>
<tr>
<td>4.1</td>
<td>Sample student data used in determining Baum-Welch parameters</td>
<td>101</td>
</tr>
<tr>
<td>4.2</td>
<td>Viterbi algorithm applied to sample student data</td>
<td>101</td>
</tr>
<tr>
<td>4.3</td>
<td>Sample student data of a student memorizing answers</td>
<td>102</td>
</tr>
<tr>
<td>4.4</td>
<td>Viterbi algorithm applied to memorizing student</td>
<td>103</td>
</tr>
<tr>
<td>4.5</td>
<td>Viterbi algorithm applied to sample student data with three hidden states</td>
<td>106</td>
</tr>
<tr>
<td>4.6</td>
<td>Viterbi algorithm applied to memorizing student with three hidden states</td>
<td>106</td>
</tr>
<tr>
<td>4.7</td>
<td>Viterbi algorithm applied to sample student data under new parameters</td>
<td>108</td>
</tr>
<tr>
<td>4.8</td>
<td>Viterbi algorithm applied to memorizing student under new parameters</td>
<td>109</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Experimental data and projected model for the number of attempts needed for complete memorization.</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Probability mass function for Example 3.14.</td>
<td>34</td>
</tr>
<tr>
<td>3.2</td>
<td>Distribution function for Example 3.14.</td>
<td>35</td>
</tr>
<tr>
<td>3.3</td>
<td>Probability mass function for Example 3.14.</td>
<td>40</td>
</tr>
<tr>
<td>3.4</td>
<td>Distribution function for Example 3.14.</td>
<td>41</td>
</tr>
<tr>
<td>3.5</td>
<td>Random walker in small town.</td>
<td>42</td>
</tr>
<tr>
<td>3.6</td>
<td>Transition state diagram for a Markov model with four states.</td>
<td>44</td>
</tr>
<tr>
<td>3.7</td>
<td>Architecture of a hidden Markov model.</td>
<td>49</td>
</tr>
<tr>
<td>3.8</td>
<td>Architecture of a hidden Markov model with associated probabilities.</td>
<td>51</td>
</tr>
<tr>
<td>4.1</td>
<td>Trellis of an observation sequence in a hidden Markov model.</td>
<td>90</td>
</tr>
<tr>
<td>4.2</td>
<td>Determination of one state at $t = 2$ in the trellis diagram.</td>
<td>91</td>
</tr>
<tr>
<td>4.3</td>
<td>Trellis diagram for example Viterbi algorithm.</td>
<td>99</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

Distance learning and online education has become an integral part of higher education over the past few decades. In such an environment all of the coursework, such as the textbook, lectures, and assignments, are presented through online systems. There is much less human interaction than compared to a traditional classroom and the students are expected to work individually [7, 16, 19, 28, 30]. Some traditional classrooms have attempted to integrate online activities into the classroom, and have seen mixed results in student performance [7, 16, 24, 30].

An online course may include some of the means for instruction listed below [7, 28, 30].

Lecture videos An instructor films a video of a lecture over a particular topic. Students then watch the video on their own time, but cannot interact with the instructor directly during the video.

Course notes or textbook Course notes from the instructor or a course textbook for students to read to review a specific topic, examine example problems written by the instructor, or work through exercises.
**Discussion forum** Instead of working together in a physical classroom, students can discuss ideas or ask questions in a forum where others can respond. A discussion forum promotes student collaboration and interaction in the course.

**Exercise assessment platform** Students can work practice problems on a topic, moderated and evaluated by a computer.

We are interested in researching the last item on the list, an exercise assessment platform, as it pertains to mathematics education through a computer-based system. One way to measure student performance in an online course is to collect individual student data pertaining to the exercises. A computer can readily track student performance: whether a student correctly answers a question, incorrectly answers a question, or uses a hint to help solve the problem [15, 28]. The time it takes for a student to answer, whether the student’s correct attempt is the first, or the number of hints used can also be monitored [15, 28, 30]. In this paper we research the possibility of using student performance data on exercises to measure student achievement as well as evaluate specific problems based on student results. In this paper we investigate the following questions:

1. How can we assess a specific exercise in an exercise platform?

2. How can we measure student learning in this system?

3. Can we identify students who are not performing on a particular exercise as expected?
Chapter 2: Personalized education

In this chapter, we will discuss personalized instruction in mathematics education. The goal of this chapter is to explore prior and current research of personalized education in order to help develop modern online platform. Several types of personalized education will be discussed here, which includes students receiving one-to-one instruction from an expert tutor, students working by themselves on classroom based activities with limited assistance from a teacher, or students learning through a computer based platform. Each scenario is linked to individualized instruction in student education.

2.1 Introduction to individualized instruction

Individualized instruction in mathematics, as defined by the research, is an instruction model where

- students complete their class work at their own level and own pace, and

- each student has a personalized agenda corresponding to their specific learning needs [7, 19, 27].
Some research argues students’ individual learning objectives are not met in a classroom, but rather when learning by themselves [26, 27]. Through individualized instruction each student’s learning objectives would be satisfied individually and the system would be able to adapt their learning style [12, 25].

Currently there are a variety of mathematical education software systems used for personalized instruction, such as ALEKS [11], MyMathLab [21], Khan Academy [15], Individually Prescribed Instruction Mathematics [7], and courses through Coursera [1] or edX [2]. Such systems have their origins in trying to serve students who could not get their educational needs met at a university or classroom setting [23]. Now more educators are seeing the potential benefits of personalized education in a computer system [23].

By investigating this variety of individualized instruction, it will be much clearer what is currently available, what to incorporate, and what to avoid in the development of a software system that promotes individualized instruction.

2.1.1 One-to-one tutoring

One-to-one tutoring refers to a setting of a student and a tutor. Here the tutor teaches the course material and provides guidance and problem solving techniques for a specific areas of the student’s course. It has been argued that one-to-one tutoring in mathematics is the most effective form of student learning [4, 17, 18]. In [4] it was found that 98% of the students receiving one-to-one tutoring performed better at a controlled assessment compared to students receiving conventional classroom instruction, and that there was a statistically significant difference between the two groups’
average scores. Moreover [17] found that a positive attitude toward the subject matter and favorability of student self-concept were also displayed in the students who received one-to-one instruction.

This leads to the question: Why is individualized tutoring so effective? Research suggests that experienced tutors maintain a balance in their instruction. Tutors encourage students to work on their own, while monitoring the student’s progress to make sure the student does not become overwhelmed or frustrated [18]. Tutors are dynamic and adaptive in their teaching, able to pick up on the intricacies of a student’s struggles real-time [18]. Not just any tutoring program is beneficial. Students show the most improvement when receiving instruction from very structured and organized instruction [17]. It’s these factors that make one-to-one tutoring the preferred method to individualized instruction [4].

2.1.2 Individually Prescribed Instruction Mathematics

The Individually Prescribed Instruction Mathematics instructional program was an individualized instruction platform for teaching elementary and middle school mathematics. It was developed by the Learning Research and Development Center at the University of Pittsburgh and first put into application in 1966 [9]. Students were given written assignments which they completed individually. Their work was then assessed either by a teacher or by the students themselves [8]. Individually Prescribed Instruction Mathematics was designed to both

1. allow the student to complete the course at their own pace with limited help from a roaming instructor and

2. adaptively tailor its instruction to each individual student [7, 25].
Although this system is not done on a computer, students still complete the work individually at their own pace with limited assistance from an instructor. This is similar to the types of individualized instruction found on computer-based systems [16, 18].

To give an idea of the types of problems used in *Individually Prescribed Instruction Mathematics*, a few sample exercises related to decimals and fractions are shown below [7]. Each student is first presented with an example on how to work the problem, and then presented with an exercise for the student to complete.

**Question #1** Fill in the blanks:

\[
3.111 = 3 + \frac{1}{10} + \frac{1}{100} + \frac{1}{1000} \quad \text{(Example)}.
\]

\[
95.015 = 95 + \frac{0}{10} + \frac{1}{100} + \frac{5}{1000} \quad \text{(Question for student)}.
\]

**Question #2** Write the correct decimal numeral for each mixed fraction.

\[
9 \frac{35}{1000} = 9.035 \quad \text{(Example)}.
\]

\[
27 \frac{15}{1000} = ____ \quad \text{(Question for student)}.
\]

**Question #3** Circle the fraction which has the same value as the digit underlined in the decimal number.

\[
0.542 : \quad \frac{4}{10} \quad \frac{4}{100} \quad \frac{4}{1000} \quad \text{(Example)}.
\]

\[
3.20 : \quad \frac{2}{100} \quad \frac{2}{2} \quad \frac{2}{10} \quad \text{(Question for student)}.
\]

In Question #1 there are three blanks for the student to fill in. The student needs to look at the denominators 10, 100, and 1000 to make a connection that these
would be the missing denominators. However, there need not be a connection made
to place value of the decimal equivalent number to the left of the equals sign. A more
challenging question would be the following.

\[ 95.015 = 95 + \frac{1}{1} + \frac{5}{1} \]

The student should realize here that this is different from the model example because
there are fewer fractions added to 95. This means that the student cannot just match
the denominators to the model example. Instead, the student must understand the
relationship between place value in a decimal number on the left of the equals sign
to the fractional equivalent on the right.

By including nearly similar worked examples for the student to follow, the system
could encourage a behavior of pattern matching, with the student constantly trying
to match their question to a model example. Students would not form the correct
mathematical concept if too simple of a heuristic has been developed.

2.1.3 Khan Academy

The Khan Academy is a computer-based system designed in 2009 with similar
in ideas of individual instruction to Individually Prescribed Instruction Mathematics
[15, 16]. Students can access the system for free online, in which they are provided
with video tutorials on a variety of subjects and practice problems to assess their
own performance. The Khan Academy “aims to encourage and enable self-paced,
mastery-based, and interactive learning” by incorporating both lecture and practice
[14]. Students are able to “use step-by-step hints, watch related videos, and find
out immediately whether they answered a question correctly” [14]. Both students
and teachers have access to performance data on every exercise worked on the system
[14]. With access to such data both students and teachers can take action to facilitate student understanding and comprehension [14, 16]. To get an idea of the types of exercises that are in the *Khan Academy* we’ve provided a few examples within the *Khan Academy*’s calculus course [15].

**Question #1**

**Example with solution** Evaluate \( \lim_{x \to 3} \frac{x^2 - 4x + 3}{x^3 - 3x^2} \) using algebraic methods.

When we try substitution, we obtain the indeterminate form \( \frac{0}{0} \). The term “indeterminate form” should be interpreted as “I have more work to do.”

Factoring the numerator and denominator will allow us to simplify the expression so that substitution will yield a value for the limit.

\[
\lim_{x \to 3} \frac{x^2 - 4x + 3}{x^3 - 3x^2} = \lim_{x \to 3} \frac{(x - 1)(x - 3)}{x^2(x - 3)} = \lim_{x \to 3} \frac{x - 1}{x^2}
\]

Using substitution we now evaluate the limit.

\[
\lim_{x \to 3} \frac{x - 1}{x^2} = \frac{3 - 1}{3^2} = \frac{2}{9}
\]

**Exercise for student** Evaluate \( \lim_{x \to -2} \frac{x^2 + 4x + 4}{x^2 + 3x + 2} \) using algebraic methods.

**Question #2**

**Example with solution** Given

\[
f(x) = \begin{cases} 
2x + 4 & \text{if } x < 1; \\
6 & \text{if } x \geq 1.
\end{cases}
\]

What is the value of \( f'(3) \)?.
The function \( f \) is a piece-wise defined function. We are interested in finding its derivative value at \( x = 3 \). For \( x \geq 1 \), the graph of \( f(x) \) is a ray passing through \((3, 6)\) with slope 0; therefore the value of \( f'(3) = 0 \).

Exercise for student Given
\[
f(x) = \begin{cases} 
-2x + 6 & \text{if } x < 2; \\
\frac{1}{2}x^2 & \text{if } x \geq 2.
\end{cases}
\]

What is the value of \( f'(2) \)?

In Question #1 above, the system provides an example that outlines exactly how the instructors would expect in solution to the exercise. This solution is provided alongside the exercise for the student to attempt. The student can follow the steps to the provided solution exactly to solve their question. However, not all limit questions are the same. For example, a student presented with the limit
\[
\lim_{x \to 0} \frac{x^2 - 2x + 1}{x^2 - 1}
\]
would not need to factor the rational function because the function is continuous at \( x = 0 \). The student misses the concept of a determinate form in regards to limit.

Question #2 provides an example of the derivative of a piece-wise function at the value \( x = 3 \) that is not the transition point \( x = 1 \). In this case only the function defined for the interval where that point exists must be considered. However, the student must work an exercise where the derivative is to be determined at the transition point \( x = 2 \). Again, the example may give the framework to approaching the exercise, but does not give all of the information needed to help the student solve the problem. The student must understand the fundamental concepts of the derivative in order to solve this exercise.
2.2 Challenges of individualized instruction platforms

Here an analysis of individualized instruction will be presented for each of the three types introduced in section 2.1. The deficits of each platform will also be discussed, primarily focusing on ways a student interacts with an exercise. We are concerned with students who arrive at the correct answer for a set of exercises with insufficient or incorrect understanding of the example.

2.2.1 Challenges of one-to-one tutoring

One-to-one tutoring has immense benefits, as described above, but it comes at a cost for most societies to bear on a large scale [4]. Teachers are not physically able to assist a class of roughly 30 students on an individual basis. There is insufficient time and resources to cover each student’s educational needs.

One-to-one tutoring follows the protocol of individualized instruction, but primarily only for individuals or small groups. One tutor cannot tend to the individual needs of more than 5–10 students in a group—something will eventually have to be excluded in the learning process. On the flip side, a computer-based system that incorporates individualized instruction can be personalized to each student within a large group. However, a concern that always arises is whether the system can be as effective in teaching as compared to a one-to-one tutoring environment.

2.2.2 Challenges of Khan Academy

Since its inception in 2009, the Khan Academy has been receiving much praise for its approach to individualized instruction [14, 16, 30]. However, some critics argue that the Khan Academy encourages uncreative, repetitive drilling [30]. In [20] it
was found that in mathematics classrooms where repetition was the primary focus, the students were the least equipped for strategy maintenance and generalization in problem solving. It is this type of behavior that we do not want to encounter in such an individualized instructional platform.

The two calculus problems presented in section 2.1.3 revealed that Khan Academy provides a similar worked example outlining the steps to each problem. The exercises left for the student may not always be in the exact same format to the model example, but all the steps are essentially identical to the student’s example. While one may be concerned that such a setup will teach students that they must be “told” a solution, and hence inhibit creative problem solving, we have found evidence for a much more serious concern.

A report surfaced of a fifth-grade teacher who began using the Khan Academy as a “helpful supplement to [the] normal instruction” in mathematics [30]. A student, referred to as Matthew, began using it not only in class to learn basic fractions, decimals, and percentages, but to also attempt some inverse trigonometry problems outside of class [30]. Matthew solved 642 inverse trigonometry problems in all. Matthew answered “10 in a row [correctly] in just a few minutes” when observed by the article’s author [30]. The report only provided one example of the problems attempted: Evaluate $\cos^{-1}(1)$. The student correctly answered $0^\circ$.

To put this into context, we present a sampling of a few inverse trigonometry problems from the Khan Academy’s exercise bank.

1. In radians, what is the principal value of $\arctan\left(\frac{\sqrt{3}}{3}\right)$?

2. In degrees, what is the principal value of $\arccos\left(-\frac{\sqrt{2}}{2}\right)$?
Let’s consider how the first problem above would be solved by trigonometry student using concepts of the unit circle or trigonometric ratios.

Example 2.1 (Inverse Trigonometry).

In radians, what is the principal value of \( \arctan \left( \frac{\sqrt{3}}{3} \right) \)?

**Solution**

Denote the solution to the equation above as angle \( \theta = \arctan \left( \frac{\sqrt{3}}{3} \right) \). The following two equations are equivalent.

\[
\theta = \arctan \left( \frac{\sqrt{3}}{3} \right) \\
\tan(\theta) = \frac{\sqrt{3}}{3}
\]

The question now becomes: What is the principal angle \( \theta \) when applied to the tangent function yields \( \frac{\sqrt{3}}{3} \)? Through knowledge of the unit circle or trigonometric ratios a student should know that the angles, \( \theta \) that satisfy this ratio with \( 0 \leq \theta < 2\pi \) are

\[ \theta = \left\{ \frac{\pi}{6} \text{ or } \frac{7\pi}{6} \right\} \]

However, the problem specifically asks for the principal angle, which for \( \theta = \arctan(x) \) must satisfy \( -\frac{\pi}{2} < \theta < \frac{\pi}{2} \). Only \( \theta = \frac{\pi}{6} \) satisfies this constraint.

The second problem would be solved similarly. There are a few necessary concepts that a student must understand before answering this question with complete conceptual understanding.

1. The student must understand the fundamental ideas of a function, which includes the definition of a function, domain and range of a function, and to be able to use an inverse trigonometric function.
2. The student must understand what an angle is in degrees.

3. The student must understand angular measures in radians and how to convert between degrees and radians. The student must also know to use the symbol π when expressing a radian angle measure.

4. The student must know that solving an inverse trigonometry problem is equivalent to solving for the angle that satisfies a particular trigonometric ratio.

5. The student must be able to use the unit circle or know how to solve for angles through trigonometric ratios.

6. The student must understand the definition of a reference angle and the range restrictions imposed for each inverse inverse trigonometric function.

After examining this list, it’s amazing that an average 10-year-old elementary school student would have the necessary conceptual understanding to solve these problems. However, Matthew was able to work 642 different inverse trigonometry problems from this section on the Khan Academy’s system.

When attempting enough problems from this section, it can be noted that the only differences between individual problems in this section are the following:

1. The use of the word *radians* versus *degrees*.

2. The inverse trigonometric function.

3. The argument inside the inverse trigonometric function.

The answer is always given to the student after any attempt, regardless of whether the student answers correctly, incorrectly, or uses a hint [15]. Once attempting enough
problems, the same problem occurs more than once, but never on subsequent attempts. The following question must be addressed. How many possible distinct problem types exist in this exercise set?

To answer this question we must have knowledge of how the problem is programmed in the system. The possible problem variations are all available within the source code of the particular webpage. After viewing the source code, we came up with the following list of variations between each problem.

1. Whether the problem is asking for degrees or radians.

2. Whether the system uses the prefix *arc* in front of the trigonometric function or whether it uses a superscript, $\Box^{-1}$.

3. The type of trigonometric function, sine, cosine, or tangent.

4. The argument inside the inverse trigonometric function.

   (a) For inverse sine and inverse cosine the argument is limited to the following values $\left\{ 0, \pm \frac{1}{2}, \pm \frac{\sqrt{2}}{2}, \pm \frac{\sqrt{3}}{2} \right\}$.

   (b) For inverse tangent the argument is limited to $\left\{ 0, \pm \frac{\sqrt{3}}{3}, \pm 1, \pm \sqrt{3} \right\}$.

For item 1, there are two possible combinations for whether the problem asks for degrees or radians. In item 2 above an adept student will recognize that $\arcsin(x) = \sin^{-1}(x)$, $\arccos(x) = \cos^{-1}(x)$, and $\arctan(x) = \tan^{-1}(x)$ by exploring the hint provided for any exercise. For example, the first line of the hint would display the following:

$$\arctan \left( \frac{\sqrt{3}}{3} \right) = \tan^{-1} \left( \frac{\sqrt{3}}{3} \right)$$
Thus item 2 does not introduce any variation. For item 3, there are three different trigonometric functions, which means there would be three different combinations in the problem statement from this item. In item 4, there are seven possible values for the argument for all three trigonometric function. Therefore the total number of possible variations in the problem statement is \(2 \times 3 \times 7 = 42\). There are 42 unique problems within this exercise set, which implies the student who answered 642 problems necessarily saw every problem multiple times.

This leads to another question: How many times must a student see the answer to a mathematics problem in order to memorize its answer? In fact this question can be answered through investigation of paired-associate learning, a topic of interest in psychology research. Paired-associate learning involves learning items from a list in pairs: first an item and its pair from the list are shown, and then at a later time one of these items is presented to recall its pair [5]. Paired-associate learning involves two distinct processes for the learner:

1. The learning of the stimulus-response pairs from the list as they are initially presented, and

2. the associative pairing of the response to the stimulus when presented at a later time [5].

The learner will usually see a particular item, and then see other items from the list of stimuli before the original item is presented again [5, 6]. The entire list of stimuli may be presented multiple times before a connection can be made to the correct response.

One major interest of a paired-associate study is the effects of list length—how the number of pairs in a given set affects the ability for one to learn all the pairs [6].
<table>
<thead>
<tr>
<th>Number of switches</th>
<th>Number of total attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1,260</td>
</tr>
<tr>
<td>8</td>
<td>2,256</td>
</tr>
<tr>
<td>10</td>
<td>3,220</td>
</tr>
<tr>
<td>12</td>
<td>4,764</td>
</tr>
</tbody>
</table>

Table 2.1: Number of total attempts for each number of switches.

The literature shows that as the length of the list increases there is also an increase in the number of times the entire list must be shown to establish a correct connection [6]. This is exactly the type of research that would pertain to Matthew and the Khan Academy. How many times must a student see the answer to a mathematics problem in order to memorize its answer? Worded differently: How many pairs of items must a student see from a list to memorize each pair?

A study was conducted that attempted to answer the question on how paired-associate learning depends on list length [6]. The set-up of the experiment was as follows. A rectangular array of light bulbs was hooked up to a rectangular array of switches. Each switch would turn on a unique pair of adjacent light bulbs, either oriented horizontally or vertically on the array. The participant would turn on a switch (the stimulus) and observe which pair of bulbs illuminated (the response) to first learn the switch-bulb pairs. After this stage, the student would go through each switch attempting to predict which pairs of light bulbs would become illuminated once the switch was flipped. The researcher tracked the number of attempts a participant needed to make in order to learn exactly four complete sets of pairs completely correctly, but not necessarily consecutively. A summary of the results is shown in Table 2.1.
<table>
<thead>
<tr>
<th>Number of switches $N$</th>
<th>Number of total attempts</th>
<th>Total number of trials</th>
<th>Average number of trials per participant</th>
<th>Avg. incorrect trials per participant</th>
<th>Avg. observations seen from a participant on incorrect trials</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1,260</td>
<td>210</td>
<td>8.75</td>
<td>4.75</td>
<td>28.50</td>
</tr>
<tr>
<td>8</td>
<td>2,256</td>
<td>282</td>
<td>11.75</td>
<td>7.75</td>
<td>62.00</td>
</tr>
<tr>
<td>10</td>
<td>3,220</td>
<td>322</td>
<td>13.42</td>
<td>9.42</td>
<td>94.17</td>
</tr>
<tr>
<td>12</td>
<td>4,764</td>
<td>397</td>
<td>16.54</td>
<td>12.54</td>
<td>150.50</td>
</tr>
</tbody>
</table>

Table 2.2: Additional experimental data on observations needed for memorization.

Table 2.1 is extended to include more insight into the experiment. There were 96 total participants and four different numbers of switches (6, 8, 10, and 12). It is assumed that the 96 participants were uniformly distributed amongst the 4 grid types, hence $\frac{96}{4} = 24$ participants per type. Table 2.2 presents the following data.

1. Total number of trials = $\frac{\text{Number of total attempts}}{\text{Number of switches}}$.

2. Average number of trials per participant = $\frac{\text{Total number of trials}}{\text{Number of participants}}$. We assume there are 24 participants per type.

3. Average number of incorrect trials per participant

$$= \text{Average number of trials per participant} - 4.$$  

The reason 4 is subtracted is because the experiment is completed once the participant correctly identifies all switch-bulb pairs on four separate trials.

4. Average number of observations seen from a participant on all incorrect trials

$$= \text{Average number of incorrect trials per participant} \times \text{Length}$$
<table>
<thead>
<tr>
<th>Number of switches $N$</th>
<th>Avg. observations seen from a participant on incorrect trials</th>
<th>Projected number of attempts necessary for complete memorization $N \times (N - 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>28.50</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>62.00</td>
<td>56</td>
</tr>
<tr>
<td>10</td>
<td>94.17</td>
<td>90</td>
</tr>
<tr>
<td>12</td>
<td>150.50</td>
<td>132</td>
</tr>
</tbody>
</table>

Table 2.3: Comparison of experimental data to proposed model for attempts to memorization.

The average number of observations seen from a participant on all incorrect trials in Table 2.2 represents the number of times a participant must attempt to make a match until the participant has committed all paired items on the list to memory. This is exactly the same as the number of problems a student must attempt to form a question-answer pair correctly for all types problems within an exercise set. Based on the data in Table 2.2 we propose that the total number of attempts, $F(N)$, is modeled by $F(N) = N \times (N - 1)$, where $N$ is the number of distinct pairs. This data is summarized in Table 2.3 and displayed in Figure 2.1.

The model makes sense intuitively. There are $N$ pairs of objects that must be memorized. Assume that the learner memorizes one pair on the first trial. On every successive trial, one more pair is memorized. Therefore it would take $N$ total trials to memorize all $N$ pairs completely. However, it’s on the $N^{th}$ trial that all $N$ pairs are memorized. Therefore there are $N - 1$ incorrect trials leading to complete memorization on the $N^{th}$ trial.
Figure 2.1: Experimental data and projected model for the number of attempts needed for complete memorization.

The model $N \times (N - 1)$ fits the data with a coefficient of determination $R^2 = 0.95$. Let’s assume that this model holds for larger list lengths, $N > 12$. There were 42 variations of problems for inverse trigonometry problems on the *Khan Academy*. For $N = 42$ in the model we’d assume that $42 \times 41 = 1,722$ attempts would be needed to achieve memorization of the entire set of 42 pairs.

Upon further investigation of the problems in this set an adept student would be able to reduce this set of 42 pairs to a smaller set through three reductions while working through the problems and noticing some patterns.

1. arcsin($x$) and arctan($x$) are both odd functions. A student would be able to recognize that arcsin($-x$) = $-\text{arcsin}(x)$ and arctan($-x$) = $-\text{arctan}(x)$. This reduces the number of choices for the argument from 7 choices down to 4 for arcsin($x$) and arctan($x$).
2. \( \arccos(x) \) is neither even nor odd, but \( \arccos(x) + \arccos(-x) = 180^\circ \). By memorizing \( \arccos(x) \) in degrees, an astute student would be able to determine \( \arccos(-x) = 180 - \arccos(x) \). This reduces the number of choices for \( \arccos(x) \) by 3.

3. The argument of 0 will yield the same result in radians and degrees for both \( \arcsin(x) \) and \( \arctan(x) \). Thus there are two fewer choices that would need to be considered.

Revising the calculation of the number of possibilities yields the following.

1. For \( \arcsin(x) \) and \( \arctan(x) \) there are \( 4 \times 2 - 1 = 7 \) possibilities for each function.

2. For \( \arccos(x) \) there are \( 7 \times 2 - 3 = 11 \) possibilities.

3. This leads to 25 total possible pairs after reduction.

For \( N = 25 \) there would be assumed to be \( 25 \times 24 = 600 \) attempts needed to memorize all 25 pairs. However, in order to get the number of pairs down from 42 to 25 a student needs to recognize the three reductions listed above. Assume that a student would cycle through the list of 42 pairs once before arriving at such a reduction. There are \( 600 + 42 = 642 \) total attempts needed to memorize all answers from this exercise set. With this in mind it is somewhat unsurprising that a fifth-grade student would be able to correctly answer 10 questions in a row after attempting the problem 642 times.
2.2.3 Challenges of Individually Prescribed Instruction Mathematics

A study was conducted of a sixth grade class using Individually Prescribed Instruction Mathematics [7]. The researcher examined a lesson focused on learning decimals and fractions. One student, referred to as “Benny” in the study, performed considerably well through the system and the teacher regarded him as one of the best mathematics students [7]. However, when Benny was interviewed by a researcher Benny showed major misconceptions with decimals and fractions.

Several examples of the questions Benny worked were provided in section 2.1.2. In each exercise, a model solution is provided to the student. The exercise posed to the student is nearly identical to the model solution. The researcher observed that Benny was always “searching for a rule or pattern” when attempting to answer the exercises. Benny “did not ask questions about the mathematical” concepts [7]. When Benny was wrong, Benny would attempt to change the answer to match the answer key [7]. This system rigidly structures the student’s experience and does not allow them to “internalize or restructure the material in [their] own way” [7].

One topic of interest to the researcher was to discover how Benny handled the addition and multiplication of decimals [7]. The following questions were given to Benny, with the answer that was provided by Benny [7].

**Question #1** Evaluate 0.3 + 0.4.

Answer: 0.07. There are two significant digits in the answer “because there’s two points: [one] at the front of the 4 and [one] at the front of the 3. So you have two numbers after the decimal” in the final answer [7].
Question #2 Evaluate 0.44 + 0.44.

Answer: 0.0088. There “have [to be] four numbers after the decimal” in the final answer because each number in the question has two numbers after the decimal [7].

Question #3 Evaluate 0.7 × 0.5.

Answer: 0.35. There are two significant digits in the answer “because there’s two points, one...in front of each number” [7].

Question #4 Evaluate 0.2 × 0.3 × 0.4.

Answer: 0.024 [7].

The following rules can be established for addition and multiplication of decimals through Benny’s work. Let a, b, c, and d be digits in each example.

1. \( 0.a + 0.b = 0.0(a + b) \),

2. \( 0.ab + 0.cd = 0.00(ab + cd) \), and

3. \( 0.a \times 0.b = 0.(a \times b) \).

Benny’s heuristic for both adding or multiplying decimals is essentially the same, the only difference being that the operation on the digits be the same operation as stated in the question. Conceptually, however, these two operations are very different. In Questions 3 and 4 the answers are indeed correct, which reinforces Benny’s heuristic that has been developed for multiplication. The incorrect extension arises when this same heuristic is applied to addition in Questions 1 and 2, which yield incorrect results.
Through the *Individual Prescribed Instruction Mathematics* system, students were only required to answer 80% correctly for each assessment [7]. After a student completed their assignment, the teacher or a teacher’s aide would grade the answers for correctness and provide the correct answers to the student. If a student did not achieve the 80% mark, the student had to work the assignment again. When Benny failed to achieve the 80% mark for a certain assessment, Benny attempted to search for a pattern among the answers provided to the questions worked.

The way the teacher graded the assessments also contributed to students learning the answers to questions, rather than learning the mathematical concepts. The teacher was only concerned with the student’s answers and not the process of arriving at the answers [7]. In order to give a proper assessment of an individual’s performance, the student’s intermediate steps leading to the answer must be monitored.

An effective individual instruction platform would be designed to identify such students who are solving mathematics problems through pattern matching. An even more robust system would both identify and adaptively correct such behavior of problem solving. There are several propositions that will be discussed later on how to identify such students who do not fundamentally understand the concepts brought out by a mathematics problem in an individual instruction platform.

### 2.3 Design considerations of Individualized Instruction

As seen from the two individualized instruction platforms outlined above, it’s possible for students to correctly answer the mathematics exercises without actually understanding the concept. It’s very interesting to see such students perform so well, when their understanding of the concept may be lacking. A thought that could be
sparked is whether students should be learning mathematics in this fashion. Most instructors would not want such students passing their class.

In order to assess student performance, it’s not just enough for the computer to count the number of correct answers within a set of attempts, as seen from the two scenarios above. A student may be answering the problems correctly without any conceptual understanding. A better analysis of student performance is necessary in order to measure student understanding.

In this paper a hidden Markov model is proposed that models student performance on exercises within the system. The hidden Markov model is composed of both observed states (correct response and incorrect response), as well as hidden states (understanding a concept and not understanding a concept). By using this model the information on student understanding (hidden states) of a particular concept can be captured from the sequential observed data for a specific exercise. In conjunction with the Viterbi algorithm, the sequence of student understanding (hidden states) can be determined that links to the sequence of observations. By determining such a sequence for each student on each exercise, inferences can be made on how the student performs over time on that exercise. This pattern of inferred hidden states is highly valuable in determining how students are learning on a particular exercise over time.

In reference to the Individually Prescribed Instruction Mathematics and Khan Academy systems, students can “learn” how to get the correct answer to a problem through repetition of the same exercise. With a hidden Markov model in place and in conjunction with the Viterbi algorithm, such student behavior can be identified within the system. This is something that could not be easily identified by a teacher.
or grader looking only at the answers to the exercises, while a computer can recognize this student behavior based upon the sequence of student responses with a *hidden Markov model* in place.
Chapter 3: Hidden Markov models and education

3.1 Introduction to hidden Markov models

This paper is concerned with individualized instruction of mathematics in a computer based system. In a computer based system, such as Khan Academy, the system keeps track of student responses in the exercises [16]. For example, some possible student observations might be the following:

Correct on first attempt  Student answers a given exercise correctly on the first try.

Correct on subsequent attempt  Student first misses a question or uses a hint before answering correctly.

Hint used  Student is provided a hint to help solve the problem.

Incorrect  Student answers the question incorrectly.

The system stores a sequence of these observed states for each exercise that a student works. From this sequence of observed states we attempt to determine whether the student is in a knowing state, an unknowing state, or as we will see later an emerging state. The states of knowing, unknowing, and emerging are not directly observable, and are thus called hidden states. The sequence of hidden states is called a Markov
chain, and the combination of the observed and hidden sequences is called a hidden Markov model [22].

3.1.1 Basics of probability

Before discussing hidden Markov models and Markov chains, one must have a basic understanding of probability. The following section will introduce the relevant terminology used within this paper (see for example [3, 10, 13]).

Definition 3.1. The set $S$ of all possible outcomes of a particular experiment is called the **sample space** for the experiment.

Definition 3.2. An **event** is a subset of the sample space.

Definition 3.3. The **complement** of an event $A$, denoted as $A^c$, is the set of all elements in the sample space $S$ that are not in $A$. That is,

$$A^c = \{ x : x \in S \text{ and } x \notin A \}.$$

Definition 3.4. The collection of all subsets of a set $S$, including the empty set and $S$ itself, is called the **power set** of $S$, denoted by $\mathcal{P}(S)$.

Definition 3.5. Given a sample space $S$ a **probability function** is a function $P : \mathcal{P}(S) \to [0,1]$ with the following properties:

1. $P(A) \geq 0$ for all $A \in \mathcal{P}(S)$.

2. $P(S) = 1$.

3. If $A_1, A_2, \ldots, A_k \in \mathcal{P}(S)$ and $A_i \cap A_j = \emptyset$ for all $i \neq j$, then

$$P(A_1 \cup A_2 \cup \cdots \cup A_k) = P(A_1) + P(A_2) + \cdots + P(A_k)$$

for each positive integer $k$. 

27
Theorem 3.6. If $S$ is a sample space, $P$ is a probability function, and $A \in \mathcal{P}(S)$, then

1. $P(A^c) = 1 - P(A)$,
2. $P(A) \leq 1$,
3. $P(\emptyset) = 0$.

Proof. By Definition 3.3 we have the following:

$$S = A \cup A^c \text{ and } A \cap A^c = \emptyset$$

From Definition 3.5 it follows that

$$P(S) = P(A) + P(A^c)$$

$$1 = P(A) + P(A^c)$$

$$\Rightarrow P(A^c) = 1 - P(A)$$

This proves item 1. Next, we prove item 2 by using the result above. Rewriting item 1 gives the following:

$$P(A^c) = 1 - P(A)$$

$$P(A) = 1 - P(A^c)$$

By Definition 3.5

$$P(A^c) \geq 0$$

Therefore

$$P(A) \leq 1$$

28
Finally, we prove item 3. Let $A = \emptyset$.

\[
S = A \cup A^c
\]

\[
P(S) = P(A \cup A^c)
\]

\[
P(S) = P(A) + P(A^c)
\]

\[
P(S) = P(\emptyset) + P(\emptyset^c)
\]

\[
P(S) = P(\emptyset) + P(S)
\]

\[
1 = P(\emptyset) + 1
\]

\[
\Rightarrow P(\emptyset) = 0
\]

Definition 3.7 (Conditional Probability). If $A$ and $B$ are events in $S$, and $P(B) > 0$, then the conditional probability of event $A$ occurring assuming that event $B$ has already occurred is

\[
P(A|B) := \frac{P(A \cap B)}{P(B)}.
\]

Definition 3.8 (Independence of Events). Two events, $A$ and $B$, are independent if

\[
P(A \cap B) = P(A) \cdot P(B).
\]

Using Definitions 3.7 and 3.8 together, we have that if events $A$ and $B$ are independent, then

\[
P(A|B) = \frac{P(A \cap B)}{P(B)}
\]

\[
= \frac{P(A) \cdot P(B)}{P(B)}
\]

\[
= P(A).
\]
Definition 3.9 (Random Variable). A random variable $X$ is a function from a sample space $S$ into the real numbers. Formally:

$$X : S \to \mathbb{R}$$

Definition 3.10. An event $A \subset S$ is said to be defined in terms of the random variable $X$ if we can determine whether $A$ has occurred from the value of $X$.

Definition 3.11. Given a sample space $S$ and random variable $X : S \to \mathbb{R}$, a probability mass function of a random variable $X$ is a function $f_X : \mathbb{R} \to [0, 1]$ defined as

$$f_X(x) = P(X = x)$$

for all $x \in \mathbb{R}$.

The probability mass function represents the probability that the random variable $X$ is equal to a particular value $x \in \mathbb{R}$. We will focus only on a finite number of possible values for $x$. A probability mass function has the following two key properties.

1. $f_X(x) \geq 0$ for all $x \in \mathbb{R}$, and

2. $\sum_x f_X(x) = 1$.

Definition 3.12. Given a sample space $S$ and random variable $X : S \to \mathbb{R}$, a distribution function is a function $F_X : \mathbb{R} \to [0, 1]$ of $X$ defined as

$$F_X(x) = P(X \leq x)$$

for all $x \in \mathbb{R}$.

Within this paper we use the word distribution to refer to the distribution function.
Definition 3.13. A random variable is **discrete** if the distribution function $F_X(x)$ is a step function of $x$.

Example 3.14 (Rolling two dice).

Two unbiased dice are rolled and the sum of the two values facing up is examined. We want to know the probability of rolling a sum of 10.

**Sample space** is, in this example, the set of all possible pairs of dice rolls. Let $(i, j)$ denote the set of two dice rolls, where $i$ is the value of the first die and $j$ the value of the second die.

$$S = \{(i, j) : 1 \leq i \leq 6, 1 \leq j \leq 6\}$$

There are 36 such pairs of dice rolls in the sample space.

**Event** The event here is all the possible dice rolls that have a sum of 10:

$$A = \{(4, 6), (5, 5), (6, 4)\}.$$  

**Power set** The collection of all possible combinations of pairs of dice rolls,

$$P(S) = \{\emptyset, \{(1, 1)\}, \{(1, 2)\}, \ldots, \{(6, 6)\}, \{(1, 1), (1, 2)\}, \ldots, \{(1, 1), (1, 2), \ldots, (6, 6)\}\}.$$  

Note that the event is an element of $P(S)$.

**Probability** We’re interested in the probability of event $A$ occurring, i.e. $P(A)$,

where $P : P(S) \rightarrow [0, 1]$ is defined as

$$P(\{(i, j)\}) = \frac{1}{36}.$$
for each $i, j$ and extended to $P(S)$ by additivity of $P$. Therefore

$$P(A) = P(\{(4,6), (5,5), (6,4)\})$$

$$= P(\{(4,6)\}) + P(\{(5,5)\}) + P(\{(6,4)\})$$

$$= \frac{1}{36} + \frac{1}{36} + \frac{1}{36}$$

$$= \frac{1}{12}$$

**Independence** Every dice roll is independent of each other.

$$P(\{(i,j)\}) = P(i) \cdot P(j)$$

$$= \frac{1}{6} \cdot \frac{1}{6}$$

$$= \frac{1}{36}$$

**Random variable** The random variable is a function $X : S \to \mathbb{R}$. Since the sample space is all possible pairs of dice rolls and the random variable is defined as the sum of the two dice rolls, then the range of the random variable is $\{2, \ldots, 12\}$

$$X : S \to \{2, \ldots, 12\}.$$  

**Event represented as a random variable** In the context of this example, we’re concerned with the sum of the two dice equal to 10, $X = 10$. The value 10 of the random variable, $X = 10$, represents the event $A = \{(4,6), (5,5), (6,4)\}$.

**Probability mass function** The probability mass function is defined as $f_X(x) = P(X = x)$. To get an idea of the behavior of this function let’s consider the possible values in the range of $X$.

- If $x = 2$ there is only one possible pair: $(1,1)$. Thus $f_X(2) = \frac{1}{36}$. The same is true for $x = 12$—the only possible pair is $(6,6)$. Thus $f_X(12) = \frac{1}{36}$.  

32
• If $x = 3$ there are two possible pairs: $(1, 2)$ and $(2, 1)$. The same is true for $x = 11$—the possible pairs are $(5, 6)$ and $(6, 5)$. Thus $f_X(3) = \frac{2}{36}$ and $f_X(11) = \frac{2}{36}$.

• The same logic continues until all values in the range of $X$ have been examined. The probability mass function is defined as follows.

$$f_X(x) = \begin{cases} \frac{6 - |x - 7|}{36} & \text{for } x = 2, \ldots, 12 \\ 0 & \text{otherwise} \end{cases}$$

For $x = 10$,

$$f_X(10) = \frac{6 - |10 - 7|}{36} = \frac{3}{36} = \frac{1}{12}$$

This of course matches the probability $P(A)$ calculated above. Figure 3.1 shows a graph of $f_X(x)$.

**Distribution** The distribution is defined as $F_X(x) = P(X \leq x)$. The probability mass function $f_X(x)$ can be used to determine the distribution function $F_X(x)$.

$$F_X(x) = P(X \leq x) = \sum_{y \leq x} P(X = y) = \sum_{y \leq x} f_X(y)$$
If $X = 2$, then

$$F_X(2) = P(X \leq 2)$$
$$= P(X = 2)$$
$$= f_X(2)$$
$$= \frac{1}{36}$$

If $X = 3$, then

$$F_X(3) = P(X \leq 3)$$
$$= P(X = 2) + P(X = 3)$$
$$= f_X(2) + f_X(3)$$
$$= \frac{3}{36}$$

These calculations continue in a similar fashion by accumulating all instances of the probability mass function less than a particular value $x$. 
The distribution function is defined for all $x \in \mathbb{R}$. If we wanted to determine $F_X(2.5)$, then

$$F_X(2.5) = P(X \leq 2)$$

$$= P(X = 2)$$

$$= f_X(2)$$

$$= \frac{1}{36}$$

Thus $F_X(x) = \frac{1}{36}$ over the interval $2 \leq x < 3$. The distribution function looks like a step function in $x$, as shown in figure 3.2. This means that the random variable $X$ is a discrete random variable by Definition 3.13.

Example 3.15 (Student answering questions).

Assume a student is making attempts on one exercise on an online calculus platform and has a probability $p = 0.70$ of answering correctly on any attempt. Assume
every attempt is independent of each other. We want to know for a sequence of three attempts, what is the probability the student answers at least two correctly?

**Sample space** is the set of all possible sequences that could occur over a set of three attempts. $C$ denotes correct and $I$ denotes incorrect,

$$S = \{(C,C,C), (C,C,I), (C,I,C), (I,C,C), (C,I,I), (I,C,I), (I,I,C), (I,I,I)\}.$$ 

**Event** The event would be all the possible sequences that have at least two correct answers out of the set of three,

$$A = \{(C,C,C), (C,C,I), (C,I,C), (I,C,C)\}.$$ 

**Power set** The collection of all possible combinations of sequences of three attempts,


**Probability** We’re interested in the probability of event $A$ occurring, i.e. $P(A)$, where $P: \mathcal{P}(S) \rightarrow [0,1],$

$$P(A) = P(\{(C,C,C), (C,C,I), (C,I,C), (I,C,C)\})$$


The sequence $(C,C,C)$ is that of three correct attempts in a row. The probability of one correct attempt is 0.70. Using the definition of independence, the probability of three correct attempts is $(0.70)^3$,

$$P(\{(C,C,C)\}) = P(C) \cdot P(C) \cdot P(C)$$

$$= (0.7)^3.$$
The sequences $(C, C, I)$, $(C, I, C)$, and $(I, C, C)$ each have two correct attempts and one incorrect attempt. The probability of one correct attempt is $0.70$, while the probability of one incorrect attempt is $1 - 0.70 = 0.30$ since $C^c = I$. Therefore the probability of each sequence is

$$P(\{C, C, I\}) = P(C) \cdot P(C) \cdot P(I)$$

$$= (0.70)^2(0.30).$$

The total probability is the sum of the three individual probabilities.

$$P(A) = P(\{(C, C, I)\}) + P(\{(C, I, C)\}) + P(\{(I, C, C)\})$$

$$= (0.70)^2(0.30) + (0.70)^2(0.30) + (0.70)^2(0.30)$$

$$= 3(0.70)^2(0.30).$$

Thus $P(A) = (0.70)^3 + 3(0.70)^2(0.30)$.

**Random variable** The random variable is the function $X : S \rightarrow \{0, 1, 2, 3\}$ defined as

$$X(\{(s_1, s_2, s_3)\}) = \text{Number of correct attempts}$$

where $s_1, s_2, s_3 \in \{C, I\}$. For example,

$$X(\{(C, C, I)\}) = 2.$$  

**Event represented as a random variable** In the context of this example, we are concerned with having at least 2 correct responses out of 3 attempts. This corresponds to $x \geq 2$. Since the range of $X$ has a maximum value of 3, then the set $\{x = 2, x = 3\}$ uniquely determines the event $A$. 

37
**Probability mass function** The probability mass function is defined as \( f_X(x) = P(X = x) \). To get an idea of the behavior of this function let us consider the possible values in the range of \( X \).

- **If \( X = 0 \)** there are no correct attempts. This would be the event

  \[ A = \{(I, I, I)\} \]

  and

  \[ f_X(0) = (0.3)^3. \]

- **If \( X = 1 \)** there is one correct attempt. This would be the event

  \[ A = \{(C, I, I), (I, C, I), (C, I, I)\} \]

  and

  \[ f_X(1) = (0.7)(0.3)^2 + (0.7)(0.3)^2 + (0.7)(0.3)^2 
  = 3(0.7)(0.3)^2. \]

- **If \( X = 2 \)** there are two correct attempts. This would be the event

  \[ A = \{(C, C, I), (C, I, C), (I, I, C)\} \]

  and

  \[ f_X(2) = (0.7)^2(0.3) + (0.7)^2(0.3) + (0.7)^2(0.3) 
  = 3(0.7)^2(0.3). \]

- **If \( X = 3 \)** there are three correct attempts. This would be the event

  \[ A = \{(C, C, C)\} \]
and

\[ f_X(3) = (0.7)^3. \]

The probability mass function is defined as

\[
 f_X(x) = \begin{cases} 
 (0.3)^3 & x = 0, \\
 3(0.7)(0.3)^2 & x = 1, \\
 3(0.7)^2(0.3) & x = 2, \\
 (0.7)^3 & x = 3, \\
 0 & \text{otherwise.} 
\end{cases}
\]

Comparing this probability to the probability of the event for \( x \geq 2 \), this means that \( x = 2 \) or \( x = 3 \) based upon the range of the random variable.

\[
f_X(2) + f_X(3) = 3(0.7)^2(0.3) + (0.7)^3
\]

This matches the probability \( P(A) \) calculated above. Figure 3.3 shows a graph of \( f_X(x) \).

**Distribution** The distribution is defined as \( F_X(x) = P(X \leq x) \). The probability mass function \( f_X(x) \) can be used to determine the distribution function \( F_X(x) \).

\[
 F_X(x) = P(X \leq x) \\
 = \sum_{y \leq x} P(X = y) \\
 = \sum_{y \leq x} f_X(y).
\]
• If $x = 0$, then

$$F_X(0) = P(X \leq 0)$$

$$= P(X = 0)$$

$$= f_X(0)$$

$$= (0.70)^3.$$  

• If $x = 1$, then

$$F_X(1) = P(X \leq 1)$$

$$= P(X = 0) + P(X = 1)$$

$$= f_X(0) + f_X(1)$$

$$= (0.70)^3 + 3(0.70)^2(0.30).$$

• These calculations continue in a similar fashion by accumulating all instances of the probability mass function less than a particular value $x$. 

Figure 3.3: Probability mass function for Example 3.14.
Figure 3.4: Distribution function for Example 3.14.

The distribution function looks like a step function in $x$. This means that the random variable $X$ is a discrete random variable by Definition 3.13. The distribution function is shown in figure 3.4.

3.1.2 Markov chains

Now we define a random process and the Markov property. These two ideas form the foundation of the theory behind hidden Markov models. In these definitions we use the variable $X_t$ to denote a random variable. $X_t$ is a discrete random variable at a specific time instance $t$. It should be understood that the probability mass function associated with $X_t$ may change with time. The focus of this paper is on investigating an individualized instruction platform with a finite number of student observations $n$. We’ll be working with discrete time $t$, where $t = 1, 2, \ldots, n$. 
Definition 3.16 (Random Process). A random process, sometimes called a stochastic process, is a sequence of random variables \( \{X_1, X_2, \ldots \} \), where \( X_t \) represents the state of a random variable at time \( t \). [10]

Example 3.17 (Random Walk). Suppose a walker is in a small town with exactly four streets as shown in Figure 3.5. Define the sample space \( S = \{s_1, s_2, s_3, s_4\} \) as the set of possible street corners. The person can only be located at one street corner at each time \( t \). At time \( t = 1 \), the person stands at street corner \( s_1 \). At each time \( t = 2, 3, \ldots, n \), the person flips a fair coin, which decides the adjacent street corner to move to next. If the coin comes up heads, the person will move to the adjacent street corner in the clockwise direction, and if the coin comes up tails, the person will move to the adjacent street corner in the counter-clockwise direction. They must move horizontally or vertically and they are not allowed to stand at the same street corner after the flip. This sequence is continued up to time \( n \).

For each time \( t \) define the random variable \( X_t : S \rightarrow \{1, 2, 3, 4\} \) as

\[
X_t(s_i) = x_i.
\]
Now let’s examine what happens to the random variable $X_t$. We are given that $X_1 = 1$ since the person begins at street corner $s_1$. For $t = 2$ the possibilities are that $X_2 = 2$ if heads comes up, or $X_2 = 4$ if tails comes up. Since each has an equal probability of $\frac{1}{2}$ of occurring we write the probability mass function as

$$P(X_2 = x_i|X_1 = 1) = \begin{cases} 0 & i = 1, \\ \frac{1}{2} & i = 2, \\ 0 & i = 3, \\ \frac{1}{2} & i = 4. \end{cases}$$

At time $t = 2$, the person can only be at $s_2$ or $s_4$. Let’s suppose that the person has moved to state $s_4$ at time $t = 2$: $X_2 = 4$. This means that at time $t = 3$, the person can only be at $s_1$ or $s_3$ based on the structure of the city. Mathematically we write the probability mass function as

$$P(X_3 = x_i|X_2 = 4 \cap X_1 = 1) = \begin{cases} \frac{1}{2} & i = 1, \\ 0 & i = 2, \\ \frac{1}{2} & i = 3, \\ 0 & i = 4. \end{cases}$$

However, the probability mass function of $X_3$ is only dependent on $X_2$. Therefore, the conditional information can be simplified to only contain information about $X_2$. Write

$$P(X_3 = x_i|X_2 = 4 \cap X_1 = 1) = P(X_3 = x_i|X_2 = 4).$$

This same property holds for $t = 4, \ldots, n$ as well. In general the probability mass function of the current state depends only on the previous state, and no other states before the previous state. This forms the basis for the Markov property.

A transition state diagram can help better visualize the Markov model transition probabilities. The transition state diagram shows the possible states and which states
each is able to transition to with a specified probability within the Markov model. Essentially this gives a visual representation of the probability mass functions for each state. The transition state diagram for this example is shown in Figure 3.6.

![Transition state diagram for a Markov model with four states](image)

Figure 3.6: Transition state diagram for a Markov model with four states

The arrows indicate which transitions between states are possible. The values next to the arrows indicate the probability of such a transition occurring. For example, the transition from state \(s_1\) to state \(s_2\) has a probability of \(\frac{1}{2}\). However, the diagram shows that a transition directly between states \(s_1\) and \(s_3\) is not possible since an arrow does not connect these two states.

It is clear from Figure 3.6 the probability of being in each state is dependent only upon the current state.

**Definition 3.18 (Markov property).** Given state space \(S = \{s_1, s_2, \ldots, s_k\}\) and a sequence of discrete random variables \(\{X_1, X_2, \ldots, X_n\}\), \(X_t : S \rightarrow \{x_1, x_2, \ldots, x_k\}\) is
defined as

\[ X_t(s_i) = x_i \]

for \( t = 1, \ldots, n \) and \( i = 1, \ldots, k \). This sequence of random variables exhibits the **Markov property** if

\[ P(X_{n+1} = x_{n+1} | X_n = x_n \cap X_{n-1} = x_{n-1} \cap \cdots \cap X_1 = x_1) = P(X_{n+1} = x_{n+1} | X_n = x_n) \]

for every \( x_j \in \{x_1, \ldots, x_k\} \) [10].

That is the probability of state \( X_{n+1} \) depends only on the previous state \( X_n \).

For the individualized instruction platforms we assume that the Markov property is satisfied. When a student is working exercises we make the assumption that the student’s performance at the current state is only dependent upon their performance in the previous state.

**Definition 3.19 (Markov chain).** Given state space \( S = \{s_1, s_2, \ldots, s_k\} \), a sequence of discrete random variables \( \{X_1, X_2, \ldots, X_n\} \). \( X_t : S \rightarrow \{x_1, x_2, \ldots, x_k\} \) is defined as

\[ X_t(s_i) = x_i \]

for \( t = 1, \ldots, n \) for \( t = 1, \ldots, n \) and \( i = 1, \ldots, k \). This sequence of random variables is a **Markov chain** if it is a random process with the Markov property [10].

The sequence \( \{X_1, X_2, \ldots, X_n\} \) in Example 3.17 is a Markov chain. Each random variable in the sequence \( X_t : S \rightarrow \{1, 2, 3, 4\} \) has its own probability mass function, and thus is a random process. The sequence also exhibits the Markov property. A sequence that has this combination comprises a **Markov chain**.
The term “chain” comes from the fact that the sequence of random variables transition between states in the state space at each time $t$. Referring to Example 3.17, if we are given the sequence $\{X_1 = 1, X_2 = 4, X_3 = 1\}$, then the model must have a transition from state $s_1$ to $s_4$ at time $t = 2$ and another transition from $s_4$ back to $s_1$ at time $t = 3$. This would represent the arrow from $s_1$ to $s_4$ between times $t = 1$ and $t = 2$ and the arrow from $s_4$ to $s_1$ between times $t = 2$ and $t = 3$ in Figure 3.6.

A Markov chain has several properties that we must consider in the analysis of hidden Markov models. A Markov chain is defined once we give the initial probabilities and transition probabilities as defined below.

**Definition 3.20 (Initial probabilities).** Given state space $S = \{s_1, \ldots, s_k\}$ and a sequence of discrete random variables $\{X_1, \ldots, X_n\}$, where $X_i : S \to \{x_1, \ldots, x_k\}$ is defined as

$$X_i(s_i) = x_i$$

for $t = 1, \ldots, n$ for $t = 1, \ldots, n$ and $i = 1, \ldots, k$. The initial probabilities of the Markov chain is a probability mass function representing the probability of how likely the Markov chain will begin in a particular state. We denote the initial probabilities as $\mu$ and represent it as a matrix with one row:

$$\mu = [\mu_1, \ldots, \mu_k]$$

where $\mu_i = P(X_1 = x_i)$, the probability that the initial state at is $x_i$ [10].

Since $\mu$ is a probability mass function, it follows that

1. $\mu_i \geq 0$ for $i = 1, \ldots, k$, and
2. \( \sum_{i=1}^{k} \mu_i = 1. \)

**Definition 3.21 (Transition probabilities).** Given state space \( S = \{s_1, \ldots, s_k\} \) and a sequence of discrete random variables \( \{X_1, X_2, \ldots, X_n\} \), where \( X_t : S \rightarrow \{x_1, \ldots, x_k\} \) is defined as

\[
X_t(s_i) = x_i
\]

for \( t = 1, \ldots, n \) and \( i = 1, \ldots, k \). The **transition probabilities** of the Markov chain is a collection of probabilities representing the probability of how likely any state in the Markov chain will transition to another state, including itself. We denote the transition probabilities as \( T \) and represent it as a \( k \times k \) matrix where \( k \) is the number of elements in the sample space.

\[
T = [q_{ij}]
\]

\[
= [P(X_t = x_j|X_{t-1} = x_i)]
\]

for \( i = 1, \ldots, k \) and \( j = 1, \ldots, k \) [10].

The initial probabilities and transition probabilities will be shown for Example 3.17.

**Example 3.22 (Initial and transition probabilities).** In Example 3.17 it was given that the person started in state \( x_1 \). Therefore the initial probabilities \( \mu \) are

\[
\begin{bmatrix}
s_1 & s_2 & s_3 & s_4 \\
1 & 0 & 0 & 0
\end{bmatrix}.
\]
Figure 3.6 shows the transition diagram, which can be used to determine the transition probabilities in matrix $T$,

$$
\begin{array}{cccc}
  & s_1 & s_2 & s_3 & s_4 \\
 s_1 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\
 s_2 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\
 s_3 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\
 s_4 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\
\end{array}
$$

The states listed on top represent the current state, while the states listed to the left represent the previous state.

### 3.1.3 Hidden Markov models

A hidden Markov model is what we wish to use as the student evaluation model in an individualized instruction environment. The goal is to determine whether a hidden Markov model can effectively represent individual exercises. With knowledge of the parameters of a hidden Markov model we can gain insight into the types of properties such a model would have.

**Definition 3.23 (Hidden Markov model).** A hidden Markov model is a set of two random processes, one that is a Markov chain and not directly observable, denoted as the hidden states, and one that is observable, denoted as the observed states [22].

A hidden Markov model has a general architecture as shown in figure 3.7. Given a sample space $S_1 = \{s_1, \ldots, s_k\}$ that represent all possible hidden states, the random variable $X_t : S_1 \to \{x_1, \ldots, x_k\}$ represents the hidden state at time $t$. The sequence $\{X_1, \ldots, X_n\}$ forms a Markov chain. Given a sample space $S_2 = \{r_1, \ldots, r_m\}$ that represents all the possible observations, the random variable $Y_t : S_2 \to \{y_1, \ldots, y_m\}$ represents the observation at time $t$. The sequence $\{Y_1, \ldots, Y_n\}$ is a random process,
but not necessarily a Markov chain. It is inferred that there is a relationship between the observations and the hidden states.

![Figure 3.7: Architecture of a hidden Markov model](image)

Two such parameters of a hidden Markov model are the initial probabilities $\mu$ and the transition probabilities $T$, which apply to Markov chains. We introduce another parameter, the emission probabilities, denoted $E$. The combination of $\mu$, $T$, and $E$ represent the parameters that define a hidden Markov model.

**Definition 3.24 (Emission probabilities).** Given state space $S_1 = \{s_1, \ldots, s_k\}$ and a Markov chain $\{X_1, X_2, \ldots, X_n\}$ of hidden states, where $X_t : S_1 \rightarrow \{x_1, \ldots, x_k\}$ is defined as

$$X_t(s_i) = x_i$$

for $t = 1, \ldots, n$ and $i = 1, \ldots, k$. Given a state space $S_2 = \{y_1, \ldots, y_m\}$ and a random process $\{Y_1, \ldots, Y_n\}$ of observations where $Y_t : S_2 \rightarrow \{r_1, \ldots, r_m\}$ is defined as

$$Y_t(r_j) = y_j$$

for $t = 1, \ldots, n$ and $j = 1, \ldots, m$. The set $\{X_1, \ldots, X_n\}$ and $\{Y_1, \ldots, Y_n\}$ forms a hidden Markov model. The emission probabilities of the hidden Markov model
is a collection of probabilities representing the probability of seeing an observation given a hidden state. We denote the emission probabilities as $E$ and represent it as a $k \times m$ matrix where $k$ is the number of possible hidden states and $m$ is the number of possible observations [22].

$$E = [e_{ij}]$$

$$= [P(Y_t = y_j|X_t = x_i)]$$

for $i = 1, \ldots, k$ and $j = 1, \ldots, m$.

Figure 3.8 shows the architecture of the hidden Markov model with the associated probabilities between states and observations. The initial probabilities $\mu$ represent the probabilities of which hidden state the hidden Markov model starts. The transition probabilities $T$ represent the probabilities of transitioning between the hidden states from time $t - 1$ to time $t$. The emission probabilities $E$ represent the probability of an observation occurring at time $t$ when the hidden Markov model is in a particular hidden state. The set of all three parameters will be denoted as $\lambda$.

$$\lambda = (\mu, T, E)$$

We present an example that summarizes all the parameters of a hidden Markov model.

**Example 3.25 (Hidden Markov model).** Suppose a student is answering problems within an individualized instruction system. The student is making multiple attempts on one particular exercise. The student does one of the following on each attempt.

**Correct on first attempt (A)** Student answers a given exercise correctly on the first try.
Correct on subsequent attempt (C) Student first misses a question or uses a hint before answering correctly.

Hint used (H) Student is provided a hint to help solve the problem.

Incorrect (I) Student answers the question incorrectly.

As time elapses the student’s performance state is also changing—this is whether the student understands the material, or does not understand. However, this information is not directly observable.

This forms the framework for a hidden Markov model. The sequence of hidden states \( \{X_1, \ldots, X_n\} \) are the performance states. Denote the hidden states as “knowing” (K) and “unknowing” (U). Define the sample space \( S_1 = \{K, U\} \) and the random variable \( X_t : \{K, U\} \rightarrow \{1, 2\} \) as

\[
X_t(x) = \begin{cases} 
1 & x = K, \\
2 & x = U.
\end{cases}
\]

The sequence of observations \( \{Y_1, \ldots, Y_n\} \) are the results of the student attempts. Define the sample space \( S_2 = \{A, C, H, I\} \) and the random variable \( Y_t : \{A, C, H, I\} \rightarrow \)
\{1, 2, 3, 4\} as
\[ Y_t(y) = \begin{cases} 
1 & y = A, \\
2 & y = C, \\
3 & y = H, \\
4 & y = I. 
\end{cases} \]

Suppose that the student initially begins in the “knowing” state with a probability of 0.2 and the “unknowing” state with a probability of 0.8. This would correspond to the initial probabilities \( \mu \). Define \( \mu \) as
\[
\begin{pmatrix}
K & U \\
0.2 & 0.8
\end{pmatrix}.
\]

Suppose that between attempts the student stays in the “knowing” state with a probability of 0.9, while the student moves from the “knowing” state to the “unknowing state” with a probability of 0.1. The student also stays in the “unknowing” state with a probability of 0.6 and moves to the “knowing” state from the “unknowing” state with a probability of 0.4. This information corresponds to the transition probabilities \( T \). Define \( T \) as
\[
\begin{pmatrix}
K & U \\
[0.9 & 0.1] & [0.4 & 0.6]
\end{pmatrix}.
\]

Suppose that if the student were in the “knowing” state, they would have a probability of 0.4 of getting the question correct on the first attempt, a probability of 0.3 of getting the question correct on a subsequent attempt, a probability of 0.2 of using a hint, and a probability of 0.1 of an incorrect attempt. If the student were in the “unknowing” state, they would have a probability of 0.1 of getting the question correct on the first attempt, a probability of 0.2 of getting the question correct on a subsequent attempt, a probability of 0.5 of using a hint, and a probability of 0.2 of an incorrect attempt.
This information corresponds to the emission probabilities \( E \). Define \( E \) as

\[
\begin{bmatrix}
A & C & H & I \\
K & 0.4 & 0.3 & 0.2 & 0.1 \\
U & 0.1 & 0.2 & 0.5 & 0.2
\end{bmatrix}.
\]

The set of parameters \( \lambda = (\mu, T, E) \) defines the parameters of the hidden Markov model.

Example 3.25 is a hidden Markov model whose parameters are given. How do we proceed if the parameters are unknown? If we use a hidden Markov model we first need to define the parameters. The Baum-Welch algorithm is an algorithm that determines the parameters of a hidden Markov model using the observations.

### 3.2 Baum-Welch algorithm

The intent of this paper is to model student performance on exercises in an individualized instruction platform using a hidden Markov model. The only information available from the platform is student data. We are not initially provided the parameters of the hidden Markov model. In this paper, we will focus on answering the following questions:

1. How can we assess a specific exercise in an exercise platform?

2. How can we measure student learning in this system?

3. Can we identify students who are not performing on a particular exercise as expected?

The first question can be answered via the Baum-Welch algorithm, while the second and third apply to the Viterbi algorithm. We must run the Baum-Welch algorithm before using the Viterbi algorithm.
The *Baum-Welch algorithm* is an algorithm designed to determine the parameters $\lambda = (\mu, T, E)$ of a *hidden Markov model* that maximize the probability of observing any sequence of observations based upon these parameters. Initially a set of parameters must be created as an input to the algorithm. The set of observed data is also an input to the algorithm.

The input observation data to the *Baum-Welch algorithm* may consist of many sequences of observations. For example each student would have their own sequence of observations. We denote a single superscript with a parenthesis $Y^{(w)}$ to denote the $w^{th}$ observed sequence in the data. The goal of the *Baum-Welch algorithm* is to determine the set of parameters $\lambda^* = (\mu^*, T^*, E^*)$ such that

$$\sum_{w=1}^{v} P(Y^{(w)} | \lambda^*) \geq \sum_{w=1}^{v} P(Y^{(w)} | \lambda)$$

for all possible parameters $\lambda$ and any set of observations $\{Y^{(1)}, \ldots, Y^{(v)}\}$ where $Y^{(w)}$ is the $w^{th}$ observation sequence.

Since the *Baum-Welch algorithm* is an iterative process we must keep track of the iteration index when making calculations. We denote a pair of superscript with a parenthesis $□^{(r,w)}$ to denote a variable under the $r^{th}$ iteration for the $w^{th}$ observed sequence $Y^{(w)}$. If a variable pertains to a combination of all observed sequences, then the notation $□^{(r,\text{all})}$ is used. To run the *Baum-Welch algorithm*, first a set of parameters $\lambda^{(0,\text{all})} = (\mu^{(0,\text{all})}, T^{(0,\text{all})}, E^{(0,\text{all})})$ must be selected. The *Baum-Welch algorithm* can be run with this set of parameters and the observed data $\{Y^{(1)}, \ldots, Y^{(v)}\}$. At each iteration a new set of parameters is calculated. For example, after one iteration of the *Baum-Welch algorithm*, the parameters $\lambda^{(1,\text{all})} = (\mu^{(1,\text{all})}, T^{(1,\text{all})}, E^{(1,\text{all})})$
are determined with the property that
\[
\sum_{w=1}^{v} P(Y^{(w)}|\lambda^{(1,\text{all})}) \geq \sum_{w=1}^{v} P(Y^{(w)}|\lambda^{(0,\text{all})}).
\]

If a particular criterion has not been met after this iteration, then the parameters \(\lambda^{(1,\text{all})}\) become the new input to the \textit{Baum-Welch algorithm} and a second iteration is run. In general we have that
\[
\sum_{w=1}^{v} P(Y^{(w)}|\lambda^{(r,\text{all})}) \geq \sum_{w=1}^{v} P(Y^{(w)}|\lambda^{(r-1,\text{all})})
\]
for each iteration \(r\). This process continues iteratively until the difference of these two sums falls below a threshold \(\varepsilon\).

\textbf{Algorithm 3.26 (Baum-Welch algorithm).}

1. Define the following beforehand:

   - \textbf{Possible hidden states} \(\{x_1, x_2, \ldots, x_k\}\), \(k\) possible hidden states.
   - \textbf{Possible observations} \(\{y_1, y_2, \ldots, y_m\}\), \(m\) possible observations.
   - \textbf{Sequences of observed data} \(\{Y^{(1)}, \ldots, Y^{(v)}\}\), \(v\) observed sequences, where \(Y^{(w)} = \{Y_1^{(w)}, \ldots, Y_n^{(w)}\}\) and \(Y_i^{(w)}\) is the \(i\)th observation in the \(w\)th sequence of \(n\) observations.
   - \textbf{Starting initial probabilities of hidden states} \(\mu^{(0,w)}\) for sequence \(Y^{(w)}\)
     \[
     \left[\mu_1^{(0,w)} \mu_2^{(0,w)} \ldots \mu_k^{(0,w)}\right].
     \]
   - \textbf{Starting transition probabilities} \(T^{(0,w)}\), a \(k \times k\) matrix where
     \[
     T^{(0,w)} = [d_{ij}^{(0,w)}] = [P(X_t = x_j|X_{t-1} = x_i)].
     \]
Starting emission probabilities $E^{(0,w)}$, a $k \times m$ matrix where

$$E^{(0,w)} = \left[ e^{(0,w)}_i(y_j) \right]$$

$$= [P(Y_t = y_j | X_t = x_i)].$$

Threshold $\varepsilon$, which is used as the criteria to when the algorithm terminates.

2. Variables calculated within algorithm at each iteration $r$ and single observation sequence $Y^{(w)}$.

Forward probability $\alpha^{(r,w)}_t(x_i)$:

$$\alpha^{(r,w)}_t(x_i) = P(Y_1 = y_1 \cap \cdots \cap Y_t = y_t \cap X_t = x_i).$$

This is the probability of seeing $\{y_1, \ldots, y_t\}$ and being in state $x_i$ at time $t$.

Backward probability $\beta^{(r,w)}_t(x_i)$:

$$\beta^{(r,w)}_t(x_i) = P(Y_{t+1} = y_{t+1} \cap \cdots \cap Y_n = y_n | X_t = x_i).$$

This is the probability of seeing the tail end observations $\{y_{t+1}, \ldots, y_n\}$ given that the model is in state $x_i$ at time $t$.

Update probabilities $\gamma^{(r,w)}_t(x_i)$ and $\xi^{(r,w)}_t(x_i, x_j)$:

$$\gamma^{(r,w)}_t(x_i) = P(X_t = x_i | Y_1 = y_1 \cap \cdots \cap Y_t = y_t).$$

This is the probability of being in state $x_i$ at time $t$ given the sequence of observations $Y^{(w)}$.

$$\xi^{(r,w)}_t(x_i, x_j) = P(X_t = x_i \cap X_{t+1} = x_j | Y_1 = y_1 \cap \cdots \cap Y_t = y_t).$$
This is the probability of being in state $x_i$ at time $t$ and state $x_j$ at time $t + 1$ given the sequence of observations $\{y_1, \ldots, y_t\}$.

**Initial probabilities** $\mu^{(r,w)}$. These are the initial probabilities of the hidden states after iteration $r$ for sequence $Y^{(w)}$.

**Transition probabilities** $T^{(r,w)}$. These are the transition probabilities between hidden states after iteration $r$ for sequence $Y^{(w)}$.

**Emission probabilities** $E^{(r,w)}$. These are the emission probabilities of observations from a hidden state after iteration $r$ for sequence $Y^{(w)}$.

3. Iteratively calculate the forward probabilities $\alpha^{(r,w)}_t(x_i)$ at iteration $r$:

   (a) $\alpha^{(r,w)}_1(x_i) = \mu^{(r-1,w)}_i \cdot e^{(r-1,w)}_i(y_1)$.
   
   (b) $\alpha^{(r,w)}_{t+1}(x_i) = e^{(r-1,w)}_i(y_{t+1}) \sum_{j=1}^k \alpha^{(r,w)}_t(x_j) \cdot q^{(r-1,w)}_{ji}$

4. Iteratively calculate the backward probabilities $\beta_t(x_i)$ at iteration $r$:

   (a) $\beta^{(r,w)}_n(x_i) = 1$.
   
   (b) $\beta^{(r,w)}_{t-1}(x_i) = \sum_{j=1}^k \beta^{(r,w)}_t(x_j) \cdot q^{(r-1,w)}_{ij} \cdot e^{(r-1,w)}_j(y_t)$

5. Iterative calculate the update probabilities $\gamma^{(r,w)}_t(x_i)$ and $\xi^{(r,w)}_t(x_i, x_j)$ at iteration $r$:

   (a) $\gamma^{(r,w)}_t(x_i) = \frac{\alpha^{(r,w)}_t(x_i) \cdot \beta^{(r,w)}_t(x_i)}{\sum_{j=1}^k \alpha^{(r,w)}_t(x_j) \cdot \beta^{(r,w)}_t(x_j)}$. The denominator can be simplified to the following (proof below):

   $\gamma^{(r,w)}_t(x_i) = \frac{\alpha^{(r,w)}_t(x_i) \cdot \beta^{(r,w)}_t(x_i)}{\sum_{j=1}^k \alpha^{(r,w)}_n(x_j)}$. 

   to the following (proof below):

   $\gamma^{(r,w)}_t(x_i) = \frac{\alpha^{(r,w)}_t(x_i) \cdot \beta^{(r,w)}_t(x_i)}{\sum_{j=1}^k \alpha^{(r,w)}_n(x_j)}$. 

   57
(b) \( \xi_t^{(r,w)}(x_i, x_j) = \frac{\alpha_t^{(r,w)}(x_i) \cdot q_{ij}^{(r-1,w)} \cdot e_i^{(r-1,w)}(y_{t+1}) \cdot \beta_t^{(r,w)}(x_j)}{\sum_{j=1}^k \alpha_t^{(r,w)}(x_j) \cdot \beta_t^{(r,w)}(x_j)} \). The denominator is the same as for \( \gamma_t^{(r,w)}(x_i) \) and can be simplified (proof below):
\[
\xi_t^{(r,w)}(x_i, x_j) = \frac{\alpha_t^{(r,w)}(x_i) \cdot q_{ij}^{(r-1,w)} \cdot e_i^{(r-1,w)}(y_{t+1}) \cdot \beta_t^{(r,w)}(x_j)}{\sum_{j=1}^k \alpha_n^{(r,w)}(x_j)}
\]

(c) Proof that the denominators in (a) and (b) can be simplified.

**Proof**

\[
\sum_{j=1}^k \alpha_t^{(r,w)}(x_j) \cdot \beta_t^{(r,w)}(x_j) = \sum_{j=1}^k e_j^{(r-1,w)}(y_t) \sum_{i=1}^k \alpha_{t-1}^{(r,w)}(x_i) \cdot q_{ij}^{(r-1,w)} \cdot \beta_t^{(r,w)}(x_j)
\]

\[
= \sum_{i=1}^k \alpha_{t-1}^{(r,w)}(x_i) \sum_{j=1}^k \beta_t^{(r,w)}(x_j) \cdot q_{ij}^{(r-1,w)} \cdot e_j^{(r-1,w)}(y_t)
\]

\[
= \sum_{i=1}^k \alpha_{t-1}^{(r,w)}(x_i) \cdot \beta_{t-1}^{(r,w)}(x_i)
\]

Therefore the denominator is independent of the time index. It we choose \( t = n \), then the simplification can be made

\[
\sum_{j=1}^k \alpha_t^{(r,w)}(x_j) \cdot \beta_t^{(r,w)}(x_j) = \sum_{j=1}^k \alpha_n^{(r,w)}(x_j) \cdot \beta_n^{(r,w)}(x_j)
\]

\[
= \sum_{j=1}^k \alpha_n^{(r,w)}(x_j)
\]

6. Compute the new initial probabilities \( \mu^{(r,w)} \) of the hidden states at iteration \( r \) for sequence \( Y^{(w)} \):

\[
[x_1 \quad \ldots \quad x_k]
\]

\[
[\gamma_1^{(r,w)}(x_1) \quad \ldots \quad \gamma_1^{(r,w)}(x_k)]
\]

58
7. Compute the new transition probability matrix $T^{(r,w)} = [q_{ij}^{(r,w)}]$ at iteration $r$ for sequence $Y^{(w)}$, where

$$q_{ij}^{(r,w)} = \frac{\sum_{t=1}^{n-1} \xi_t^{(r,w)}(x_i, x_j)}{\sum_{t=1}^{n-1} \gamma_t^{(r,w)}(x_i)}.$$

8. Compute the new emission probability matrix $E^{(r,w)} = \{e_i^{(r,w)}(y_j)\}$ at iteration $r$ for sequence $Y^{(w)}$, where

$$e_i^{(r,w)}(y_j) = \frac{\sum_{t=1}^{n} \gamma_t^{(r,w)}(x_i) \cdot I(Y_t = y_j)}{\sum_{t=1}^{n} \gamma_t^{(r,w)}(x_i)}$$

and $I(Y_t = y_j)$ is an indicator function defined as

$$I(Y_t = y_j) = \begin{cases} 1 & Y_t = y_j \\ 0 & \text{otherwise} \end{cases}$$

9. Define $\lambda^{(r,w)}$ as

$$\lambda^{(r,w)} = (\mu^{(r,w)}, T^{(r,w)}, E^{(r,w)})$$

The probability of the sequence of observations, $Y^{(w)}$ is computed under the new set of parameters $\lambda^{(r,w)}$.

$$p^{(r,w)} = P(Y_1 = y_1 \cap \cdots \cap Y_n = y_n | \lambda^{(r,w)})$$

$$= \sum_{j=1}^{k} a_n^{(r,w)}(x_j)$$

10. Steps 3–9 are repeated under iteration $r$ for every observation sequence $Y^{(w)}$ for $w = 1, \ldots, v$ to compute $\lambda^{(r,w)}$ and $p^{(r,w)}$.

11. The parameters $\lambda^{(r, all)}$ and $p^{(r, all)}$ are computed using all of the observed sequences under iteration $r$. 
Initial probabilities $\mu^{(r, all)}$.

$$\mu^{(r, all)} = \frac{1}{v} \sum_{w=1}^{v} \mu^{(r, w)}$$

Transition probabilities $T^{(r, all)}$.

$$T^{(r, all)} = \frac{1}{v} \sum_{w=1}^{v} T^{(r, w)}$$

Emission probabilities $E^{(r, all)}$.

$$E^{(r, all)} = \frac{1}{v} \sum_{w=1}^{v} E^{(r, w)}$$

Total observation probability $p^{(r, all)}$.

$$p^{(r, all)} = \sum_{w=1}^{v} p^{(r, w)}$$

12. The process terminates once $|p^{(r, all)} - p^{(r-1, all)}| < \varepsilon$ for iteration $r > 1$ and a previously specified threshold $\varepsilon$. The process must iterate at least twice to test for convergence. If the process has converged for iteration $r$, then set $\lambda^* = \lambda^{(r, all)}$. If the process has not converged at iteration $r$, then repeat steps 3–11 for the next iteration $r + 1$.

To gain a better insight into the Baum-Welch algorithm, an example is provided outlining the calculations required to determine the final parameters $\lambda^*$ of a hidden Markov model. Detailed calculations are provided for one iteration of the algorithm on one observation sequence.

Example 3.27 (Baum-Welch algorithm). Suppose a student is working exercises through an individualized instruction platform. We have observed data for a few students making attempts on one exercise. This system only monitors whether the
student is correct or incorrect. We also assume that the student is only either in a knowing state or an unknowing state.

1. Define the following beforehand for observation $Y^{(w)}$:

**Possible hidden states** Knowing (K) or Unknowing (U): $\{x_1, x_2\} = \{K, U\}$, $k = 2$ possible hidden states.

**Possible observations** Correct (C) or Incorrect (I), $m = 2$ possible observations.

**Sequences of observed data**

$Y^{(1)} = \{I, C, C\}$

$Y^{(2)} = \{C, I, I, C\}$

This example will give the detailed calculations for $Y^{(1)}$.

**Starting initial probabilities of hidden states** $\mu^{(0,1)}$ are

$\begin{bmatrix}
K & U \\
0.2 & 0.8
\end{bmatrix}$.

**Stating transition probabilities** $T^{(0,1)}$ is

$\begin{bmatrix}
K & U \\
K & 0.9 & 0.1 \\
U & 0.4 & 0.6
\end{bmatrix}$.

**Starting emission probabilities** $E^{(0,1)}$ is

$\begin{bmatrix}
C & I \\
K & 0.7 & 0.3 \\
U & 0.3 & 0.7
\end{bmatrix}$.

**Threshold** $\varepsilon = 10^{-5}$.
2. Variables calculated within algorithm at iteration \( r = 1 \) and observation sequence \( Y^{(1)} \).

**Forward probability** \( \alpha^{(1,1)}_t(x_i) \)

\[
\begin{align*}
\alpha^{(1,1)}_1(K) &= P(Y_1 = I \cap X_1 = K) \\
\alpha^{(1,1)}_1(U) &= P(Y_1 = I \cap X_1 = U) \\
\alpha^{(1,1)}_2(K) &= P(Y_1 = I \cap Y_2 = C \cap X_2 = K) \\
\alpha^{(1,1)}_2(U) &= P(Y_1 = I \cap Y_2 = C \cap X_2 = U) \\
\alpha^{(1,1)}_3(K) &= P(Y_1 = I \cap Y_2 = C \cap Y_3 = C \cap X_3 = K) \\
\alpha^{(1,1)}_3(U) &= P(Y_1 = I \cap Y_2 = C \cap Y_3 = C \cap X_3 = U)
\end{align*}
\]

**Backward probability** \( \beta^{(1,1)}_t(x_i) \)

\[
\begin{align*}
\beta^{(1,1)}_3(K) &= 1 \\
\beta^{(1,1)}_3(U) &= 1 \\
\beta^{(1,1)}_2(K) &= P(Y_3 = C | X_2 = K) \\
\beta^{(1,1)}_2(U) &= P(Y_3 = C | X_2 = U) \\
\beta^{(1,1)}_1(K) &= P(Y_2 = C \cap Y_3 = C | X_1 = K) \\
\beta^{(1,1)}_1(U) &= P(Y_2 = C \cap Y_3 = C | X_1 = U)
\end{align*}
\]
Update probabilities $\gamma_t^{(1,1)}(x_i)$ and $\xi_t^{(1,1)}(x_i, x_j)$

\[
\begin{align*}
\gamma_1^{(1,1)}(K) &= P(X_1 = K| Y_1 = I) \\
\gamma_1^{(1,1)}(U) &= P(X_1 = U| Y_1 = I) \\
\gamma_2^{(1,1)}(K) &= P(X_2 = K| Y_1 = I \cap Y_2 = C) \\
\gamma_2^{(1,1)}(U) &= P(X_2 = U| Y_1 = I \cap Y_2 = C) \\
\gamma_3^{(1,1)}(K) &= P(X_3 = K| Y_1 = I \cap Y_2 = C \cap Y_3 = C) \\
\gamma_3^{(1,1)}(U) &= P(X_3 = U| Y_1 = I \cap Y_2 = C \cap Y_3 = C) \\
\xi_1^{(1,1)}(K, K) &= P(X_1 = K \cap X_2 = K| Y_1 = I) \\
\xi_1^{(1,1)}(K, U) &= P(X_1 = K \cap X_2 = U| Y_1 = I) \\
\xi_1^{(1,1)}(U, K) &= P(X_1 = U \cap X_2 = K| Y_1 = I) \\
\xi_1^{(1,1)}(U, U) &= P(X_1 = U \cap X_2 = U| Y_1 = I) \\
\xi_2^{(1,1)}(K, K) &= P(X_2 = K \cap X_3 = K| Y_1 = I \cap Y_2 = C) \\
\xi_2^{(1,1)}(K, U) &= P(X_2 = K \cap X_3 = U| Y_1 = I \cap Y_2 = C) \\
\xi_2^{(1,1)}(U, K) &= P(X_2 = U \cap X_3 = K| Y_1 = I \cap Y_2 = C) \\
\xi_2^{(1,1)}(U, U) &= P(X_2 = U \cap X_3 = U| Y_1 = I \cap Y_2 = C)
\end{align*}
\]

Initial probabilities $\mu^{(1,1)}$. These are the initial probabilities of hidden states after the first iteration for sequence $Y^{(1)}$.

Transition probabilities $T^{(1,1)}$. These are the transition probabilities between hidden states after the first iteration for sequence $Y^{(1)}$.

Emission probabilities $E^{(1,1)}$. These are the emission probabilities of observations from a hidden state after the first iteration for sequence $Y^{(1)}$. 
3. Iteratively calculate the forward probabilities $\alpha_{1}^{(1,1)}(x_i)$ for the first iteration:

(a) Initialize $\alpha_{1}^{(1,1)}(x_i)$

$$\alpha_{1}^{(1,1)}(x_i) = \mu_{i}^{(0,1)} \cdot e_{i}^{(0,1)}(y_1)$$

$$\alpha_{1}^{(1,1)}(K) = \mu_{K}^{(0,1)} \cdot e_{K}^{(0,1)}(I)$$

$$= (0.2)(0.3)$$

$$= 0.06$$

$$\alpha_{1}^{(1,1)}(U) = \mu_{U}^{(0,1)} \cdot e_{U}^{(0,1)}(I)$$

$$= (0.8)(0.7)$$

$$= 0.56$$
(b) Iteratively calculate $\alpha^{(1,1)}_{t+1}(x_i)$

\[
\alpha^{(1,1)}_{t+1}(x_i) = e^{(0,1)}_i(y_{t+1}) \sum_{j=1}^{k} \alpha^{(1,1)}_t(x_j) \cdot q^{(0,1)}_{ji}
\]

\[
\alpha^{(1,1)}_2(K) = e^{(0,1)}_K(C) \left( \alpha^{(1,1)}_1(K) \cdot q^{(0,1)}_{KK} + \alpha^{(1,1)}_1(U) \cdot q^{(0,1)}_{UK} \right)
\]

\[
= (0.7) ((0.06)(0.9) + (0.56)(0.4))
\]

\[
= 0.1946
\]

\[
\alpha^{(1,1)}_2(U) = e^{(0,1)}_U(C) \left( \alpha^{(1,1)}_1(K) \cdot q^{(0,1)}_{KU} + \alpha^{(1,1)}_1(U) \cdot q^{(0,1)}_{UU} \right)
\]

\[
= (0.3) ((0.06)(0.1) + (0.56)(0.6))
\]

\[
= 0.1026
\]

\[
\alpha^{(1,1)}_3(K) = e^{(0,1)}_K(C) \left( \alpha^{(1,1)}_2(K) \cdot q^{(0,1)}_{KK} + \alpha^{(1,1)}_2(U) \cdot q^{(0,1)}_{UK} \right)
\]

\[
= (0.7) ((0.1946)(0.9) + (0.1026)(0.4))
\]

\[
= 0.151326
\]

\[
\alpha^{(1,1)}_3(U) = e^{(0,1)}_U(C) \left( \alpha^{(1,1)}_2(K) \cdot q^{(0,1)}_{KU} + \alpha^{(1,1)}_2(U) \cdot q^{(0,1)}_{UU} \right)
\]

\[
= (0.3) ((0.1946)(0.1) + (0.1026)(0.6))
\]

\[
= 0.024306
\]

4. Iteratively calculate the backward probabilities $\beta^{(1,1)}_t(x_i)$ for the first iteration:

(a) Initialize $\beta^{(1,1)}_t(x_i)$

\[
\beta^{(1,1)}_t(x_i) = 1
\]

\[
\beta^{(1,1)}_3(K) = 1
\]

\[
\beta^{(1,1)}_3(U) = 1
\]
(b) Iteratively calculate $\beta_{t-1}^{(1,1)}(x_i)$

\[
\beta_{t-1}^{(1,1)}(x_i) = \sum_{j=1}^{k} \beta_t^{(1,1)}(x_i) \cdot q_{ij}^{(0,1)} \cdot e_j^{(0,1)}(y_t)
\]

\[
\beta_2^{(1,1)}(K) = \beta_3^{(1,1)}(K) \cdot q_{KK}^{(0,1)} \cdot e_K^{(0,1)}(C) + \beta_3^{(1,1)}(U) \cdot q_{KU}^{(0,1)} \cdot e_U^{(0,1)}(C)
\]

\[
= (1)(0.9)(0.7) + (1)(0.1)(0.3)
\]

\[
= 0.66
\]

\[
\beta_2^{(1,1)}(U) = \beta_3^{(1,1)}(K) \cdot q_{UK}^{(0,1)} \cdot e_K^{(0,1)}(C) + \beta_3^{(1,1)}(U) \cdot q_{UU}^{(0,1)} \cdot e_U^{(0,1)}(C)
\]

\[
= (1)(0.4)(0.7) + (1)(0.6)(0.3)
\]

\[
= 0.46
\]

\[
\beta_1^{(1,1)}(K) = \beta_2^{(1,1)}(K) \cdot q_{KK}^{(0,1)} \cdot e_K^{(0,1)}(C) + \beta_2^{(1,1)}(U) \cdot q_{KU}^{(0,1)} \cdot e_U^{(0,1)}(C)
\]

\[
= (0.66)(0.9)(0.7) + (0.46)(0.1)(0.3)
\]

\[
= 0.4296
\]

\[
\beta_1^{(1,1)}(U) = \beta_2^{(1,1)}(K) \cdot q_{UK}^{(0,1)} \cdot e_K^{(0,1)}(C) + \beta_2^{(1,1)}(U) \cdot q_{UU}^{(0,1)} \cdot e_U^{(0,1)}(C)
\]

\[
= (0.66)(0.4)(0.7) + (0.46)(0.6)(0.3)
\]

\[
= 0.2676
\]

5. Iteratively calculate update probabilities $\gamma_t^{(1,1)}(x_i)$ and $\xi_t^{(1,1)}(x_i, x_j)$ for the first iteration:

(a) Calculate denominator that is the same for all terms, $\sum_{j=1}^{k} \alpha_n(x_j)$

\[
\alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U) = 0.151326 + 0.024306
\]

\[
= 0.175632
\]

(b) Calculate the update probabilities $\gamma_t^{(1,1)}(x_i)$

\[
\gamma_t^{(1,1)}(x_i) = \frac{\alpha_t^{(1,1)}(x_i) \cdot \beta_t^{(1,1)}(x_i)}{\sum_{j=1}^{k} \alpha_n^{(1,1)}(x_j)}
\]
\[
\begin{align*}
\gamma_{1,1}^{(1,1)}(K) &= \frac{\alpha_3^{(1,1)}(K) \cdot \beta_1^{(1,1)}(K)}{\alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U)} \\
&= \frac{(0.06)(0.4296)}{0.175632} \\
&= 0.146761
\\
\gamma_{1,1}^{(1,1)}(U) &= \frac{\alpha_3^{(1,1)}(U) \cdot \beta_1^{(1,1)}(U)}{\alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U)} \\
&= \frac{(0.56)(0.2676)}{0.175632} \\
&= 0.853239
\\
\gamma_{2,1}^{(1,1)}(K) &= \frac{\alpha_3^{(1,1)}(K) \cdot \beta_2^{(1,1)}(K)}{\alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U)} \\
&= \frac{(0.1946)(0.66)}{0.175632} \\
&= 0.731279
\\
\gamma_{2,1}^{(1,1)}(U) &= \frac{\alpha_3^{(1,1)}(U) \cdot \beta_2^{(1,1)}(U)}{\alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U)} \\
&= \frac{(0.1026)(0.46)}{0.175632} \\
&= 0.268721
\\
\gamma_{3,1}^{(1,1)}(K) &= \frac{\alpha_3^{(1,1)}(K) \cdot \beta_3^{(1,1)}(K)}{\alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U)} \\
&= \frac{(0.151326)(1)}{0.175632} \\
&= 0.861608
\\
\gamma_{3,1}^{(1,1)}(U) &= \frac{\alpha_3^{(1,1)}(U) \cdot \beta_3^{(1,1)}(U)}{\alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U)} \\
&= \frac{(0.024306)(1)}{0.175632} \\
&= 0.138392
\end{align*}
\]
(c) Calculate the update probabilities $\xi_{t}^{(1,1)}(x_i, x_j)$ for the first iteration and first observation:

$$ \xi_{t}^{(1,1)}(x_i, x_j) = \frac{\alpha_{t}^{(1,1)}(x_i) \cdot q_{ij}^{(0,1)} \cdot e_i(y_{t+1}) \cdot \beta_{t+1}(x_j)}{\sum_{j=1}^{k} \alpha_{n}^{(1,1)}(x_j)} $$

$$ \xi_{t}^{(1,1)}(K, K) = \frac{\alpha_{1}^{(1,1)}(K) \cdot q_{KK}^{(0,1)} \cdot e_K(C) \cdot \beta_{2}^{(1,1)}(K)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} $$

$$ = \frac{(0.06)(0.9)(0.7)(0.66)}{0.175632} $$

$$ = 0.142047 $$

$$ \xi_{t}^{(1,1)}(K, U) = \frac{\alpha_{1}^{(1,1)}(K) \cdot q_{KU}^{(0,1)} \cdot e_U(C) \cdot \beta_{2}^{(1,1)}(U)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} $$

$$ = \frac{(0.06)(0.1)(0.3)(0.46)}{0.175632} $$

$$ = 0.004714 $$

$$ \xi_{t}^{(1,1)}(U, K) = \frac{\alpha_{1}^{(1,1)}(U) \cdot q_{UK}^{(0,1)} \cdot e_K(C) \cdot \beta_{2}^{(1,1)}(K)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} $$

$$ = \frac{(0.56)(0.4)(0.7)(0.66)}{0.175632} $$

$$ = 0.589232 $$

$$ \xi_{t}^{(1,1)}(U, U) = \frac{\alpha_{1}^{(1,1)}(U) \cdot q_{UU}^{(0,1)} \cdot e_U(C) \cdot \beta_{2}^{(1,1)}(K)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} $$

$$ = \frac{(0.56)(0.6)(0.3)(0.46)}{0.175632} $$

$$ = 0.264007 $$
\[\xi_{2}^{(1,1)}(K, K) = \frac{\alpha_{2}^{(1,1)}(K) \cdot q_{KK}^{(0,1)} \cdot e_{K}^{(0,1)}(C) \cdot \beta_{3}^{(1,1)}(K)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} = \frac{(0.1946)(0.9)(0.7)(1)}{0.175632} = 0.698039\]

\[\xi_{2}^{(1,1)}(K, U) = \frac{\alpha_{2}^{(1,1)}(K) \cdot q_{KU}^{(0,1)} \cdot e_{U}^{(0,1)}(C) \cdot \beta_{3}^{(1,1)}(U)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} = \frac{(0.1946)(0.1)(0.3)(1)}{0.175632} = 0.033240\]

\[\xi_{2}^{(1,1)}(U, K) = \frac{\alpha_{2}^{(1,1)}(U) \cdot q_{UK}^{(0,1)} \cdot e_{K}^{(0,1)}(C) \cdot \beta_{3}^{(1,1)}(K)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} = \frac{(0.1026)(0.4)(0.7)(1)}{0.175632} = 0.163569\]

\[\xi_{2}^{(1,1)}(U, U) = \frac{\alpha_{2}^{(1,1)}(U) \cdot q_{UU}^{(0,1)} \cdot e_{U}^{(0,1)}(C) \cdot \beta_{3}^{(1,1)}(K)}{\alpha_{3}^{(1,1)}(K) + \alpha_{3}^{(1,1)}(U)} = \frac{(0.1026)(0.6)(0.3)(1)}{0.175632} = 0.105152\]

6. Compute the new initial probabilities of the hidden states for the first iteration and first observation:

\[\mathbf{\mu}^{(1,1)} = \left(\gamma_{1}^{(1,1)}(x_{1}), \ldots, \gamma_{1}^{(1,1)}(x_{k})\right)\]

\[= \left(\gamma_{1}^{(1,1)}(K), \gamma_{1}^{(1,1)}(U)\right)\]

\[= (0.146761, 0.853239)\]
7. Compute the new transition probability matrix \( T^{(1,1)} = [q_{ij}^{(1,1)}] \) for the first iteration:

\[
q_{ij}^{(1,1)} = \frac{\sum_{t=1}^{n-1} \xi_t^{(1,1)}(x_i, x_j)}{\sum_{t=1}^{n-1} \gamma_t^{(1,1)}(x_i)}
\]

\[
q_{KK}^{(1,1)} = \frac{\xi_1^{(1,1)}(K, K) + \xi_2^{(1,1)}(K, K)}{\gamma_1^{(1,1)}(K) + \gamma_2^{(1,1)}(K)}
\]

\[
= \frac{0.142047 + 0.698039}{0.146761 + 0.731279}
\]

\[
= 0.956774
\]

\[
q_{KU}^{(1,1)} = \frac{\xi_1^{(1,1)}(K, U) + \xi_2^{(1,1)}(K, U)}{\gamma_1^{(1,1)}(K) + \gamma_2^{(1,1)}(K)}
\]

\[
= \frac{0.004714 + 0.033240}{0.146761 + 0.731279}
\]

\[
= 0.043226
\]

\[
q_{UK}^{(1,1)} = \frac{\xi_1^{(1,1)}(U, K) + \xi_2^{(1,1)}(U, K)}{\gamma_1^{(1,1)}(U) + \gamma_2^{(1,1)}(U)}
\]

\[
= \frac{0.589232 + 0.163569}{0.853239 + 0.268721}
\]

\[
= 0.670970
\]

\[
q_{UU}^{(1,1)} = \frac{\xi_1^{(1,1)}(U, U) + \xi_2^{(1,1)}(U, U)}{\gamma_1^{(1,1)}(U) + \gamma_2^{(1,1)}(U)}
\]

\[
= \frac{0.264007 + 0.105152}{0.853239 + 0.268721}
\]

\[
= 0.329030
\]
8. Compute the new emission probability matrix \( E = [e_i^{(1,1)}(y_j)] \) for the first iteration and first observation.

\[
e_i^{(1,1)}(y_j) = \frac{\sum_{t=1}^{n} \gamma_t^{(1,1)}(x_i) \cdot I(Y_t = y_j)}{\sum_{t=1}^{n} \gamma_t^{(1,1)}(x_i)}
\]

\[
e_K^{(1,1)}(C) = \frac{\gamma_2^{(1,1)}(K) + \gamma_3^{(1,1)}(K)}{\gamma_1^{(1,1)}(K) + \gamma_2^{(1,1)}(K) + \gamma_3^{(1,1)}(K)}
= \frac{0.731279 + 0.861608}{0.146761 + 0.731279 + 0.861608}
= 0.915638
\]

\[
e_K^{(1,1)}(I) = \frac{\gamma_1^{(1,1)}(K)}{\gamma_1^{(1,1)}(K) + \gamma_2^{(1,1)}(K) + \gamma_3^{(1,1)}(K)}
= \frac{0.146761}{0.146761 + 0.731279 + 0.861608}
= 0.084362
\]

\[
e_U^{(1,1)}(C) = \frac{\gamma_2^{(1,1)}(U) + \gamma_3^{(1,1)}(U)}{\gamma_1^{(1,1)}(U) + \gamma_2^{(1,1)}(U) + \gamma_3^{(1,1)}(U)}
= \frac{0.268721 + 0.138392}{0.853239 + 0.268721 + 0.138392}
= 0.323015
\]

\[
e_U^{(1,1)}(I) = \frac{\gamma_1^{(1,1)}(U)}{\gamma_1^{(1,1)}(U) + \gamma_2^{(1,1)}(U) + \gamma_3^{(1,1)}(U)}
= \frac{0.853239}{0.853239 + 0.268721 + 0.138392}
= 0.676985
\]
9. Define $\lambda^{(1,1)} = (\mu^{(1,1)}, T^{(1,1)}, E^{(1,1)})$ as the updated parameters after one iteration

$$
\mu^{(1,1)} = \begin{bmatrix} K & U \\ 0.146761 & 0.853239 \end{bmatrix}.
$$

$$
T^{(1,1)} = \begin{bmatrix} K \\ U \end{bmatrix} = \begin{bmatrix} 0.956774 & 0.043226 \\ 0.670970 & 0.329030 \end{bmatrix}.
$$

$$
E^{(1,1)} = \begin{bmatrix} C \\ I \end{bmatrix} = \begin{bmatrix} 0.915638 & 0.084362 \\ 0.323015 & 0.676985 \end{bmatrix}.
$$

Compute the probability of seeing sequence $Y^{(1)}$.

$$
p^{(1,1)} = \sum_{j=1}^{k} \alpha_n^{(1,1)}(x_j)
= \alpha_3^{(1,1)}(K) + \alpha_3^{(1,1)}(U)
= 0.175632
$$

10. Steps 3–9 are repeated for $Y^{(2)}$, yielding the following parameters after one iteration.

$$
\mu^{(1,2)} = \begin{bmatrix} K & U \\ 0.240658 & 0.759343 \end{bmatrix}.
$$

$$
T^{(1,2)} = \begin{bmatrix} K \\ U \end{bmatrix} = \begin{bmatrix} 0.872928 & 0.127072 \\ 0.334234 & 0.665766 \end{bmatrix}.
$$

$$
E^{(1,2)} = \begin{bmatrix} C \\ I \end{bmatrix} = \begin{bmatrix} 0.562981 & 0.437019 \\ 0.448691 & 0.551309 \end{bmatrix}.
$$

72
Compute the probability of seeing sequence $Y^{(2)}$.

\[ p^{(1,2)} = \sum_{j=1}^{k} \alpha^{(1,2)}_{n}(x_j) \]
\[ = \alpha^{(1,2)}_{3}(K) + \alpha^{(1,2)}_{3}(U) \]
\[ = 0.044140 \]

11. The parameters $\lambda^{(1, all)}$ and $p^{(1, all)}$ are computed using both observed sequences under the first iteration.

Initial probabilities $\mu^{(1, all)}$:
\[ \mu^{(1, all)} = \frac{1}{2} \left( \mu^{(1,1)} + \mu^{(1,2)} \right) \]
\[ = \begin{bmatrix} K \\ U \end{bmatrix} \begin{bmatrix} 0.193709 \\ 0.806291 \end{bmatrix} \]

Transition probabilities $T^{(1, all)}$:
\[ T^{(1, all)} = \frac{1}{2} \left( T^{(1,1)} + T^{(1,2)} \right) \]
\[ = \begin{bmatrix} K \\ U \end{bmatrix} \begin{bmatrix} 0.914851 & 0.085149 \\ 0.502602 & 0.497398 \end{bmatrix} \]

Emission probabilities $E^{(1, all)}$:
\[ E^{(1, all)} = \frac{1}{2} \left( E^{(1,1)} + E^{(1,2)} \right) \]
\[ = \begin{bmatrix} K \\ U \end{bmatrix} \begin{bmatrix} 0.739309 & 0.260691 \\ 0.385853 & 0.614147 \end{bmatrix} \]

Total observation probability $p^{(1, all)}$:
\[ p^{(1, all)} = p^{(1,1)} + p^{(1,2)} \]
\[ = 0.219772 \]
12. Another iteration is run to determine whether the process terminates. After
the second iteration, the following parameters have been updated.

Initial probabilities $\mu^{(2,\text{all})}$:

$$
\mu^{(2,\text{all})} = \begin{bmatrix} K & U \\ 0.164380 & 0.835621 \end{bmatrix}
$$

Transition probabilities $T^{(2,\text{all})}$:

$$
T^{(2,\text{all})} = \begin{bmatrix} K & U \\ 0.920537 & 0.079463 \\ 0.540063 & 0.459937 \end{bmatrix}
$$

Emission probabilities $E^{(2,\text{all})}$:

$$
E^{(2,\text{all})} = \begin{bmatrix} C & I \\ 0.733961 & 0.266039 \\ 0.384876 & 0.615124 \end{bmatrix}
$$

Total observation probability $p^{(2,\text{all})}$:

$$
p^{(2,\text{all})} = 0.251260
$$

Notice that $p^{(2,\text{all})} \geq p^{(1,\text{all})}$. The difference is $0.031488 > \varepsilon = 10^{-5}$. Thus we
must iterate again. After convergence the parameters are equal to the following.

Final initial probabilities $\mu^*$:

$$
\mu^* = \begin{bmatrix} K & U \\ 0 & 1 \end{bmatrix}
$$

Final transition probabilities $T^*$:

$$
T^* = \begin{bmatrix} K & U \\ 1.00 & 0.00 \\ 0.64 & 0.36 \end{bmatrix}
$$
Final emission probabilities $E^*$:

$$E^* = K \begin{bmatrix} C & I \\ 0.74 & 0.26 \\ 0.34 & 0.66 \end{bmatrix}$$

Final total observation probability $p^*$:

$$p^* = 0.313452$$

The Baum-Welch algorithm is an excellent process to determining the parameters of a hidden Markov model. However, the final parameters from the algorithm are only a local maximum to maximizing the probability $p^*$. The algorithm is highly dependent upon the input parameters $\lambda(0, \text{all})$. In Example 3.27 $p^* = 0.313452$. Is it possible to achieve a higher value for $p^*$ by modifying the input parameters? This is a very important question that will be a future research consideration. In this paper we wish to provide the framework for an exercise assessment model to an individualized instruction platform through a hidden Markov model.

3.3 Baum-Welch algorithm in assessment of exercises

The Baum-Welch algorithm can be used in an individual instruction exercise platform with similar structure as the above example. For the example above there were only two observed states and two hidden states, but the algorithm is dynamic to account for a variable number of observed and hidden states. By monitoring the observations in the background the system has a wealth of exercise data that can be input into the Baum-Welch algorithm in order to analyze a particular exercise. All calculations performed in these examples were done in $R$ [29].

To get a better idea of how the Baum-Welch algorithm can be used to analyze an exercise a few examples will be presented showcasing how student data drives the
parameters of the *Baum-Welch algorithm*. The goal here is to determine if a *hidden Markov model* that is used to model a specific exercise has unique parameters as a result of the *Baum-Welch algorithm*. If we run the *Baum-Welch algorithm* on the data set for a particular exercise, can we make any inference into the overall student performance on that particular exercise?

We define the following parameters beforehand for the *hidden Markov models* in each example below.

**Possible hidden states** Knowing (K) or Unknowing (U): \( \{x_1, x_2\} = \{K, U\}, k = 2 \) possible hidden states.

**Possible observations** Correct (C) or Incorrect (I), \( m = 2 \) possible observations.

**Sequences of observed data** A sample of 100 student data was generated for each example. The sequence length varied between 10 and 20 observations for each sequence. The specifics of the data selection are outlined in each example.

**Starting initial probabilities of hidden states** \( \mu^{(0,w)} \) for every sequence \( Y^{(w)} \) are

\[
\begin{bmatrix}
K \\
U
\end{bmatrix}=egin{bmatrix}
0.2 \\
0.8
\end{bmatrix}.
\]

**Stating transition probabilities** \( T^{(0,w)} \) for every sequence \( Y^{(w)} \) are

\[
\begin{bmatrix}
K & U \\
K & 0.9 \\
U & 0.1 \\
U & 0.4 \\
U & 0.6
\end{bmatrix}.
\]

**Starting emission probabilities** \( E^{(0,w)} \) for every sequence \( Y^{(w)} \) are

\[
\begin{bmatrix}
C & I \\
K & 0.7 \\
K & 0.3 \\
U & 0.3 \\
U & 0.7
\end{bmatrix}.
\]

76
Threshold $\varepsilon = 10^{-5}$.

Example 3.28 (Students “guessing”). The data for this exercise shows that every student in the system is answering correct answer with a probability of roughly $\frac{1}{2}$ and the incorrect answer with a probability of roughly $\frac{1}{2}$. For this example the data for each of the the 100 students was generated as follows.

1. The number of attempts $n$ for a single student was selected uniformly between 10 and 20.

2. For each student attempt either correct or incorrect was selected with equal probability.

After running this set of 100 students through the Baum-Welch algorithm, the following parameters were generated after convergence.

Final initial probabilities of hidden states $\mu^{(*)}$:

$$
\begin{bmatrix}
    K & U \\
    0.32 & 0.68 \\
\end{bmatrix}
$$

Final transition probabilities $T^{*}$:

$$
\begin{bmatrix}
    K & U \\
    K & 0.84 & 0.16 \\
    U & 0.41 & 0.59 \\
\end{bmatrix}
$$

Final emission probabilities $E^{*}$:

$$
\begin{bmatrix}
    C & I \\
    K & 0.50 & 0.50 \\
    U & 0.48 & 0.52 \\
\end{bmatrix}
$$
The most interesting result from the final parameters is that of the final emission probability matrix. Essentially each probability is equal to \( \frac{1}{2} \), regardless of the hidden state. This corresponds exactly to the type of data that was generated for this example where students are correctly or incorrectly with equal probability. The final initial probabilities and transition probabilities do not vary much from the starting parameters. Since the student data is correct or incorrect with equal probability that this would not have much of an effect on the starting parameters to change one of these starting or transition probabilities.

**Example 3.29 (Students “mostly incorrect”).** The exercise observation data shows that there is a high percentage of students getting the problem incorrect most of the time. For this example the data for each of the 100 students was generated as follows:

1. The number of attempts \( n \) for a single student was selected uniformly between 10 and 20.

2. The number of incorrect responses \( i \) for a student was selected uniformly between \( \left\lfloor \frac{n}{2} \right\rfloor \) and \( n - 1 \).

3. The number of correct responses \( c \) was set equal to \( c = n - i \).

4. The student data was a random permutation of the \( i \) incorrect and \( c \) correct responses.

After running this set of 100 students through the Baum-Welch algorithm, the following parameters were generated after convergence.
Final initial probabilities of hidden states $\mu^{(e)}$:

\[
\begin{pmatrix}
K & U \\
0.10 & 0.90
\end{pmatrix}.
\]

Final transition probabilities $T^*$:

\[
\begin{pmatrix}
K & U \\
K & [0.77, 0.23] \\
U & [0.31, 0.69]
\end{pmatrix}.
\]

Final emission probabilities $E^*$:

\[
\begin{pmatrix}
C & I \\
K & [0.31, 0.69] \\
U & [0.27, 0.73]
\end{pmatrix}.
\]

Firstly the final initial probabilities have the students starting with even greater probability in the unknowing state as compared to the starting initial probabilities.

Secondly the change in the transition probability matrix also shows a decline in the probabilities for the student ending in the knowing state and a rise in the probabilities for the student ending in the unknowing state. Finally the emission probability matrix shows that the majority of the students are answering incorrectly, regardless of whether they know the material.

**Example 3.30 (Students “mostly correct”).** The exercise observation data shows that there is a high percentage of students getting the problem correct most of the time. For this example the data for each of the the 100 students was generated as follows:

1. The number of attempts $n$ for a single student was selected uniformly between 10 and 20.
2. The number of correct responses $c$ was selected uniformly between $\left\lceil \frac{n}{2} \right\rceil$ and $n - 1$.

3. The number of incorrect responses $i$ was set equal to $i = n - c$.

4. The student data was a random permutation of the $i$ incorrect and $c$ correct responses.

After running this set of 100 students through the Baum-Welch algorithm, the following parameters were generated after convergence.

**Final initial probabilities of hidden states $\mu^*$:**

\[
\begin{bmatrix}
K & U \\
0.38 & 0.62
\end{bmatrix}
\]

**Final transition probabilities $T^*$:**

\[
\begin{bmatrix}
K & U \\
K & [0.91 \ 0.09] \\
U & [0.54 \ 0.46]
\end{bmatrix}
\]

**Final emission probabilities $E^*$:**

\[
\begin{bmatrix}
C & I \\
K & [0.71 \ 0.29] \\
U & [0.74 \ 0.26]
\end{bmatrix}
\]

The results of this example are nearly opposite of example 3.29. The final initial probabilities show that the probability of starting in the knowing state has increased, while the probability of starting in the unknowing state has decreased. The transition probability matrix shows a higher probability of moving to the knowing state regardless of the previous state. The emission probability matrix shows similar results in that the probability of getting the answer correct is larger for both knowing and unknowing states, and is virtually independent of which state the student is in.
Example 3.31 (Students show “evidence of learning”). The exercise observation data shows that the students are first getting the problem incorrect, but quickly get it correct consistently after several incorrect attempts. For this example the data for each of the the 100 students was generated as follows:

1. The number of attempts \( n \) for a single student was selected uniformly between 10 and 20.

2. If \( n \) was even the number of correct \( c \) and incorrect \( i \) responses were set equal to \( \frac{n}{2} \).

3. If \( n \) was odd, then the two numbers \( \frac{n-1}{2} \) and \( \frac{n+1}{2} \) were randomly assigned to each the number of correct \( c \) and incorrect \( i \) with equal probability.

4. The first \( \left\lfloor \frac{n}{3} \right\rfloor \) attempts were set to incorrect, while the last \( \left\lfloor \frac{n}{3} \right\rfloor \) attempts were set to correct.

5. The remaining \( n - \left( 2 \cdot \left\lfloor \frac{n}{3} \right\rfloor \right) \) attempts of correct and incorrect were randomly permuted to make up the middle attempts.

After running this set of 100 students through the Baum-Welch algorithm, the following parameters were generated after convergence.

**Final initial probabilities of hidden states** \( \mu^{(s)} \):

\[
\begin{pmatrix}
K & U \\
0.00 & 1.00
\end{pmatrix}
\]

**Final transition probabilities** \( T^* \):

\[
\begin{pmatrix}
K & U \\
K & [1.00 \quad 0.00] \\
U & [0.28 \quad 0.72]
\end{pmatrix}
\]
Final emission probabilities $E^*$:

$$

\begin{bmatrix}
  C & I \\
  K & [0.75 \ 0.25] \\
  U & [0.00 \ 1.00]
\end{bmatrix}

$$

The results of the initial probabilities show that all students start in the unknowing state. This makes sense because the data was constructed such that all students answered at least the first two attempts incorrectly. The transition probability also makes sense. When the student is in the knowing state, they always stay in the knowing state. The data was constructed so that initially every student is mostly incorrect, but finally the student is mostly correct at the end of their attempts. The probability of moving from the unknowing state into the knowing state has also decreased, which means that the students are not entering into the knowing state as quickly. The emission probability matrix shows that it’s impossible for a student to answer a question correctly when in the unknowing state, but the student would still answer incorrectly if in the knowing state.

**Example 3.32 (Examples 3.28–3.31 combined).** A combination of Examples 3.28–3.31 where $\frac{1}{4}$ of the students are taken from each group. For this example 25 students were randomly generated from the previous four example data sets to make one data set of 100 students. The results of the Baum-Welch algorithm is as follows.

Final initial probabilities of hidden states $\mu^{(*)}$:

$$

\begin{bmatrix}
  K & U \\
  [0.13 \ 0.87]
\end{bmatrix}

$$

Final transition probabilities $T^*$:

$$

\begin{bmatrix}
  K & U \\
  K & [0.86 \ 0.14] \\
  U & [0.38 \ 0.62]
\end{bmatrix}

$$
Final emission probabilities $E^*$:

\[
\begin{array}{cc}
C & I \\
K & [0.58 \ 0.42] \\
U & [0.39 \ 0.61]
\end{array}
\]

The final initial and transition probabilities do not change much from the starting parameters. This is similar to what was seen in example 3.28. This primarily means that the variability in the data does not pull these probabilities in a certain direction, i.e. either toward a knowing state or an unknowing state. The emission probabilities did change for the knowing state. Each probability has become closer to 0.50. The emission probabilities for the unknowing state did not change much at all. Essentially the variation in the data that has been generated drove all these probabilities closer to 0.50.

Table 3.1 summarizes the parameters for each example hidden Markov model. Each example presented has a slightly different variation in the hidden Markov model parameters. Insight into the exercise can be gained by examining the student data as a whole. The parameters generated by the Baum-Welch algorithm can be compared to model examples such as these and one would be able to understand student performance on that particular exercise as a whole. The final parameters could suggest that most students are struggling with an exercise, are completely guessing at an answer, are not challenged enough, or are indeed learning as the course intends. The best way to fully understand the results of the Baum-Welch algorithm would be to generate such model examples beforehand using randomly generated data, and then compare the parameters generated from actual student data to these examples.

However, we suggest that only knowing the parameters of the hidden Markov model is not enough to understand how students are performing on a particular exercise.
<table>
<thead>
<tr>
<th>Example</th>
<th>Type</th>
<th>$\mu^*$</th>
<th>$T^*$</th>
<th>$E^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Starting parameters</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.20  0.80]</td>
<td>[0.90  0.10]</td>
<td>[0.70  0.30]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.40  0.60]</td>
<td>[0.30  0.70]</td>
<td></td>
</tr>
<tr>
<td>3.28</td>
<td>Students “guessing”</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.32  0.68]</td>
<td>[0.84  0.16]</td>
<td>[0.50  0.50]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.41  0.59]</td>
<td>[0.48  0.52]</td>
<td></td>
</tr>
<tr>
<td>3.29</td>
<td>Students “mostly</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
</tr>
<tr>
<td></td>
<td>incorrect”</td>
<td>[0.10  0.90]</td>
<td>[0.77  0.23]</td>
<td>[0.31  0.69]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.31  0.69]</td>
<td>[0.27  0.73]</td>
<td></td>
</tr>
<tr>
<td>3.30</td>
<td>Students “mostly</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
</tr>
<tr>
<td></td>
<td>correct”</td>
<td>[0.38  0.62]</td>
<td>[0.91  0.09]</td>
<td>[0.71  0.29]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.54  0.46]</td>
<td>[0.74  0.26]</td>
<td></td>
</tr>
<tr>
<td>3.31</td>
<td>Students “learning”</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.00  1.00]</td>
<td>[1.00  0.00]</td>
<td>[0.75  0.25]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.28  0.72]</td>
<td>[0.00  1.00]</td>
<td></td>
</tr>
<tr>
<td>3.32</td>
<td>Combined</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
<td>$K$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.13  0.87]</td>
<td>[0.86  0.14]</td>
<td>[0.58  0.42]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
<td>$U$  $U$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0.38  0.62]</td>
<td>[0.39  0.61]</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Summary of example hidden Markov model parameters.
The Viterbi algorithm gives insight into individual student performance by uncovering their hidden performance state. It is the combination of both results that gives way to effective student performance analysis through the hidden Markov model.
Chapter 4: Individualized instruction assessment

One goal of this paper is to determine how to effectively assess student performance in a computer-based individualized instruction mathematics environment. As seen in Chapter 2, some students are able to answer the questions correctly as assessed by the Individually Prescribed Instruction Mathematics and Khan Academy platforms; however, the literature suggests that the students actually have not learned the fundamental concepts [7, 30]. In this chapter it will be proposed that the Baum-Welch algorithm and Viterbi algorithm be used to identify such behavior within the system. The Viterbi algorithm can first determine a sequence of knowledge states of unkowning and knowing for an example student who may be pattern matching or memorizing answers, and then match the students’ sequences to this particular sequence to identify a potential student who may not completely understand the fundamental concepts the problem wishes to bring onto the student.

4.1 Viterbi algorithm

The Viterbi algorithm is used to estimate the sequence of hidden states with the sequence of observations known. In order to apply the algorithm we must know the parameters of the hidden Markov model $\lambda^* = (\mu^*, T^*, \text{and } E^*)$. These probabilities are determined beforehand through the Baum-Welch algorithm.
The Viterbi algorithm takes a single sequence of observations and parameters $\lambda^*$ to determine the most likely sequence of hidden states in a hidden Markov model, called the Viterbi path, that would produce such a sequence of observations. The algorithm is iterative in nature. The first step is to determine the hidden state that maximizes the probability of the first hidden state given the first observation. Once that state has been determined, the second hidden state is found that maximizes the probability of seeing the first two observations. The first hidden state determined prior is used in the maximization of the probability in the hidden state sequence of the first two states.

Algorithm 4.1 (Viterbi Algorithm).

1. Define the following beforehand:

   **Possible hidden states** $\{x_1, x_2, \ldots, x_k\}$, $k$ possible hidden states.

   **Possible observations** $\{y_1, y_2, \ldots, y_m\}$, $m$ possible observations

   **Sequence of hidden states** $\{X_1, \ldots, X_n\}$, where $X_t \in \{x_1, \ldots, x_k\}$ is the $t^{th}$ hidden state in the sequence for $t = 1, \ldots, n$.

   **Sequence of observed data** $Y = \{Y_1, \ldots, Y_n\}$ and $Y_t \in \{y_1, \ldots, y_n\}$ is the $t^{th}$ observation in the sequence of $n$ observations.

   **Final initial probabilities of hidden states** $\mu^*$ is a result of the Baum-Welch algorithm.

   $$\begin{bmatrix} \mu^*_1 \\ \mu^*_2 \\ \cdots \\ \mu^*_k \end{bmatrix}.$$
Final transition probabilities $T^*$, a $k \times k$ matrix that is a result of the 
Baum-Welch algorithm where

$$T^* = [q_{ij}^*]$$

$$= [P(X_t = x_i | X_{t-1} = x_j)].$$

Final emission probabilities $E^*$, a $k \times m$ matrix this is a result of the Baum-
Welch algorithm where

$$E^* = [e^*_i(y_j)]$$

$$= [P(Y_t = y_j | X_t = x_i)].$$

2. Variables used within algorithm

Partial probabilities $\delta_t(x_i)$ is the maximum probability of reaching hidden 
state $x_i$ at time $t$.

Maximal states $\phi_t(x_{it}) = x^*_{it-1}$. For hidden state $x_{it}$ at time $t$, this represents 
The hidden state $x^*_{it-1}$ at time $t - 1$ that produces $\delta_t(x_{it})$ where $x^*_{it} \in 
\{x_1, \ldots, x_k\}$ represents the $t^{th}$ hidden state in the Viterbi path that ends 
at hidden state $x_i$.

Viterbi path $\phi(x_i) = \{x^*_{i1}, \ldots, x^*_{in}\}$. This sequence represents the sequence 
of hidden states produced by the Viterbi algorithm that ends at state $x_i$.

3. Goal: Find the sequence of hidden states that maximizes the probability

$$P(Y_1 = y_1 \cap \cdots \cap Y_n = y_n | X_1 = x_1 \cap \cdots \cap X_n = x_n).$$
By using Definition 3.7, this is equivalent to finding the sequence of hidden states that maximizes the probability

\[
P(Y_1 = y_1 \cap \cdots \cap Y_n = y_n \cap X_1 = x_1 \cap \cdots \cap X_n = x_n).
\]

4. Calculate initial partial probability for state \(x_i\).

\[
\delta_1(x_i) = \mu_i \cdot e_i(y_1).
\]

5. Calculate the partial probability at each time \(t = 1, \ldots, n\) and for each state \(x_i\).

\[
\delta_t(x_i) = \max_{1 \leq j \leq k} \{\delta_{t-1}(x_j) \cdot q_{ji} \cdot e_i(y_t)\}.
\]

6. For each state \(\delta_t(x_i)\) at time \(t\), record the state \(x_j^*\) at time \(t-1\) that produced the maximum value. The hidden state \(x_j^*\) at \(t-1\) that resulted in the maximum partial probability sequence is set equal to \(\phi_t(x_i)\).

\[
\phi_t(x_i) = \arg \max_{1 \leq j \leq k} \{\delta_{t-1}(x_j) \cdot q_{ji} \cdot e_i(y_t)\} = x_j^*.
\]

7. At time \(t = n\) determine the state with the largest partial probability. Denote this as \(x_n^*\).

\[
x_n^* = \arg \max_{1 \leq i \leq k} \{\delta_n(x_i)\}.
\]

8. For each time starting with \(t = n\), work backwards until the state \(x_1^*\) has been reached by iteration. The construction of a trellis diagram is useful.

\[
x_{t-1}^* = \phi_t(x_t^*).
\]

9. The sequence \(\phi(x_n^*) = \{x_1^*, \ldots, x_n^*\}\) is the **Viterbi path**.
The Viterbi algorithm is usually accompanied by a trellis diagram to help determine the Viterbi path. An example if a trellis diagram is provided in Example 4.2.

Example 4.2 (Trellis diagram). Consider a hidden Markov model with four possible hidden states \( \{x_1, x_2, x_3, x_4\} \), two possible observations \( \{y_1, y_2\} \), and a sequence with three observed values \( \{Y_1 = y_1, Y_2 = y_2, Y_3 = y_1\} \). The goal of the Viterbi algorithm is to determine the sequence of hidden states \( \{X_1, X_2, X_3\} \).

Figure 4.1 shows a trellis diagram for a hidden Markov model under the sequence of observations. Connections are made between each pair of hidden states in the time sequence since it is possible to move between any two hidden states. One possible sequence of hidden states is outlined in Figure 4.1: \( \{X_1 = x_2, X_2 = x_1, X_3 = x_1\} \).

The partial probability \( \delta_t(x_i) \) is computed for every \( t = 1, 2, 3 \) and \( i = 1, 2, 3, 4 \). The previous state that yielded the largest such \( \delta_t(x_i) \) for \( x_i \) is set equal to \( \phi_t(x_i) \).
For example suppose we want to determine $\phi_2(x_1)$, which is the state at time $t = 1$ that produces the largest $\delta_2(x_1)$. Figure 4.2 outlines the possible connections between all hidden states at time $t = 1$ to hidden state $x_1$ at time $t = 2$. Suppose that

$$\delta_2(x_1) = \max_{1 \leq j \leq 4} \{ \delta_1(x_j) \cdot q_{j1} \cdot e_2(y_2) \}$$

$$= \delta_1(x_2) \cdot q_{21} \cdot e_2(y_2).$$

This means that state $x_2$ at time $t = 1$ produced the largest possible value $\delta_2(x_1)$ for hidden state $x_1$. Thus $\phi_2(x_1) = x_2$. These calculations are done for all states at $t = 2$ and $t = 3$.

$$t = 1 \quad t = 2 \quad t = 3$$

$\begin{align*}
  x_1 & \quad x_1 & \quad x_1 \\
  x_2 & \quad x_2 & \quad x_2 \\
  x_3 & \quad x_3 & \quad x_3 \\
  x_4 & \quad x_4 & \quad x_4 \\
\end{align*}$

$Y_1 = y_1 \quad Y_2 = y_2 \quad Y_3 = y_1$

Figure 4.2: Determination of one state at $t = 2$ in the trellis diagram.

At the final time $t = 3$ the most likely final state is determined by looking for the largest value of $\delta_3(x_i)$. Suppose

$$x_1 = \arg \max_{1 \leq i \leq 4} \{ \delta_3(x_i) \}.$$ 

This means that hidden state $x_1$ is the most likely ending state. This is the last state in the Viterbi path and our goal is to determine $\phi(x_1)$. From this ending state $x_1$ the
most likely previous state is determined by finding the state at \( t = 2 \) that produces \( \delta_3(x_1) \). Suppose that

\[
\delta_3(x_1) = \max_{1 \leq j \leq 4} \{\delta_2(x_j) \cdot q_{j1} \cdot e_3(y_1)\}
\]

\[
= \delta_1(x_1) \cdot q_{11} \cdot e_3(y_1).
\]

This means that \( x_1 \) at time \( t = 2 \) is the most likely state at \( t = 2 \) of the Viterbi path. From the calculations above we also saw that \( x_2 \) at time \( t = 1 \) is the most likely state to start the Viterbi path. Thus the Viterbi path is defined to be

\[
\phi(x_1) = \{X_1 = x_2, X_2 = x_1, X_3 = x_1\}
\]

shown in Figure 4.1.

**Example 4.3 (Viterbi Algorithm).** Let’s continue example 3.27 of a student attempting problems for a single exercise in an individualized instruction system. For the Viterbi algorithm we must have a predetermined set of initial, transition, and emission probabilities to actually use the algorithm. These parameters come from the Baum-Welch algorithm. The Viterbi algorithm will take one specific sequence of observations and determine the sequence of hidden states whose probability is maximized under the final parameter set.

1. Define the following beforehand:

   **Possible hidden states** Knowing (K) or Unknowing (U): \( \{x_1, x_2\} = \{K, U\} \),
   
   \( k = 2 \) possible hidden states.

   **Possible observations** Correct (C) or Incorrect (I), \( m = 2 \) possible observations.
Sequence of observed data

\[ Y^{(2)} = \{C, I, I, C\} \]

Sequence of hidden states \( \{X_1, \ldots, X_4\} \) where \( X_t \in \{K, U\} \).

**Initial probabilities** The initial probabilities of hidden states \( \mu \) are

\[
\begin{pmatrix}
K & U \\
0 & 1
\end{pmatrix}.
\]

**Transition probabilities** The transition probability matrix \( T \) is

\[
\begin{pmatrix}
K & U \\
K & [1.00 \ 0.00] \\
U & [0.64 \ 0.36]
\end{pmatrix}.
\]

**Emission probabilities** The emission probability matrix \( E \) is

\[
\begin{pmatrix}
C & I \\
K & [0.74 \ 0.26] \\
U & [0.34 \ 0.66]
\end{pmatrix}.
\]

2. Variables used within algorithm

**Partial probabilities** \( \delta_t(K) \) and \( \delta_t(U) \) are the maximum probability of being at hidden state \( K \) and \( U \) respectively at time \( t = 1, 2, 3, 4 \).

**Maximal states** \( \phi_t(x_{i_t}) = x_{t-1}^* \). For hidden state \( x_{i_t} \) at time \( t \), this represents the hidden state \( x_{t-1}^* \) at time \( t-1 \) that produces \( \delta_t(x_{i_t}) \) where \( x_{i_t}^* \in \{K, U\} \) represents the \( t \)th hidden state in the Viterbi path that ends at hidden state \( x_i \).

**Viterbi path** \( \phi(x_i) = \{x_{i_1}^*, \ldots, x_{i_n}^*\} \). This sequence represents the sequence of hidden states produced by the Viterbi algorithm that ends at state \( x_i \in \{K, U\} \).
3. Goal: Find the sequence of hidden states \( \{X_1, X_2, X_3, X_4\} \) that maximizes the probability

\[
P(Y_1 = C \cap Y_2 = I \cap Y_3 = I \cap Y_4 = C \cap X_1 = x_1 \cap X_2 = x_2 \cap X_3 = x_3 \cap X_4 = x_4)
\]

where \( x_t \in \{K, U\} \) for \( t = 1, 2, 3, 4 \).

By using Definition 3.7, this is equivalent to finding the sequence of hidden states that maximizes the probability

\[
P(Y_1 = y_1 \cap \cdots \cap Y_n = y_n \cap X_1 = x_1 \cap \cdots \cap X_n = x_n).
\]

4. Calculate initial partial probability for each hidden state,

\[
\delta_1(K) = \mu_1 \cdot e_K(C)
\]

\[
= (0)(0.74)
\]

\[
= 0.00
\]

\[
\delta_1(U) = \mu_2 \cdot e_U(C)
\]

\[
= (1)(0.34)
\]

\[
= 0.34.
\]

94
5. Calculate the partial probability at each time $t = 2, 3, 4$ and for each hidden state $\{K, U\}$,

$$
\delta_t(K) = \max_{x_j \in \{K, U\}} \{\delta_{t-1}(x_j) \cdot q_{jK} \cdot e_K(Y_t)\}
$$

$$
\delta_t(U) = \max_{x_j \in \{K, U\}} \{\delta_{t-1}(x_j) \cdot q_{jU} \cdot e_U(Y_t)\}
$$

$$
\delta_2(K) = \max\{\delta_1(K)q_{KK}e_K(I), \delta_1(U)q_{UK}e_K(I)\}
= \max\{(0.00)(1.00)(0.26), (0.34)(0.64)(0.26)\}
= \max\{0.0000, 0.0566\}
= 0.0566
$$

$$
\delta_2(U) = \max\{\delta_1(K)q_{KU}e_U(I), \delta_1(U)q_{UU}e_U(I)\}
= \max\{(0.00)(0.00)(0.66), (0.34)(0.36)(0.66)\}
= \max\{0.0000, 0.0808\}
= 0.0808
$$

$$
\delta_3(K) = \max\{\delta_2(K)q_{KK}e_K(I), \delta_2(U)q_{UK}e_K(I)\}
= \max\{(0.0566)(1.00)(0.26), (0.0808)(0.64)(0.26)\}
= \max\{0.0147, 0.0134\}
= 0.0147
$$

$$
\delta_3(U) = \max\{\delta_2(K)q_{KU}e_U(I), \delta_2(U)q_{UU}e_U(I)\}
= \max\{(0.0566)(0.00)(0.66), (0.0808)(0.36)(0.66)\}
= \max\{0.0000, 0.0192\}
= 0.0192
$$
\[ \delta_4(K) = \max\{\delta_3(K)q_{KK}e_K(C), \delta_3(U)q_{UK}e_K(C)\} \]
\[ = \max\{(0.0147)(1.00)(0.74), (0.0192)(0.64)(0.74)\} \]
\[ = \max\{0.0109, 0.0091\} \]
\[ = 0.0109 \]

\[ \delta_4(U) = \max\{\delta_2(K)q_{KU}e_U(C), \delta_2(U)q_{UU}e_U(C)\} \]
\[ = \max\{(0.0147)(0.00)(0.34), (0.0192)(0.36)(0.34)\} \]
\[ = \max\{0.0000, 0.0023\} \]
\[ = 0.0023 \]

6. Determine the state \( \phi_t(x_i) = x_{i_{t-1}}^* \) that produced the value \( \delta_t(x_i) \) for each \( x_i \) for \( t = 2, 3, 4 \).

- \( t = 2 \)

\[ \delta_2(K) = \max\{0.0000, 0.0566\} \]
\[ = 0.0566 \]

\[ \phi_2(K) = U \]

\[ \delta_2(U) = \max\{0.0000, 0.0808\} \]
\[ = 0.0808 \]

\[ \phi_2(U) = U \]

Therefore at time \( t = 2 \) for both \( K \) and \( U \) the transition must have come from state \( U \) at time \( t = 1 \).
• $t = 3$

\[
\delta_3(K) = \max\{K, U\} = \max\{0.0147, 0.0134\} = 0.0147
\]

\[
\phi_3(K) = K
\]

\[
\delta_3(U) = \max\{K, U\} = \max\{0.0000, 0.0192\} = 0.0192
\]

\[
\phi_3(U) = U
\]

Therefore at time $t = 3$ hidden state $K$ must have come from hidden state $K$ at time $t = 2$ and at time $t = 3$ hidden state $U$ must have come from hidden state $U$ at time $t = 2$.

• $t = 4$

\[
\delta_4(K) = \max\{K, U\} = \max\{0.0109, 0.0091\} = 0.0109
\]

\[
\phi_4(K) = K
\]

\[
\delta_4(U) = \max\{K, U\} = \max\{0.0000, 0.0023\} = 0.0023
\]

\[
\phi_4(U) = U
\]

Therefore at time $t = 4$ hidden state $K$ must have come from hidden state $K$ at time $t = 3$ and at time $t = 4$ hidden state $U$ must have come from hidden state $U$ at time $t = 3$. 
7. At time \( t = 4 \) determine the state with the largest partial probability, \( x_4^* \).

\[
x_4^* = \text{arg max}\{\delta_4(K), \delta_4(U)\}
\]

\[
= \text{arg max}\{0.0109, 0.0023\}
\]

\[
= K
\]

Thus the final state of the Viterbi path is \( x_4^* = K \).

8. Starting with \( t = 4 \) work backwards until the state \( x_1^* \) has been reached through iteration,

\[
x_3^* = \phi_4(x_4^*)
\]

\[
= \phi_4(K)
\]

\[
= K
\]

\[
x_2^* = \phi_3(x_3^*)
\]

\[
= \phi_3(K)
\]

\[
= K
\]

\[
x_1^* = \phi_2(x_2^*)
\]

\[
= \phi_2(K)
\]

\[
= U.
\]

The trellis diagram for this example is provided in Figure 4.3.

9. The sequence \( \phi(K) = \{U, K, K, K\} \) is the Viterbi path for the observation \( Y^{(2)} = \{C, I, I, C\} \).
Steps 1–9 can also be applied to the sequence $Y^{(1)} = \{I, C, C\}$. The Viterbi path for this sequence is $\phi(K) = \{U, K, K\}$.

This example highlights how the Viterbi algorithm determines the hidden states associated with an observable sequence of a hidden Markov model. It’s an algorithm that is very sensitive to the parameters $\lambda^*$ that come from the Baum-Welch algorithm. If the transition or emission probabilities vary, this will result in a different determination of a hidden state at a certain time, thereby leading to a different Viterbi path.

From this point on any calculations will be done in the statistical software R [29]. We present several examples demonstrating the Viterbi algorithm applied to various hidden Markov models that represent student performance data.

4.1.1 Viterbi algorithm in assessment of students

By using the Viterbi algorithm the hidden knowledge time-series data of each individual can be determined at each observation. Insights into how the student is learning can be made through these hidden states. Both the number of observations
seen for a single exercise, along with the time at which the student switches from an unknowing state to a knowing state.

To showcase how the Viterbi algorithm can assess student performance let’s revisit example 3.31, in which the hidden Markov model parameters were calculated for a sample of student data where learning was observed.

**Example 4.4.** Suppose that the majority of students followed a learning-like pattern in which they initially were making incorrect attempts, but quickly made improvement and began making correct attempts. The final parameters from the Baum-Welch algorithm in example 3.31 were the following.

**Final initial probabilities** The final initial probabilities of hidden states $\mu^{(*)}$ are

$$
K \begin{bmatrix} 0.00 & 1.00 \end{bmatrix}.
$$

**Final transition probabilities** The final transition probability matrix $T^*$ is

$$
K \begin{bmatrix} 1.00 & 0.00 \\ 0.28 & 0.72 \end{bmatrix}.
$$

**Final emission probabilities** The final emission probability matrix $E^*$ is

$$
C \begin{bmatrix} 0.75 & 0.25 \\ 0.00 & 1.00 \end{bmatrix}.
$$

These parameters would become the inputs to the Viterbi algorithm for any given observation. The Viterbi algorithm would determine the hidden states, knowing $K$ and unknowing $U$, which would be the most probable for a set of observations.

One student’s observations were the following sequence outlined in table 4.1.
Table 4.1: Sample student data used in determining Baum-Welch parameters

Using the final Baum-Welch parameters, the Viterbi algorithm generates the following sequence of hidden states, shown in table 4.2.

Table 4.2: Viterbi algorithm applied to sample student data

This student transitions from the unknowing state to the knowing state at attempt #7, which is the first occurrence of a correct attempt. Based on these parameters the instance of a correct attempt is highly favorable toward the knowing state as seen in the emission probabilities. Nearly all of the students that were used to generate the Baum-Welch parameters have a sequence of hidden states that is always unknowing at first for the first few hidden states, and then immediately transitions to the knowing state without transitioning back to the unknowing state. The few students who do not follow this pattern will transition between unknowing and knowing at first, but then are always in the knowing state after at least half of the attempts have been used.
up. For example, a student with 16 attempts would always be in the knowing state after at least the 8th attempt.

Suppose there was another student who attempted this exercise, but instead of actually learning through working the problems this student was attempting to memorize the answers or pattern match, similar to the students discussed in chapter 2. Let’s give this student the following observations, shown in table 4.3.

<table>
<thead>
<tr>
<th>Time</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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observation</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observation</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>I</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

Table 4.3: Sample student data of a student memorizing answers

The data for this student was generated as follows.

1. Assume the student had 5 different types of problems to memorize the answer for. This would mean that the student would need $5 \times 4 = 20$ attempts to memorize all the answers.

2. The student initially saw the first 5 problems on the first 5 attempts. The student got all of them incorrect because they had not yet learned the answers for any problem.

3. For the next successive 15 attempts the student would first have 4 consecutive incorrect responses, and then a correct response. Next the student would have 3 consecutive incorrect responses, and then a correct response. This pattern continued until all 20 attempts had been used.
Using this data, the student’s hidden states uncovered by the Viterbi algorithm are given in table 4.4.

<table>
<thead>
<tr>
<th>Time</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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden state</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>K</td>
</tr>
<tr>
<td>Time</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Hidden state</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
</tr>
</tbody>
</table>

Table 4.4: Viterbi algorithm applied to memorizing student

There are a few points to include about this sequence of hidden states. Firstly, this student is primarily in the unknowing state for the first half of their attempts. This shows that the student does not understand the concepts initially, which most students would also fall into this category. However, the sequence is essentially the same structure as the first student.

Reconsidering the example above, one may question whether the first student’s sequence of hidden states (whose data was constructed for learning) is actually different from the second student’s sequence (whose data was constructed for memorizing). A point to make here is that the knowing state and the unknowing state may not be the only states involved with student learning in such a hidden Markov model. Could there be a transition state in between unknowing to knowing? Example 4.4 is extended to include three hidden states instead of two. This three state model is presented in example 4.5.

**Example 4.5.** In example 4.4 we assumed that there were two hidden states comprising the hidden Markov model: the knowing state and the unknowing state. Here
we include a third transition state between the unknowing and knowing state, denoted as an “emerging” state E. The following assumptions are made for the initial Baum-Welch parameters.

**Starting initial probabilities** The starting initial probabilities of hidden states $\mu^{(0,u)}$ are

$$
\begin{bmatrix}
K & E & U \\
0.1 & 0.3 & 0.6 \\
\end{bmatrix}.
$$

**Starting transition probabilities** The starting transition probabilities $T^{(0,u)}$ are

$$
\begin{bmatrix}
K & E & U \\
K & 0.8 & 0.2 & 0.0 \\
E & 0.3 & 0.4 & 0.3 \\
U & 0.0 & 0.5 & 0.5 \\
\end{bmatrix}.
$$

**Starting emission probabilities** The starting emission probabilities $E^{(0,u)}$ are

$$
\begin{bmatrix}
C & I \\
K & 0.7 & 0.3 \\
E & 0.5 & 0.5 \\
U & 0.3 & 0.7 \\
\end{bmatrix}.
$$

R is used to run the Baum-Welch algorithm to determine the hidden Markov model final parameters using the above parameters as the inputs to the algorithm. The same data set from examples 3.31 and 4.4 is used for this model as well. The final parameters are the following.

**Final initial probabilities** The final initial probabilities of hidden states $\mu^*$ are

$$
\begin{bmatrix}
K & E & U \\
0.00 & 0.00 & 1.00 \\
\end{bmatrix}.
$$
Final transition probabilities. The final transition probability matrix $T^*$ is

$$
\begin{bmatrix}
K & E & U \\
K & 1.00 & 0.00 & 0.00 \\
E & 0.20 & 0.80 & 0.00 \\
U & 0.00 & 0.33 & 0.67
\end{bmatrix}
$$

Final emission probabilities. The final emission probability matrix $E^*$ is

$$
\begin{bmatrix}
C & I \\
K & 0.94 & 0.06 \\
E & 0.44 & 0.56 \\
U & 0.00 & 1.00
\end{bmatrix}
$$

The final initial parameters indicate that all students begin in the unknowing state. The transition probabilities indicate that once students enter the knowing state, they stay in the knowing state. From the emerging state the student either stays in the emerging state or moves to the knowing state, while from the unknowing state the student either stays in the unknowing state or moves to the emerging state. The model confirms that the student must transition through the emerging state in order to move from the unknowing state to the knowing state. The final emission probabilities indicate that while a student is in the knowing state they will answer correctly with high probability. While in the emerging state the student will answer correctly or incorrectly with equal probability. While in the unknowing state the student will always be incorrect. The way the data was constructed supports all of this evidence since all students show signs of progressing from unknowing to knowing very quickly.

The Viterbi algorithm was performed on each of the two example students from example 4.4. The unknown states for each are shown in tables 4.5 and 4.6.
Table 4.5: Viterbi algorithm applied to sample student data with three hidden states

<table>
<thead>
<tr>
<th>Time</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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden state</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
</tr>
</tbody>
</table>

Excessive

Table 4.6: Viterbi algorithm applied to memorizing student with three hidden states

<table>
<thead>
<tr>
<th>Time</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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden state</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
</tr>
</tbody>
</table>

Examining the two sequences we see a noticeable difference between the hidden states. Table 4.5 displays the student in the unknowing state for about one-third of the total attempts, then in the emerging state for the next one-third of the total attempts, and finally in the knowing state for the last one-third of the total attempts. This exactly corresponds to the type of expected behavior from a student who is progressively learning a concept. The student begins in a state of not knowing, begins to make progress gradually until they build themselves up to the knowing state. Now focusing on to the student who appeared to be memorizing based on their data. As seen from their hidden states this student remains in the unknowing state for nearly one-half of the total attempts, and then remains in the emerging state for nearly the next one-half of the total attempts. The student does not enter into the
knowing state until the last two attempts. This is clearly indicative of a student who is learning at a much slower rate than the previous student.

As seen from example 4.5 the Viterbi algorithm clearly distinguished two types of students’ learning patterns through a model utilizing three hidden states: unknowing, emerging, and knowing. This stage in the second student’s development would require human interaction to help correct their behavior. In all, this algorithm effectively clusters students into several groups within the system, which an instructor can examine to make further decisions about how to improve the students’ learning. This may involve correcting student behavior individually, or even changing the exercise all together.

**Example 4.6.** Consider now a situation where all students are attempting to memorize the answers to the exercise, with data exactly as the data in table 4.3. A valid question would be: What would the parameters of such a hidden Markov model look like for this exercise? Could we differentiate from the final parameters in example 4.5. A sampling of 100 students with data exactly shown in table 4.3 was generated as input to the Baum-Welch algorithm. We initialized a three hidden state hidden Markov model with starting parameters from example 4.5.

**Final initial probabilities** The final initial probabilities of hidden states $\mu^*$ are

$$\begin{bmatrix} K & E & U \\
0.00 & 0.00 & 1.00 \end{bmatrix}.$$ 

**Final transition probabilities** The final transition probability matrix $T^*$ is

$$\begin{bmatrix} K & E & U \\
K & 1.00 & 0.00 & 0.00 \\
E & 0.30 & 0.70 & 0.00 \\
U & 0.00 & 0.28 & 0.72 \end{bmatrix}.$$
Final emission probabilities The final emission probability matrix $E^*$ is

\[
\begin{pmatrix}
C & I \\
K & 0.33 & 0.67 \\
E & 0.00 & 1.00 \\
U & 0.00 & 1.00
\end{pmatrix}.
\]

If we compare these parameters to those in example 4.5 we see that the initial probabilities and transition probabilities are very similar. However, there’s a significant difference between the emission probabilities. The parameters in example 4.5 indicate that a student in a knowing state would primarily get correct answers, a student in an emerging state would get questions correct half the time and incorrect half the time, and a student in the unknowing state would primarily be incorrect. However, the parameters listed above referring to students memorizing answers signify that students in the knowing state primarily answer questions incorrectly, and students in the emerging and unknowing states always answer incorrectly.

A follow up question to this would be: How do the students’ hidden states compare to example 4.5? The two sample students listed above are passed through the Viterbi algorithm to determine what the hidden states would be under this new set of parameters. These are outlined in tables 4.7 and 4.8.

<table>
<thead>
<tr>
<th>Time</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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden state</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>E</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden state</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7: Viterbi algorithm applied to sample student data under new parameters
Table 4.8: Viterbi algorithm applied to memorizing student under new parameters

<table>
<thead>
<tr>
<th>Time</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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden state</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>E</td>
<td>K</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden state</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>K</td>
</tr>
</tbody>
</table>

The hidden states for both students show a significant difference than the hidden states in example 4.5. The states in tables 4.7 and 4.8 indicate that the students are initially in the unknowing state, then transition to the emerging state for one attempt, and then stay in the knowing state for the remainder of their attempts. This is quite the contrary to the previously stated hidden states in example 4.5.

Let’s reason why these hidden states are different.

1. Firstly, the transition matrix does not allow a student from the knowing state to revert back into the emerging or unknowing states. The student will never leave the knowing state once they enter, even if it’s early in the sequence of attempts. Here the algorithm assumed both students entered the knowing state within the first 10 attempts.

2. Secondly, the first six observations are all incorrect. Since the student began in the unknowing state there’s no reason to believe that the student has moved into the emerging state during this time. Thus the student will remain in the emerging state until a correct response is seen. This correct response would trigger a move from the unknowing state to the emerging state, which may occur before the correct response is actually observed.
3. Thirdly, the emission matrix assumes that the student must be in the knowing state if a correct response is observed. In table 4.1 the student makes their first correct response on attempt #7. The Viterbi algorithm would want to move the student directly into the knowing state at attempt #7. In order to move into the knowing state, the student can either be in the knowing state or the emerging state at attempt #6. Since the student cannot transition directly from the unknowing state to the knowing state at any attempt, then this student must be in the emerging state at attempt #6.

To summarize, the hidden states determined by the Viterbi algorithm are able to be explained by the hidden Markov model parameters. What we can conclude from this observation is that we need both the hidden Markov model parameters determined by the Baum-Welch algorithm and the calculated hidden states determined by the Viterbi algorithm to effectively model student learning.
Chapter 5: Conclusions and further study

This paper has developed a way to assess both the students and the quality of the exercise found in an individualized instruction platform.

Our intent was to answer a few questions.

Question #1  How can we assess a specific exercise in an exercise platform?

Answer: Through the hidden Markov model we can understand how students generally perform on an exercise. If the experimental parameters can be matched to predefined model parameters, as was done in section 3.3. The Baum-Welch algorithm is used to determine these parameters.

Question #2  How can we measure student learning in this system?

Answer: For student assessment, the Viterbi algorithm must be used to uncover the hidden performance states for an individual student. Both the model parameters and the hidden states allow us to understand student performance for an exercise.

Question #3  Can we identify students who are not performing on a particular exercise as expected?

Answer: The Viterbi algorithm will uncover the hidden performance states of an individual student making multiple attempts a single exercise. In section
4.1.1 we showed examples where students could and could not be distinguished, all the while knowing that the observations were produced differently. Before using the Viterbi algorithm we must have an understanding of the parameters of the hidden Markov model first. For example, in Example 4.6 the parameters of the hidden Markov model indicated that the students as a whole showed serious misunderstandings with the exercise. It is strongly discouraged to run the Viterbi algorithm on such an exercise’s data. What this is telling us is that there could be a flaw with the exercise itself, rather than student misunderstandings. The most telling story of this is in Example 4.5. A student with data constructed to fit the profile of someone memorizing answers had a sequence of hidden states that did not meet the profile of a student who was learning. However, the only way to make this distinction is through observation of the hidden sequences. Future research is needed to find a way to have the computer systematically locate these students. We intend to research metrics on how to quantify student performance by using the hidden states.

We intend to continue this research in online exercise platforms through the following areas of further study.

1. *What is the ideal number of hidden states within the hidden Markov model?*

   Based on the examples presented in this paper the hidden Markov model with three hidden states modeled student performance better than a two state model. We did not propose more than three hidden states, but this is something that should be investigated. It is possible that different exercises have a different
number of hidden states. We intend to develop models with at least three hidden states using actual student observations to answer this question.

2. *How can the input parameters to the Baum-Welch algorithm be selected to optimize the algorithm?*

The *Baum-Welch algorithm* calculates the parameters $\lambda^*$ that achieve a local maximum in the probability of the observations. It’s possible that the parameters it converges to do not reach a global maximum in the probability. The input parameters $\lambda^{(0,\text{all})}$ are a major contributor to which local maximum the algorithm converges. More research is needed to address the selection of these parameters so that the final parameters from the algorithm represent the model as closely as possible.

3. *Can a higher-order hidden Markov model represent such an exercise platform?*

This paper only concerned a first-order *hidden Markov model*—one that satisfies the *Markov property*. A $k^{\text{th}}$ order *hidden Markov model* would mean that the current hidden state depends on the previous $k$ hidden states. We intend to investigate this question by developing higher-order models using actual student observations.

4. *Can the hidden Markov model be developed inside the system to provide real-time student assessment data?*

We recommended implementation outside the system in R because we are not as familiar with the software development side of an exercise system. However, having such real-time data is beneficial for both the student and the instructor. Students could monitor their own progress as seen by the system to better
gauge their own understanding of a particular topic. Instructors could have reports automatically generated of the entire class including students that the system believes need additional help. We plan to investigate how to incorporate this model into the system to produce such an environment. We also plan to investigate metrics to measure student performance to identify outlier students in the system.
Bibliography


