An Investigation of Knowledge Tracing Algorithms as Learner Simulators

Billy McKenna

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Computer Engineering, MAI

Supervisor: Prof. Vincent Wade

April 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Billy McKenna

April 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Billy McKenna

April 19, 2022

An Investigation of Knowledge Tracing Algorithms as Learner Simulators

Billy McKenna, Master of Computer Engineering, MAI

University of Dublin, Trinity College, 2022

Supervisor: Prof. Vincent Wade

The development and research of adaptive learning systems is greatly hindered by the requirement for significant amounts of learners to interact with the systems. Many adaptive learning systems are built using machine learning technologies and require a significant number of learners' interactions to train and evaluate the systems. The use of knowledge tracing algorithms as learner simulators offers a potential solution to this problem, in particular for recommendation systems that recommend questions to a learner according to how the learner responds to other questions. Knowledge tracing algorithms can be used as learner simulators to simulate learners' responses to questions. It is hoped that knowledge tracing algorithms can realistically simulate how learners respond to questions and reduce the number of real learners and responses per learner required to train and evaluate these adaptive learning recommendation systems.

Current research fails to evaluate knowledge tracing algorithms as learner simulators. Research focuses on evaluating the performance of knowledge tracing algorithms for knowledge tracing as opposed to learner simulation. It is currently unknown how accurately learner simulators can simulate the responses of learners to questions. This dissertation investigates the use of knowledge tracing algorithms as learner simulators to simulate the responses of learners to questions. The research seeks to discover how well knowledge tracing algorithms can perform as learner simulators.

In an extensive evaluation of knowledge tracing algorithms as learner simulators, it was found that the best performing learner simulators can simulate the responses of learners with an average accuracy of 76% for the two learner simulation tasks defined.

The two learner simulation tasks defined reflect the number of responses that would be required to be simulated during training and evaluation of these adaptive learning recommendation systems. The evaluation and analysis conducted also produced a number of other interesting findings including how the accuracy of learner simulators can be increased further, how the performance of knowledge tracing algorithms for knowledge tracing can inform us about their performance as learner simulators and which knowledge tracing algorithms are most suitable for the simulation of learners' responses.

Overall, the accuracy achieved by the best performing learner simulators is a promising indication that knowledge tracing algorithms have the ability to realistically simulate learners' responses to questions. It is hoped that this level of accuracy is suitable for reducing the number of real learners and responses per learner required for the training and evaluation of adaptive learning recommendation systems that recommend questions according to learners' responses.

Acknowledgments

I would like to extend my sincere gratitude to my supervisor, Prof. Vincent Wade, for his continued support and guidance throughout the course of this research project. I would also like to thank my friends and family for their support.

Billy McKenna

University of Dublin, Trinity College
April 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Personalized education is the tailoring of educational content to an individual's needs. It is a contrast to the traditional "one-size-fits-all" approach and offers great potential for improvements to educational experiences (U.S. Department of Education Office of Educational Technology (2010)).

Adaptive learning is a method of delivering personalized education. Adaptive learning aims to offer personalized learning experiences by adapting the learning experience according to information about the learner. Examples of learner information that can be used to adapt the learning experience include a learner's responses to questions, a learner's preferences or a learner's learning style (Xie et al. (2019)). Adaptive learning can be used to adapt the learning experience in different ways such as the provision of adaptive feedback (Awais Hassan et al. (2019)), the early detection of at-risk students (Wolff et al. (2014)) or the recommendation of content (Nurjanah (2016);Raghuveer et al. (2014)). The current ability and future potential of adaptive learning to improve educational experiences is evident in research. Learner performance has shown to be improved when adaptive feedback is provided to the learner (Awais Hassan et al. (2019)), students that complete problems adapted to their personal interests have been shown to solve the problems faster and more accurately (Walkington (2013)) and the recommendation of content according to different information known about the learner has been observed to have great potential (Raghuveer et al. (2014)).

Evidence has shown that adaptive learning is a promising method of improving educational experiences and is being used to transform education. However, a major problem hindering the development and evaluation of adaptive learning systems is the requirement for significant amounts of learners. Many of the adaptive learning systems are built using machine learning technologies. During training, the underlying machine learning models learn how to adapt the learning experiences by observing learner's behaviour from

interactions with the systems. Furthermore, in the evaluation of most adaptive learning systems many learners are also required. To fairly compare how learners are affected by a non-adaptive system and an adaptive system, or to compare how learners are affected by different adaptive systems, many learners must be sourced to interact with the systems.

The problem of machine learning technologies requiring significant amounts of data for training and testing has been addressed in other research areas with the generation of synthetic data. An example of a technology that can generate synthetic data for various research areas is a generative adversarial network (GANs) (Goodfellow et al. (2014)). GANs can create synthetic images expanding datasets for computer vision problems or generate synthetic text producing data for natural language processing applications (Pan et al. (2019)). However when it comes to the generation of synthetic data for adaptive learning, generating synthetic data becomes a very difficult task. As opposed to generating a static synthetic image or a static synthetic sentence, dynamic synthetic learner behaviour must be generated. The type of synthetic learner behaviour required to be generated also depends on the training and evaluation requirements of the adaptive learning system. For example, in order to train or evaluate an adaptive learning system it may need to observe whether a learner answers different questions correctly or incorrectly or which learning material a learner would indicate preferences for. As a result, the generation of synthetic data for adaptive learning is a very difficult task.

The use of knowledge tracing algorithms as learner simulators offers a potential solution in the generation of synthetic learner behaviour. Knowledge tracing algorithms can predict the response of a learner to a question given the responses of the learner to questions previously completed. As a result, knowledge tracing algorithms can be used to simulate learners' responses to questions. In the area of knowledge tracing, a learner's response is defined as an indication of whether a learner answers a question correctly or incorrectly. It is a Boolean variable. This dissertation uses the same definition for a learner's response. The term is used throughout this dissertation to refer to a Boolean indication of whether a learner answers a question correctly or incorrectly. As mentioned previously, there are many different types of adaptive learning systems and the learner behaviour required to be simulated for each can differ greatly. A popular adaptive learning system that requires learners' responses for training and evaluation is a recommendation system that recommends questions to a learner according to how the learner responds to other questions (Farrell (2020); Liu et al. (2019)). As a result, this dissertation focuses on the use of knowledge tracing algorithms as learner simulators for the training and evaluation of this type of adaptive learning system. The synthetic learner behaviour being considered is a learner's response to questions. While the potential for learner simulators to be designed and implemented to simulate different learner behaviours is acknowledged,

when referring to learner simulators or the use of knowledge tracing algorithms as learner simulators this dissertation is referring to software that can simulate learners' responses to questions.

It is hoped that the learner simulators being considered can realistically mimic how a real learner would respond to questions and be used to train and evaluate the recommendation systems similar to how a real learner would. The goal is not to eliminate the requirement for real learners in the training and evaluation of adaptive learning recommendation systems but to significantly reduce the number of real learners required and reduce the number of questions each real learner must respond to. Furthermore, through the use of knowledge tracing algorithms as learner simulators a learner's responses to many different sequences of questions can be observed independent of each other. The use of knowledge tracing algorithms as learner simulators offer a potential solution in the generation of synthetic learner behaviour for adaptive learning, in particular for the training and evaluation of adaptive learning recommendation systems that recommend questions according to learners' responses. If these learner simulators can realistically simulate this learner behaviour they could solve a major problem hindering the development and research of adaptive learning systems.

1.1 Motivation

The potential benefits of using knowledge tracing algorithms as learner simulators is evident. Knowledge tracing algorithms possess the ability to simulate the learner behaviour required to train and evaluate adaptive learning recommendation systems that recommend questions according to learners' responses. If the algorithms can realistically simulate learners' responses, they can significantly reduce the number of real learners required to train and evaluate these systems. However, currently there is little research into the use of knowledge tracing algorithms as learner simulators. Farrell (2020) and Liu et al. (2019) used knowledge tracing algorithms as learner simulators to train and evaluate adaptive learning recommendation systems that recommend questions according to learners' responses. However, their research does not investigate the accuracy of the learner simulators used. It is unknown how realistically the learner simulators used respond to questions in comparison to how real learners would respond. Their research relies on the current evaluation of knowledge tracing algorithms in literature.

Current evaluation of knowledge tracing algorithms evaluates their performance for a knowledge tracing task. A knowledge tracing task involves the modelling of a learner's knowledge state as they answer questions and predicting the next response of the learner as each question is answered. For the knowledge tracing task, the learner's response to

each question in a sequence of questions is predicted. As the learner's response to each question is predicted, the knowledge state of the learner is updated by observing the learner's true response to the question. As a result, a knowledge tracing task can be described as predicting the next response of a learner given their past responses. The recommendation systems being investigated require learners' responses to a sequence of questions during training and evaluation (Farrell, 2020; Liu et al., 2019). As a result, a knowledge tracing algorithm used as a learner simulator must be able to predict and simulate a learner's response to each question in a sequence of questions. However, the learner's true response to each question is not available. As a result, the knowledge state of the learner can not be updated as it is for a knowledge tracing task. The simulated responses must be used to update the knowledge state of the learner. Since the knowledge state of the learner will never be updated with the true responses of a learner, some initial knowledge state of a learner is required. The learner simulator would simulate responses according to how a learner with that knowledge state would respond to questions. Since a knowledge state is inferred from the observed responses of a learner, the responses of a learner to some questions are required. As a result, this dissertation defines a learner simulation task as simulating the responses of a learner to a sequence of questions given their past responses to questions. The performance of a knowledge tracing algorithm for a learner simulation task will be informative as to how realistically the algorithm can simulate learners' responses when training and evaluating the recommendation systems being considered. Evaluation and comparative analysis of knowledge tracing algorithms for knowledge tracing tasks is present in research (Farrell (2020);Gervet et al. (2020)). However, there is currently no evaluation of knowledge tracing algorithms for learner simulation tasks. As a result, it is unclear whether the results of research such as that produced by Farrell (2020) and Liu et al. (2019) would be similar to the results achieved if real learners were used. Furthermore, it is unclear whether knowledge tracing algorithms can realistically simulate how a real learner would respond to a sequence of questions. Investigating the performance of knowledge tracing algorithms as learner simulators is the first step in discovering whether they are a solution to the data problem facing adaptive learning.

1.2 Research Question

The problems outlined in the motivation inspired the research question of "How well do knowledge tracing algorithms perform as learner simulators?". [1]

This overall research question is tackled by addressing the following project objectives:

- *Investigate and carry out a literature review of state-of-the-art knowledge tracing algorithms and their potential use in simulating learners' responses.* This objective will review adaptive learning recommendation systems that recommend questions according to learners' responses, current state-of-the-art use of knowledge tracing algorithms to simulate learners' responses, state-of-the-art knowledge tracing algorithms, the evaluation methodologies used for knowledge tracing algorithms, the reported performance of knowledge tracing algorithms and the different properties and data requirements of knowledge tracing algorithms.

- *Conceive, design, and implement learner simulators using knowledge tracing algorithms and an experimental framework for the evaluation of the learner simulators.*

- *Conduct evaluation and comparative analysis of the knowledge tracing algorithms as learner simulators.*

1.3 Research Methodology

The research methodology followed for this research project was design-based research. This research methodology consists of the investigation of a problem, the development of a solution to the problem and the evaluation of the proposed solution. The problem this research project is addressing is the requirement for large amounts of learners' responses for the training and evaluation of adaptive learning recommendation systems that recommend questions according to learners' responses. The proposed solution to this problem is the use of knowledge tracing algorithms as learner simulators to simulate learners' responses. The solution is developed by designing and implementing learner simulators built using knowledge tracing algorithms. The method of evaluating this solution is to investigate how well the implemented knowledge tracing algorithms perform as learner simulators. This is achieved through the implementation of an evaluation framework and conducting evaluation and comparative analysis of the implemented learner simulators.

[1] In this dissertation, the performance of a knowledge tracing algorithm as a learner simulator refers to the simulation of learners' responses for the purpose of training and evaluating the adaptive learning recommendation systems being considered

1.4 Contributions

This research looks to contribute to the research area in several ways. It is hoped that the research conducted will provide insight into whether the use of knowledge tracing algorithms as learner simulators offers a solution to the data problem facing adaptive learning, allow for the results of past research to be better understood, allow for future research to be conducted, and offer inspiration for the generation of complex dynamic synthetic data in other research areas.

The evaluation of the performance of knowledge tracing algorithms as learner simulators will indicate whether they offer a solution to the data problem facing adaptive learning. It is hoped that the learner simulators can realistically simulate learners' responses and be used to train and evaluate the recommendation systems similar to how a real learner would. The goal is not to eliminate the requirement for real learners in the training and evaluation of adaptive learning systems but to significantly reduce the number of real learners required and reduce the number of questions each real learner must respond to. The performance of knowledge tracing algorithms for knowledge tracing tasks is a promising indication that they may perform well as learner simulators. However, it is expected that the performance of knowledge tracing algorithms will decrease as they are used as learner simulators. Given some number of past responses of a learner to questions, the algorithms are required to simulate the responses of the learner for a sequence of questions as opposed to a single question. Measuring the accuracy of knowledge tracing algorithms for learner simulation tasks will be informative of whether they can be used for the training and evaluation of adaptive learning recommendation systems that recommend questions according to learners' responses.

By evaluating the performance of knowledge tracing algorithms as learner simulators the results of past research such as that conducted by Farrell (2020) and Liu et al. (2019) will be better understood. Liu et al. (2019) used knowledge tracing algorithms as learner simulators to train and evaluate a number of adaptive learning recommendation systems. The results of their research demonstrated that the learner simulators learned more effectively using their recommendation system, CSEAL, than the other recommendation systems evaluated. In the research conducted by Farrell (2020), a recommendation system was trained and evaluated with a learner simulator. The results of this research showed the learner simulator learned more effectively when recommended questions by the author's recommendation system than it did from baseline recommendation strategies. In both cases, it is unknown how realistically the learner simulators used respond to questions in comparison to real learners. As a result, it is unknown whether the results of their research would be similar to the results achieved if real learners were used. By

evaluating the performance of knowledge tracing algorithms as learner simulators it will be better understood whether the results of their research would be similar to the results achieved if real learners were used. If knowledge tracing algorithms are found to simulate learners' responses with a high accuracy, we can assume the recommendation systems that allow learner simulators to learn more effectively are the recommendation systems that would allow a real learner to learn more effectively.

A third contribution of this research project is that it will allow for future research to be conducted. Once the accuracy of knowledge tracing algorithms as learner simulators is understood, further research can be conducted investigating what level of accuracy is suitable for the training of the adaptive learning recommendation systems being considered. It is likely that the learner simulators do not have to perfectly replicate how a real learner would have responded to a question in order to train an adaptive learning recommendation system. It is likely that the learner simulator instead must respond to questions in a similar manner to how real learners would. Understanding the performance of the knowledge tracing algorithms as learner simulators is the first step in discovering whether they are a solution to the data problem facing adaptive learning. Once the accuracy of the learner simulators is understood, the level of accuracy required can then be investigated.

Finally, conducting an investigation into the use of knowledge tracing algorithms as learner simulators may offer inspiration for the generation of complex dynamic synthetic data in other research areas. As mentioned previously, the availability of training and test data is a problem faced by machine learning technologies in many research areas. Some research areas such as computer vision and natural language processing solve the problem through the generation of static synthetic data Pan et al. (2019). However, generating synthetic data for human behaviour in dynamic adaptive environments, such as for adaptive learning, is much more difficult. Other research areas such as medicine, transport and e-commerce may be able to generate the synthetic behaviour of patients, commuters and consumers in a similar manner to how synthetic learner behaviour is being generated for adaptive learning.

1.5 Dissertation Outline

The next chapters of this dissertation are organised as follows:

- Chapter 2 presents the literature review of state-of-the-art knowledge tracing algorithms and their potential use in simulating learners' responses.
- Chapter 3 documents the design and implementation of the learner simulator and

the underlying knowledge tracing algorithms used.

- Chapter 4 documents the evaluation of the knowledge tracing algorithms as learner simulators. This chapter contains details of the datasets used for evaluation, the metrics used for evaluation, the experimental setup of each experiment, the results of each experiment and a discussion of the results.
- Chapter 5 presents the conclusions drawn from the research, identifies limitations of the research and discusses potential future work.

Chapter 2

State of the Art

2.1 Adaptive Learning Recommendation Systems

As mentioned previously, this dissertation focuses on adaptive learning recommendation systems that recommend questions according to learners' responses. The adaptive learning recommendation systems designed and implemented by Farrell (2020) and Liu et al. (2019) are examples of such systems.

Both systems use reinforcement learning to train an agent to recommend questions. Reinforcement learning is an example of a machine learning technology that requires a significant amount of learners' responses when used for adaptive learning recommendation systems that recommend questions according to learners' responses. Reinforcement learning consists of an intelligent agent learning a desired behaviour in an interactive environment. Through a trial-and-error process, an agent learns using feedback from its actions. This feedback takes the form of rewards assigned to the agent. The goal for the agent is to find a suitable policy that would maximize the total reward of the agent. A policy is a function that returns the actions the agent should take. Reinforcement learning is a machine learning technology that has seen great success in recent years in many areas such as gaming and robotics (Arulkumaran et al., 2017; Li, 2017). Reinforcement learning has been used to create AI that can outperform human players when playing games. An example of this is AlphaGO (Silver et al., 2016), the first computer program to defeat a human professional player in the game of Go. In the area of robotics, reinforcement learning is being used to enable a robot to learn from its actions in an environment to create its own control system (Levine et al., 2015, 2016). Reinforcement learning is hoped to achieve similar success in adaptive learning. As a result, it is being used for adaptive learning recommendation systems such as those implemented by Farrell (2020) and Liu et al. (2019). However as mentioned previously, it requires a significant amount

of learners' responses. In application areas such as gaming and robotics, reinforcement learning benefits from the availability of large amounts of dynamic data. The dynamic data is the feedback from the agent's actions in the interactive environment. In gaming, an agent can explore countless different moves and the game will respond according to the moves made. In robotics, a robot can make countless different movements and observe the response of the physical environment to its movements. However for the adaptive learning recommendation systems being considered, an agent needs to recommend many different questions and observe the response of learners to those questions. Sourcing lots of learners' responses to many different questions is very difficult.

Reinforcement learning has been applied to provide adaptive learning solutions such as the reinforcement learning system RILS proposed by Raghuveer et al. (2014). However, this system was built for a Massive Open Online Courses (MOOC) environment where there are a significant number of existing users. The significant number of users available to interact with the system would provide the required environment to train and evaluate the reinforcement learning agents. It is very difficult for developers and researchers to get access to this number of learners, in particular in the early stages of development of adaptive learning systems. Furthermore, the research conducted by Raghuveer et al. (2014) was still limited by the number of learners available for training and testing. Where data containing learners' responses to questions is available, the data is static. Datasets generally contain learner sequences. A learner sequence is a sequence of questions completed by a learner and the learner's responses to those questions. The data is static as it can't be updated to contain the responses of a learner to new questions. Since a reinforcement learning agent learns using a trial-and-error process, it would recommend many different questions to a learner. Its policy would be updated according to the responses of the learner. The response of the learner to any question recommended would be required. As a result, the static data cannot be directly used to train the reinforcement learning agents of the recommendation systems being considered. A means of generating learners' responses to different questions is required. As a result, the use of knowledge tracing algorithms as learner simulators is particularly beneficial for training and evaluating these recommendation systems built using reinforcement learning.

In terms of how these recommendation systems recommend questions, the system implemented by Farrell (2020) seeks to maximize the educational gain of a learner while reducing information overload. Maximizing the educational gain of a learner consists of improving their proficiency in skills in the course. Reducing information overload consists of reducing the number of questions the learner must complete. The goal of the recommendation system is to recommend the least number of questions required to best improve the performance of the learner in the course. The recommendation process is

separated into a pre-test, recommended questions and post-test. The pre-test is taken by the learner prior to the recommendation of questions. This allows the recommendation agent to evaluate the initial knowledge state of the learner. Once the pre-test is completed, the agent can recommend a question. The agent then analyses the response of the learner to the question. The agent continues to recommend questions until it decides the learner is ready to complete the post-test. The responses of the learner to the questions of the post-test allows the agent to evaluate the change in performance caused by the recommended questions being completed. The agent is rewarded for improvements in the learner's performance from pre-test to post-test. The agent is penalized for the number of questions it recommended. The recommendation system implemented by Liu et al. (2019) seeks to recommend learning paths that maximize learning effectiveness. A learning path is a sequence of question recommended to a learner. Similar to the recommendation system designed and implemented by Farrell (2020), the recommendation process is separated into a pre-test, recommended questions and post-test. The learning effectiveness is a measure of how the learner's score improved from the pre-test to the post-test. This system does not only recommend questions according to learners' responses but also according to the knowledge structure of the questions and skills in the course. However, since the knowledge structure is not updating as questions are being recommended this is not data that would need to be simulated. In both cases, the recommendation process consists of a pre-test, recommended questions and post-test. As a result, the training and evaluation of both systems requires learners' responses to the pre-test, recommendation questions and post-test.

2.2 Learner Simulators

As mentioned previously, the use of knowledge tracing algorithms as learner simulators offers a potential solution in simulating learners' responses for training and evaluating adaptive learning recommendation systems that recommend questions according to learners' responses. However, there is currently limited research in the area.

The research of Farrell (2020) investigated knowledge tracing algorithms and the use of knowledge tracing algorithms as learner simulators for training an adaptive learning recommendation system. As mentioned above, an adaptive learning recommendation system was implemented by Farrell (2020). As part of this research, a comparative analysis of knowledge tracing algorithms was conducted and a learner simulator was implemented using the knowledge tracing algorithm Bayesian Knowledge Tracing (BKT). BKT is discussed further in section 2.3. The implemented recommendation system was trained and evaluated using the learner simulator. The results of this research showed the learner

simulator learned more effectively when recommended questions by the author's recommendation system than it did from baseline recommendation strategies. While a learner simulator was implemented and used to train and evaluate an adaptive learning recommendation system, the performance of the learner simulator was not investigated. It is unknown how realistically the learner simulator used simulated the responses of learners. The comparative analysis of knowledge tracing algorithms conducted evaluated their performance for knowledge tracing as opposed to learner simulation. The research relies on the evaluation of knowledge tracing algorithms for knowledge tracing to give an indication of how realistically knowledge tracing algorithms can simulate learners' responses.

The challenge of training an adaptive learning recommendation system was acknowledge by Liu et al. (2019) in the implementation of their recommendation system. The authors solved the data problem by implementing two learner simulators. One simulator was rule based using Item Response Theory (IRT) (van der Linden and Hambleton, 1997). IRT assumes that the knowledge state of a learner is constant as they complete questions. In reality a learner's knowledge state evolves as they answer questions. As a result, assuming the knowledge state of a learner is constant as they answer questions would likely lead to unrealistic simulation of the learner's responses. The second simulator used the knowledge tracing algorithm Deep Knowledge Tracing (DKT) Piech et al. (2015). DKT is discussed further in section 2.3. Knowledge tracing algorithms model the change in the knowledge state of a learner as they complete questions. Modelling a learner's knowledge state as a dynamic variable that updates with the completion of each question is a more realistic representation than modelling a constant knowledge state. As a result, it would likely provide more realistic simulation of learners' responses. Despite the use of learner simulators to train and evaluate different recommendation systems in this research, the research offers a limited insight into the operation of the learner simulators and lacks evaluation of their performance as learner simulators. Again it is unknown how realistically the learner simulators used simulated learners' responses.

In both cases the research demonstrates the use of knowledge tracing algorithms as learner simulators to train and evaluate adaptive learning recommendation systems. However, the performance of the learner simulators used is not investigated. It is hoped that learner simulators can realistically simulate how a real learner would respond to questions and can be used to train and evaluate recommendation systems similar to how a real learner would. However, it is unknown how realistically the learner simulators used responded to questions in comparison to how real learners would respond. Their research relies on the evaluation of knowledge tracing algorithms for knowledge tracing tasks. As detailed in section 2.3.2 this method of evaluation does not necessarily provide an accurate indication of the performance of knowledge tracing algorithms as learner simulators. As

a result, it is unclear whether the results of the research would be similar to the results achieved if real learners were used. Furthermore, it is still unclear whether knowledge tracing algorithms can realistically simulate how a learner would respond to a sequence of questions.

2.3 Knowledge Tracing Algorithms

This section includes an overview of knowledge tracing algorithms, a description of the current evaluation methodologies used for knowledge tracing algorithms and detailed descriptions of the algorithms BKT and DKT. DKT, a variation of DKT and a variation of BKT are implemented as learner simulators for this research project. As a result, detailed descriptions of BKT and DKT are provided in this section. Further details on design choices are outlined in chapter 3.

2.3.1 Overview of Knowledge Tracing Algorithms

Knowledge tracing is the modelling of a learner's knowledge state as the learner completes questions. The knowledge state of a learner is a latent variable that is inferred from the observable variable, a learner's responses to questions. As mentioned previously, a learner's response refers to whether the learner answered a question correctly or incorrectly. Knowledge tracing algorithms are often used to predict the response of a learner to a question given past responses.

The first knowledge tracing algorithm was introduced by Corbett and Anderson (1994). The knowledge tracing algorithm introduced was BKT. Since the introduction of BKT in 1994, there has been lots of research conducted in the area of knowledge tracing and predicting learners' responses. This resulted in the implementation of many new algorithms as well as improved variants of the original BKT algorithm. BKT and its variants are often considered benchmark knowledge tracing algorithms. New algorithms are often compared with BKT or its variants. The initial advancements in knowledge tracing and predicting learners' responses consisted of what are commonly considered probabilistic or logistic algorithms. An example of a popular logistic algorithm that was implemented to compete with the original BKT algorithm was Performance Factor Analysis (PFA) introduced by Pavlik et al. (2009). The algorithm was introduced as an alternative to knowledge tracing for predicting learners' responses. Pavlik et al. (2009) reported that PFA slightly outperformed their implementation of BKT for the prediction of learners' responses to questions. However, future research such as the comparative analysis of the two algorithms conducted by Farrell (2020) suggests the contrary. BKT was found to outper-

form or have equal performance to PFA . Piech et al. (2015) sparked the beginning of the implementation of many deep learning models for knowledge tracing with the introduction of Deep Knowledge Tracing (DKT). This paper reported that DKT achieved a 25% gain in AUC over the benchmark knowledge tracing algorithm BKT. In the subsequent years following the introduction of DKT, variations of DKT and new deep knowledge tracing algorithms were implemented. When a new knowledge tracing algorithm is introduced, the authors of the newly introduced algorithm often report the algorithm they have introduced to outperform benchmark and state-of-the-art algorithms such as BKT and DKT. However, comparative analysis later conducted often finds performance differences to be overstated or even finds the current benchmark or state-of-the-art algorithms to outperform the newly introduced algorithm.

The comparative analysis conducted by Gervet et al. (2020) finds many of the performance differences between knowledge tracing algorithms in literature to be overstated. Furthermore, classical algorithms such as BKT were found to outperform the state-of-the-art algorithms for some datasets when newer variations were used. The particular focus of their research was an investigation of when deep knowledge tracing algorithms should be used as opposed to probabilistic or logistic algorithms. A variety of different algorithms were compared and their performance in predicting learners' responses was compared across nine different datasets. The general findings were that DKT peformed best on five of the nine datasets while a logistic regression model named Best-LR performed best on the remaining datasets. From the results it was concluded that logistic regression models perform best on smaller datasets as they underfit the larger datasets. Meanwhile, deep learning models performed best on large datasets as they overfit the smaller ones. Another interesting finding of this research was that Self-Attentive Knowledge Tracing (SAKT), a knowledge tracing algorithm presented by Pandey and Karypis (2019), was outperformed by DKT on all datasets. This was a contrast to the results reported in the paper introducing SAKT (Pandey and Karypis, 2019). The comparative evaluation conducted by Gervet et al. (2020) also found BKT to outperform DKT on two of the datasets, This is an interesting finding considering the introductory paper of DKT reported a 25% gain in AUC over BKT (Piech et al., 2015).

The work of Gervet et al. (2020) provides an adequate summary of the state-of-the-art knowledge tracing algorithms through their comparative analysis. However, there are several other recently invented knowledge tracing algorithms not evaluated in their research. Knowledge tracing algorithms introduced in the last two years include Context-Aware Attentive Knowledge Tracing (AKT) (Ghosh et al., 2020), Relation-Aware Self Attention for Knowledge Tracing (RKT) (Pandey and Srivastava, 2020), a variation of Separated Self-Attentive Neural Knowledge Tracing (SAINT+) (Shin et al., 2021), Deep

Knowledge Tracing with Transformers (DKTT) (Pu et al., 2020) and Deep Self-Attentive Knowledge Tracing DSAKT (Zeng et al., 2021). While these models were not included in the research conducted by Gervet et al. (2020), all of the models are compared with SAKT in their respective papers with results suggesting they outperform the model. Furthermore, all models except DKTT are compared to DKT with results suggesting they outperform DKT. None of these knowledge tracing algorithms from 2020 or 2021 are compared with each other. As a result, each of these algorithms could be considered state-of-the-art knowledge tracing algorithms. However, as seen with the comparative evaluation conducted by Gervet et al. (2020) it is often the case that the reported performance differences between new knowledge tracing algorithms and current state-of-the-art or benchmark algorithms can be overstated.

When considering the use of knowledge tracing algorithms as learner simulators there are a number of factors to be considered. Most knowledge tracing algorithms require a dataset containing the responses of learners to questions for training. The parameters of the knowledge tracing algorithms are calculated during training using the data. These parameter values are used to predict the response of a learner to questions. The knowledge tracing algorithms must be trained on the responses of learners to questions that they will be used to predict responses for. As a result, when used for learner simulation the knowledge tracing algorithms will need to be trained on learners' responses to questions in the course, i.e., questions that can be recommended. As mentioned previously, the goal of learner simulators is not to eliminate the requirement for real learners but to reduce the number of real learners required and to reduce the number of questions each real learner would be required to respond to. For example the responses of learners' in a particular class could be recorded, and the knowledge tracing algorithms could use this data to simulate responses for questions related to the same content. These simulated responses could be used to train a recommendation system that will be used to recommend that content in the near future. Each knowledge tracing algorithm also requires different data to be present in the dataset. Algorithms such as SAINT+ and DKTT use the time taken by a learner to answer a question when predicting the response of a learner. This is not suitable for a learner simulator as the learner simulator would be required to also simulate the time taken to answer the question. Many recent deep knowledge tracing algorithms also focus on improving the interpretability of DKT. This is not necessarily the most important feature of a knowledge tracing algorithm that is required to simulate learners' responses as realistically as possible.

2.3.2 Evaluation of Knowledge Tracing Algorithms

As mentioned previously, current research evaluates knowledge tracing algorithms for a knowledge tracing task. A knowledge tracing task involves the modelling of a learner's knowledge state as they complete questions and predicting the next response of the learner as each question is completed.

Knowledge tracing algorithms are trained and evaluated on a dataset containing learner sequences. As mentioned previously, a learner sequence is considered a sequence of questions answered by a learner and the learner's response to each question. The datasets are split into training and test sets, typically using a 80:20 partition. During training, the parameters of the knowledge tracing algorithms are calculated. The trained knowledge tracing algorithms are evaluated by predicting the responses of the learners present in the test to each question in their corresponding learner sequence, i.e., each question they answered. Predicting the learner's response consists of predicting the probability that the learner answers the question correctly. For each prediction, the algorithm observes all past questions and responses in the learner sequence and infers the knowledge state of the learner. The predicted responses to each question in each of the learner sequences are compared with the true responses. AUC is the metric generally calculated and reported in literature. AUC is a measure of how well the algorithm can distinguish between the two classes, correct and incorrect, when making predictions. To summarize the evaluation of knowledge tracing algorithms for a knowledge tracing task, a knowledge tracing algorithm predicts the probability that a learner's answer to the next question is correct given their past questions and responses. This evaluation methodology is used in the evaluation of knowledge tracing algorithms in literature.

The results achieved using this evaluation methodology are not necessarily an accurate indication of how knowledge tracing algorithms perform as learner simulators. During a knowledge tracing task, a knowledge tracing algorithm uses the true responses of the learner to all past questions to update the learner's knowledge state and predict a response. When simulating learners' responses for the training and evaluation of a recommendation system, the true responses of a learner to all past questions will not be present. As a learner simulator, the knowledge tracing algorithms are required to simulate the responses of a learner to questions that true responses of the learner are not available for. As a result, the knowledge state of the learner must be updated by simulated responses as opposed to true responses. This is the key difference between knowledge tracing and learner simulation. For knowledge tracing, an incorrectly predicted response will not have an effect on the next prediction made. For learner simulation, an incorrectly simulated response can have a negative effect on the future responses simulated. As a result, simulating learners'

responses is a much more difficult task than knowledge tracing. Since the knowledge state of the learner will never be updated with true responses as questions are answered, some initial knowledge state must be inferred for the learner and used to inform the simulations. This would take the form of a limited amount of a learner's responses to previously completed questions. This dissertation defines a learner simulation task as simulating the responses of a learner to a sequence of questions given their past responses to questions. The performance of a knowledge tracing algorithm for a learner simulation task will be informative as to how realistically the algorithm can simulate learners' responses when training and evaluating the recommendation systems being considered. Currently, there is no evaluation of knowledge tracing algorithms for learner simulation tasks present in literature.

2.3.3 Bayesian Knowledge Tracing

BKT models the latent knowledge state of the learner as a set of binary variables. Each variable represents whether a learner knows or does not know a particular skill. As a result, the learner has a knowledge state for each skill for the range of skills being examined. The knowledge state of the learner for each skill is considered to be independent of one another. A Hidden Markov Model (HMM) is used to update the set of binary variables using a learner's responses to questions related to skills. BKT assumes each question being completed by a learner has one skill associated with it.

The BKT algorithm has four model parameters:

- $p(L_0)$, the initial probability that the learner knows the skill.
- $p(G)$, the probability of the learner answering a question correctly despite not knowing the skill. This is considered a guess by the learner.
- $p(S)$, the probability of the learner answering a question incorrectly despite knowing the skill. This is considered a slip by the learner.
- $p(T)$ the probability the learner learns the skill having answered a question related to the skill.

These parameters must be set for each skill being considered. The parameters are often set using historical logs containing learners' responses to questions related to the skills being considered. Badrinath et al. (2021) offers a popular python library for Bayesian knowledge tracing algorithms. This library uses expectation maximization to fit the parameters from historical logs containing learners' responses.

Once the parameters are set for each skill, they can be used to update the knowledge state of the learner as the learner responds to questions related to the skills. Before the first question is completed by learner u, the probability of the learner learning the skill k is set to the value of $p(L_0)$ for skill k as seen in Equation 2.1. As each question t is answered by learner u, Equation 2.2, Equation 2.3 and Equation 2.4 are used to update the knowledge state of the learner. Equation 2.2 and Equation 2.3 use the observed response of the learner to the question. Having observed the responses of the learner to questions up to t, the probability that the learner answers the question $t+1$ correct is given by $p(C_{t+1})$ as seen in Equation 2.5. This allows for BKT to be used to predict the response of learner u to a question related to skill k given the past responses of learner u to questions related to skill k.

$$p(L_1)_u^k = p(L_0)^k \tag{2.1}$$

$$p(L_t|obs = correct)_u^k = \frac{p(L_t)_u^k \cdot (1 - p(S)^k)}{p(L_t)_u^k \cdot (1 - p(S)^k) + (1 - p(L_t)_u^k) \cdot p(G)^k} \tag{2.2}$$

$$p(L_t|obs = incorrect)_u^k = \frac{p(L_t)_u^k \cdot p(S)^k}{p(L_t)_u^k \cdot p(S)^k + (1 - p(L_t)_u^k) \cdot (1 - p(G)^k)} \tag{2.3}$$

$$p(L_{t+1})_u^k = p(L_t|obs)_u^k + (1 - p(L_t|obs)_u^k) \cdot p(T)^k \tag{2.4}$$

$$p(C_{t+1})_u^k = p(L_{t+1})_u^k \cdot (1 - p(S)^k) + (1 - p(L_{t+1})_u^k) \cdot p(G)^k \tag{2.5}$$

Some limitations of the BKT algorithm are that it requires each question being completed to be explicitly tagged with a knowledge state and the knowledge state of the learner for each skill is modelled independently. The explicit tagging of questions with skills can cause unknown relationships between questions and skills to not be modelled. Modelling the knowledge state of the learner for each skill independently can be an unrealistic modelling of knowledge states for learners in real world environments. It is often the case that the knowledge state of the learner for one skill is strongly related to another.

2.3.4 Deep Knowledge Tracing

DKT models the latent knowledge state of the learner "using large vectors of artificial 'neurons', and allows the latent variable representation of student knowledge to be learned from data rather than hard-coded" Piech et al. (2015, p. 1). A Recurrent Neural Network (RNN) is used by DKT to model the latent knowledge state of the learner. A RNN is a

type of artificial neural network used for sequential data, e.g., a sequence of questions and a learner's responses to those questions. Using an artificial network to model the latent knowledge state of the learner means DKT can model more complex representations of the learner's knowledge state in comparison to BKT. While BKT models the knowledge state of the learner for each skill independently, DKT models all knowledge states jointly.

The DKT algorithm can be used to predict the response of a learner to a question, given the responses of the learner to previously completed questions. For DKT the knowledge tracing task can be formalized as $p(a_{t+1} = 1|X_t, q_{t+1})$, namely predict the probability that the learner's answer to the next question a_{t+1} is correct given their past t interactions $X_t = (x_1, x_2, ..., x_t)$ and information about the next question q_{t+1}. Each past interaction $x_i = (q_i, a_i)$ is a tuple containing the question information and the learner's answer to the question where q_i is the question information relating to the ith question and a_i is the learner's answer to the ith question. The learner's answer to the ith question, a_i, is either 0 or 1 corresponding to a incorrect or correct answer. Question information q is typically represented by a question ID or a skill ID. When a question ID is used, DKT is predicting the response of the learner to a specific question. When a skill ID is used, DKT is predicting the response of the learner to a question related to a specific skill.

Chapter 3

Design & Implementation

3.1 Overview of the Approach

This research focuses on designing a learner simulator that is capable of using different knowledge tracing algorithms to simulate learners' responses. This learner simulator is designed to be appropriate for the training and evaluating of adaptive learning recommendation systems that recommend questions according to learners' responses.

As mentioned in section 2.1, the recommendation systems the learner simulator is being designed for recommend a sequence of questions and also provide the learner with some form of examination. Examples of these recommendation systems are the systems designed and implemented by Farrell (2020) and Liu et al. (2019). Both these systems provide the learner with a pre-test, recommended questions and post-test. In order to be trained and evaluated the systems require learners' responses to the pre-test, recommended questions and post-test.

The pre-test contains a learner's responses to questions that were completed before any questions have been recommended. It represents the initial knowledge state of the learner. As mentioned previously, a knowledge tracing algorithm used as a learner simulator requires an initial knowledge state for the learner. As a result, the decision was made that a learner's responses to the questions in the pre-test should be sourced and used to initialize the knowledge state of the learner simulator. The learner simulator can then simulate the responses of the learner to the recommended questions and post-test using this initial knowledge state. The responses of the learner to the pre-test can be manually configured or a record of a real learner's responses to the questions could be used. Manually configuring the learner's responses to the questions of the pre-test allows for the creation of a fictional learner. By manually selecting the responses of the learner to the questions, the initial knowledge state of the learner simulator is being configured for spe-

cific questions or skills. The learner simulator should respond to questions similar to how a learner with that knowledge state would. Alternatively, using the recorded responses of a real learner to the questions initializes the learner simulator with the knowledge state of that real learner. The learner simulator should then respond to questions similar to how that real learner would have.

3.2 Requirements of the Learner Simulator

Since different knowledge tracing algorithms are being investigated, the learner simulator must be configurable with different knowledge tracing algorithms. As mentioned above, the underlying knowledge tracing algorithm of the learner simulator requires an initial knowledge state of the learner in order to make predictions. As a result, the learner simulator requires a sequence of questions and the responses of a learner to those questions from which the initial knowledge state of the learner can be inferred, i.e., the pre-test. The knowledge state of a learner is dynamic and changes as a learner completes questions. As a result, a learner simulator must update the knowledge state of the learner as it simulates responses to questions.

3.3 Design

The learner simulator was designed according to the requirements listed above. Figure 3.1, displays an illustration of the inputs and outputs of the learner simulator.

In order to allow the learner simulator to be configured with different knowledge tracing algorithms, the first input of the learner simulator is a trained knowledge tracing algorithm. The knowledge tracing algorithm must be trained on data containing the responses of learners to questions or questions associated with skills that responses may need to be simulated for.

To allow for the learner simulator to contain an initial knowledge state of the learner that can be used by the knowledge tracing algorithm to make predictions, the second input is an initialization sequence. This initialization sequence is equivalent to the pre-test of the recommendation process. It is a sequence of questions and a learner's responses to the questions. As mentioned previously, the responses of the learner to these questions can be manually configured or a record of a learner's responses to a sequence of questions can be used.

The final input of the learner simulator is the simulation sequence. The simulation sequence consists of a sequence of questions that responses must be simulated for. This is equivalent to the recommended questions and the post-test of the recommendation

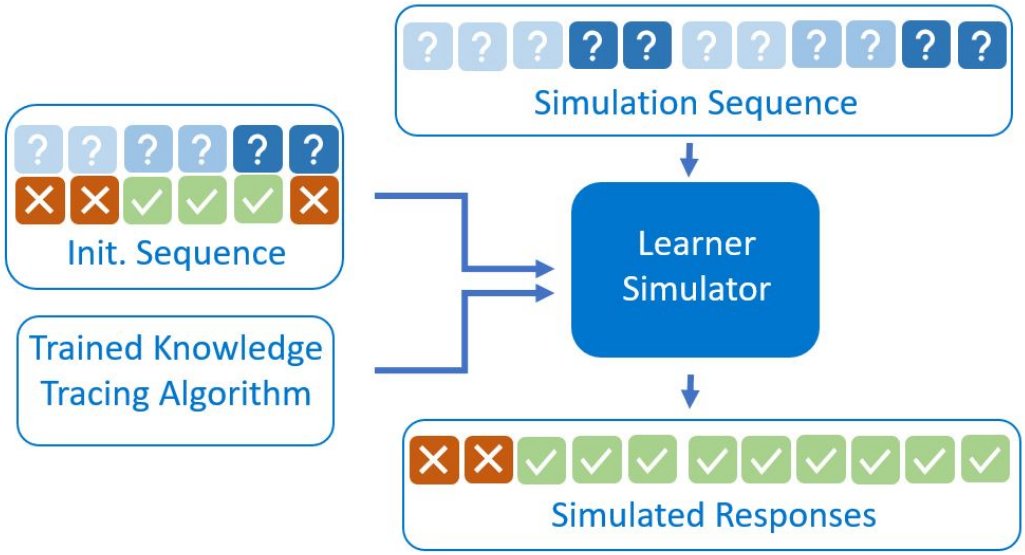


Figure 3.1: Illustration of the inputs and outputs of the learner simulator

process. Fig 3.1 depicts the learner simulator receiving a sequence of questions to be simulated. This assumes that the recommendation system is recommending a sequence of questions as opposed to recommending a single question, analyzing the response of the learner, and repeating the process. However, the learner simulator is still appropriate for simulating a response to a single question, outputting the simulated response and repeating the process.

The output of the learner simulator is a sequence of simulated responses to each question in the simulation sequence. The simulated response is either 1 or 0 corresponding to the learner answering the question correctly or incorrectly respectively.

3.3.1 Simulating Responses

The trained knowledge tracing algorithm is used to simulate the learner's response to each question. For each question in the simulation sequence the knowledge tracing algorithm receives the initialization sequence and the current question as inputs. The knowledge tracing algorithm infers the knowledge state of the learner from the initialization sequence and predicts the response of the learner to the current question. The predicted response is a value between 0 and 1 that represents the probability of whether the learner answered the question correctly. The predicted response is mapped to a simulated response by converting the prediction to a value of 0 or 1. A threshold of 0.5 is used for mapping the probability to a learner's response. For predictions greater than 0.5, the simulated

response is 1. For predictions less than or equal to 0.5, the simulated response is 0. The value of 0 indicates the learner answered the question incorrectly. A value of 1 indicates the learner answered the question correctly. As mentioned in the requirements, the learner simulator must update the knowledge state of the learner as it simulates responses to questions. As a result, the question and simulated response are appended to the initialization sequence updating the knowledge state of the learner. The process is repeated for each question in the simulation sequence.

3.4 Implementation

The learner simulator was implemented according to the outlined designs using Python. Three knowledge tracing algorithms were chosen to be evaluated as learner simulators for this research project. As a result, the learner simulator was implemented according to the operation of the knowledge tracing algorithms. Each knowledge tracing algorithm models the latent knowledge state of the learner differently and predicts the response of the learner to a question differently. The knowledge tracing algorithms chosen were Bayesian Knowledge Tracing with Forgetting (BKT+F), DKT and a modified version of DKT that has been named DKT+ for this dissertation.

3.4.1 BKT+F

BKT+F is a variation of BKT that includes an extra parameter $p(F)$, the probability that a learner forgets a skill they know. Once standard BKT infers that a learner has learned a skill, the inferred knowledge state will not be changed. Even if a long sequence of incorrect answers for questions related to a particular skill are observed, the learner's knowledge state will still be modelled as knowing the skill. However, when the forgetting parameter is included the knowledge state of the learner can move from the learner knowing the skill to the learner not knowing the skill. As a result, if the learner performs poorly for a skill after already learning it they can forget the skill. Furthermore, if the learner completes a number of question unrelated to a particular skill that skill can be forgotten by the learner. The forgetting parameter allows for more realistic modelling of a learner's knowledge state. Results of comparative evaluations show BKT+F outperforms standard BKT for knowledge tracing tasks (Farrell, 2020; Khajah et al., 2016). Since Bayesian knowledge tracing algorithms predict the learner's response to a question according to the skill associated with it, each question is required to be represented by a skill ID. Skill ID is the input of the algorithm in terms of previously completed questions and the question a response must be predicted for. As a result, BKT+F predicts the response of

a learner to a question related to a specific skill.

BKT+F was chosen to be investigated as a learner simulator because it is an improved variation of the benchmark knowledge tracing algorithm BKT. It offers a simplistic and easily interpreted method for modelling the knowledge state of a learner when compared with deep knowledge tracing algorithms. Furthermore, comparative analysis of knowledge tracing algorithms such as that conducted by Farrell (2020) and Gervet et al. (2020) have shown BKT and its variations are competitive with deep knowledge tracing algorithms when evaluated on certain datasets.

BKT+F was implemented using the python library pyBKT (Badrinath et al. (2021)). This library provides different variations of the BKT model that can be initialized and trained. The training of the model consists of using expectation maximization to fit the model parameters from inputted historical logs containing learners' responses as explained in section 2.3.3. The library also contains helper functions for evaluating the model and for making predictions. The pyBKT library is focused on providing the ability to train and test knowledge tracing algorithms for knowledge tracing tasks as opposed to learner simulation tasks. As a result, the code is not optimized for simulating responses. This results in slow simulation times as discussed later in section 4.1.5. This research was focused on the operation of the learner simulators and an investigation of their performance in terms of the accuracy of simulated responses. As a result, significant time was not spent enhancing the implementation of BKT+F to reduce simulation times.

3.4.2 DKT

As mentioned in section 2.3.4, the question information inputted into DKT to represent the previously completed questions and the question a response must be predicted for can be the question ID or the skill ID of the skill associated with the question. When question ID is used, DKT is predicting the response of the learner to a specific question. When a skill ID is used, DKT is predicting the response of the learner to a question related to a specific skill.

DKT was chosen to be investigated as a learner simulator as the comparative analysis of knowledge tracing algorithms conducted by Gervet et al. (2020) demonstrated it is a state-of-the-art knowledge tracing algorithm. It offers a complex method of modelling the knowledge state of the learner that allows for the relationships between different questions and skills to be learned by the algorithm. However, the modelling of the learner's knowledge state is not easy to interpret. This offers a contrast to the BKT+F algorithm.

The source code provided by Gervet et al. (2020) was used for DKT. [2] The DKT model

[2]https://github.com/theophilee/learner-performance-prediction

used for the learner simulator was trained using the default hyper-parameters provided, which were selected by the original authors through 5-fold nested cross validation. Their implementation of DKT was updated for the learner simulator.

3.4.3 DKT+

DKT+ is a variation of DKT implemented by Gervet et al. (2020). The modifications to standard DKT were inspired by the research of Lee and Yeung (2019). Their research proposed a new method for encoding the knowledge states and skills. While DKT represents the question with either the question ID or the skill ID, DKT+ represents the question with both the question ID and the skill ID. Both the question ID and the skill ID are inputted to the model for the previously completed questions and for the question a prediction must be made. As a result, DKT+ predicts the response of the learner to a specific question related to a specific skill.

DKT+ was chosen to be investigated as a learner simulator as it is unclear whether the results presented for DKT in the extensive comparative analysis of knowledge tracing algorithms conducted by Gervet et al. (2020) was for standard DKT or this variation. Furthermore, the research of Lee and Yeung (2019) suggests the new method for encoding the knowledge states and skills improves the performance of DKT.

The source code provided by Gervet et al. (2020) was used for DKT+. [2] The default hyper-parameters selected by the original authors through 5-fold nested cross validation were again used. Their implementation of DKT+ was updated for the learner simulator.

3.5 Summary

To summarize, a learner simulator capable of training and evaluating adaptive learning recommendation systems that recommend questions according to learners' responses was designed and implemented. The learner simulator can be configured with the knowledge tracing algorithms BKT+F, DKT or DKT+. The learner simulator receives an initialization sequence, equivalent to the pre-test, and simulates responses to each question in the simulation sequence, equivalent to a combination of the recommended questions and post-test. The initialization sequence can be manually configured or a historic record of a learner's responses to questions can be used.

Chapter 4

Evaluation

In the pursuit of answering the overall research question of, "How well do knowledge tracing algorithms perform as learner simulators?", four experiments were conducted corresponding to the following four evaluation questions:

- How well do knowledge tracing algorithms perform for knowledge tracing tasks?
- How well do knowledge tracing algorithms perform for learner simulation tasks?
- How does the length of the simulation sequence effect the performance of the learner simulators?
- How does the length of the initialization sequence effect the performance of the learner simulators?

The first experiment looks to answer the question of "How well do knowledge tracing algorithms perform for a knowledge tracing task?". As mentioned previously, the current evaluation of knowledge tracing algorithms in literature answers this question. These results are reproduced to allow for the analysis of the difference in performance between the implemented knowledge tracing algorithms for a knowledge tracing task. The difference in performance for a knowledge tracing task is compared with the difference in performance for learner simulation tasks. Comparing these results, it can be investigated whether the performance of knowledge tracing algorithms for a knowledge tracing task can be informative about their performance as learner simulators.

The second evaluation question looks to directly evaluate the performance of knowledge tracing algorithms as learner simulators. While knowledge tracing algorithms are currently evaluated in literature for knowledge tracing tasks, they are not evaluated for learner simulation tasks. In order to understand how realistically a knowledge tracing

algorithm can simulate the responses of learners, the knowledge tracing algorithms are evaluated for learner simulation tasks. This experiment evaluates the performance of the implemented learner simulators for two different learner simulation tasks.

The third evaluation question investigates how the performance of a learner simulator changes as the simulation sequence length changes. As mentioned previously, the simulation sequence contains the questions a learner simulator must simulate responses for. It is expected that the performance of the learner simulator will decrease as it is required to simulate more responses.

Finally, the effect of the initialization sequence length on the performance of the learner simulator is investigated. The initialization sequence contains the questions and responses used to initialize the learner simulator with an initial knowledge state. It is expected that the performance of the learner simulator will increase as the number of questions and responses used to initialize the knowledge state of the learner increases.

4.1 Datasets and Data Processing

To answer the four evaluation questions, experiments were run on datasets containing learner sequences. As mentioned previously, a learner sequence is a sequence of questions answered by a learner and the learner's responses to those questions. As outlined in section 3.4, the knowledge tracing algorithms implemented represent the questions with either the question ID, the skill ID of the skill associated with the question, or a combination of both. As a result, each dataset is required to have a question ID, skill ID, and response of the learner, 0 or 1, available for each question. Furthermore, datasets that are publicly available, used in the evaluation of knowledge tracing algorithms in literature and that cover a range of different subjects and learner types were considered requirements for the datasets used in this evaluation.

The evaluation of knowledge tracing algorithms in past research, such as that conducted by Piech et al. (2015), has produced results containing errors. The errors were caused by the data processing methods used. In particular, the errors were caused by the repetition of multi-skill questions in learner sequences. When a question had more than one skill associated with it, the question was repeated multiple times in a learner sequence with a single skill ID associated with it each time. The repetition of questions in a learner sequence results in improved performance for deep knowledge tracing algorithms such as DKT. Xiong et al. (2016) identified this problem and proposed a solution by generating a new skill for any combination of skills associated with a single question in the data. Gervet et al. (2020) adopted this solution in their evaluation of knowledge tracing algorithms. With this solution, a repeated question in the learner sequence associated with

multiple skills is replaced with a single question associated with the newly generated skill. While this solves the problem of removing repeated questions, it creates a new problem.

By introducing new skills to represent combinations of skills the data is being altered and information is being lost. The deep knowledge tracing algorithms, DKT and DKT+, may not suffer significantly from this loss of information as they have the ability to discover the underlying relationships between questions and skills. These models may be able to learn that the newly generated skill is closely related to each individual skill it is a combination of. As a result, they may be able to use the responses of the learner to questions associated with the individual skills to inform their prediction of the learner's response to a question associated with the newly generated skill. However, BKT+F would likely experience significant decreases in performance. As mentioned in section 2.3.3, Bayesian knowledge tracing algorithms model the knowledge state of the learner for each skill independently. When predicting a learner's response to a question associated with a skill, Bayesian knowledge tracing algorithms only use the responses of the learner to questions associated with that skill to inform their prediction. As a result, BKT+F would not be able to use the responses of the learner to questions associated with the individual skills to inform its prediction of the learner's response to a question associated with the newly generated skill. Generating new skills for the combination of skills removes valuable information for BKT+F that would cause a decrease in its performance. As a result, this solution to the multi-skill question problem was deemed unsatisfactory. In order to avoid the problem presented by multi-skill questions but faithfully maintain the structure of the data, datasets containing only single-skill questions were considered. As a result, each question in the three datasets selected is associated with a single skill.

4.1.1 ASSISTments 2015

This data was collected from the ASSISTments online learning system. The ASSISTments 2015 dataset contains the repsonses of grade school students across the United States of America to a variety of math questions. The dataset contains 14,228 learners, 100 different questions and 100 different skills. There are a total of 656,154 learners' responses to questions in the dataset.

4.1.2 Spanish 2013

The Spanish 2013 dataset was collected as part of the research conducted by Lindsey et al. (2014). The dataset contains the responses of middle school students in Colorado to a variety of Spanish exercises. The dataset contains 182 learners, 409 questions, 221 skills and 578,726 total responses.

4.1.3 Statics 2011

The data of the Statics 2011 dataset was collected from students enrolled in a statics course at Carneige Mellon University. The dataset contains the responses of the university students to statics questions. Statics is an engineering subject focused on the study of particles and rigid bodies that are in equilibrium. The dataset contains 282 learners, 1,223 questions, 98 skills and 189,297 learners' responses to questions.

4.1.4 Comparison of Datasets

Table 4.1 displays the number of learners, questions, skills and responses for each dataset. The ASSISTments 2015 dataset contains significantly more learners than Spanish 2013 and Statics 2011 as seen in Table 4.1. However as seen from the number of responses, each learner in the Spanish 2013 and Statics 2011 datasets responds to more questions.

Datasets	**Learners**	**Questions**	**Skills**	**Responses**
ASSIStments 2015	14,228	100	100	656,154
Spanish 2013	182	409	221	578,726
Statics 2011	282	1,223	98	189,297

Table 4.1: The number of learners, questions, skills and responses for each dataset

4.1.5 Data Processing

The data processing methods used by Gervet et al. (2020) were used for this evaluation. This data processing consisted of removing learners with less than 10 responses and removing responses to questions associated with skills labelled "NAN". The datasets were split into training and test sets according to a 80:20 partition.

One problem faced during experimentation was the slow run-time of BKT+F. BKT+F was implemented using the pyBKT library (Badrinath et al., 2021). For both the training and testing of the algorithm for knowledge tracing BKT+F was significantly slower than DKT and DKT+. The slow training and testing of BKT+F is likely due to the independent modelling of each learner's knowledge state for each skill in the dataset. BKT+F is essentially creating a model for each learner for each skill. Despite the slow training and testing times for knowledge tracing, evaluation of the algorithms could be carried out in a reasonable time. However, when BKT+F is applied to learner simulation tasks the time taken to test the algorithm is increased further. For each response to be simulated the underlying knowledge tracing algorithm must update the knowledge state with the simulated response. Current implementations of knowledge tracing algorithms, such as

those provided by the pyBKT library, are focused on providing the ability to train and test knowledge tracing algorithms for knowledge tracing tasks. As mentioned in section 3.4.1, the code is not optimized for learner simulation tasks. This research project was focused on the operation of the learner simulators and an investigation of their performance in terms of accuracy of simulated responses. Significant time was not spent optimizing the simulation times of the algorithms. As a result, evaluation of BKT+F for learner simulation tasks on the full test sets could not be carried out in reasonable times.

To address the slow simulation times of BKT+F and ensure experiments could be run in a reasonable time each experiment was run twice. The top five learner sequences of each test set for each of the knowledge tracing algorithms were extracted from the test sets. The top five learner sequences for a respective algorithm are the five learner sequences the algorithm performs best on for knowledge tracing. The learner sequences considered as top five learner sequences were required to contain more than 60 questions and confined to a maximum of 500 questions. Since BKT+F could not be evaluated as a learner simulator on the full test set in a reasonable time, it was evaluated on the top five learner sequences. To allow for comparisons each experiment was run twice, once for the top five learner sequences of each test set and once for the full test set.

4.2 Metrics

The two metrics used to evaluate the performance of the knowledge tracing algorithms for knowledge tracing and learner simulation were AUC and accuracy. The knowledge tracing algorithms can be considered classifiers. The task they are being applied for is binary classification. The two classes are whether the response of a learner to a question is a correct or incorrect answer. The knowledge tracing algorithms predict the probability that the learner answered the question correctly, a value between 0 and 1. The learner simulator uses the knowledge tracing algorithm to make this prediction and then converts the prediction to a response according to the threshold 0.5 as explained in section 3.3.1.

Accuracy is the percentage of correct classifications for the total number of classifications made. As a result, it offers a useful metric when analysing the performance of the learner simulators. The metric presents how many simulated responses were the same as the true responses of the learners. It gives an indication of how realistically the learner simulators can simulate the responses of learners. As a result, this metric was used for Experiment 2, where the performance of the knowledge tracing algorithms for learner simulation tasks was evaluated. Since accuracy considers the classification made by the model as opposed to the predicted probability, it offers limited insight into the performance of the knowledge tracing algorithms. For example for a question that the learner

answered correctly, a prediction close to the threshold, such as 0.4, and a prediction far from the threshold, such as 0.1, are considered equally poor predictions when accuracy is used. However, an algorithm that predicts a probability of 0.4 is closer to achieving the correct classification in this example.

AUC stands for area under the ROC curve and is a metric commonly used to measure performance in classification problems. A ROC curve is a plot of true positive rate vs. false positive rate for various decision thresholds of a classifier. The decision threshold of a classifier is the boundary that separates the classification of a point as a certain class. As mentioned above, a threshold of 0.5 was used for simulating the learners' responses. Overall, AUC is a measure of how well the algorithm can distinguish between the two classes, correct or incorrect. Unlike accuracy which considers the class predicted, AUC considers the probability predicted. For example for a question that the learner answered correctly, an algorithm that produces a prediction close to the threshold, such as 0.4, would have a higher AUC than an algorithm that produces a prediction far from the threshold, such as 0.1. As a result, AUC can provide a better insight into the predictive performance of the knowledge tracing algorithms. As a result, this metric was used for each experiment.

4.3 Experiment 1: How well do knowledge tracing algorithms perform at knowledge tracing tasks?

4.3.1 Experimental Setup

Each knowledge tracing algorithm was trained on the training set of each dataset. Each knowledge tracing algorithm was evaluated twice, once on the top five learner sequences of the test set and once on the full test set. For each learner sequence the knowledge tracing algorithm predicted a response to each question in the sequence. For each prediction, the algorithm received all available past questions and responses in the sequence as inputs and received the current question as an input. The algorithm then predicted a response to the current question. This was repeated for each question in the learner sequence. The predicted responses of the learners for each question in the learner sequences were compared with the true responses of the learners for those questions and the AUC of the predictions was calculated. As outlined in section 3.4, the representation of the questions as inputs, depends on the knowledge tracing algorithm. BKT+F uses the skill ID of the skill associated with a question. DKT+ uses both the question ID and skill ID. DKT can use either the question ID or skill ID. The input that resulted in the best performance for each dataset was used by DKT. The inputs used were skill ID for ASSISTments 2015

and Spanish 2013, and question ID for Statics 2011.

4.3.2 Results

The AUC of each knowledge tracing algorithm for the top five learner sequences of each test set is displayed in Fig. 4.1. For the ASSISTments 2015 dataset, DKT+ is the best performing model with an AUC of 0.956 while BKT+F performs worst with an AUC of 0.900. For Spanish 2013, DKT is the best performing algorithm with an AUC of 0.863 while DKT+ is the worst performing algorithm with an AUC of 0.834. When the algorithms are applied to the dataset Statics 2011, DKT+ is the best performing algorithm with an AUC of 0.886 and BKT+F is the worst performing algorithm with an AUC of 0.793. A complete breakdown of the AUC achieved by each knowledge tracing algorithm for the top five learner sequences of each test set can be found in Appendix A.1. The average AUC of each knowledge tracing algorithm across the top five learner sequences of each test set is displayed in Table 4.2. When considering the average AUC achieved across the top five learner sequences of each test set, DKT is the best performing algorithm for knowledge tracing. It achieved an average AUC of 0.895. DKT closely follows in performance with an average AUC of 0.892. Finally, BKT+F achieved an average AUC of 0.849 across the three datasets. The difference in average AUC between the best performing algorithm, DKT, and the worst performing algorithm, BKT+F, is 0.046.

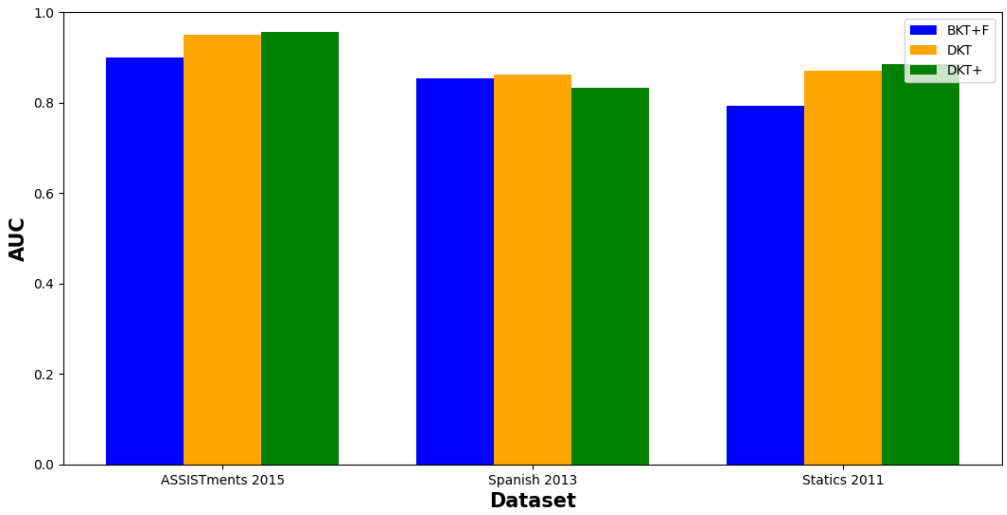


Figure 4.1: The AUC of each knowledge tracing algorithm for the top five learner sequences of each test for the knowledge tracing task

Knowledge Tracing Algorithm	Average AUC
BKT+F	0.849
DKT	0.895
DKT+	0.892

Table 4.2: The average AUC of each knowledge tracing algorithm for the top five learner sequences of the test sets for the knowledge tracing task

The AUC of each knowledge tracing algorithm when evaluated for the full test sets is displayed in Figure 4.2. DKT is the best performing algorithm on the ASSISTments 2015 dataset achieving an AUC of 0.730. BKT+F is the worst performing algorithm with an AUC of 0.703. For the Spanish 2013 dataset, BKT is the best performing algorithm with an AUC of 0.839 while DKT+ is the worst performing algorithm with an AUC of 0.830. For Statics 2011, DKT+ achieves the highest AUC, 0.827, while BKT+F is the worst performing algorithm with an AUC of 0.713. A complete breakdown of the AUC achieved by each knowledge tracing algorithm for each of the full test sets can be found in Appendix A.1. The average AUC of each knowledge tracing algorithm across the three test sets is displayed in Table 4.3. DKT+ is the best performing knowledge tracing algorithm when considering the average AUC achieved across the three datasets when all of the test set is used. It achieved an average AUC of 0.794. DKT closely follows in performance with an average AUC of 0.793. BKT+F achieved an average AUC of 0.752. The difference in average AUC between the best performing algorithm, DKT+, and the worst performing algorithm, BKT+F, is 0.042.

Knowledge Tracing Algorithm	Average AUC
BKT+F	0.752
DKT	0.793
DKT+	0.794

Table 4.3: The average AUC of each knowledge tracing algorithm for the full test sets for the knowledge tracing task

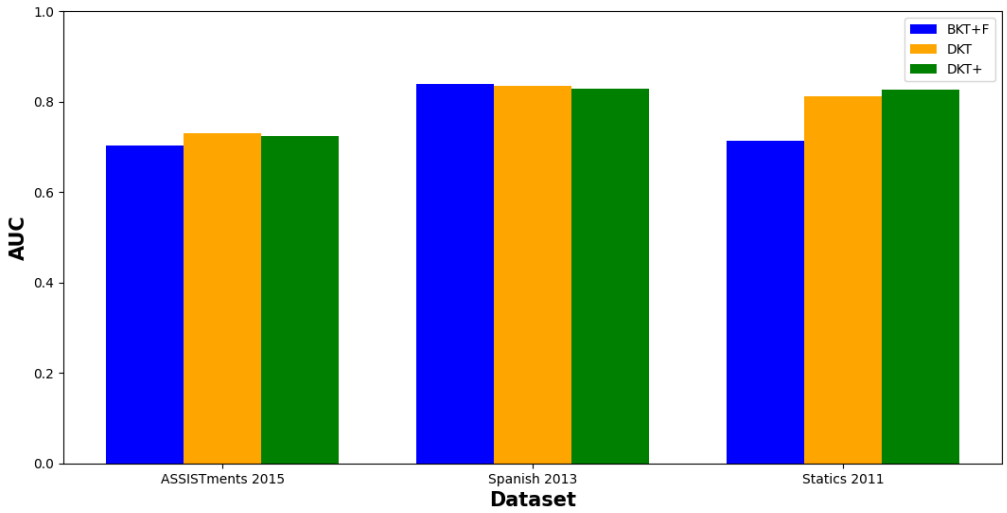


Figure 4.2: The AUC of each knowledge tracing algorithm for each full test set for the knowledge tracing task

4.4 Experiment 2: How well do knowledge tracing algorithms perform at learner simulation tasks?

4.4.1 Experimental Setup

As mentioned in chapter 3, a learner simulator has three inputs; a trained knowledge tracing algorithm, an initialization sequence and a simulation sequence. Each of the knowledge tracing algorithms trained for the first experiment are inputted to a learner simulator. This creates three learner simulators, one for each knowledge tracing algorithm.

A learner simulation task consists of simulating the responses of a learner to a sequence of questions given an initial knowledge state for the learner. As a result, the learner simulation task depends on the initialization sequence and the simulation sequence. A configuration of the initialization sequence length, the number of questions and responses used to initialize the learner, and the simulation sequence length, the number of questions responses must be simulated for, defines a learner simulation task.

For this experiment two learner simulation tasks were defined by defining two combinations of initialization and simulation sequence lengths. The adaptive learning recommendation systems presented by Farrell (2020) and Liu et al. (2019) influenced the choice of initialization and simulation sequence lengths. As mentioned in section 2.1, the recommendation system designed by Farrell (2020) provides a learner with a pre-test, recommended questions and a post-test. The pre-test is equivalent to the initialization

sequence and the recommended questions and post-test are equivalent to the simulation sequence. In this recommendation system the pre-test consisted of 20 questions, the maximum number of questions that could be recommended was 20 and the post-test consisted of 20 questions. The recommendation systems evaluated by Liu et al. (2019) were configured to recommend sequences of 20 questions for comparative experiments. Further experiments investigated sequences of questions of lengths ranging between 5 and 40 questions. Influenced by these adaptive learning recommendation systems, the lengths of initialization and simulation sequences for two learner simulation tasks were defined. Learner Simulation Task 1 was defined to have an initialization sequence length of 10 questions and responses and a simulation sequence length of 20 questions. Learner Simulation Task 2 was defined to have an initialization sequence length of 20 questions and responses and a simulation sequence length of 40 questions.

DKT and DKT+ were evaluated twice for each learner simulation task, once on the top five learner sequences of the test sets and once on the full test sets. BKT+F was evaluated for each learner simulation task on the top five learner sequences of the test sets. Each learner sequence was divided into sub-sequences. Each sub-sequence contained 60 questions. The sub-sequences were equal in length to the maximum number of initialization questions, 20, and the maximum number of question for which responses must be simulated, 40. These sub-sequences were further divided into an initialization sequence and simulation sequence according to the learner simulation task. For each sub-sequence the learner simulator receives the initialization sequence as input and outputs a simulated response to each question in the simulation sequence. The simulated responses of the learner for the questions in the simulation sequence were compared with the true responses of the learner for those questions and the AUC of the predictions and the accuracy of the simulated responses were calculated.

4.4.2 Results

The performance of each knowledge tracing algorithm as a learner simulator on the top five learner sequences of each test set is displayed in Fig. 4.3 and Fig. 4.4. Fig. 4.3 displays the AUC and accuracy of each algorithm for Learner Simulation Task 1. Fig. 4.4 displays the AUC and accuracy for Learner Simulation Task 2.

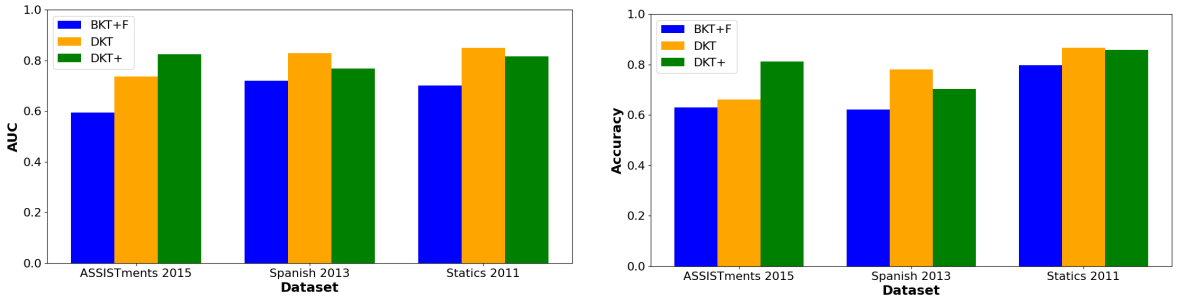


Figure 4.3: The AUC and accuracy of each learner simulator for the top five learner sequences of each test set for Learner Simulation Task 1

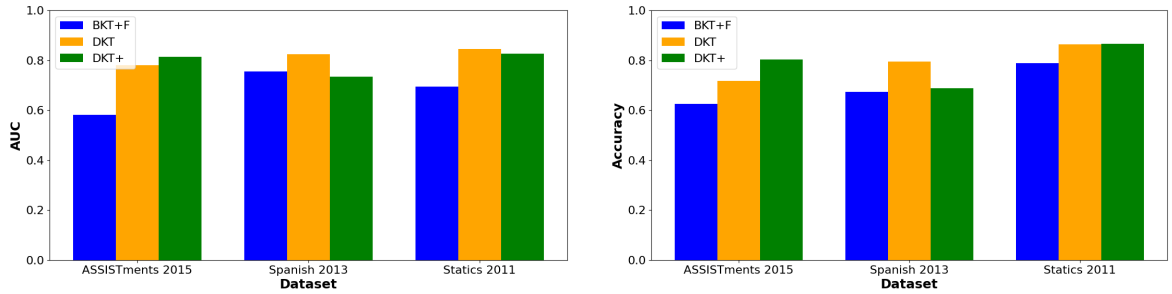


Figure 4.4: The AUC and accuracy of each learner simulator for the top five learner sequences of each test set for Learner Simulation Task 2

The average AUC of each learner simulator across the three datasets for the two learner simulation tasks is presented in Table 4.4. DKT is the best performing algorithm for learner simulation when considering the average AUC achieved across the the top five learner sequences of each test set for the two learner simulation tasks. The algorithm achieved an average AUC of 0.811. DKT+ performs second best with an average AUC of 0.798. Finally, BKT+F achieved an average AUC of 0.675. The difference in average AUC between the best performing algorithm, DKT, and the worst performing algorithm, BKT+F, is 0.136.

The average accuracy of each learner simulator across the three datasets for the two learner simulation tasks is presented in Table 4.5. In terms of the average accuracy of each learner simulator across the top five learner sequences of each test set for the two learner simulation tasks, DKT+ is the best performing model. It achieved an average accuracy of 0.789. DKT achieved the second highest average accuracy, 0.781. BKT+F was the least accurate learner simulator with an average accuracy of 0.690. In terms of how the algorithms accuracy varied across the datasets and the learner simulation tasks, the accuracy of BKT ranged from 0.621 to 0.797, the accuracy of DKT ranged from 0.661 to 0.867 and the accuracy of DKT+ ranged from 0.688 to 0.866.

A complete breakdown of the AUC and accuracy of each knowledge tracing algorithm

as a learner simulator on the top five learner sequences of each test set for each learner simulation task can be found in Appendix A.2.

Learner Simulator	Average AUC
BKT+F	0.675
DKT	0.811
DKT+	0.798

Table 4.4: The average AUC of each learner simulator for the top five learner sequences of the test sets and the two learner simulation tasks

Learner Simulator	Average Accuracy
BKT+F	0.690
DKT	0.781
DKT+	0.789

Table 4.5: The average accuracy of each learner simulator for the top five learner sequences of the test sets and the two learner simulation tasks

The performance of DKT and DKT+ as learner simulators on each of the datasets for the full test set is displayed in Fig. 4.5 and Fig. 4.6. Fig. 4.5 displays the AUC and accuracy of each algorithm for Learner Simulation Task 1. Fig. 4.6 displays the AUC and accuracy for Learner Simulation Task 2.

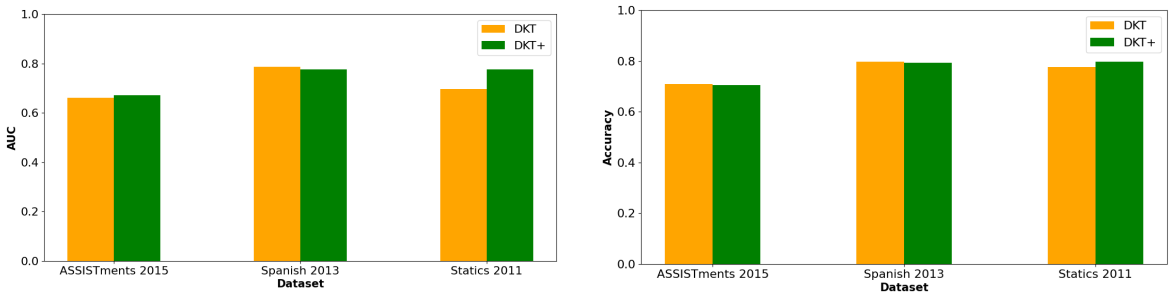


Figure 4.5: The AUC and accuracy of each learner simulator for each test set for Learner Simulation Task 1

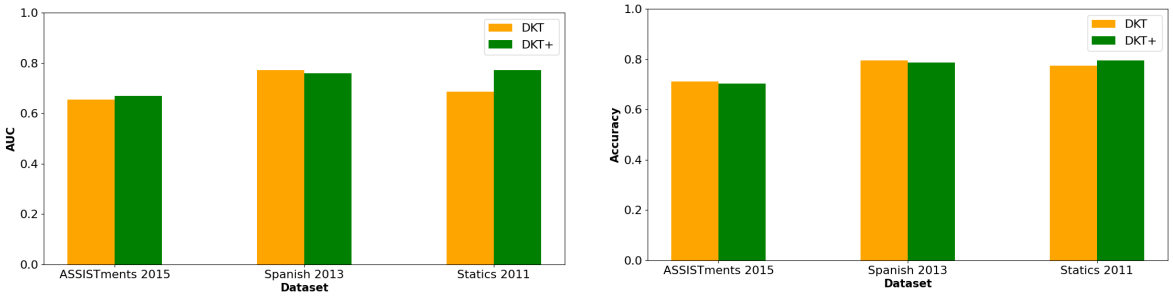


Figure 4.6: The AUC and accuracy of each learner simulator for each test set for Learner Simulation Task 2

DKT+ is the best performing algorithm for learner simulation when considering the average AUC achieved across the three datasets and two learner simulation tasks when the full test set was used. The algorithm achieved an average AUC of 0.738. DKT achieves an an average AUC of 0.711. The average AUC of each learner simulator across the three datasets for the two learner simulation tasks is displayed in Table 4.6.

The average accuracy of each learner simulator across the three datasets for the two learner simulation tasks is displayed in Table 4.7. In terms of the average accuracy of each learner simulator across the three datasets for the two learner simulation tasks, DKT+ is the best performing model. It achieved an average accuracy of 0.764. DKT is less accurate with an accuracy of 0.761. In terms of how the algorithms accuracy varied across the datasets and the learner simulation tasks, the accuracy of DKT ranged from 0.710 to 0.797 and the accuracy of DKT+ ranged from 0.704 to 0.797.

A complete breakdown of the AUC and accuracy of each knowledge tracing algorithm as a learner simulator on the full test sets for each learner simulation task can be found in Appendix A.2.

Knowledge Tracing Algorithm	**Average AUC**
DKT	0.711
DKT+	0.738

Table 4.6: The average AUC of each learner simulator for the full test sets and the two learner simulation tasks

Learner Simulator	Average Accuracy
DKT	0.761
DKT+	0.764

Table 4.7: The average accuracy of each learner simulator for the full test sets and the two learner simulation tasks

4.5 Experiment 3: How does the length of the simulation sequence effect performance?

4.5.1 Experimental Setup

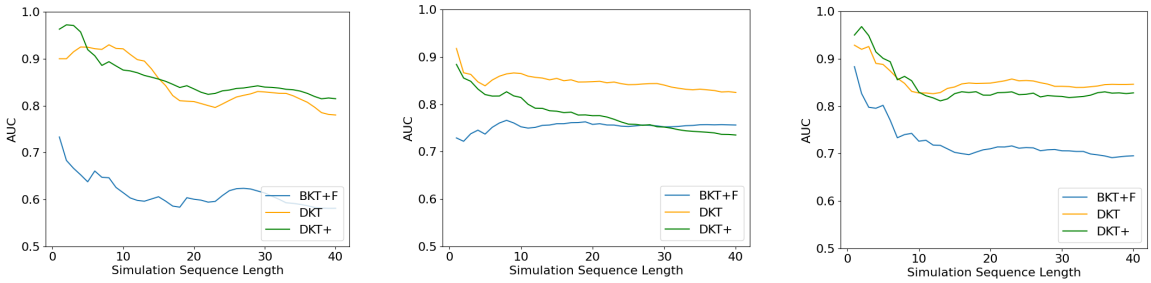
For this experiment each learner sequence was divided into a sub-sequence of 60 questions as it was for the second experiment. The sub-sequence was divided into an initialization and simulation sequence. The initialization sequence was fixed at a length of 20 questions. The simulation sequence length was varied from 1 to 40 questions. The simulated responses of the learner simulators for the questions in the simulation sequence were compared with the true responses of the learner for those questions and the AUC was calculated. The experiment was conducted for the top five learner sequences of each test set using all three learner simulators and for the full test sets using DKT and DKT+.

4.5.2 Results

The effect of the simulation sequence length on the AUC of each knowledge tracing algorithm as a learner simulator is displayed in Fig. 4.7 and Fig. 4.8.

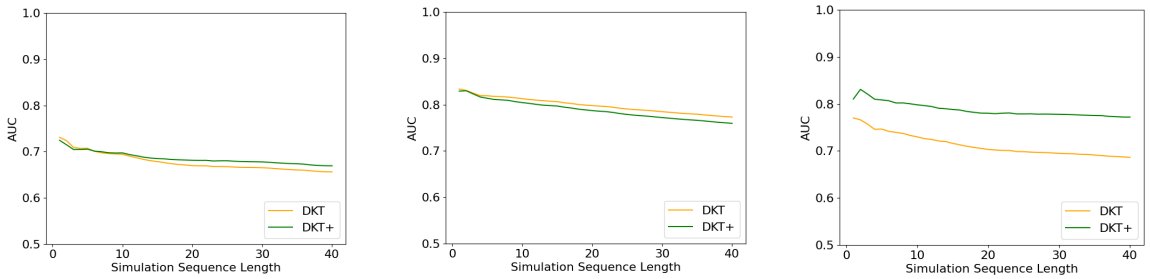
Fig. 4.7 displays the change in AUC for each algorithm for the top five learner sequences as the simulation sequence length is increased. In general, there is an overall decrease in AUC for an increase in simulation sequence length. The overall decrease in AUC exhibits noisy behaviour. In general, the decrease in AUC is greater for the initial increase in simulation sequence length. The effect of the simulation sequence length on the AUC of BKT+F for the top five learner sequences of Spanish 2013 is an exception to the general trend. The AUC of the learner simulator increases for an initial increase in simulation sequence length.

Fig. 4.8 displays the change in AUC for DKT and DKT+ when evaluated on the full test sets. An overall decrease in AUC for an increase in simulation sequence length is again observed. The decrease experienced for an increase in simulation sequence length for the full test set is less than the decrease experienced for the top five learner sequences. The decrease in AUC as simulation sequence length increases experiences less noise.



(a) ASSISTments 2015 (b) Spanish 2013 (c) Statics 2011

Figure 4.7: The AUC of each learner simulator vs. simulation sequence length for the top five learner sequences of each test set



(a) ASSISTments 2015 (b) Spanish 2013 (c) Statics 2011

Figure 4.8: The AUC of each learner simulator vs. simulation sequence length for each full test set

4.6 Experiment 4: How does the length of the initialization sequence effect performance?

4.6.1 Experimental Setup

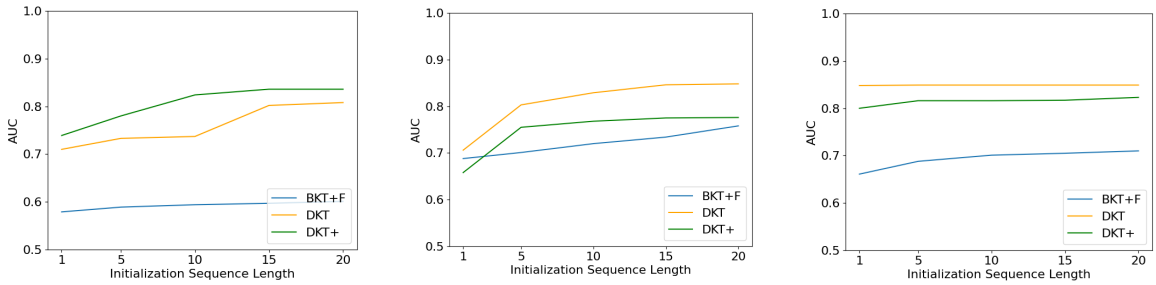
For this experiment, each learner sequence was divided into a sub-sequence of 60 questions as it was for the second and third experiment. The sub-sequence was divided into an initialization and simulation sequence. The simulation sequence was fixed at a length of 20 questions. The initialization sequence length was varied from 1 to 20 questions in intervals of 5 questions. The simulated responses of the learner simulators for the questions in the simulation sequence were compared with the true responses of the learner for those questions and the AUC was calculated. The experiment was conducted for the top five learner sequences of each test set for all three learner simulators and for the full test sets for DKT and DKT+. The evaluation of DKT for the top five learner sequences of Statics 2011 was rerun using different inputs due to the results produced.

4.6.2 Results

The effect of the initialization sequence length on the AUC of each knowledge tracing algorithm as a learner simulator is displayed in Fig. 4.9, Fig. 4.10 and Fig. 4.11.

Fig. 4.9 displays the change in AUC for each learner simulator for the top five learner sequences as the initialization sequence length is increased. In general, there is an overall increase in AUC for an increase in initialization sequence length. In general, the increase in AUC is greater for the initial increase in initialization sequence length. The effect of the initialization sequence length on the AUC of DKT for the top five learner sequences of Statics 2011 is an exception to the general trend. There is no change in AUC for an increase in initialization sequence length. Fig. 4.10 displays the change in AUC for an increase in initialization sequence for the different inputs of DKT. When DKT uses skill ID as the input there is an initial increase in AUC for an increase in initialization sequence length.

Fig. 4.11 displays the change in AUC for DKT and DKT+ when evaluated on the full test sets. An overall increase in AUC for an increase in initialization sequence length is again observed. The increase experienced for an increase in initialization sequence for the full test set is less than the increase experienced for the top five learner sequences for ASSISTments 2015 and Statics 2011.



(a) ASSISTments 2015 (b) Spanish 2013 (c) Statics 2011

Figure 4.9: The AUC of each learner simulator vs. initialization sequence length for the top five learner sequences of each test set

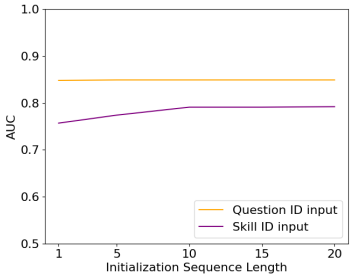
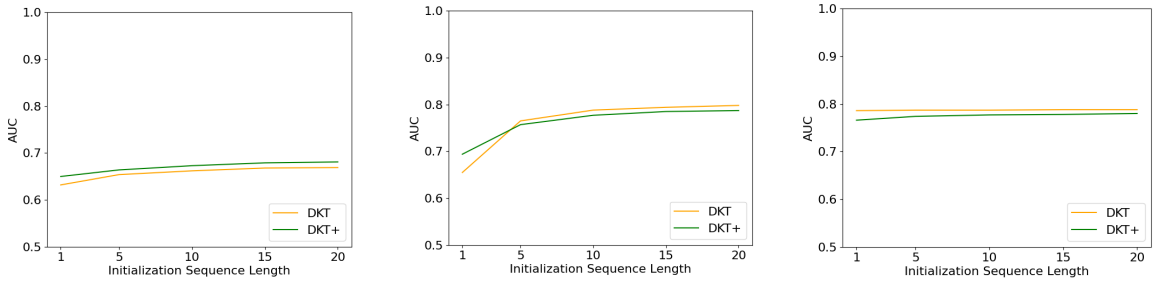


Figure 4.10: The AUC of DKT with different inputs vs. initialization sequence length for the top five learner sequences of Statics 2011



(a) ASSISTments 2015 (b) Spanish 2013 (c) Statics 2011

Figure 4.11: The AUC of each learner simulator vs. initialization sequence length for each full test set

4.7 Discussion

4.7.1 Comparing Knowledge Tracing Algorithms for Knowledge Tracing and Learner Simulation

In terms of knowledge tracing, the deep knowledge tracing algorithms, DKT and DKT+, are the best performing algorithms in terms of average AUC across the three datasets. Despite DKT and DKT+ achieving a greater average AUC than BKT+F, BKT+F is still competitive with the deep knowledge tracing algorithms for knowledge tracing. When considering the full test sets, the difference in average AUC between BKT+F and the best performing algorithm, DKT+, is 0.042. Furthermore, the performance of the knowledge tracing algorithms depends on the dataset. BKT+F was the best performing algorithm for Spanish 2013 slightly outperforming the deep knowledge tracing algorithms. Meanwhile, the deep knowledge tracing algorithms slightly outperformed BKT+F for ASSISTments 2015 and significantly outperformed BKT+F for Statics 2011.

When the algorithms are tested on the top five learner sequences of the test set for knowledge tracing, performance increases as expected. The deep knowledge tracing al-

gorithms outperform BKT+F to a similar degree in terms of average AUC across the three datasets. The difference in average AUC between BKT+F and the best performing algorithm, DKT+, is 0.046.

The average AUC of the algorithms for learner simulation presented in the second experiment can be compared with the average AUC of the algorithms for knowledge tracing. A comparison of these results gives an indication of whether the performance of knowledge tracing algorithms for knowledge tracing can inform us about their performance as learner simulators.

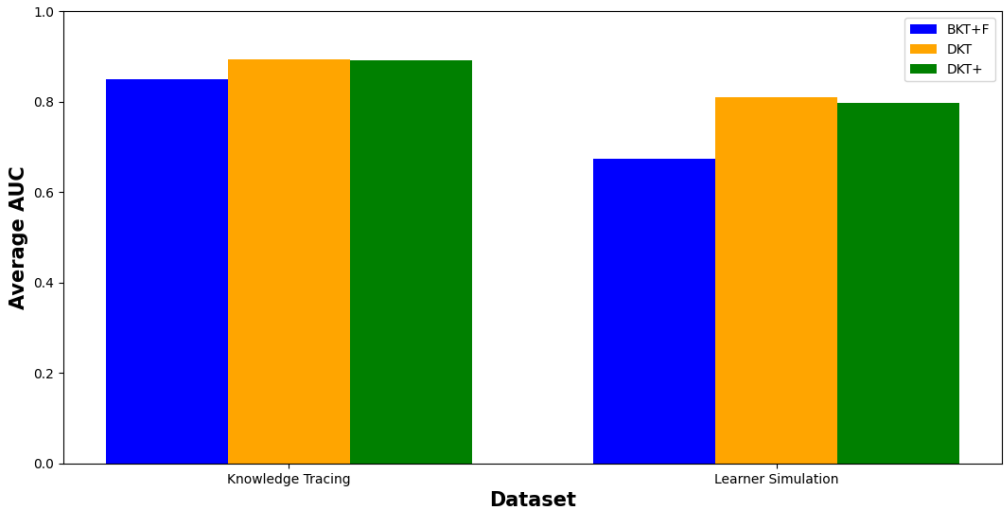


Figure 4.12: The average AUC of each algorithm for the top five learner sequences of each test set for knowledge tracing and learner simulation

DKT and DKT+, are the best performing algorithms in terms of average AUC across the top five learner sequences of each test set for the two simulation tasks. BKT+F is less competitive with the deep knowledge tracing algorithms for learner simulation than it was for knowledge tracing. The difference in average AUC between BKT+F and the best performing algorithm, DKT, is 0.137. Fig. 4.12 displays the average AUC achieved by each algorithm for knowledge tracing and learner simulation across the top five learner sequences of each test set. Comparing the AUC of the algorithms for knowledge tracing and learner simulation, it appears the performance of the algorithms for knowledge tracing can inform us to a degree about which algorithms will be best for learner simulation. The deep knowledge tracing algorithms outperform BKT+F in both experiments. Furthermore, the best performing algorithm in terms of average AUC is the same for learner simulation as it is for knowledge tracing. DKT achieved the highest average AUC for the top five learner sequences for both knowledge tracing and learner simulation. DKT+

achieved the highest average AUC for the full test sets for both knowledge tracing and learner simulation. However, the difference in average AUC between BKT+F and the deep knowledge tracing algorithms is significantly greater for learner simulation than for knowledge tracing as seen in Fig. 4.12. BKT+F is competitive with the deep knowledge tracing algorithms for knowledge tracing, in particular when applied to certain datasets. However, as learner simulators the deep knowledge tracing algorithms outperform BKT+F to a much greater degree.

BKT+F being less competitive with the deep knowledge tracing algorithms for learner simulation than it was for knowledge tracing may be due to BKT+F modelling the knowledge state of the learner for each skill independently. When predicting a learner's response to a question associated with a skill, Bayesian knowledge tracing algorithms only use the responses of the learner to questions associated with that skill to inform their prediction. For the evaluation of the algorithms for knowledge tracing the knowledge state of the learner continuously increases as the responses for questions further in the learner sequence are predicted. As a result, for predictions of responses to questions further in the learner sequence some information about the knowledge state of the learner for the skill of the current question is likely to be available. In contrast, for learner simulation the initial knowledge state is fixed and the knowledge state is only updated with simulated responses. As a result, BKT+F relies on questions in the simulation sequence being associated with skills that questions in the initialization sequence are also associated with in order to make informed predictions. In contrast, the deep knowledge tracing algorithms have the ability to learn the underlying relationships between different skills and questions. Questions associated with different skills than those present in the simulation sequence can still inform the predictions of the deep knowledge tracing algorithms if underlying relationships are present.

4.7.2 The Performance of Knowledge Tracing Algorithms as Learner Simulators

The results of the second experiment inform us how accurately knowledge tracing algorithms can simulate the responses of a real learner given an initial knowledge state of the learner. The results indicate that the best performing learner simulators, DKT and DKT+, can simulate learners' responses across the three datasets with an average accuracy of 76% for the two learner simulation tasks defined. The two learner simulation tasks defined reflect the number of questions that would be required to be simulated during training and evaluation of the adaptive learning recommendation systems being considered. This is a promising result that gives an indication of how realistically

learner simulators built using knowledge tracing algorithms can mimic the responses of real learners when responding to questions for these adaptive learning recommendation system. When the learner simulators DKT and DKT+ are applied to the top five learner sequences of each test set, their accuracy increases as expected. The average accuracy of DKT+ increases from 0.764 to 0.789 while the average accuracy of DKT increases from 0.761 to 0.781. When considering the maximum accuracy in the range of accuracy achieved by DKT and DKT+ for the top five learner sequences and two learner simulation tasks, DKT reached an accuracy of 0.867 and DKT+ reached an accuracy of 0.866. The increase in accuracy when learner simulators are applied to the top five learner sequences and the maximum accuracy achieved demonstrates that the accuracy of learner simulators can increase depending on the learner sequences it is simulating responses for. This indicates that their is potential for the accuracy of learner simulators to be increased further when used for training and evaluating adaptive learning recommendation systems. It is likely that the top five learner sequences contain positive sequence structure characteristics that are allowing more accurate predictions to be made by the knowledge tracing algorithms. Examples of these positive sequence structure characteristics might be the repetition of questions associated with the same skill or the repetition of questions associated with strongly related skills throughout the sequence. Since the adaptive learning recommendation systems will be generating the sequence of questions to be completed by the learner simulators, they can ensure the sequences contain positive sequence structure characteristics which will allow for increased learner simulator accuracy. Overall, the accuracy of the deep knowledge tracing learner simulators presented in these results is a promising indication that knowledge tracing algorithms can realistically mimic the learner behaviour required to train and evaluate adaptive learning recommendation systems that recommend questions according to learners' responses. It is hoped that this level of accuracy is suitable for reducing the number of real learners and the number of responses per real learner required for the training and evaluation of these adaptive learning recommendation systems.

In terms of the accuracy of the BKT+F learner simulator, it achieved an average accuracy of 0.690 when simulating responses for the top five learner sequences. BKT+F does not outperform the other algorithms as a learner simulator for any of the datasets or for either of the simulation tasks. The results indicate that the deep knowledge tracing algorithms are the best choices as learner simulators when simulating the responses of learners. As mentioned previously, the significant difference in performance between BKT+F and the deep knowledge tracing algorithms for learner simulation is likely due to BKT+F modeling the knowledge state of the learner for each skill independently.

Evaluation of the performance of the knowledge tracing algorithms as learner simula-

tions is focused on their simulation of learners' responses for the purpose of training adaptive learning recommendation systems that recommend questions according to learners' responses. As a result, the evaluation methodology looked to replicate the initialization and simulation sequences that would be present in the environment of these recommendation systems. Evaluation involved evaluating knowledge tracing algorithms as learner simulators for two learner simulation tasks. These simulation tasks reflected the number of questions that could be used for initialization and the number of responses that would need to be simulated when training and evaluating these systems. While the number of questions being used for initialization and the number of responses being simulated may be an accurate reflection of what would occur during training and evaluation of these systems, the structure of the sequences used for evaluation may not be an accurate reflection of the structure of the sequences that would be presented to the learner simulator during training and evaluation of these systems. The datasets used contain the responses of learners to questions from different tutoring systems and courses. The structure of the sequence of questions being completed by learners in these systems may differ to the structure of the sequence of questions that would be completed by learners using the recommendation systems being considered. As a result, the reported learner simulator performance may differ when learner simulators simulate responses in the environment of the recommendation system. However, it is likely that the structure of the sequence of questions present will be beneficial for learner simulator performance. The pre-test will likely contain questions associated with a limited range of skills. The questions that can be recommended may be confined to questions associated with the skills present in the pre-test. The post-test will contain questions related to the same skills as those present in the pre-test. There will be a high repetition of the same skills and strongly related questions throughout the sequence. As a result, it is likely that the knowledge state inferred from the pre-test or initialization sequence will be very informative for predicting the responses of the learner to questions in the simulation sequence, i.e., the recommended questions and post-test. The evaluation setup may not be a perfect reflection of the environment present in the recommendation systems being considered. However, it provides insight into how the learner simulators would perform in that environment. Furthermore, it is believed the accuracy of the learner simulators would be higher in the real environment due to the structure of the sequences of questions present.

4.7.3 The Effect of Simulation Sequence Length and Initialization Sequence Length on Performance

As seen from the results of the third experiment, in general, AUC decreases as simulation sequence length increases. Prior to experimentation it was expected that the performance of the knowledge tracing algorithms would decrease as the number of responses to be simulated increases. An exception to the general trend observed is the initial increase in the AUC of BKT+F as simulation sequence length increases for the top five learner sequences of the Spanish 2013 dataset. A possible explanation for this increase in performance as the number of responses to be simulated increases is due to BKT+F modeling the knowledge state of the learner for each skill independently. As stated previously, BKT+F relies on questions in the simulation sequence being associated with skills that questions in the initialization sequence are associated with in order to make informed predictions. It may be the case that for the Spanish 2013 dataset, the initial questions the algorithm must simulate responses for are not associated with many skills present in the initialization sequence. As the algorithm simulates more responses it may increase the chance of the algorithm encountering questions associated with skills that questions in the initialization sequence are also associated with. As a result, the algorithm can make more informed predictions.

The results of the final fourth experiment demonstrate the general trend that performance increases as the initialization sequence increases. This trend was expected to be observed. Increasing the length of the initialization sequence provides more information for the algorithms to use to model the initial knowledge state of the learner allowing for more informed predictions to be made. An exception to this general trend was the change in AUC of DKT for an increase in initialization sequence length for the dataset Statics 2011 when DKT uses question ID as an input. DKT experienced no change in AUC as the initialization sequence length increased.

When using question ID as an input, DKT has the ability to learn about the underlying relationships between different questions. The strongest relationships between questions would likely be the skill associated with the questions. DKT can use the response of the learner to one question to inform its prediction for another question provided there is some underlying relationship present between the questions, e.g., they are both associated with the same skill. When DKT is operating in this manner, increasing the initialization sequence would be expected to increase performance. This operation of DKT assumes that the response of the learner to similar questions, the knowledge state of the learner, is the feature of the model that has the greatest impact on the prediction. However, it may be the case that for this particular dataset the specific question being answered may be

the feature of the model that has the greatest impact on the prediction. This situation may occur if questions in the dataset vary greatly in terms of difficulty. The difficulty of the question may be more informative than a learner's knowledge state in terms of how a learner would respond to a question. This offers a potential explanation as to why increasing the initialization sequence has no effect on performance when question ID is used as the input for Statics 2011. When skill ID is used the model no longer has the specific question being answered as a feature. Instead a prediction is being made for some question related to a specific skill. As a result, the length of the initialization sequence has a greater impact on performance.

Chapter 5

Conclusions & Future Work

5.1 Conclusions

The research conducted consisted of an investigation of knowledge tracing algorithms as learner simulators. It was noted that there are many different types of adaptive learning systems and the learner behaviour required to train and evaluate each can differ greatly. Furthermore, the potential for learner simulators to be designed and implemented to simulate different learner behaviours was also acknowledged. However, for the purposes of this research, learner simulators were defined as software that can simulate learners' responses to questions, where a learner's response was defined as an indication of whether a learner answers a question correctly or incorrectly. The use of knowledge tracing algorithms as learner simulators was identified as a potential solution for simulating learners' responses which can be used to train and evaluate adaptive learning recommendation systems that recommend questions according to learners' responses. However, it was recognised that current research lacks evaluation of knowledge tracing algorithms as learner simulators. As a result it was unknown how realistically learner simulators built using knowledge tracing algorithms can simulate learners' responses.

The research conducted was aimed at answering the question of "How well do knowledge tracing algorithms perform as learner simulators?" In the pursuit of answering this question a literature review of state-of-the-art knowledge tracing algorithms was first conducted. Learner simulators that use knowledge tracing algorithms were designed and implemented as well as an evaluation framework for the learner simulators. The learner simulators were designed and implemented to use the benchmark and state-of-the-art knowledge tracing algorithms BKT+F, DKT, and DKT+. Finally, an evaluation and comparative analysis of the knowledge tracing algorithms as learner simulators was conducted. As a result, each of the research objectives were carried out.

The evaluation of the learner simulators found that the best performing learner simulators, DKT and DKT+, can simulate learners' responses across the three datasets with an average accuracy of 76% for the two learner simulation tasks defined. The two learner simulation tasks the learner simulators were evaluated for reflect the number of questions that would be required to be simulated for adaptive learning recommendation systems that recommend questions according to learners' responses. This level of accuracy is promising and gives an indication of how realistically learner simulators built using knowledge tracing algorithms can simulate learners' responses for the adaptive learning recommendation systems investigated. Evaluation of the learner simulators also found that the accuracy of the learner simulators depends on the initialization and simulation sequence. There is the potential for the accuracy of the learner simulators to be increased by using initialization and simulation sequences containing positive sequence structure characteristics. Examples of positive sequence structure characteristics may include the repetition of questions associated with the same skill or the repetition of questions associated with strongly related skills.

In terms of comparing the different learner simulators, the average accuracy of the learner simulators, DKT and DKT+, for the simulation of the responses of each learner in the test sets across the two learner simulation tasks is approximately equal. BKT+F is outperformed by these algorithms for learner simulation. It is speculated that the difference in performance is due to BKT+F modelling the knowledge state of the learner for each skill independently. It lacks the ability to learn the underlying relationships between different questions and skills, an ability the deep knowledge tracing algorithms possess. These results suggest knowledge tracing algorithms that can learn the underlying relationships between different skills and questions are more appropriate learner simulators than algorithms that model skills independently.

Finally, it was observed that the performance of knowledge tracing algorithms for knowledge tracing can inform us to a degree about their performance as learner simulators. For both knowledge tracing and learner simulation the deep knowledge tracing algorithms, DKT and DKT+, outperformed BKT+F. However, the difference in performance between the deep knowledge tracing algorithms and BKT+F was much greater for learner simulation tasks.

Overall, the performance of knowledge tracing algorithms as learner simulators presented in this research is a promising indication that knowledge tracing algorithms can realistically mimic the behaviour of real learners required to train and evaluate adaptive learning recommendation systems that recommend questions according to learners' responses. It is hoped that this level of accuracy is suitable for reducing the number of real learners and responses per learner required for the training and evaluation of these

adaptive learning recommendation systems.

5.2 Limitations of Approach and Research

The research conducted investigated how well knowledge tracing algorithms performed as learner simulators in terms of simulating the responses' of learners to questions. In particular the research focused on simulating learners' responses for the purpose of training and evaluating adaptive learning recommendation systems that recommend questions according to learners' responses. As a result, the evaluation methodology looked to replicate the environment that would be present in the recommendation system being considered. While the number of questions being used for initialization and the number of responses being simulated is an accurate reflection of what would occur during training and evaluation of these systems, the sequences of questions present in the datasets used for evaluation may not be an accurate reflection of the sequences of questions that would be presented to the learner simulator during training and evaluation of these systems. As a result of this limitation, the reported performance of the learner simulators may be different than their performance simulating learners' responses for the recommendation system being considered. However, it is believed performance may increase when the learner simulators are applied to the real environment due to the structure of the questions that would be present.

A second limitation of this research was the slow simulation time of the learner simulator configured with BKT+F. As a result of this slow simulation time, BKT+F could not be evaluated as a learner simulator on the full test sets. Despite this limitation, the results produced in the research provided insight into the performance of BKT+F as a learner simulator.

A final limitation of this research is the evaluation of the knowledge tracing algorithms on datatsets containing questions only associated with a single skill. The exclusion of datatsets containing multi-skill questions was necessary in order to prevent the repetition of past errors in research. However, the restriction for questions to only be labeled with a single skill is a limitation of the knowledge tracing algorithms used and the learner simulators implemented. In reality, questions often have multiple skills associated with them. As a result, having the ability to label questions with multiple skills allows for more realistic representation of questions and skills.

5.3 Future Work

The research carried out evaluated the performance of BKT+F, DKT and DKT+ as learner simulators. BKT+F is a benchmark knowledge tracing algorithm. It offers a simplistic modelling of a learner's knowledge state that is easily interpreted. DKT and DKT+ are state-of-the-art knowledge tracing algorithms that offer a complex modelling of a learner's knowledge state that is difficult to interpret. Future research could investigate the performance of more state-of-the-art knowledge tracing algorithms as learner simulators. Examples of algorithms that could be investigated are Best-LR presented by Gervet et al. (2020), AKT introduced by Ghosh et al. (2020), RKT implemented by (Pandey and Srivastava, 2020) and DSAKT invented by Zeng et al. (2021).

Another area for future work is an in-depth analysis of the effect of sequence structure on the performance of learner simulators. This research has identified the effect of sequence structure on the accuracy of simulated learners' responses. It is believed certain characteristics of the sequence of questions being used for initialization and simulation, such as the repetition of many questions with the same or similar skills, could increase the performance of the learner simulators. These effects of the sequence structure could be investigated, in particular considering the structure of the pre-tests, recommended questions and post-tests that would be present in adaptive learning recommendation systems.

The results of this research present how accurately learner simulators configured with different knowledge tracing algorithms can simulate the responses of learners for the defined simulation tasks. The accuracy achieved by the best performing learner simulators is a promising indication that knowledge tracing algorithms have the ability to realistically simulate learners' responses to questions. It is hoped that this level of accuracy is suitable for reducing the number of real learners and responses per learner required for the training and evaluation of adaptive learning recommendation systems that recommend questions according to learners' responses. Future research could now investigate this. An adaptive learning recommendation system that recommends questions according to learners' responses could be trained with learner simulators. Different recommendation systems could be trained with different learner simulators varying in accuracy, e.g, BKT+F, DKT and DKT+. Real learners could then be used to evaluate the performance of the trained recommendation system. Through this process it could be determined what level of accuracy is required for learner simulators.

Bibliography

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38.

Awais Hassan, M., Habiba, U., Khalid, H., Shoaib, M., and Arshad, S. (2019). An adaptive feedback system to improve student performance based on collaborative behavior. *IEEE Access*, 7:107171–107178.

Badrinath, A., Wang, F., and Pardos, Z. A. (2021). pybkt: An accessible python library of bayesian knowledge tracing models. *ArXiv*, abs/2105.00385.

Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278.

Farrell, D. (2020). Investigating knowledge tracing algorithms and learner simulators for training adaptive recommendation agents in education. Master's dissertation, University of Dublin, Trinity College.

Gervet, T., Koedinger, K., Schneider, J., and Mitchell, T. (2020). When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54.

Ghosh, A., Heffernan, N., and Lan, A. S. (2020). Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2330–2339, New York, NY, USA. Association for Computing Machinery.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Khajah, M., Lindsey, R. V., and Mozer, M. C. (2016). How deep is knowledge tracing?

Lee, J. and Yeung, D.-Y. (2019). Knowledge query network for knowledge tracing. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. ACM.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2015). End-to-end training of deep visuomotor policies.

Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D. (2016). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection.

Li, Y. (2017). Deep reinforcement learning: An overview.

Lindsey, R. V., Shroyer, J. D., Pashler, H., and Mozer, M. C. (2014). Improving students' long-term knowledge retention through personalized review. *Psychological Science*, 25(3):639–647. PMID: 24444515.

Liu, Q., Tong, S., Liu, C., Zhao, H., Chen, E., Ma, H., and Wang, S. (2019). Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Nurjanah, D. (2016). Good and similar learners' recommendation in adaptive learning systems. In *CSEDU*.

Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., and Zheng, Y. (2019). Recent progress on generative adversarial networks (gans): A survey. *IEEE Access*, 7:36322–36333.

Pandey, S. and Karypis, G. (2019). A self-attentive model for knowledge tracing.

Pandey, S. and Srivastava, J. (2020). Rkt: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1205–1214, New York, NY, USA. Association for Computing Machinery.

Pavlik, P. I., Cen, H., and Koedinger, K. R. (2009). Performance factors analysis –a new alternative to knowledge tracing. page 531–538, NLD. IOS Press.

Piech, C., Spencer, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., and Sohl-Dickstein, J. (2015). Deep knowledge tracing.

Pu, S., Yudelson, M., Ou, L., and Huang, Y. (2020). *Deep Knowledge Tracing with Transformers*, pages 252–256.

Raghuveer, V. R., Tripathy, B. K., Singh, T., and Khanna, S. (2014). Reinforcement learning approach towards effective content recommendation in mooc environments. In *2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)*, pages 285–289.

Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B., and Choi, Y. (2021). Saint+: Integrating temporal features for ednet correctness prediction. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, LAK21, page 490–496, New York, NY, USA. Association for Computing Machinery.

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489.

U.S. Department of Education Office of Educational Technology (2010). *Transforming American Education: Learning Powered by Technology. National Education Technology Plan 2010*.

van der Linden, W. and Hambleton, R. (1997). *Handbook of modern item response theory*. Springer, Netherlands.

Walkington, C. (2013). Using adaptive learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. *Journal of Educational Psychology*, 105:932.

Wolff, A., Zdráhal, Z., Herrmannova, D., Kuzilek, J., and Hlosta, M. (2014). Developing predictive models for early detection of at-risk students on distance learning modules. *CEUR Workshop Proceedings*, 1137.

Xie, H., Chu, H.-C., Hwang, G.-J., and Wang, C.-C. (2019). Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017. *Computers & Education*, 140:103599.

Xiong, X., Zhao, S., Inwegen, E. V., and Beck, J. E. (2016). Going deeper with deep knowledge tracing. In *EDM*.

Zeng, J., Zhang, Q., Xie, N., and Yang, B. (2021). Application of deep self-attention in knowledge tracing.

Appendix A

Experimental Results

A.1 Experiment 1: How well do knowledge tracing algorithms perform at knowledge tracing tasks?

Dataset	Knowledge Tracing Algorithm	AUC
ASSISTments 2015	BKT+F	0.900
ASSISTments 2015	DKT	0.950
ASSISTments 2015	DKT+	0.956
Spanish 2013	BKT+F	0.855
Spanish 2013	DKT	0.863
Spanish 2013	DKT+	0.834
Statics 2011	BKT+F	0.793
Statics 2011	DKT	0.871
Statics 2011	DKT+	0.886

Table A.1: The AUC of each knowledge tracing algorithm for the top five learner sequences of each test set

Dataset	Knowledge Tracing Algorithm	AUC
ASSISTments 2015	BKT+F	0.703
ASSISTments 2015	DKT	0.730
ASSISTments 2015	DKT+	0.725
Spanish 2013	BKT+F	0.839
Spanish 2013	DKT	0.836
Spanish 2013	DKT+	0.830
Statics 2011	BKT+F	0.713
Statics 2011	DKT	0.813
Statics 2011	DKT+	0.827

Table A.2: The AUC of each knowledge tracing algorithm for each full test set

A.2 Experiment 2: How well do knowledge tracing algorithms perform at learner simulation tasks?

Dataset	Knowledge Tracing Algorithm	AUC	Accuracy
ASSISTments 2015	BKT+F	0.594	0.631
ASSISTments 2015	DKT	0.737	0.661
ASSISTments 2015	DKT+	0.824	0.813
Spanish 2013	BKT+F	0.720	0.621
Spanish 2013	DKT	0.829	0.78
Spanish 2013	DKT+	0.768	0.703
Statics 2011	BKT+F	0.701	0.797
Statics 2011	DKT	0.849	0.867
Statics 2011	DKT+	0.816	0.859

Table A.3: The AUC and accuracy of each learner simulator for the top five learner sequences of each test set for Learner Simulation Task 1

Dataset	Knowledge Tracing Algorithm	AUC	Accuracy
ASSISTments 2015	BKT+F	0.582	0.625
ASSISTments 2015	DKT	0.780	0.719
ASSISTments 2015	DKT+	0.815	0.804
Spanish 2013	BKT+F	0.756	0.675
Spanish 2013	DKT	0.825	0.795
Spanish 2013	DKT+	0.735	0.688
Statics 2011	BKT+F	0.695	0.789
Statics 2011	DKT	0.846	0.864
Statics 2011	DKT+	0.828	0.866

Table A.4: The AUC and accuracy of each learner simulator for the top five learner sequences of each test set for Learner Simulation Task 2

Dataset	Knowledge Tracing Algorithm	AUC	Accuracy
ASSISTments 2015	DKT	0.662	0.710
ASSISTments 2015	DKT+	0.673	0.705
Spanish 2013	DKT	0.788	0.797
Spanish 2013	DKT+	0.777	0.794
Statics 2011	DKT	0.698	0.777
Statics 2011	DKT+	0.777	0.797

Table A.5: The AUC and accuracy of each learner simulator for each full test set for Learner Simulation Task 1

Dataset	Knowledge Tracing Algorithm	AUC	Accuracy
ASSISTments 2015	DKT	0.656	0.712
ASSISTments 2015	DKT+	0.670	0.704
Spanish 2013	DKT	0.773	0.795
Spanish 2013	DKT+	0.760	0.788
Statics 2011	DKT	0.686	0.774
Statics 2011	DKT+	0.772	0.796

Table A.6: The AUC and accuracy of each learner simulator for each full test set for Learner Simulation Task 2