Project Proposal: Knowledge Tracing Model

Project Description

Problem to be Solved

Traditional flashcard applications without knowledge tracing allow you to record responses (i.e. know/unsure/don't know) and sort cards for future study based on these responses. However, it's not clear what the optimal curriculum should be—how many times do you need to correctly answer a question to be sure you know it? Furthermore, students would need to complete all flashcards in order for the application to be confident they have mastered the content. This can result in the student studying flashcards unnecessarily (for example, when skills are redundant or an already mastered skill encapsulates another unattempted skill).

Knowledge tracing solves these problems: By estimating students' proficiencies in real time across all possible skills, we can make predictions about what a student knows and does not. Therefore, if a student has mastered a concept (according to the model), they do not need to study redundant questions or encapsulated skills.

A knowledge tracing modeling pipeline and API can not only improve the mobile flashcard app, but also supplement recommendations in other ITS applications.

Proposed Solution

Functionality

Mobile application that delivers simple quiz questions to users. We can display different formats: multiple choice, vocab/concept flashcards, and short answers (developed by mobile subteam). Based on which questions the student gets correct, it can recommend what they should study next. These recommendations will be determined using a knowledge tracing model.

1. Modeling pipeline: We need a pipeline that trains a KT model from historical learner interactions data.
    - Get a sample dataset (ASSISTments). The required columns are:
        - Learner ID
        - Timestamp
        - Skill ID
        - Proficiency (binary "correctness" value)
    - Initial prototype in Jupyter
        - Preprocess into KT format (tf.data.dataset)

                - Split into training, validation and test sets
                - X shape = (batches, samples, embeddings)
                - y shape = (batches, samples, proficiencies)
            - Training pipeline
                - Specify TF model, loss, optimizer, and hyperparameters
                - Train the model
                - Evaluate on test set
    - Training pipeline (Airflow)
        - Split jupyter notebook code into separate functions
        - Specify functions as tasks in Airflow DAG
2. Model API: We need an API to serve that model, allowing other applications to request predictions and send new data to use for future training.
    - Request predictions Endpoint
        - Receive input data (history of learner's interactions)
        - Feed data through model to get predictions
        - Send back predictions
    - Update data endpoint
        - Receive new (training) data and save to database

Benefits

- Efficiently provides questions based on estimating proficiency across many skill sets with the knowledge tracing model
    - Can better estimate whether a student has mastered a skill without unnecessary flashcard repetition
- Provides a targeted learning plan for students who are struggling on, for example, how to start a particular assignment
    - Can serve as a supplement/reinforcer to TutorJS application, also can be an alternative for students to chatbot - another form of guidance

Other Areas for Research

- What data can augment the core KT data to improve model quality?
    - For example, is it helpful to include additional time metadata like how long a student spent on a question or how much time has passed since they last attempted a skill?
- There are a variety of KT modeling approaches—which is most appropriate for our application (flashcard app)?
    - We can investigate the pros and cons of IRT, Bayesian, and deep-learned approaches.

Potential Problems and Pitfalls

- Insufficient training data to train an initial model on GA Tech students' interactions. Nonetheless, we can still implement the modeling pipeline and API using a sample dataset (i.e. ASSISTments).
- Inexperience in machine learning. Some team members are new to this concept and would be spending a good portion of the semester learning this content
    - This may lead to readjusting our plan on what our final product will be

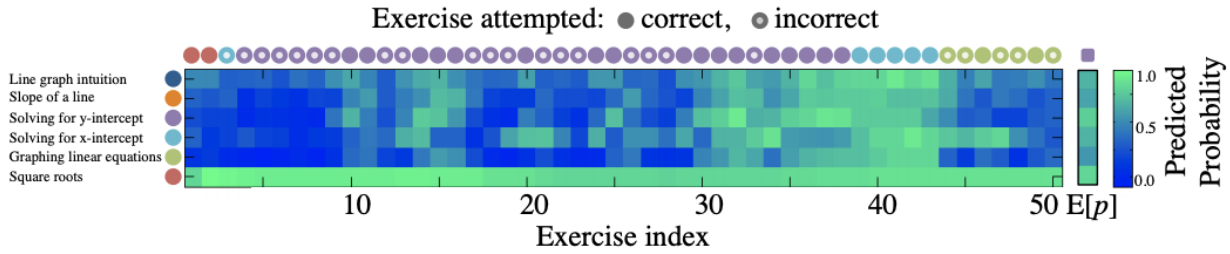
Implementation

Software

- Python libraries
    - Tensorflow
    - Pandas
    - Jupyter notebooks
    - Airflow
    - Opyrator
- Docker

Knowledge Tracing (KT)

KT is the task of predicting a student's knowledge state over time given a sequence of skill-proficiency pairs. Given the history of all prior interactions, the model predicts the probability that a student will correctly answer questions from each skill at the next time step. Together, the vector of predicted probabilities represents the student's knowledge state:



This describes only the most basic form of the model. If more elaborate data is available—i.e. time spent on each question, number of attempts, perception of difficulty, etc.—this can be included in the models as well.

A few common applications for this class of models:

- adaptive testing
- curriculum recommendation

- concept map validation/generation

There are 3 main types of KT models, with some overlap:

- Bayesian KT
- Deep KT
- Logistic models (IRT)

File Management

We would be using GitHub for code/file management and task collaboration.

Project Goals

Since we have a few members that are not experienced in machine learning, our goal for the first month will be to acquaint ourselves with deep neural networks, more specifically RNN and LSTM neural nets, and deep knowledge tracing. This will be done through researching these topics and sharing resources we find with our team.

For the implementation of the KT model, we plan on implementing the functions to train the model by the start of November. Then, we will implement the API, so that we can access the model to update the data and request predictions.

Milestone	**Date Due**	**Status**
Project Proposal Draft	**Friday 9/17/21**	Done
Intro to machine learning	Thursday 9/23/21	In progress
Final Project Proposal	**Friday 9/24/21**	In progress
Deep neural networks	Sunday 9/26/21	Not started
RNN, LSTM	Sunday 10/3/21	Not started
Midterm peer review	**Friday 10/8/21**	Not started
Deep knowledge tracing	Sunday 10/10/21	Not started
Environment Setup	Sunday 10/10/21	Not started
Implement Preprocessing Function	Sunday 10/17/21	Not started
Implement DKT Model and Train Function	Sunday 10/24/21	Not started

Implement Evaluation Function	Sunday 10/31/21	Not started
API: Request Predictions Endpoint	Sunday 11/7/21	Not started
API: Update Data Endpoint	Sunday 11/14/21	Not started
(Stretch) Implement Pipeline as Airflow DAG	Sunday 11/14/21	Not started
Final Code Improvements and Check-ins	Sunday 11/21/21	Not started
(skip for Thanksgiving Break)	Sunday 11/28/21	Not started
Project Cleanup and Final Documentation	Sunday 12/5/21	Not started
Project Presentation	**Wednesday 12/8/21**	Not started

Group Membership

Member skill sets

Member	**Time Commitment**	**Skills and Interests**
Marjorie Ivy	3-4 hours/week(1 credit)	Python, Java, C#, Android/iOS basic app dev, Quality Assurance/Automation Testing. Interested in Machine Learning.
Lukas Olson	3-4 hours/week (1 credit)	Python, Javascript, ML. Interested in content-based recommendation using deep learned knowledge tracing models.
Roshni Dhanasekar	7-8 hours/week (2 credits)	Java, Python, HTML, CSS, JavaScript, React
Varun Krishnaswamy	3-4 hours/week (1 credit)	Python, Java, Javascript, basic ML Interested in doing more ML

Task assignment and participation

- Marjorie Ivy:
    - Facilitator/Developer. Responsible for scheduling meetings and providing insight on knowledge tracing model development.
- Lukas Olson:
    - Team lead/mentor. Responsible for updating tasks and timelines for the project based on the progress made.
- Roshni Dhanasekar:
    - Scribe/Developer. Responsible for taking notes during meetings and aiding in knowledge tracing model implementation. Also will provide the team with insight on previous ITS projects (what worked and what didn't).
- Varun Krishnaswamy:
    - Developer/Scholar. Responsible for reading up on topics that are unfamiliar and aiding in the implementation of the model.

Communication

- Weekly Meetings: Sundays 12pm - 1pm EST via Microsoft Teams
- Scheduler: Lettuce Meet
- Google Drive: meeting notes, design documents
- WhatsApp for quick communications
- Task Collaboration: Github, Kanban Boards:
    - Repo: https://github.gatech.edu/VIP-ITS/Knowledge-Tracing
    - Project Kanban Board: https://github.gatech.edu/VIP-ITS/Knowledge-Tracing/projects/1

Resources

Knowledge Tracing

Here is an excellent overview of knowledge tracing: Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques

And a few other good papers in order of publication date:

- Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge
- Performance Factors Analysis -- A New Alternative to Knowledge Tracing
- Individualized Bayesian Knowledge Tracing Models
- Deep Knowledge Tracing
- Going Deeper with Deep Knowledge Tracing

- A Self-Attentive model for Knowledge Tracing
- Context-Aware Attentive Knowledge Tracing

Videos

- Knowledge Tracing High Level: https://youtu.be/CzRmRZNpB1Y

Deep Learning

- Neural Networks and Deep Learning: https://www.youtube.com/playlist?list=PLkDaE6sCZn6Ec-XTbcX1uRg2_u4xOEky0
- Intro to Machine Learning: https://omscs.gatech.edu/cs-7641-machine-learning-course-videos