A REPORT ON

Student Performance Prediction using Knowledge Tracing

By

Gauransh Sawhney 2018A3PS0325P

Divyansh Agarwal 2018A3PS0791P

Utkarsh Kumar Singh 2018A3PS0368P

Prepared in partial fulfilment of the Course - CS F407 (Artificial Intelligence)

At

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani (Rajasthan)

21 November, 2021

Problem Statement

Following the natural progression of development of education methods and the disruption caused by Covid-19 which only accelerated this process, the need for holistic Intelligent Tutoring Systems(ITSs) has grown manifold.

Intelligent Tutoring Systems are computer systems that aim to provide personalised instruction and feedback to users, often through AI technology and without a human teacher. As part of our project, we have proposed an improved Knowledge Tracing system that will provide a way to predict student performance and tune the course and concept structures in ITSs catered to each individual student based on their predicted performance. A strong KT method can greatly improve an ITS with regards to planning and reorganization of content and evaluative materials. This is exactly the target of our focus as part of the ITS we have designed.

Implementation details

We implement the graph-based Knowledge Tracing model in the popular PyTorch framework. Other helpful libraries such as NumPy, Pandas are used for preprocessing the data and performing necessary transformations in intermediate learning steps. The GNN model is trained on the ASSISTment dataset which(after necessary preprocessing) had data for 1000 students, evaluated over ~63000 assessments on 101 skills.

Preprocessing was done majorly in two steps:

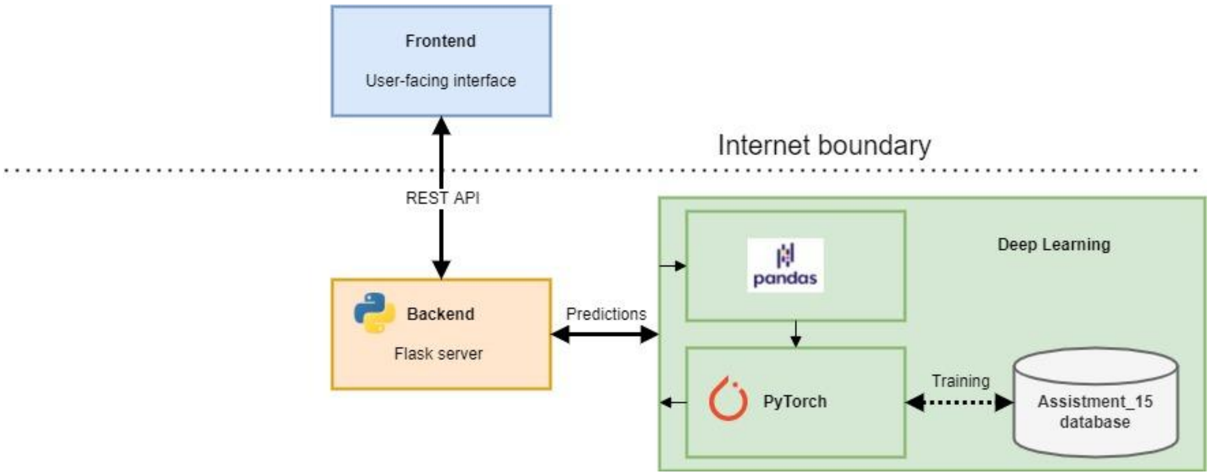
- Simultaneous answer logs were combined into one to prevent the model from cheating by leveraging frequently co-occurring answer tags in the raw dataset,
- Dummy skill tags were discarded to minimize the noise in the training dataset.

In the preprocessed dataset, every student was referred to by a unique “student_id” and every skill was referred to by a unique “skill_id”, these two identifiers are used to query a student’s performance on our WebApp.

After experimentation, we found that 32-dimensional hidden vectors worked best for the GNN. Usual deep learning techniques such as dropout(rate=0.5) on the hidden to output vectors, batch normalization on the output layers were used. Adam optimization method was used for the learning process(learning rate set to 0.01) with a batch size of 16(optimally chosen after experiments based on the compute resources at our disposal).

A student would typically sit for assessing their skill levels on the set of skills available in the dataset by taking a dedicated set of assessments. The pretrained graph-based model would output the predicted performance of the student after every assessment which could be used by the human/AI teacher to alter the course of offering future assessments and concepts. The model would also update after every assessment taking into account the assessment offered and the student's answer at any given time.

Block diagram



How to execute

- Hardware Requirements: Nvidia GPU with CUDA support.
- Install Python>=3.7
- Install required Python and NodeJS dependencies:
    - Python: `pip3 install -r requirements.txt`
- Run `python server.py`
- The Backend module (server.py) is a Flask app that runs on `localhost:5000`.
- Open the proposed ITS application on `localhost:5000` in a browser.

Comparison with existing tools and software

Most current knowledge tracing implementations use either Deep Learning models(Deep Knowledge Tracing; DKT) or Bayesian models(Bayesian Knowledge Tracing; BKT) at their core. While these have been shown to perform well on qualitatively and on quantitative metrics, one major drawback of such implementations is the lack of balance between accuracy and interpretability. Bayesian methods have high interpretability but don't perform very well quantitatively. Deep Learning methods perform well quantitatively but lack the inherent interpretability which Bayesian methods provide. Our approach builds on a recent line of work that employs graph-based knowledge tracing methods to model student performance, at the core of such methods are a class of neural networks called Graph Neural Networks(GNNs). GNNs are able to match and even improve the performance of DKT methods and are inherently interpretable. To improve upon the current line of work using GNNs we use Graph Convolutional Networks instead of simple Graph Neural Networks to better propagate the relationships between learned concepts leading to better prediction performance.

The use of GNNs/GCNs as the backbone of the Knowledge Tracing model makes it easy to visualise the change of a student's knowledge states over time, this can in-turn help the designated teacher(AI/human) to alter the course of future learning to maximise the expected performance of the student.

Deep learning models typically used in DKT methods use the same single hidden state vector for all the concepts which make modelling longer-term temporal dependencies between concepts very difficult. These models are only able to learn the dependencies of those concepts that are answered in temporally close order. Our work with graph-based models such as GNNs/GCNs is not faced with such issues.

Analysis and Conclusions

Student Proficiency Modelling is a vital aspect of ITS. As mentioned above, the improved performance of GNN enables the content to be more personalized which allows the students to master the coursework more efficiently.

In this work, we proposed and improved graph-based knowledge tracing methods based on GNNs/GCNs. Casting the knowledge structure as a graph, we reformulated the knowledge tracing task as a time-series node-level classification problem in the graph. Validation on the open ASSISTments 2009 “skill-builder” dataset, showed the method's capability to provide accurate and interpretable predictions. Our graph-based models give higher AUC values(0.723) than deep learning based DKT models.

Due to the inherent interpretability of graph-based models, it becomes easy to visualise the evolution of a student's knowledge states over the course of offering concepts and assessments.

Future Improvements

Future iterations of the application can incorporate more advanced graph-based models such as GraphSAGE and GAT(Graph Attention Network) which are able to model both spatial and temporal dependencies in the longer term more efficiently than GNNs/GCNs. Improvements on the overall application could see the addition of a visual representation of the knowledge states of the students on the said number of skills such that both the student and the teacher can have live feedback of the student's proficiency. Once the dataset grows large enough, distributed training and processing techniques could be employed to deploy this system at a large scale

Team

Gauransh Sawhney	2018A3PS0325P
Divyansh Agarwal	2018A3PS0791P
Utkarsh Kumar Singh	2018A3PS368P