CS230

Deep Knowledge Tracing and its Variants

Nuntanut Raksasri
nr2747@stanford.edu

Tianqi Yan
tianqi@stanford.edu

Chao Xu
xc0125@stanford.edu

Abstract

Knowledge tracing is the task of modeling a student's knowledge acquisition and loss process based on the student's past trajectories of interactions with a learning system. The ability to model student knowledge has a high educational impact. With the advent of deep learning, lately there has been significant performance improvement in the RNN based methods. one example is the Deep Knowledge Tracing (DKT). In this paper we continued on the work of DKT, by exploring new features and applying the latest technique of embedding, showing a potential new approach to the knowledge tracing problem.

1 Introduction

Knowledge tracing is a task of modeling student knowledge over time so as to predict a student's future performance. A simple way to formalize this task is to take a sequence of students' interaction on a task, i.e. students' attempt on related questions as well as the outcome of the attempt, and try to predict the outcome of the future interactions. This set-up suggests a solution of a sequence model. C. Piech showed an RNN-based model, the Deep Knowledge Tracing, with remarkable performance. One issue of the set-up is how to capture knowledge. While formalizing the task, we have been using skill tagged to a question as a proxy of knowledge. However, such tags are heavily subjective to the annotators, and they might not be able to capture the true characteristics of knowledge involved.
In this paper, we tried a novel approach of embedding the skill features. Namely, convert from the dimensions of skill tags to the dimension of "knowledge", and in turn to achieve better prediction.

2 Related work

There are several related works, as we know, there are four best-known modeling methods for estimating student performance. The first one is Item Response Theory(IRT), which assumes the student knowledge state is static and represented by her proficiency when completing an assessment during an exam. The second one is Bayesian Knowledge Tracing(BKT), which introduces the learning environment into the model. The third one is Performance Factor Analysis(PFA), it is similar to BKT, but it includes multiple skills simultaneously with its basic structure. The last one is our project working on– Deep Knowledge Tracing, it is totally different from IRT, it uses a Long Short-TermMemory (LSTM) to represent the latent knowledge space of students dynamically.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Features

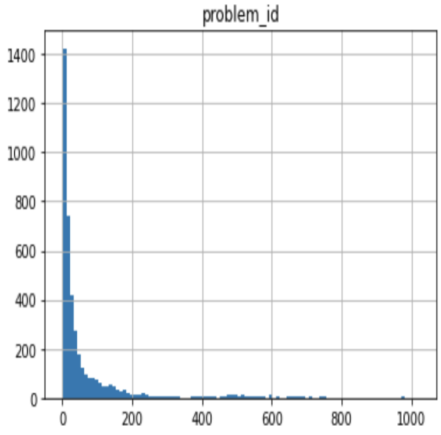
The data used in this project is the 2009-2010 ASSISTment skill builder data set. Here are some basic statistics for the data:

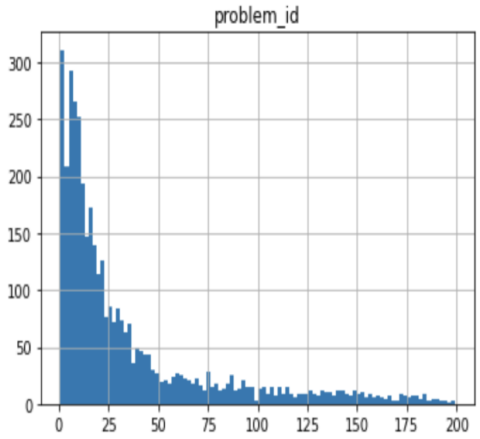
- Number of rows : 459208
- Number of distinct records : 283105
- Number of students : 4163
- Number of problem sets : 620
- Number of problems : 17751
- Number of skills : 123
- Number of answer types : 5

The original dataset contains many columns. We shortlisted the few columns that are relevant in the project:

- correct (the output to predict as well information to feed in the network)
- skill_id (the skill tag)
- answer_type (type of answers, i.e. open-ended vs MCQ)
- ms_first_response (time to deliver the answer)
- hint_count (number of hints used)

Correct, skill_id and answer_type are categorical variables while ms_first_response and hint_count are continuous variables. The original dataset contains questions that are without any skill tag. As we considered skill tags being the most important feature, the question attempts without any skill tag were filtered out. For the multi-skill problem, we allow the one-hot vector representing the skill tags to have value 1 for all the dimensions of its skill tags.
As different students behave differently, it does not make sense to combine different students' attempts to the same sequence. Hence we constructed the sequence within each student's attempts. According to the distribution of the number of questions answered by students (see figure below), we think about 15 to 50 is a fair value of length of the sequence. For students answering less, their sequence will be padded with 0 and masked from training, while for those answering more, their attempts will be truncated into multiple sequences.





4 Methods

4.1 Model

Instead of comparing the performance between different training algorithms, this project focused on deep dive in a model of LSTM and feature embedding. The nature of the problem implies the

solution being a sequence model. In addition, previous work done by C. Piech suggested better performance of LSTM than a normal RNN, hence we decided to adopt the LSTM structure. To avoid forward looking biases, the LSTM is unidirectional. To enhance the results of existing work, we tried to add two novel steps: feature embedding and feature enrichment.

Feature Embedding

Embedding is a technology developed later, mainly used in the area of natural language processing, to capture characteristics of works. We think this technique is helpful too in finding characteristics of skills.

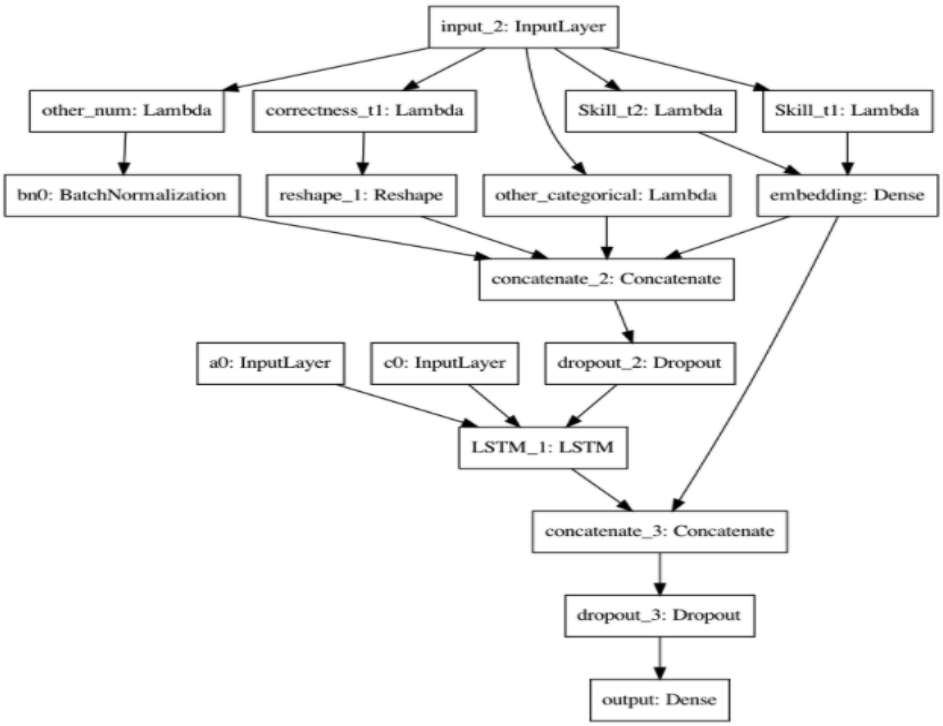
Conventionally, the skill tags feeded into the model are represented as a one-hot encoded vector, which implies the skill tags are orthogonal to each other, mathematically. In reality, such an assumption is not true. For example, the skill of computing area of a rectangle is much more related to another skill of computing area of parallelogram. Hence, it is meaningful to map such one-hot encoded skill set vectors to other dimensions of features representing the skills' characteristics.

Feature Enrichment

The work done by C. Piech mainly focuses on how past interactions imply future behaviour of students, while the past interactions are solely based on the type of question and students' response. We think there are also other features that provide information reflecting on students' grasp of the knowledge. For example, we think the question types should be relevant, with the hypothesis that a short structure question should reflect more than a Multiple Choice question. Furthermore, the time used by a student on solving the problems also reflects how well the student masters the skills. Therefore, we also introduced the features of question types, time taken as well as hint counts to the model.

Hyperparameters

The set of hyperparameters we tried to fine tune are: length of the sequence model; number of hidden states of the LSTM layer, output dimension of the skill feature embedding, A visual representation of the final model is presented in Figure below.



4.2 Input and Ouput

The input to the model is a time series of the metadata (the features specified previously) of questions tried as well as its correctness (binary with value 1 or 0). The output is a time series of the question correctness, but shifted one step to the future, so that we are not predicting what is known.

Refer to the model design in above figure, at each step, the one-hot vectors representing the skills will go through an embedding, together with other categorical and numerical variables, as well as the answer correctness of the same question will be feeded into a classic designed LSTM layer. The output activation state, together with the next question's metadata, will be feeded into another sigmoid layer, to predict the next question's answer correctness.
To avoid overfitting, dropout layers are introduced for regularization. The model is optimized using Adam optimization with a binary cross entropy loss on the predicted answer correctness.

5 Evaluation of Results

The dataset was divided into train/dev/test with the ratio of 90/5/5. We used a train/dev set validation for hyperparameter tuning, from a relatively small network of 8 steps LSTM with 64 dimensions of embedding output and 64 hidden states to a large 48 steps LSTM with 128 dimensions of embedding output and 256 hidden states.

Tx	embedded_size	n_a	epoch	loss	AUC	accuracy
24	64	128	1000	0.523837	0.794549	0.735237
24	64	256	1000	0.551081	0.806699	0.739751
24	128	64	1000	0.521523	0.789170	0.733407
24	128	128	1000	0.525625	0.802338	0.737311
24	128	256	1000	0.610221	0.811663	0.748536
32	64	256	2000	0.515792	0.848573	0.777174
32	64	64	1000	0.488892	0.816296	0.749660
32	64	128	1000	0.482036	0.827879	0.752774
32	64	256	1000	0.513598	0.848195	0.777797
32	128	64	1000	0.488088	0.819202	0.748019
32	128	128	1000	0.491482	0.833430	0.760983
32	128	256	1000	0.512912	0.831631	0.759171
48	64	256	2000	0.462076	0.886407	0.807788
48	64	256	2000	0.462076	0.886407	0.807788
48	64	256	2000	0.462076	0.886407	0.807788
48	64	256	2000	0.462076	0.886407	0.807788
48	128	256	2000	0.491512	0.893681	0.814137

An observation is that the validation AUC generally increases as the model becomes more complex, and the performance is most sensitive to the length of the sequence, which is aligned with our expectation. The longer the sequence represents more interactions we captured of a student,

hence more accurate in predicting the future performance. Choosing the model with a sequence length of 48, embedding output dimension of 128 as well as 256 activation states, we manage to achieve an AUC of 0.89, a slight improvement of the baseline DKT performance (0.86).

6 Discussion

In this paper, we extended the work of DKT by applying an LSTM with feature embedding, and achieved a slight improvement in the performance.
Though the focus of the paper is on the prediction of future student performance, we think the trained embedding matrix itself contains information that is interesting to study. Since it maps the skills to other characteristics dimensions, it may reveal some interesting relations between the skill tags, like those in word2vec. We will leave this in future studies.

References

[1] Piech, C., Bassen, J., et al. "Deep Knowledge Tracing"
https://stanford.edu/ cpiech/bio/papers/deepKnowledgeTracing.pdf

[2] https://github.com/ckyeungac/deep-knowledge-tracing-plus

[3] https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data

[4] Yeung, C., Yeung, D., "Addressing Two Problems in Deep Knowledge Tracing via Prediction-Consistent Regularization"
https://arxiv.org/pdf/1806.02180.pdf, https://github.com/ckyeungac/deep-knowledge-tracing-plus

[5] Jiani Zhang, Xingjian Shi., et al. "Dynamic Key-Value Memory Networks for Knowledge Tracing"

[6] Liang Zhang, Xiaolu Xiong., et al. "Incorporating Rich Features into Deep Knowledge Tracing"