How to Predict Students' Interactions with Diagnostic Questions: from A Perspective of Recommender System

Anonymous Author(s)
Affiliation
Address
email

Abstract

In the NeurIPS 2020 Education Challenge, competitors focused on the students' answer records to these multiple-choice diagnostic questions, with the aim of task 1&2: accurately predicting which answers the students provide; task 3: accurately predicting which questions have high quality; and task 4: determining a personalized sequence of questions for each student that best predicts the student's answers. In this study, we describe the solution of our team N&E, using which we won the second and fourth places, in task 1 and 2 respectively. We treat the two tasks as a Recommender Systems (RS) problem, solve them with three different methods including GBDT, a multi-head attention based network and a transformer based network, both of them can achieve competitive results respectively. And at last, we integrate their results to get our best predictive accuracy.

1 Introduction

Diagnostic questions, which can reveal key information about the specific nature of misconceptions that the students may hold, are hence important education resources. Analyzing the massive quantities of data stemming from students' interactions with these diagnostic questions can help us more accurately understand the students' learning status and thus allow us to automate learning curriculum recommendations. The task 1&2 of NeurIPS 2020 Education Challenge focused on this issue. It aimed to build models that could infer if a student can answer a given question correctly or not (in task 1), even can predict which answer the student would choose for that question (in task 2).

Recommender systems (RS) is one of traditional task in the community of Machine Learning (ML), and many novel methods have been developed for it. Computationally, the task 1 of NeurIPS 2020 Education Challenge shares three features with RS: 1) the interactions can be categorized into two types, for example the answer from a student to a question is right or wrong in task 1, while a user is interested in a target item or not in RS; 2) the participants of interaction can be split into two completely independent groups, for example student and question in task 1, user and item in RS; 3) the attributes in given data of task 1 are similar with RS, for example both of student and user have birthday and gender details, question and item have categories information. Due to these apparent similarities between task 1 and RS, it seems natural to integrate various successful techniques which has been widely used in RS to this challenge.

Here, three methods have been utilized in our solution for task 1: first, we extract some features and apply a GBDT method to construct our baseline; then, we implement a multi-head attention based neural network model to enrich the interactions between different features; at last, use the powerful transformer model to capture the sequential signals underlying students' answer sequences. Both of them can achieve competitive results respectively. After integrating their results using a multi-layer

Submitted to the 1st NeurIPS Education Challenge (NeurIPS Education Challenge 2020). Do not distribute.

perceptron (MLP), we got our best score. For task 2, we use the same methods while changing the objective function from binary to multi-classification.

Our work is organized as follows. In Section 2, we summarize the related works in RS. Section 3 details the three methods and the structure of final model-ensemble MLP. In Section 4, we present the experimental results of different methods.

2 Related work

Recommender systems (RS) have evolved into a fundamental tool for helping users make informed decisions and choices, especially in the era of big data in which customers have to make choices from a large number of products and services. One challenge in recommender systems is to achieve both memorization and generalization [6].

A lot of RS models and techniques have been proposed, among them generalized linear or decision tree-based models with nonlinear feature transformations are widely used for large-scale regression and classification problems with sparse inputs. To achieve memorization and generalization effectively, some manual feature engineering must be applied before feeding into model, beyond that, they do not generalize to query-item feature pairs that have not appeared in the training data [2].

Embedding-based models, such as factorization machines [4] or deep neural networks [2, 3, 5], can generalize to previously unseen query-item feature pairs by learning a low-dimensional dense embedding vector for each query and item feature, with less burden of feature engineering.

Despite the success of those RS framework, it is inherently far from satisfying since it ignores one type of very important signals in practice, i.e., the sequential signal underlying the users' behavior sequences. Therefore, recent works try to incorporate sequential signal of users' behavior sequences into RS, such as DIN [7], BST [1].

In our final solution, we utilize the aforementioned three types of RS techniques.

3 Approach

In our final solution for task 1&2, we integrate the results from three methods: feature engineering & GBDT, a multi-head attention based neural network and a transformer-based model.

3.1 Feature engineering & GBDT

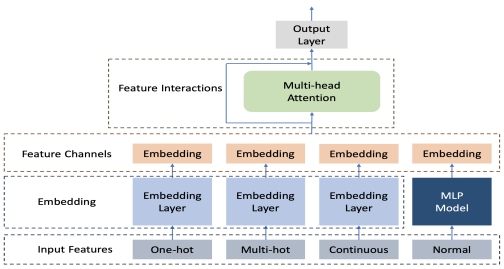
Feature engineering & GBDT is one most widely used methods in RS. Here, we extract 17 features for task1&2, including some features which can be obtained directly from data (for example: UserId, QuestionId, SubjectId, Gender, PremiumPupil, GroupId, QuizId, SchemeOfWorkId, Confidence), and some other features which need further processed (see Table 1 for details).

Table 1: Features

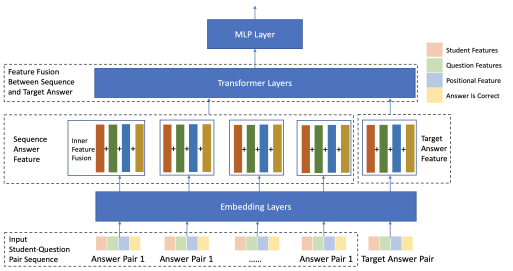
Feature Name	Description
HourAnswered	The answer date, split in hour.
AgeAnsweredHalfYear	The age of student when answer the given question, split in half year.
SubjectId_Level0-3	The topic IDs associated with a question. Only on subject level 1-4.
SubjectHistoryAnswer	The answer accuracy on each subject before answer current question.
AnswerCostTime	The cost time for a student to answer current question.

3.2 Multi-head attention based neural network

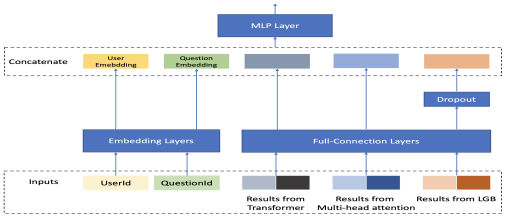
Attention can help to capture high-order feature interactions without applying special manual feature engineering, therefore have been utilized in RS recently. As shown in Figure 1a, Our multi-head attention based neural network is similar to AutoInt, except that it has four feature types and its normal features are embedded by an MLP model. In details, the input features in table 1 can be categorized as following four types:



(a) Transformer-based neural network



(b) Transformer-based neural network



(c) Transformer-based neural network

Figure 1: Model structures

- One-hot feature: mutually exclusive category features. For example, “UserId”, “QuestionId”, “Gender”, “PremiumPupil” and so on.
- Multi-hot feature: category features which may occur together. For example, “SubjectId”.
- Continuous feature: a single numerical value. For example, “AnswerCostTime”.
- Normal feature: a vector of numerical values. For example, “SubjectHistoryAnswer”, whose length is 388, denotes the number of all the subjects.

All of the feature embeddings are organized as different channels for an inputting Student-Question pair, and feed into multi-head attention feature interaction module.

3.3 Transformer-based neural network

Inspired by the great success of the transformer for machine translation task in natural language processing (NLP), we apply the self-attention mechanism to learn a better representation for each item in a user’s behavior sequence by considering the sequential information in embedding stage, and then feed them into MLPs to predict students’ responses to candidate questions. As shown in Figure 1b, Our Transformer-based model is similar to BST, while it has three differences in feature fusion:

- Input sequences contain both student and question features, while only consider item features in BST.
- We add all the embeddings in our inner feature fusion module, while in BST, concatenate is applied.
- We only keep the interactions of sequence answer and target answer features, while ignoring the interactions between sequence answer features. Different from BST, in which, all of the feature interactions are considered and concatenated to feed into MLP layer.

In transformer-based neural network, we utilize four types of features:

- Student Features, including UserId, GroupId, QuizId and Confidence.
- Question Features, including QuestionId and a categorical feature that calculates the similarity between questions based on subjectid.
- Positional Feature, is computed as $pos(v_i) = t(v_t) - t(v_i)$, where $t(v_t)$ represents the answer time of target Student-Question pair and $t(v_i)$ the timestamp when student answer question v_i.

- Answer is correct: indicating if student answer question vi correctly or not, note that it must be set as 0.5 (denotes unknown) for validation and test datasets.

3.4 Model ensemble

As show in Figure 1c. The results from aforementioned three models are ensembled using following models. All of prediction probabilities including the "UserId" and "QuestionId" are feed into model. After passing embedding layer and full-connection layers, the results are concatenated to decide the final prediction results. Note that, we only adopt dropout for the results from LightGBM since in our scenarios it outperforms other strategies.

4 Experiments

4.1 Settings

Table 2: Model Setting for Multi-head attention and Transformer based model

Parameters	Description	In Multi-head attention	In Transformer
Embedding size	Embedding layer output size.	64	128
Attention dim	Attention layer hidden size.	128	768
head number	Self-attention head number.	4	12
Interaction layer number	Interaction layer number.	3	1

For LightGBM, we just used its default model parameters. The settings of Multi-head attention and Transformer-based models are given in Table 2. In ensemble model, we set embedding size to 4, hidden size to 32 and dropout=0.95.

4.2 Results

The results of each methods on the public leaderboard denotes that both of three mothods can achieve competitive results respectively, for emample LightGBM got 0.7581, Multi-head attention got 0.7594 and Transformer-based model got 0.7661 on the public leaderboard of task1. And our best prediction accuracy on the private leaderboard is 0.7706 and 0.6708 in task 1&2 respectively.

References

[1] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pages 1–4, 2019.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[3] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He, and Zhenhua Dong. Deepfm: An end-to-end wide & deep learning framework for ctr prediction. *arXiv preprint arXiv:1804.04950*, 2018.

[4] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[5] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1161–1170, 2019.

[6] Shoujin Wang, Longbing Cao, and Yan Wang. A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864*, 2019.

[7] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068, 2018.