ADL, Fall 2022

Final Project - Report

郭宜婷	廖政華	王浚亦	廖郁珊
R10922193	R11922133	R11922159	R11944053
資工所	資工所	資工所	網媒所

January 4, 2023

Abstract

In this project, we try to use various methods and techniques to recommend Hahow courses to their users. After many discussions and attempts, we finally decided to use the Hybrid method in seen domain and DeepFM in unseen domain, which seem to be the most suitable and have the highest accuracy.

1 Introduction

Hahow is an online course platform, teachers can upload their courses on the web, and users can buy these courses and get various skills. In this project, they provide their own datasets which contain user information, purchase history, and course information. Hope to find the correlation between user information and courses. Our task is to do the prediction on the courses that the users may buy in the future.

The task is divided into two parts, seen domain and unseen domain. The users in seen domain can predict courses based on their purchase history, while users in unseen domain have no purchase record, we can only compare the users' similarity to infer their purchase tendency. Therefore, we try to apply different methods on two domains, which the result of seen domain may have better performance than unseen domain.

2 Related Work

2.1 DeepFM

The DeepFM[1] model consists of two parts: FM and DNN. The FM model is responsible for extracting low-order features, and the DNN model is responsible for extracting high-order features. Different from Wide&Deep, DeepFM doesn't need feature engineering, it can map each field of the sparse matrix to k-dimensional embedding, which we think that is suitable for the dataset in this final project. By integrating low-order feature with high-order features, the performance of our recommended system achieved high accuracy in the experiment.

2.2 Recommender Systems

The common approach to build a recommender system is using collaborative filtering (CF). However, we face a problem that how can we convert the course and user information into embeddings which can represent themselves. DNCF[2] utilizes dual embeddings which combine the primary embeddings with additional embeddings obtained from historical interactions to get final users' and items' representation. This method can use historial information to get latent factor matrix for users and items. And DLRM[3] uses multi-hot vector to handle categorical features. It utilizes embedding tables for mapping categorical features to dense representations which generalize the latent factors used in matrix factorization. Inspiring by those methods, we apply similar concept to do our data processing.

3 Approach

This section introduces our approach for course recommendation. For subgroup recommendation, we simply aggregate the subgroup labels of the recommended courses.

3.1 seen

The main strategy here is to recommend courses similar to user's purchase history. Specifically, we first obtain the course-course similarity matrix. Then multiply it by the user vector, a 728-dimension one hot encoding by his/her purchase history, as the user's preference value over the courses.

The course-course similarity is the linear combination of two components, co-purchase frequency and content-similarity. The former calculates the co-occurrence in the same user's purchase history for each course-course pair. The latter builds a TF-IDF index for each courses on its textual columns (such as `description`).

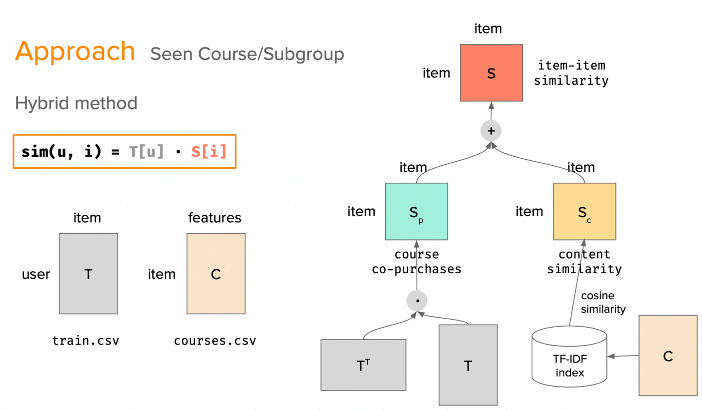

Figure 1: Illustration of seen-domain approach.

3.2 unseen

For the unseen domain, we treat it as a multi-label classification task given user profile (i.e. `gender`, `interests`, `occupation_titles`, `recreation_names`). Since these columns are categorical and multi-valued, they are first mapped to a dense embedding, mean-pooled, and concatenated as the model input. Then, the input embedding is fed to the factorization-machine module and a deep-neural network. Finally, their outputs are concatenated and fed to a linear layer to obtain output logits. For training, we use Adam as optimizer with learning rate of 5e-2, batch size of 256, and binary cross entropy loss as loss function. For inference, we select the top-50 courses in terms of output logits as our recommendation.

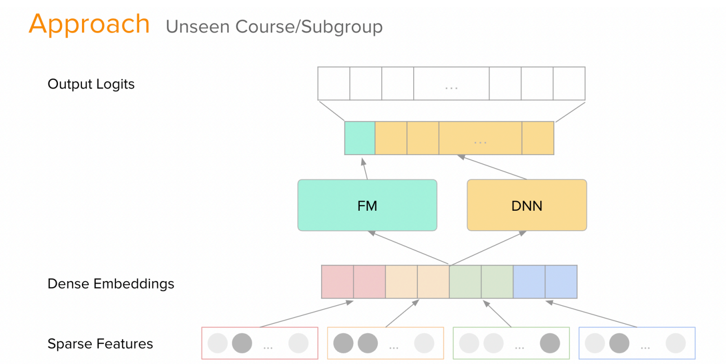

Figure 2: Illustration of unseen-domain approach.

4 Experiments

Tabel 1 shows the performance of our final recommended system. In course prediction, we use the Hybrid method in seen domain and DeepFM in unseen domain. In subgroup prediction, we use outputs of recommended courses to predict the subgroup.

	Kaggle Score	Validation Score
Seen Course	0.07489	0.0802
Seen Subgroup	0.27152	0.2934
Unseen Course	0.07143	0.0798
Unseen Subgroup	0.25386	0.2646

Table 1: Accuracy of our system

Next, we want to know whether the accuracy can be improved by using different user's datasets to train the model. First, we try to delete some users(about 8%) who only have one feature, which seems hard to make a prediction. Second, we try to generate extra users for training, the detail of the method will be introduced in section 5.2. Table 2 compares these results.

	original	extra users	less users
Unseen Course	0.0714	0.0685	0.0641

Table 2: Accuracy of the models using different user's datasets in unseen domain.

5 Discussion

In this section, we discuss some issues we find and methods we try to solve these problems.

5.1 User Preprocessing

First issue is about how to encode user's features, we try two approachs, one-hot encoding and frequency encoding. If we use one-hot encoding, the feature vector will be too sparse which lead the training fail. Then, we try frequency encoding to count the number of times the features occur together, but still can't successfully train the model.

We also try to divide users into several groups by using their feautres, so when a new user join, the model can divide him/her into similar groups, then predict the course's subgroup by those users who already in training data.

Unfortunately, above approaches don't get good result in experiment. Finally we use deepFM, which take the sparse features as inputs, then the model will learn how to generate 2-dimension dense embedding by these input features.

5.2 Generate Extra Users

In our model, if some courses don't appear in training data, then it will be hardly to predict when inferencing. To solve this problem, we try to create extra users to buy those courses by picking features from users who buy the similar courses. However, it might be some flaws in creating users, this approach doesn't get better score in experiment.

5.3 Seen Subgroup

Comparing to conventional matrix factorization model, which uses mean square error as loss function, the regularized matrix factorization model can learn the embeddings of irrelevant items by adding some regularization terms. We apply such RMF model to predict the subgroups of courses that the seen users

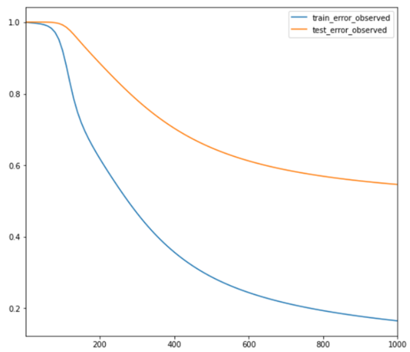

Figure 3: Training curve of training by courses

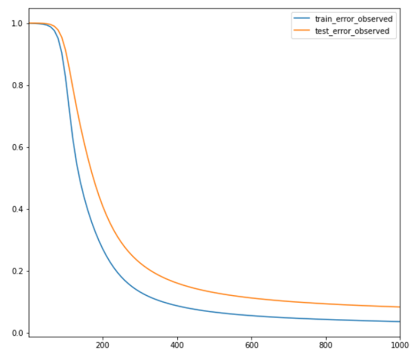

Figure 4: Training curve of training by subgroups

would buy. Figure 3is the training loss and validation loss in 1000 steps by the training data of courses. And Figure 4 is training by subgroups directly. The figures show that training by subgroups will achieve better performance in this case.

In terms of prediction, computing the scores with the dot product of query embedding and each of the item embeddings will result better rather than computing cosine similarity. The possible reason is that using dot product have a tendency of predicting more popular courses or subgroups, which in this case is great for those testing users.

5.4 Unseen Course

We tried different approaches in this section, the first method is that we compute the similarity between the unseen user and the train user, after that we can push back the courses that the unseen may like. The way of computing the similarity is that we first encode the interests of the user into the one-hot format, then compute the cosine similarity of each unseen user and total train user, and select the top few train users with the highest similarity. By matrix factorization, we can obtain possible course information that train users may buy. According to this process, we can calculate the courses that the train user may purchase as the possible purchase courses of the unseen user. However, this method didn't perform so well. We assume the reason is that when calculating cosine similarity, we only use the one-hot format with only 0 and 1, so it cannot return the real similar users.

The second method is that we test the BM25 text retrieval model as suggested by the TA. Since it is a kind of retrieval method, it is needed to generate documents and query lists. We package course information into a document, including teacher introduction, course description, what will learn in the course, target user, and the subgroup of the courses, and do the tokenization. For the query form, we make the interest, occupation, and recreation of users into the query form. This method indeed gives us a better result compared to the first method.

However, in the second method, we don't use the information from the training dataset, so we think it is quite a pity. Hence, our idea is that we want to cluster the user information, trying to find out which group the unseen user and the training user belong to. In the query of unseen users, we add the course information that train users have bought into it. But in this test, it didn't perform so well.

6 Conclusion

A recommendation system trained by real world data is hard to develop. In this task, making predictions using only user information mostly get low accuracy. Therefore, after experimenting with multiple models, we apply a hybrid method to conduct this real world prediction task. The experimental results show that predicting subgroup by using the result of course prediction always better than other methods. Hence, we apply the same approaches for predicting courses and subgroups in the same domain. Finally, we obtain the result that predicting seen users is better than predicting unseen users with our proposed method, which is meet our anticipation.

7 Work Distribution

The work distribution is shown in table 3.

	Seen Domain	Unseen Domain	Presenation+Report
郭宜婷	✓		✓
廖政華	✓	✓	✓
王浚亦		✓	✓
廖郁珊		✓	✓

Table 3: Work Division

References

[1] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for ctr prediction," in *26th International Joint Conference on Artificial Intelligence*, pp. 1–8, 2017.

[2] G. He, D. Zhao, and L. Ding, "Dual-embedding based neural collaborative filtering for recommender systems," *arXiv preprint arXiv:2102.02549*, 2021.

[3] M. Naumov, D. Mudigere, H.-J. M. Shi, J. Huang, N. Sundaraman, J. Park, X. Wang, U. Gupta, W. Carole-Jean, A. Alisson G., D. Dmytro, A. Mallevich, I. Cherniavskii, Y. Lu, R. Krishnamoorthi, A. Yu, V. Kondratenko, S. Pereira, X. Chen, W. Chen, V. Rao, B. Jia, L. Xiong, and M. Smelyanskiy, "Deep learning recommendation model for personalization and recommendation systems," *arXiv preprint arXiv:1906.00091*, 2019.