CSIT 6000M Project-2 Proposal

Hexiao Zhang
20932780
hzhangea@connect.ust.hk

Ke Zheng
20882179
kzhengah@connect.ust.hk

1 Background

A recommendation system is a retrieval-ranking system, and a key step in the recommendation task is the prediction of user action probabilities using user, contextual and item features. Features play a central role in the success of many recommendation systems. Using raw features, however, can rarely lead to optimal results. That's the reason why data scientists spend a lot of work on the transformation of raw features to generate combinatorial features to improve recommendation performance. Manual feature engineering methods are often very expensive due to the variability and size of the data. Traditional Factorization Machines (FM) embed each raw feature to a latent vector, pairwise feature interactions are modeled as the inner product of latent vectors. However, traditional FMs modeling the interactions with useless features may introduce noises and degrade the performance. As computing power grows rapidly these years, deep neural networks (DNNs) has shown strong ability in learning useful high-order feature representations. However, DNNs model high-order feature interactions in an implicit fashion. The final function learned by DNNs can be arbitrary, and there is no theoretical conclusion on what the maximum degree of feature interactions is. In this project, we plan to reproduce xDeepFM which learn feature interactions in an explicit, vector-wise fashion. Based on xDeepFM we will make some improvements.

2 Problem Definition

A recommendation system is a retrieval-ranking system. The original training data acquisition process of a recommendation system is as follows: the user visits the relevant website or application to generate user features and context features (we call them queries); The recommendation system returns a list of items (called impressions); the user performs an action on the item, which can be clicking, buying, etc. User features, context features, and item features are used as the input data, and user operations are used as labels, which together constitute the training data of the recommendation system.

The complete recommendation process is as follows: a user visits a website to generate a query, including user and context features; the retrieval system obtains a short list of items that best match the query from a large database based on a machine learning model or human-specified rules; , context and item features predict scores, and use scores for ranking. Scores are usually conditional probabilities of user actions given the acquired features.

The recommendation task discussed in this paper refers to the score prediction task of the ranking system. The input is feature x, including user features (such as age, gender, language, etc.), contextual features (access time, equipment, access records, etc.) and item features (item category, item history statistics, etc.). Typically, the output is designed as the conditional probability P(y|x) of the user performing an operation y given the feature x.

3 Related Work

This section briefly introduces the development of click-through rate (CTR) prediction algorithms and the application of deep learning in this field.

Before the rise of deep learning, CTR prediction was often achieved through manual feature engineering. This required a lot of expert experience, could not be generalized across different datasets, and was unable to extract high-order feature interactions. Factorization Machine (FM) [1] maps original features to embeddings and models the correlation between features through the inner product of the embeddings. The weighted sum of the original features is used to obtain the final prediction output. The formula is shown below:

$$y = \sum_{i,j} < V_i, V_j > x_i \cdot x_j \tag{1}$$

where x_i and x_j represent original features, and $< Vi, Vj >$ represents the inner product of the embeddings of the original features. FM can automatically implement pairwise feature interactions and is simple and effective. It is the foundation of many feature interaction models that follow. In theory, FM can be extended to high-order feature interactions, but this can lead to high complexity. Additionally, in recommendation systems, FM maps these zero vector features to non-zero embeddings, which can lead to learning too many useless interaction terms and introducing noise.

With the development of Deep Learning, researchers began to apply the powerful feature-learning ability of neural networks to recommendation systems. [2] and [3] respectively used Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for rating prediction in recommendation systems. CNN focuses more on the interaction between adjacent features due to the property of convolution. RNN has better modeling performance for sequential feature interactions due to its time characteristics.

The above DNN-based recommendation system models for learning high-order feature interactions ignore low-order feature interaction relationships [4].

DeepFM [5] proposes an end-to-end network structure that combines FM and DNN. The FM module is a fusion of FM and linear prediction models, which can learn both low-order and high-order feature interactions and does not require manual construction of cross-product transformations. xDeepFM [6] further improves by proposing a Compressed Interaction Network (CIN) to learn feature interactions, and the degree of feature interactions deepens as the network layers increase. At the same time, each layer of CIN interacts at the vector level.

In this project, we plan to reproduce xDeepFM and make some improvements.

4 Dataset

The Criteo Advertising Dataset[1] is a publicly available and classic dataset used for predicting ad click-through rates. In 2014, the globally renowned advertising company Criteo sponsored the Display Advertising Challenge competition. In the Criteo dataset, each row represents a sample, where each sample includes:

- a label indicating whether the ad was clicked (1) or not clicked (0);
- 13 dense features (numerical features) named I1-I13;
- 26 sparse features (categorical features) named C1-C26, where the values of these features are hashed to 32 bits for privacy reasons. When a feature is missing, the field is empty.

The dataset consists of a training set (10.38 GB) and a test set (1.35 GB), and is widely used as a benchmark in recommendation system research.

[1] https://labs.criteo.com/2014/02/download-kaggle-display-advertising-challenge-dataset/

Table 1: Samples from the Criteo Dataset

label	I1	I2	I3	...	C23	C24	C25	C26
0	1.0	1	5.0	...	3a171ecb	c5c50484	e8b83407	9727dd16
0	2.0	0	44.0	...	3a171ecb	43f13e8b	e8b83407	731c3655
0	2.0	0	1.0	...	3a171ecb	3b183c5c	Na	Na
0	Na	893	Na	...	3a171ecb	9117a34a	Na	Na

5 Research Plan

The research plan for this project includes the following steps:

- Read relevant papers on click-through rate prediction, summarize from different periods based on the development of the field, and the main methods proposed in these papers. Additionally, we also want to identify improvements made in subsequent algorithms compared to previous ones.
- Reproduce the xDeepFM model with the assistance of open-source code[2], and try to make some improvements in terms of feature engineering and evaluation metrics.
- Test the model using commonly used evaluation metrics in the field (AUC and logloss), provide visualized results of the model training.

We hope that through this assignment, we can practice our ability to read papers, summarize algorithms, and reproduce relevant codes.

6 Feasibility

We have an RTX 3090 GPU available. Nevertheless, it would still take much time to train on the entire Criteo dataset. We may take a portion of the training set without affecting the reproduction result.

References

[1] Rendle, S. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[2] Liu, Q., F. Yu, S. Wu, et al. A convolutional click prediction model. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1743–1746. 2015.

[3] Zhang, Y., H. Dai, C. Xu, et al. Sequential click prediction for sponsored search with recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28. 2014.

[4] Zhang, W., T. Du, J. Wang. Deep learning over multi-field categorical data. In *European conference on information retrieval*, pages 45–57. Springer, 2016.

[5] Guo, H., R. Tang, Y. Ye, et al. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[6] Wang, R., B. Fu, G. Fu, et al. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. 2017.

[2] https://github.com/shenweichen/DeepCTR-Torch