2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR) | 979-8-3503-3426-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICBAR58199.2022.00038

Fast Recommender System Combining Global and Local Information

Construction of large-scale commodity information recommendation system

Xinyan Liu
Department of Mathematics in University of Washington
University of Washington
Seattle, United States
hathawayxinyan@gmail.com

***Abstract*—Large-scale recommender systems have important applications in online shopping and other fields. . In this paper, we propose a system that can balance the speed and accuracy of large-scale personalized recommender systems. To improve speed, we use GCN instead of the previous dense operation to obtain global information. At the same time, we use an attention mechanism so that various information can be fully integrated. The experimental results show that the method has obvious advantages in speed and accuracy, and the inference speed of only 109ms can achieve an accuracy of 83.1%.**

Keywords-recommender system, GCN, attention

I. INTRODUCTION

In nowadays technology era with explosive information, people are receiving more and more information at a progressively faster pace, and the diversity of information provides more possibilities for people to make choices in various aspects of their lives. But at the same time, along with the interchanging speed of receiving information, people are facing the problems of making meaningless choices, not being able to find the required information on time which increase the time cost, and unavailability of information. Therefore, personalized recommendation systems are born in the search for solutions to these problems.

Through filtering information, a personalized recommendation system uses the existing history of users' choices or other similar relationships to explore the potential interests of individual users and, through these interests, establish the relationship between users and corresponding products for further recommendation so as to improve the efficiency of personalized recommendation in the future by improving the user profile. In addition to improving the efficiency of user selection and receiving information, a personalized recommendation system also plays an important role in filtering invalid information among massive information and avoiding unnecessary selection.

Nowadays, personalized recommendation methods are mainly based on collaborative filtering methods, content-based recommendation methods, topic-based recommendation methods, or hybrid systems that combine any of the two. The main methods currently in use allow the current personalized recommendation system to achieve a certain degree of accuracy, but there are still many drawbacks with the above methods:

1. Long runtime and low efficiency of the algorithm. Most of the current personalized recommendation methods heavily rely on user information and product information to analyze user similarity for constructing users' portfolios, which reduces the overall efficiency of the algorithm since the database is too large to filter the initial information, which prolongs subsequent user portrait time.

2. Low accuracy of filtering. The current personalized recommendation method does not clearly categorize the initial information during the process of classification, which results in a large amount of useless information after filtering, or a large amount of useful information filtered out, which both result in a decrease in the accuracy of the algorithm.

3. Over-focus on the overall characteristics or on local information. Take collaborative filtering and content-based recommendation methods as examples; both of them focus more on individual information and tend to ignore the relationship between users during the process of improving individual information, which leads to the loss of the overall characteristics of the system; furthermore, in light of the topic-based model recommendation method, this method usually obtains certain features of a user or product to calculate a vector and uses this vector to calculate the connection between users and user or user-product connection. When focusing on the overall relationship, this recommendation method inclines to neglect user information and lead to missing information, which leads to user churn and loss of original user features.

In view of the above existing methods and their drawbacks, we propose our framework GConv-Transform to fulfill the imperfections of the existing methods by integrating the well-known Transformer (or attention in general) and graph convolution blocks. Particularly, we process the targeted data by using the ChebConv layer and followed by our own stack of multi-headed attention and the GConv layer by LAM-Gconv.

Our new framework intends to make the following improvements:

Instead of focusing on overall or details, we will adopt a parallel method of overall and details analysis to predict the user's interests and implement personalized recommendations

979-8-3503-3426-5/22/$31.00 ©2022 IEEE
DOI 10.1109/ICBAR58199.2022.00038

for the user's needs through the filtering of crucial information in general, combined with the user's personalized information.

In the process of information filtering, we will consider the following three parts: the availability of products/commodities, the balance of overall and user interests, and user-personalized information/personalized features. We aim to efficiently recommend customized information while fulfilling the overall users' requirements.

II. RELATED WORK

Based on the existing methods, the two most frequently used methods for organizing neighbor information and calculating similarities in between are Attention-based methods and Graph Convolution-based methods. Our proposed framework is constructed based on the research that had been conducted according to these two major methods, which will be discussed below.

A. Attention-based

Early in 2017, GAT [1, 2] uses self-attention to calculate neighbor information by learning every weight and aGCN aggregate neighbor information similarly while learning neighbor joints to obtain the weight of each node. In the same year, Transformer [3] connect the encoder and decoder through attention, combing with convolutions, achieves a faster speed in translation task, which proves the effectiveness of attention mechanism. Yet the limitation of slow program runtime and ignorance of graph structure of data remain challenging to attention-based methods.

B. Graph Convolution based

Similarly, some achievements had been made by previous works using graph convolution networks related to graph-structured data. For example, [4] uses semantic graph convolution to learn weights between neighbors to draw connections between neighbors. [5] proposes an hourglass network to find out useful features from images, which the similar idea is also useful for selecting useful features from customer information. Moreover, in the same year, [6, 7, 8, 9] conduct a comprehensive survey on graph convolution networks, which demonstrates how different types of graph neural networks could potentially aid different research in aspects. On the other hand, the content-based recommendation system measures the similarity between users based on the selected product information and makes recommendations accordingly. However, past research showed that the growing database of user information in above two systems will lead to longer time to profile all users and reduce the efficiency of recommendation, and the overly focus on user information will make us ignore the overall relationship between users and products in the prediction process and lose the overall characteristics.

Most graph convolutions have their own attention, but they are based on fully connected attention, which is obviously not enough.In this paper, we combine the idea of Transformer and Graph Convolution to improve personalized recommendation system in different perspectives.

III. METHOD

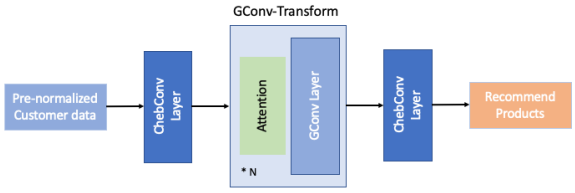

Fig.1 Overview of the whole process.

A. Overview

In light on above concerns, we proposed a method that integrate users' information that preserves details and overall users' relationship at the same time, while filtering out unrelated information that decrease the efficiency. Our method takes purchasing information and history as inputs and predicts the potential products in favor of customers. Our proposed framework combines attention and graph convolution, which constructs relationship between different customers and products for further enforcing personalized recommendation; high prediction accuracy and efficiency will be maintained in the meanwhile. GConv-Transform is constructed by pre-processing data through ChebConv layer, then applying to main body of GConv-Transfrom which is a combination of multi-head attention and GCN layer. The further details of our proposed framework are described in below sections.

The overall organizational diagram is shown in Fig. 1. First normalize the input data, and then put it into a layer of Chebyshev Spectral Graph Convolution layer, which will be introduced in B. Then go through N layers of GConv-Transform layer. Finally, the result is obtained through the Chebyshev Spectral Graph Convolution layer. Among them, GConv-Transform includes attention (will be introduced in III.C) and Graph Convolution (will be introduced in III.D), and the combination of the two will be introduced in III.D.

B. Chebyshev Spectral Graph Convolution

Before inputting into the main building blocks of the proposed framework, it is necessary to have a preliminary process on the inputs. We will perform preliminaries by using Chebyshev Spectral Graph Convolution layer (ChebConv). Assume h_i^l as input features and h_i^{l+1} as output features. Let k stands for the Chebyshev filter size K of the ChebConv, the output of the ChevConv layer could be formulated as:

$$h_i^{l+1} = \sum_{k=0}^{K-1} W^{k,l}, z_i^{k,l}$$

where it takes in dimension and filter size k as parameter for outputs. Here W is learnable weight of ChebConv. In our specific framework, it represents purchasing information of

customers in form of graph and outputted as previous, which is in form of feature vectors.

C. Attention (Transformer)

A rough description of what attention is is "focus on what you're doing". This mechanism is the same as when the human brain thinks. For example, when we think about translating the sentence "I am a student "into English, the first thing that comes to mind is such a mapping of "I" and "I" (instead of thinking of other parts first), which is the performance of the attention mechanism in the human brain. That is, as we translate the sentence phrase by phrase, when we translate to a certain part, our brain will only associate (focus on) the most relevant (similar) English mapping. Therefore, we can apply this same thought process to neural networks.

This section will mainly discuss Attention block in our framework. As shown in previous section, we pre-process purchasing information using ChebConv layer, which results in feature vectors from purchasing information. After normalization by layer norm (LN) , the feature vectors and its dimension are inputted into the multi-head self-attention in our attention block, which is originated from the transformer encoder layers. Then, the output from self-attention is regularized by a dropout layer and contains relationship between different purchase information.

D. Graph Convolution

After the attention block, Chebyshev Graph Convolution is introduced to the proposed framework, where it is more suitable to process graph-liked data, which matches the graph-liked purchase information that we have in our dataset. Output after self-attention will go through normalization by LN layer, which is followed by GCN layers and a ReLU activation function, where GCN layers are resulted from LAM-Gconv. Finally, a dropout layer will be applied.

E. GConv-Transform

Using the provided Attention and Graph Convolution in past research, our proposed framework – Gconv-Transform works by combining multi-head self-attention block and GCN layer from Graph Convolution. Utilizing these two blocks only increase complexity slightly but process our data at a lower cost and make our framework more compatible from data in different structures.

We first enter a norm layer, and then go through N GConv_transformers. Finally, decode to the original ground space. The whole process is shown in Figure 1. The process is simple and efficient. The attention we use is transformer, and graph convolution is the most common graph convolution.

IV. EXPERIMENTS

In this section, we will mainly introduce the experiment of our proposed framework. We will firstly explain implementation details of this experiment, followed by the comparison of former models and analysis of results. At the end, we will enforce the improvement of our framework.

A. Dataset

We used a consumer website dataset containing 18 million consumer behaviors for the year 2012, and from 2018 to 2022. All personal information had been removed and all input data are pre-normalized. Our experiment divides the dataset into training set and test set, where the data are randomly selected in the ratio of 9:1. In particular, the number of training set is 90% in which 20% are selected as validation data, and the number of test set is 10%. In addition, a 0 operation is performed on the missing values of the features, and the problematic data are filtered to prevent the wrong data from negatively impacting in the experiment.

B. Implementation Details.

Training Setting In order to train our framework, we used eight graphic cards where all of them are GTX3090 with 24 GPU memory. In addition, the batch size of this experiment is 96.

C. Results

Our framework GConv-Transform shows significant improvements on finding similarities between customers for personalized recommendations. In particular, GConv-Transform performs relatively better on computational speed, program complexity and prediction accuracy than most of the past methods including pure Attention and graph convolution.

To be specific, according to the figure, although graph convolution maintains relatively high computation speed due to major operations on details, it obtains a high program complexity because of inability of sharing parameters during each operation, which means that every parameter could not be used into other operation in different parts of computation in graph convolution. The high complexity contributes to the high cost of graph convolution.

Conversely, Attention has a high accuracy in prediction but uses too much time. The reason is that Attention focus on the overall relationship and similarity of customers, which requires the process of all customer information each time, thus contributes to the longtime of computation. Another drawback is that Attention’s main focusing on overall operation causes its lack of attention on details, which makes Attention generally costly in performance.

Moreover, other than Attention and graph convolution, we also test the traditional convolution method as shown in the figure. Unlike the method we mentioned above, traditional convolution only focusses on the relationship between neighbors’ data rather than differences between every data and similarities and effectiveness between those data, which decreases efficiency in computation in general.

After carefully looking at above results, it is obvious that our proposed GConv-Transform has overall good performance and achievement in increasing efficiency in prediction and recommendation, which furthermore demonstrate that how combination of Attention [10] and Graph Convolution rather than a single method could boost up accuracy and speed to a considerable value.

Table 1 Results

Method	Accuracy (%)	Times	complexity
SVM	14.6	1.2ms	0.2M
DNN	44.3	232ms	44.8M
CNN	46.2	123ms	42.3M
DenseNet	78.9	1048ms	123.2M
DenseCNN	81.2	542ms	87.9M
Graph	64.2	45ms	74.2M
Attention	52.3	4654ms	327.0M
Ours	83.1	109ms	23.4M

D. Analysis

The key argument is, based on the graph volume. The recommendation method of product network has received great attention in the field of recommendation system because of its powerful learning ability and expression abilityOf concern. However, the existing recommendation methods based on graph convolutional networks face the following challenges: (1) operate on the whole graphLaplacian smoothing is very time-consuming; (2) Most aggregators ignore strong synergy signals in knowledge graphs. It can extract valuable features. (3) The existing recommendation methods based on graph convolutional networks do not consider different neighborhoods.Importance of layers, thus negatively affecting the recommendation performance. To address these issues, this chapter first focuses on knowledge.

A fixed size receptive field is introduced into the atlas to control the amount of computation. Next, we elaborate a simplified version.To capture strong synergy signals in the knowledge graph. Finally, we design a new kind of layer-oriented.The message measurement mechanism measures the information of each network layer in finer granularity to form the final representation. We were at three.A large number of comparative experiments and ablation experiments are carried out on open source datasets to prove the effectiveness of the proposed algorithm. Experimental results show that the proposed algorithm outperforms the state of the art algorithms.

E. Ablation Study

Table 2 Ablation study

Method	Accuracy(%)
Without Chebyshev Convolution	44.6
Without Graph	74.3

The results of Ablation study are shown in Table 2. It can be seen that graph convolution and attention are very important to the final result. In comparison, the attention mechanism has a greater impact on the final result.

Table 3 Influence of N

N	Accuracy(%)
12	74.6
16	79.3
22	83.1

F. The Infuence of N

In Figure 1, there are N GConv-Transfrom, and we are here to verify the impact of N on the final result. The results of the experiment are shown in Table 3. It can be seen that the best results are obtained when N=22.

V. CONCLSUION

Personalized recommendation based on knowledge graph and graph neural network is a challenging research problem in the field of recommendation algorithm. This paper focuses on the analysis of user historical interaction behavior data and the knowledge graph of the item side, and uses graph neural network to mine the semantic information in it to recommend items of interest to users. This paper aims to design efficient and accurate recommendation algorithms to optimize user experience. Therefore, in view of the problems existing in the existing recommendation algorithms in various scenarios, this paper proposes a new recommendation method.

REFERENCES

[1] Resnick P, Varian H R. Recommender systems[J]. Communications of the ACM, 1997, 40(3): 56-58.

[2] Shan L, Wang W. DenseNet-Based Land Cover Classification Network With Deep Fusion[J]. IEEE Geoscience and Remote Sensing Letters, 2021, 19: 1-5.

[3] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives[J]. ACM Computing Surveys (CSUR), 2019, 52(1): 1-38.

[4] Shani G, Heckerman D, Brafman R I, et al. An MDP-based recommender system[J]. Journal of Machine Learning Research, 2005, 6(9).

[5] han L, Li M, Li X, et al. UHRSNet: A Semantic Segmentation Network Specifically for Ultra-High-Resolution Images[C]//2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021: 1460-1466.

[6] He J, Chu W W. A social network-based recommender system (SNRS)[M]//Data mining for social network data. Springer, Boston, MA, 2010: 47-74.

[7] Tejeda-Lorente A, Porcel C, Peis E, et al. A quality based recommender system to disseminate information in a university digital library[J]. Information Sciences, 2014, 261: 52-69.

[8] Zhang H R, Min F, He X, et al. A hybrid recommender system based on user-recommender interaction[J]. Mathematical Problems in Engineering, 2015, 2015.

[9] Ma T, Zhou J, Tang M, et al. Social network and tag sources based augmenting collaborative recommender system[J]. IEICE transactions on Information and Systems, 2015, 98(4): 902-910.

[10] Zhao W, Tian Y, Ye Q, et al. Graformer: Graph convolution transformer for 3d pose estimation[J]. arXiv preprint arXiv:2109.08364, 202