Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/73/20240406

Recommendation system model based on graph neural network combined with support vector machine and comparative learning for filtering

Nazhi Guo

Guang Dong Ocean University, Zhanjiang, Guangdong, 524088, China

Mafia3Freeze@gmail.com

Abstract. To solve the problem that it is difficult to distinguish between noisy data and data that users do not score and considering the great impact of noise on the performance of graph neural networks, a recommendation system based on graph neural networks and comparative learning is proposed. Firstly, the user-item labels are standardized by using SVM, and all user-item data are constructed as graphs. The high-order connection between users and items is established through a graph neural network. Then, the low-weighted node representation is filtered out by using the attention mechanism with a noise filter, and the global embedded representation of denoising enhancement is obtained. At the same time, the comparative learning technology is introduced to set up the optimization strategy guidance model, and the noise data is taken as the negative sample, The supervised training model generates a recommendation list through the prediction layer and carries out experimental tests on the Tmall data set. The results show that the recommendation effect of the model is significantly improved compared with the non-denoising model and the model without self-supervised learning. The NDCG score on the Tmall data set reaches 3.88%, which proves that the performance of the model is improved by preprocessing the label first and combining comparative learning denoising.

Keywords: Graph Neural Network, Support Vector Machine, Comparative Learning, Recommendation System, Denoising Enhancement.

1. Introduction

The recommendation system aims to recommend goods to users by understanding users' personalized interests. It provides important web services to alleviate the problem of information overload. Recently, compared with Self-Supervised Learning (SSL), it has achieved good results in generating representations with a small amount of labeled data. In the context of graph learning, Graph Neural Networks (GNNS) have become one of the best frameworks for establishing high-order connections between users and objects, such as NGCF [1], and LightGCN[2].

Since the recommendation system essentially has a graph structure, it is considered to use the recommendation system based on a graph neural network, but GNN is easy to be deceived by noise, resulting in greatly reduced performance [3], So far, it is still a problem to remove the noise in the network. It is seen that there are some users in a large number of data sets who interact with the project but do not give a score, which will produce a lot of noise on the system, because the system cannot determine whether these items are of interest to users, that is, most users are interested in the project but

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/73/20240406

do not give a score. According to the research of Wu [4], He[5], and others, great progress has been made in the model and structure, but the noise has not been well handled. Considering that most users are interested in the items but do not score, which makes it difficult to distinguish the items that users are interested in, not interested in, and noisy items, we should consider doing a data preprocessing before using the network training, and pre classify the tag data according to the scoring habits of different users, so as to avoid affecting the performance of the network in the subsequent training of noisy data. Then through comparative learning, the noise samples are taken as negative samples, which can reduce the impact of noise on the model and highlight those important features.

2. Literature Review

2.1. Graph Neural Network (GNN)

Graph Neural Network is a deep learning model designed specifically for graph-structured data. GNN uses the information of nodes and edges in the graph and maps the graph structure data to low-dimensional vector space by the backpropagation algorithm of a multi-layer neural network. GNN has been widely used in recommendation systems because most of the information in recommendation systems has a graph structure in essence. Graph Neural Networks transmit information through multi-layer networks, which can encode higher-order signals in user interaction more conveniently. With the development of Deep Learning, Graph Neural Networks have become one of the best methods for Graph Computation and Graph Mining [6]. By comparing the graph model and the traditional recommendation model, it can be seen that the traditional model generally only uses the first-order neighbor node information on the graph, so the performance of the model is not accurate enough. [7]. The Graph Neural Network model can make nodes aggregate the information of higher-order neighbors through the message propagation mechanism, it can effectively capture higher-order relationships to achieve more accurate representation learning [8,9]According to Wu et al. [10]Graph Neural Networks have a wide range of applications in recommendation systems, but due to data heterogeneity, users in the graph have diverse and multiple uncertain interests, so it is difficult to capture the features of users. Graph attention mechanism can allocate the weight of each neighbor node instead of aggregating neighbor nodes with the same weight, which can be used to solve the problem of data heterogeneity.

2.2. Comparative Learning

Comparative learning is a self-supervised learning method that captures the consistency of feature representations between views and generates self-supervised signals by establishing comparative views. In recommendation tasks, contrastive learning serves as an auxiliary task that can utilize unlabeled data to alleviate data sparsity issues. Comparative Learning helps models learn representations with greater similarity in different views by comparing views that information maximizes, making a graph or node with rich coding. [11]. By comparing the samples before denoising with the samples after denoising, we can effectively combat the noise in the data.

2.3. Recommendation System

Because the features in the recommendation system are mostly graph structures, it is considered to apply graph neural network training in the recommendation system. In recent years, with the development of social platforms, the social network between users has rapidly grown social relationships reflect the connections between users, so users often have similar preferences and mutual influence on certain items. Therefore, when users have less rating data on items, they can use the behavioral data of other users who have deeper connections with the user to predict. For recommendation systems, using only raw information for prediction results in poor performance. To provide users with more accurate personalized recommendations, it is necessary to consider the interaction between information. By learning the information that user-item representations, when the timestamp of a user's historical behavior is known or neighboring node's historical behavior is known, user representations can be enhanced through the behavior patterns or historical behavior order of neighboring nodes.

DOI: 10.54254/2755-2721/73/20240406

Graph Neural Networks (GNNs) can efficiently learn graph structures and extract features and have shown good performance in user-item representation learning. In recent years, recommendation algorithms based on GNN have become a new direction of widespread academic attention.

3. Methodology

By utilizing the node representations learned from Graph Neural Networks, combined with the similarity measures learned from comparative learning methods, more accurate and personalized recommendations can be achieved. To address the challenge of existing social recommendation methods in accurately obtaining user preferences from noisy anonymous interactions, establish a recommendation model by combining comparative learning and graph neural networks. Due to the impact of noise on graph neural networks, data that carries noise is used as negative samples for comparative learning with filtered data to achieve a better noise-filtering effect. Take comparative learning partially and globally to reduce the impact of unlabeled or missing data on target nodes.

Let U=$\{u_1, u_2, \ldots, u_n\}$ be all of users and I=$\{i_1, i_2, \ldots, i_m\}$ be all of items. $V = U \cup I$ be node,edge between I and U $\varepsilon = \{e_{ij}, i \epsilon U, j \epsilon I\}$,$G = \{V, \varepsilon\}$ be the user-item graph.

To preprocess labels, in order to distinguish items that users do not mark, items that users are not interested in, and noise items, SVM is considered. Because the amount of data for a single user is not large, the SVM model is very suitable for classification, which is essentially a single classification problem. SVM has great advantages in dealing with this problem.

Using SVM to classify items that users only clicked on but did not purchase or bookmark, as most of a user's data has a label of 0, the purpose is to select items that they like but did not purchase or bookmark for various reasons, so that they can make recommendations in similar items next time. However, a single user has a small amount of data, so using One class SVM, items with a label>0 are used as positive samples, and items with a label=0 but similar to label>0 are selected.

Because the labels in the sample data only have 0 (Click Behavior), 1 (bookmark behavior), and 2 (purchase behavior), recommendation systems are usually essentially predicting items with non-zero label data. This paper believes that recommendation systems should not be a simple binary classification problem, but a multi-classification problem, with varying degrees of user preference for a certain item. Additionally, due to the limited data observed for a single user, it is difficult to adopt large deep learning models, so it is proposed to use SVM to pre-classify user data and standardize labels to (0,1), to make the data more diverse during network training and reducing overfitting of the model.

$$i_{k \epsilon u_q} = \begin{cases} i^+, & label > 0 \\ i^-, & else \end{cases}$$

$$\underbrace{min}_{r,o} V(r) + C \sum \zeta$$

$$\|i^+ - o\| \leq r + \zeta_j$$

Where r is the minimized radius of the hypersphere, o is the center of the target hypersphere, and C is the penalty coefficient.

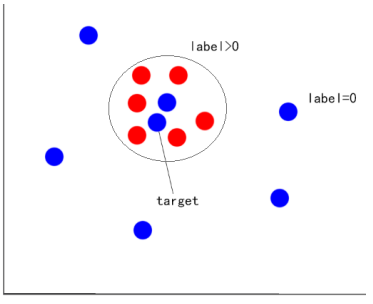

Figure 1. SVM Data classification diagram

DOI: 10.54254/2755-2721/73/20240406

$$i^{-} = \begin{cases} label.replace(\gamma)\ , \gamma\epsilon(0.5,1), if\ \|i^{-} - o\| \leq r + \zeta_j \\ label.replace(\gamma)\ , \gamma\epsilon(0,0.5), if\ \|i^{-} - o\| > r + \zeta_j \end{cases}$$

Observing that most of the user's data is click behavior, with only a few data having interaction. Considering that training such data will be subject to significant noise interference and there is actually less useful data, the data is pre-classified. Using user interaction data as a positive sample, a hypersphere is generated based on its characteristics to process data with only click behavior, The purpose of this behavior is to identify items that users click on and are interested in but have not interacted with due to various reasons. This early classification can also reduce or even remove some noise caused by incorrect clicks or other behaviors while filling in some missing data through classification.

$$h_i^{'} = \sigma\left(\sum_{j\epsilon N(i)} Wh_j\right)$$

h is the embedding of item, W is a trainable parameter,σ is activation function

$$e_{ij} = a\left(Wh_i, Wh_j\right)$$

Introducing attention[12] mechanism makes training more focused on important samples, and considering the importance of aggregating neighboring nodes, aggregating considers neighboring nodes to reduce the impact of noise. α represents the specific attention correlations between entities and relationships in the process of knowledge aggregation.

$$a_{ij} = softmax\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k\in N_i}\exp\left(e_{ik}\right)}$$

$$a_{ij} = \frac{\exp\left(LeakyReLU(a[Wh_i||Wh_j])\right)}{\sum_{k\in N_i}\exp\left(LeakyReLU(a[Wh_i||Wh_k])\right)}$$

Where Ni is the neighboring entities of item i, where i,j,k is in Graph G. LeakyReLU is the activation function. Where parameters W are trainable parameters.

$$h_i = \sigma\left(\sum_{j\epsilon N(i)} \alpha_{ij}^{'} Wh_j\right)$$

Small-weight noise still accumulates a large amount of noise.[13] Therefore, filtering out data with smaller weights, although small weight data will be given smaller weights in the attention network, as the network iterates, will still generate noise that has an impact on the network.

$$\alpha_i^{'} = \begin{cases} \alpha_i\ , & \alpha_i > \lambda \\ 0\ , & else \end{cases}$$

$\alpha_i^{'}$ is the filtered weight parameter. λ is the filtering threshold.

$$\widehat{y_i} = softmax(h_i)$$

$$l_r = -\sum_{i=1}^{n} \widehat{y_i}\, log_e(\widehat{y_i}) + (1 - \widehat{y_i}) log_e\ (1 - \widehat{y_i})$$

Normalize the data, and then use the Cross-Entropy Loss Function to eliminate system uncertainty with the minimum cost.

By comparing the denoised and pre-denoised samples through comparative learning, the pre-denoised samples become negative samples for supervised training, and the data is not artificially noisy,

DOI: 10.54254/2755-2721/73/20240406

which can make the noisy data more realistic. The data processed with noise can provide better feature extraction, better highlight model features, and reduce the impact of noise on the model.

$$l_{cl} = -\log \frac{\exp\left(sim(S, \tilde{S})/\tau\right)}{\sum_U \exp\left(\mathrm{sim}(\mathrm{S}, \tilde{S}/\tau\right)}$$

$$S = \sum h_i, \tilde{S} = \sum \tilde{h}$$

Considering the comparison of noisy data with the original data, the Info NCE loss function is used.[14] Due to the use of noisy data as negative samples, there may be multi-dimensional noise, so it cannot be simply treated as binary classification. τ is a hyperparameter.

$$L = l_r + \beta l_{cl}$$

The final loss of the model is determined by both the comparative learning loss function and the network training loss function. β is parameter.

4. Results

4.1. Dataset

The dataset which we is the Tmall dataset, sourced from the IJCAI-15 competition. It is the user behavior data recorded by China's largest online shopping platform, with approximately 960000 users having 4452812 interactions with 2353207 products, including timestamp and product category information. Remove session records with missing data and fewer than 10 interactions. Use 80% of the data for the training set, 10% for the validation set, and 10% for the testing set.

4.2. Evaluation method

To evaluate the effectiveness of the model applied to recommendation systems, this article adopts Recall@k and NDCG@k These two indicators are commonly used to evaluate the accuracy of recommendation systems for testing. For recommendation systems, we aim to maximize the recall rate of Recall results, while NDCG takes into account the order of user recommendations, often ranking according to user interest will have better results.

4.3. Hyper-parameters

The training batch size is set to 100, the initial learning rate is set to 0.001, and the regularization coefficient L2=10^{-5}, We use the Adam function to optimize the model parameters. The initial epoch is set to 500, the embedding size is 100, the batch size for the mini-batch is 100, and the number of network layers is set to three. If there is no change in the evaluation index for 10 epochs within the validation set, the training will be terminated early.

4.4. Comparison of benchmark models

NGCF [11]: It is a GNN-based CF method that performs graph convolution on embedded user project interaction graphs.

LightGCN [12]. it is a method that simplifies the message-passing rules of GCN by linearly propagating user-item embeddings on the interaction graph for collaborative filtering.

NCF [15]. This is the method that replaces the inner product in MF with a multi-layer perceptron to estimate user-item interactions.

SR-GNN [16]: This is the method that applies a gated graph convolutional layer to learn item transitions.

AutoR [17]: This is the method that applies a three-layer automatic encoder with fully connected layers to encode user interaction vectors.

To test the effectiveness of the model applied to the recommendation system, we used Recall and NDCG as evaluation criteria and set the range of recommended number k to 40 for testing. The results of the model and benchmark model in the dataset Tmall are shown in the table.

DOI: 10.54254/2755-2721/73/20240406

Table 1. Comparing Recall and NDCG with other models.

	NGCF	*Light GCN*	NCF	SRGNN	AutoR	*p*
Recall	0.0751	0.0895	0.0647	0.0704	0.0611	0.0760
NDCG	0.0365	0.0499	0.0344	0.0313	0.0295	0.0388

Comparing the results of using SVM to classify data labels in advance with those of not using them, we can see that using SVM to pre-classify data in advance can improve the ability of the model and make good use of those unlabeled samples. And the model can be effectively improved at a low cost. The purpose of noise filter is to filter out the data with low actual influence, and then make the original sample data as a negative sample through comparative learning, to better highlight the data characteristics. An attention mechanism is introduced to make the model focus more on the data with greater impact. Compared with the graph neural network without denoising, it can be seen that there is a great improvement, which shows that the influence of noise on the graph neural network can be reduced by classifying data and comparative learning, and the network can be further improved.

Due to the introduction of self-supervised learning, the model performs better than the model without self-supervised learning in this data set, and due to the pre-classification of the model before training, the model has greater advantages than some models without denoising, because the noise will have a greater impact in the graph network. Due to the introduction of the attention mechanism, it is better to compare some models that ignore the importance of neighbors and only focus on themselves. The model itself is trained by a graph neural network, and the training effect is better than that of the ordinary neural network, which is due to the advantages of the graph structure of the data itself. However, because the model itself is relatively simple, the denoising method also uses implicit denoising, which has defects compared with the high aggregation and high synergy models.

4.5. Impact of Model Depth.

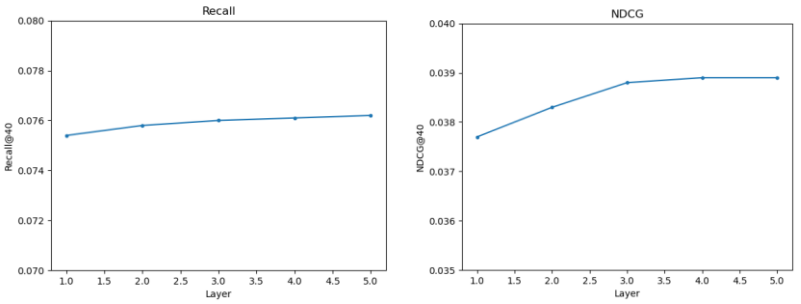

Figure 2. The variation of Recall@40 and NDCG@40 with the number of network layers

To study the influence of the depth of the graph neural network on the recommendation system, the number of layers is controlled to be 1, 2, 3, 4, and 5 respectively. With the increase of the network layer, the curve tends to be flatter. The model is not very sensitive to the number of network layers of the data set, but with the increase of the network layer, the training time will also increase, resulting in poor performance of the model, which is not conducive to our research, so the network layer is set as three layers.

DOI: 10.54254/2755-2721/73/20240406

4.6. Impact of Self-Supervised Learning.

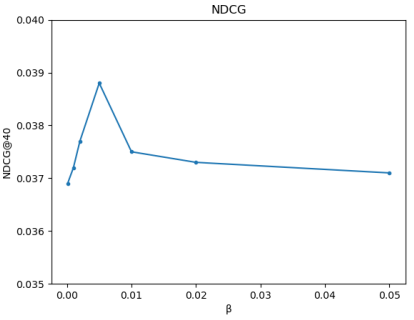

Figure 3. The variation of NDCG@40 with the value of β

This paper uses super parameters β to control the weight of the comparative learning loss in the training, here we hold the other parameters, and change β.The effect of the value of β on the results can be seen, β When the values of {0.001,0.002,0.005,0.01,0.02,0.05} are taken respectively, the results of the indicators are as shown in the Figure 3. By changing the values of β, the best effect is achieved when the value of β is 0.005, then tends to be flat. That means When β smaller, it can promote the fitting of the model, with the increase of β in training may make the model overlearn or remove important information, resulting in poor effect of the model.

5. Discussion

The graph neural network is superior to other network structures for processing the recommendation system. This is because the recommendation system itself has the characteristics of graph structure, and the performance of the recommendation system has been greatly improved whether using the comparative learning combined noise filter for noise reduction or label preprocessing. For noisy clicks in user behavior, the traditional solution usually uses the attention mechanism weighted by SoftMax to weaken the impact of items unrelated to the user's main purpose, that is, giving a small weight proportion to unimportant items, which belongs to an implicit denoising method. However, after many iterations, such small weight noisy clicks will still accumulate a lot of noise, It is easy to form invalid intentional learning [18]. The solution proposed in this paper is to preprocess the label data before training. In the beginning, the data that does not conform to the user's other characteristics and habits is given a lower weight when it does not pass the network, and then gradually filters out the parts with low weight in the training process. Finally, through comparative learning, we can learn those important features, and take the unimportant features as negative samples to supervise the training. During the research process, it was found that due to the low weight of the data that users are not interested in, the rough noise filter will filter out some data that users are not interested in as noise. However, the data that users do not like is an important sample for the recommendation system. In the next step, more accurate noise filters will be considered to avoid wrong filtering.

6. Conclusions

This paper designs a recommendation system based on graph neural networks and comparative learning. The purpose is to compare the advantages of graph neural networks over other networks in the recommendation system and to solve the influence of noise on graph neural networks in the recommendation system. The adaptability of comparative learning in the recommendation system and graph neural network is discussed, and SVM is used to classify the data before training, get a set of data tags that are more suitable for the recommendation system, and then use the graph neural network to train the features. This model is equipped with a noise filter. After filtering out the noise of the sample, combined with the comparative learning strategy, the data before denoising is taken as the negative sample for joint denoising. The setting comparison experiment on the real recommendation system data set proves that using SVM to classify data labels before training is helpful to the performance of the

DOI: 10.54254/2755-2721/73/20240406

recommendation system. Filtering out the noise items in the session can promote the training of graph neural networks more effectively and help to mine the real intention of users. The research broadens the understanding of the recommendation system based on graph neural networks and improves the accuracy of the system in predicting user behavior. In the next step, we will study how to optimize the model structure, introduce the convolution layer, or improve the efficiency of the model in other ways, and study more accurate noise filtering functions to avoid filtering some data incorrectly, to achieve a faster and more accurate recommended model design.

References

[1] WANG, Xiang, JIN, Hongye, ZHANG, An, HE, Xiangnan, XU, Tong, CHUA, Tat-Seng. Disentangled Graph Collaborative Filtering. In: SIGIR, 2020, pp. 1001-1010.

[2] HE, Xiangnan, DENG, Kuan, WANG, Xiang, et al. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In: SIGIR, 2020, pp. 639-648.

[3] WU, Shiwen, SUN, Fei, ZHANG, Wentao, XIE, Xu, CUI, Bin. Graph Neural Networks in Recommender Systems: A Survey. In: ACM Computing Surveys, 2022, Vol. 55, No. 5, Article 97, 37 pages.

[4] WU, J., WANG, X., FENG, F., HE, X., CHEN, L., LIAN, J., XIE, X. Self-Supervised Graph Learning for Recommendation. In: SIGIR, 2021, pp. 726-735.

[5] FAN, Wenqi, MA, Yao, LI, Qing, HE, Yuan, ZHAO, Eric, TANG, Jiliang, YIN, Dawei. Graph Neural Networks for Social Recommendation. In: The World Wide Web Conference, 2019, pp. 417-426.

[6] HE, Xiangnan, DENG, Kuan, WANG, Xiang, et al. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Xi'an, China: Association for Computing Machinery, 2020, pp. 639-648.

[7] FAN, Wenqi, MA, Yao, LI, Qing, HE, Yuan, et al. Graph Neural Networks for Social Recommendation. In: The World Wide Web Conference, 2019, pp. 417-426.

[8] HAMILTON, William L., YING, Rex, LESKOVEC, Jure. Inductive Representation Learning on Large Graphs. In: Advances in Neural Information Processing Systems, 2017, Vol. 30, No. 6, pp. 567-347.

[9] LIN, Xin, SHAO, Xinhui. A Recommendation System Model Based on Graph Neural Network. In: Computer Application and Software, 2023, Vol. 40, No. 03, pp. 325-330.

[10] WU, Shiwen, SUN, Fei, ZHANG, Wentao, XIE, Xu, CUI, Bin. Graph Neural Networks in Recommender Systems: A Survey. In: ACM Computing Surveys, 2022, Vol. 55, No. 5, Article 97, 37 pages. Published in May 2023.

[11] HASSANI, Kaveh, KHASAHMADI, Amir Hosein. Contrastive Multi-View Representation Learning on Graphs. In: International Conference on Machine Learning, PMLR, Honolulu, Hawaii: The International Machine Learning Society, 2020, pp. 4116-4126.

[12] VELIČKOVIĆ, Petar, CUCURULL, Guillem, CASANOVA, Arantxa, ROMERO, Adriana, LIÒ, Pietro, BENGIO, Yoshua. Graph Attention Networks. arXiv:1710.10903v3, 2017.

[13] ZHENG, Xiaoli, WANG, Wei, ZHANG, Chuang. A Graph Neural Network Session Recommendation Model Based on Fusion of Comparative Learning. In: Applied Science, 2023, 1-9. Published on October 8, 2023.

[14] HE, Kaiming, FAN, Haoqi, WU, Yuxin, XIE, Saining, GIRSHICK, Ross. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9729-9738.

[15] HE, Xiangnan, LIAO, Lizi, ZHANG, Hanwang, NIE, Liqiang, HU, Xia, CHUA, Tat-Seng. Neural Collaborative Filtering. In: WWW, 2017, pp. 173-182.

[16] WU, S., TANG, Y., ZHU, Y., WANG, L., XIE, X., TAN, T. Session-Based Recommendation with Graph Neural Networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, Volume 33, 2019, pp. 346-353.

DOI: 10.54254/2755-2721/73/20240406

[17] SEDHAIN, Suvash, MENON, Aditya Krishna, SANNER, Scott, XIE, Lexing. AutoRec: Autoencoders Meet Collaborative Filtering. In: The Web Conference (WWW), 2015, pp. 111-112.

[18] QIN, Yuqi, WANG, Pengfei, LI, Chenliang. The World Is Binary: Contrastive Learning for Denoising Next Basket Recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. Waseda, University: Association for Computing Machinery, 2021, pp. 859-868.