

Article

Not peer-reviewed version

Recommendation Algorithm for Graph Convolutional Collaborative Filtering Based on Multivariate Sampling

Hai Yan , [Zheng Ji Tao](#) ^{*} , [Liu Zhi Zhong](#) ^{*} , [Wang Dong Yang](#) ^{*} , Ding Cheng Rui

Posted Date: 17 October 2024

doi: 10.20944/preprints202410.1410.v1

Keywords: graph convolutional networks; multivariate sampling; contrast learning; collaborative filtering; hard to negative sample



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Recommendation Algorithm for Graph Convolutional Collaborative Filtering Based on Multivariate Sampling

Hai Yan ¹, Zheng Jitao ^{1,*}, Liu Zhizhong ², Wang Dongyang ¹ and Ding Chengrui ¹

¹ College of Information Engineering, North China University of Water Resources and Electric Power

² College of Computer and Control, Yantai University

* Correspondence: zjtlm@126.com

Abstract: In the field of recommender systems, difficult negative samples that are easily misclassified by classifiers have higher learning value. In order to solve the problem that the existing graph convolutional network lacks the exploration of negative sampling strategy, a multivariate sampling based on graph convolution collaborative filtering recommendation model was proposed. Firstly, part of the information in the positive sample was injected into the negative sample, and the high quality difficult negative sample was selected by the inner product method. Then, contrastive learning is used to maximize the similarity between the difficult negative samples and the positive samples, so that the difficult negative samples are as close to the positive samples as possible in the feature space. Ultimately, a multi-task approach was employed to concurrently enhance both the recommendation supervision task and the contrastive learning task. To validate the effectiveness of our approach, we conducted comparative evaluations utilizing three publicly accessible datasets. The experimental outcomes indicate that our proposed model surpasses the baseline model in terms of performance.

Keywords: graph convolutional networks; multivariate sampling; contrast learning; collaborative filtering; hard to negative sample

1. Introduction

With the continuous evolution of the digital era, the amount of information shows explosive growth, and it becomes more and more difficult for people to find the required content in the huge amount of information [1]. Recommender systems have come out of nowhere and become one of the powerful tools to effectively solve the problem of information overload [2]. Traditional recommendation algorithms can be categorized into content-based, collaborative filtering and hybrid selection, and all have a specific technical basis and technical means to complete the recommendation task.

Collaborative filtering algorithm is a widely used algorithm in recommender system, which is based on the historical behavioral data between users and items, and by analyzing and comparing the similar characteristics between users or items, it can predict the interest of the items that users have not touched, so as to speculate the degree of their preference for these items [3–5]. Collaborative filtering algorithms can be categorized into two types: user-based collaborative filtering and item-based collaborative filtering [6]. User-based collaborative filtering approach focuses on the analysis of behavioral patterns among users, which identifies other groups of users with similar interests by comparing their historical behaviors and accordingly recommends items preferred by those similar users to the target user [7,8]. The core of the item-based collaborative filtering strategy is to measure the degree of similarity between different items. The method finds and recommends new products that are similar in features or attributes by analyzing the user's favorite items in the past, in order to more accurately satisfy the user's personalized interests and needs.

In recent years, in order to further improve the embedding quality and mitigate the effects of data sparsity, an important research direction is to utilize the non-Euclidean structure of Graph Convolutional Networks (GCN) to characterize the nodes [9]. Graph Convolutional operations are able to aggregate information within a node's neighborhood to learn an embedded representation of each node that contains not only attribute information about users and items, but also incorporates information about their structure in the graph [10–12]. In this way, the algorithm is able to understand user preferences and item properties more comprehensively, thus generating more accurate recommendations and alleviating the data sparsity problem to a certain extent.

Contrastive learning is an effective unsupervised learning method, which learns feature representations by comparing positive and negative samples, so that similar samples are closer in the feature space and different samples are further away [13–15]. In recommendation algorithms, the application of comparative learning is mainly reflected in improving the performance of recommendation systems by learning the interaction features between users and items [16,17]. Specifically, the recommender system can use the user's historical behavioral data to construct positive sample pairs and negative sample pairs, and then optimize the feature representation through contrast learning, so that the positive sample pairs are more compact in the feature space, while the negative sample pairs are more dispersed. In this way, contrastive learning not only improves the accuracy and efficiency of the recommender system, but also enhances its robustness and generalization ability [18–20].

In graph convolutional network recommendation, a common practice when training a recommender system model is to use items that users have actually interacted with as positive samples, while selecting items that users have not interacted with as negative samples. These positive and negative samples are then used to optimize the loss function of the model, which in turn learns embedding vectors that accurately represent the characteristics of the user and the item, and finally completes the model training process. Although negative sampling improves the efficiency of the recommendation system, it is still necessary to consider how to optimize the sampling strategy and improve the quality of negative sampling in order to obtain more accurate recommendations in practical applications.

To address the above problems, this paper proposes a Graph Convolution Collaborative Filtering Recommendation Algorithm Based On Multivariate Sampling (MGCF), the The main contributions of the algorithm are as follows:

- (1) A new negative sampling strategy is proposed, firstly, the original negative samples are randomly selected, secondly, the positive sample embedding information is randomly introduced into the original negative samples using the interpolated multivariate technique, and then the final difficult-negative samples are synthesized by using the inner-product method selection strategy in the process of aggregation in the graph convolutional network to lay the foundation for the subsequent model training.
- (2) Introducing the contrast learning method, by mining more feature information in the positive samples and the difficult-negative samples, the difficult-negative samples are made closer to the positive samples in the feature space, which further improves the model's ability to recognize the boundary of the positive and negative samples.
- (3) In order to validate the overall performance of the collaborative filtering recommendation algorithm based on multivariate sampling for graphical convolutional networks, the experimental results comparing it with a variety of state-of-the-art algorithms on three publicly available datasets, Yelp2018, Alibaba, and Gowalla, are conducted significantly demonstrating that the proposed model has a significant performance enhancement with respect to the baseline model. This improvement not only validates the effectiveness of the multivariate negative sampling module, but also confirms the positive role of the contrast learning module in the model.

2. Related work

2.1. Collaborative Filtering Recommendation Algorithm Based on Graph Convolutional Networks

Compared with traditional deep learning methods, GCN shows unique advantages in processing non-Euclidean structured data, and these advantages have led to a wide range of applications in various fields such as computer vision and recommender systems. Wang et al [21] proposed Neighborhood Based Collaborative Filtering (NGCF) proposed by Wang et al. applies GCN to collaborative filtering algorithms for the first time, and explicitly learns the high-dimensional information of user-item interactions through message passing; He et al. [22] simplify the traditional GCN framework for the collaborative filtering by omitting the feature transformations and nonlinear activation steps, and constructs a lightweight Light Graph Convolution Network (LightGCN) to provide a more effective and efficient collaborative filtering algorithm. Convolution Network (LightGCN). This simplified version of the GCN model shows higher accuracy in recommender system applications, and has significant performance improvement compared to models such as NGCF; Ying et al [23] propose a random wandering strategy for graph convolutional networks named (PinGraph with Sample and Aggregate, PinSAGE), which has excellent scalability and can effectively handle large-scale network nodes.

2.2. Negative Sampling Strategy

In the field of recommender systems, most of the collected data belongs to the implicit feedback category, and these forms of feedback include user behaviors such as clicking, browsing, favoriting, and purchasing [24]. Given that users usually interact with only a limited number of items, an intuitive and efficient strategy is to consider the items that the user directly interacts with as positive samples, which directly reflects the user's interests and preferences. A common negative sampling approach is to perform Random Negative Sample (RNS), which randomly selects samples from all non-interacted samples as negative samples according to a uniform distribution [25]. Cai et al. [26] suggest that about 95% of negative samples belong to the category of easy negative samples, which are not similar to positive samples in terms of features. Therefore, just relying on these easily negative samples to train the model is not enough to get a model with superior performance. Zhang et al [27] proposed a dynamic negative sampling strategy applied in collaborative filtering recommendation task, which dynamically selects negative samples based on the currently generated ranked list and improves the quality of negative sampling.

2.3. Comparative Learning

Contrast learning, as an effective self-supervised learning method, mainly relies on comparing the differences between positive and negative examples in the feature space in order to capture the consistency of feature representations across different views. The SGL model [28] employs three graph-based data augmentation techniques that utilize the multi-view information followed by contrast learning to increase the differences between different node representations. This data augmentation and pre-training approach helps to mine more supervised signals from raw graph data. Yu et al [29] proposed a recommendation algorithm SimGCL that adds uniform noise to the embedding null to create a contrasting view, which implicitly mitigates popularity bias and smoothly adjusts the uniformity of the learned representations by learning a more uniform representation of the user or item to make recommendations. Chuang et al [30] improved on the Infonce model by proposing the RINCE model, which is robust to noise and does not require explicit estimation of noise.

3. Algorithm Design

3.1. Overall Framework

The recommendation model framework of graph convolutional network based on multivariate sampling is shown in Figure 1. It can be mainly divided into three major modules: (1) Multivariate

negative sampling module: by randomly introducing positive sample information into the negative samples, the negative samples with higher quality are selected to generate the final negative examples in the aggregation process of graph convolutional network. (2) Lightweight graph convolutional network module: the multivariate negative sampling module provides the difficult negative sample data required for model training, and the lightweight graph convolutional network module performs feature extraction based on the graph structure and the sample data obtained from sampling. (3) Optimization module: the positive samples and the obtained difficult-negative samples are compared and learned as sample pairs so as to narrow the distance between the difficult-negative samples and the positive samples in the semantic feature space, and optimized with the BPR loss function for joint training.

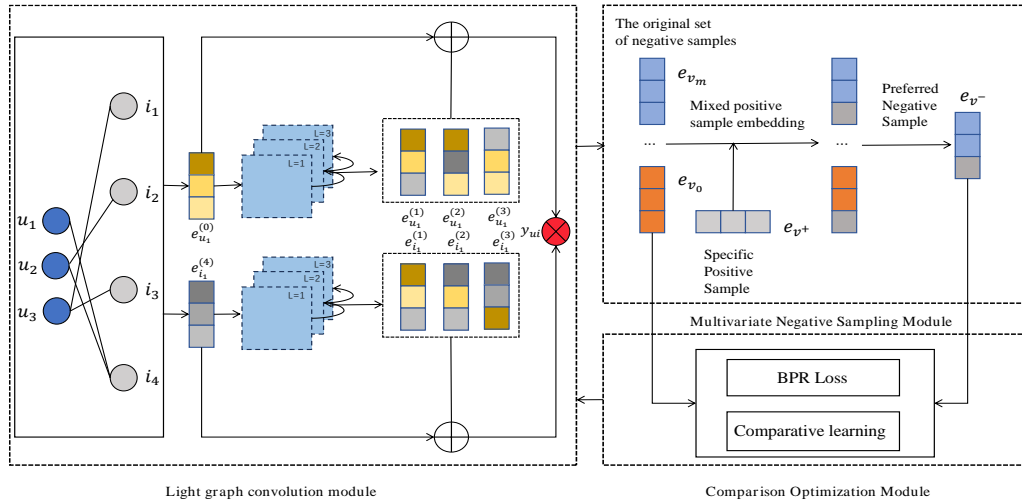


Figure 1. MGCF modeling framework.

3.2. Multivariate Negative Sampling

In the model training phase, in order to learn user preferences, the model receives both positive and negative samples as input. By adjusting the loss function, the model is trained to amplify the difference between positive and negative samples. The traditional approach is to directly utilize the samples that are real and have been identified as negative for training. However, this paper proposes a new strategy of artificially constructing those difficult negative samples that are challenging for the model in a continuous embedding space. This strategy helps the model to capture and distinguish user preferences more accurately during training. The method is divided into two main steps: generating enhanced candidate negative samples and sample preference aggregation.

3.2.1. Generate an Augmented Negative Sample Candidate Set

The first step of this method generates an enhanced negative sample candidate set by introducing positive sample information into the original negative samples. In an L-layer graph convolutional network, each item generates L+1 embedding vectors. Each embedding vector $e_v^{(l)}$ integrates the information of the previous layer, in order to obtain the embedding w' , this paper firstly draws M negative samples randomly from an original set of negative samples to form a candidate set. In this paper, we first randomly draw M negative samples from an original set of negative samples to form a candidate set w . Inspired by the mixup method, this paper further randomly injects the positive sample information e_v^+ into the original set of candidate negative samples, and finally generates the augmented candidate negative sample set w' , which is generated as follows:

$$e_{v_m}'^{(l)} = \alpha^{(l)} e_{v^+}^{(l)} + (1 - \alpha^{(l)}) e_{v_m}^{(l)}, \alpha^{(l)} \in (0, 1) \quad (1)$$

In Eq. A is the newly generated augmented candidate negative samples, and B is the mixing coefficient used to regulate the amount of positive sample information injected into the model. In order to reduce the potential impact of this coefficient on the generalization ability of the model, its value is selected uniformly in the range of (0,1) in each layer by randomization, thus ensuring that the introduction of this coefficient does not bias a particular data distribution or pattern. The identification ability of the model is enhanced by injecting information from the positive samples into the negative samples, thus making it more difficult for the model to distinguish between decision boundaries.

3.2.2. Sample Optimization and Aggregation

After the augmented candidate negative sample generation process, the augmented candidate negative sample set w' is obtained, and then through the aggregation mechanism of GCN, the candidate set is filtered and refined to determine the final pseudo-negative sample v^- and obtain its corresponding embedding representation e_{v^-} . For the layered GCN, the set of difficult negative samples generated by each layer is denoted as $w^{(l)}$. Firstly, the hard-negative sample embeddings are sampled from $w^{(l)}$, and these embeddings are combined by pooling operation to obtain the final hard-negative sample embeddings:

$$e_{v^-} = f_{pool}(e_{v_x}^{(0)}, \dots, e_{v_x}^{(l)}) \quad (2)$$

In Eq. f_{pool} is the aggregation function to synthesize the final embedding, and $e_{v_x}^{(l)}$ denotes the embedding of the difficult-to-negative samples sampled in the first layer. In this paper, we are inspired from MCNS [31] to utilize the difficult-negative sample selection strategy $w^{(l)}$ with inner product approximating the positive sample distribution to aggregate the negative sample embeddings with the largest inner product in each layer to synthesize the final difficult-negative samples. The inner product between user vectors and negative vectors is first computed, and then the negative samples corresponding to the largest inner product are selected. Therefore, the difficult negative sample vector selection strategy for the first layer is:

$$e_{v_x}^{(l)} = \arg \max_{e_{v_m}^{(l)} \in \mathcal{E}^{(l)}} f_Q(u, l) \cdot e_{v_m}^{(l)} \quad (3)$$

In Eq. $f_Q(u, l)$ is a query mapping function that returns the representation vector of the user's layer; denotes the inner product operation, the form of which depends on the pooling aggregation.

3.3. Contrastive Learning

In the multivariate sampling stage, due to the use of linear interpolation of the positive sample information randomly injected into the negative sample, which will lead to the generation of difficult-negative samples semantic features will have a certain bias, and difficult-negative samples in the semantic features of the positive samples bias away from the negative samples can improve the model's recognition and classification ability. To address this problem, based on the work in literature [15], this paper further uses augmented robustness comparison learning, after multivariate sampling to obtain the difficult-negative samples, the positive samples and the difficult-negative samples are regarded as positive pairs, and the negative samples and the difficult-negative samples are regarded as negative pairs, and the distance between the positive samples and the difficult-negative samples in the feature space is further narrowed through the augmented robustness comparison learning, which can help the model to learn a better representation of the data. In this paper, the Robust InfoNCE (RINCE) loss function is used as the core function of the contrast learning module, the core of which is to use the Wasserstein dependency measure as the lower bound of mutual information in RINCE, which, due to the strong ensemble property of its Wasserstein distance, makes the RINCE in terms of anti-sample noise stronger, which can improve the robustness of the model as well as the generalization ability, and the RINCE loss function is:

$$\mathcal{L}_{RINCE}^{\lambda, q}(s) = \frac{-e^{q \cdot s^+}}{q} + \frac{\left(\lambda \cdot \left(e^{s^+} + \sum_{i=1}^K e^{s_i^-} \right) \right)^q}{q} \quad (4)$$

In Eq. λ , q are the hyperparameters, s^+ positive sample pairs of scores, and s^- negative sample pairs of scores.

4. Experiments and Analysis of Results

4.1. Data Sets and Evaluation Indicators

In this paper, three public datasets Yelp2018, Alibaba, and Gowalla are used and the datasets are divided into training, validation, and testing sets for comparative experiments according to the ratio of 7:1:2, and the detailed information of the datasets is shown in Table 1.

Table 1. Data information of Yelp 2018, Alibaba, Gowalla.

Dataset name	Number of users	Number of products	Number of interactions	data density
Yelp2018	31668	38048	1561406	0.00130
Alibaba	106042	53591	907407	0.00130
Gowalla	29858	40981	1027370	0.00084

In this paper, based on a popular top-K recommendation algorithm, Recall@20 and NDCG@20, which are widely recognized in the industry, are chosen as the core evaluation metrics in order to evaluate the performance of the recommender system. In accordance with the convention in the field of recommender systems, the value of K is set to 20 to conform to the conventional setting for the length of recommendation lists in the field. Recall is one of the important metrics for evaluating the performance of recommender systems. It measures how many of all truly relevant positive samples are correctly predicted or recognized by the recommender system. Normalized Discounted Cumulative Gain (NDCG) is a key metric for measuring the performance of recommender systems, which focuses specifically on the relevance of the recommendations to the user's preferences and the ordering of these recommendations in the list.

4.2. Benchmarking Model

In order to evaluate the performance of the MGCN recommendation model in a more comprehensive way, this paper selects the better-performing models for comparison tests, and the comparison models are as follows:

- (1) NGCF [21]: graph convolutional neural network is applied to collaborative filtering recommendation task by constructing a bipartite graph of user-item interactions in order to capture high-dimensional interactions between users and items.
- (2) LightGCN [22]: discarded the traditional complete graph convolution process and removed the feature transformation and nonlinear activation steps. Reduces the computational complexity of the model.
- (3) PinSage [23]: the core idea is to learn the embedded representations of the nodes in the graph through efficient random wandering and graph convolution operations to capture high-dimensional interactions between users and items.
- (4) RNS [25]: A negative sampling module is added to the traditional network to select uninteracted items as negative examples to enrich the training data and improve the model performance.
- (5) SGL [27]: Adding an auxiliary self-supervised task to the classical supervised recommendation task to enhance node representation learning through self-recognition.
- (6) GTN [32]: innovatively introduces graph trend collaborative filtering technique and constructs a new graph trend filtering network framework. This dynamically captures the adaptive

reliability of interactions between users and items to provide more accurate and personalized recommendation services.

4.2. Experimental Environment and Parameter Settings

In order to validate the model performance, experiments were conducted using the PyTorch machine learning framework in PyCharm 2020, NVIDIA GeForce GTX 3090, and Python 3.6. In the experiments the embedding size is set to 64, the learning rate is set to 0.001, the batch size is 2048, the graph convolution layer L is set to 3, the hyperparameters $\alpha=0.5$, $\lambda=0.025$, the parameters are initialized using Xavier, and the optimizer is set to Adam.

4.3. Analysis of Experimental Results

The results of the comparison between the model proposed in this paper and the six base models on the public dataset are shown in Table 2, where underlining is used to indicate the best performance achieved by the baseline model, bolding denotes the optimal results, and enhancement/ represents the % of enhancement of the model proposed in this paper with respect to the optimal baseline model in each of the metrics.

Table 2. Experimental results comparing MGCF and baseline models.

Dataset name	Yelp2018		Alibaba		Gowalla	
model name	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
NGCF	0.0543	0.0443	0.0426	0.0197	0.1573	0.1332
LightGCN	0.0637	0.0531	0.0585	0.0275	0.1822	0.1543
PinSage	0.0470	0.0392	0.0411	0.0332	0.1380	0.1195
RNS	0.0625	0.0516	0.0506	0.0232	0.1532	0.1319
SGL	0.0675	0.0553	<u>0.0688</u>	<u>0.0338</u>	0.1611	0.1398
GTN	<u>0.0676</u>	<u>0.0555</u>	0.0450	0.0336	<u>0.1871</u>	<u>0.1589</u>
MGCN	0.0733	0.0603	0.080	0.0381	0.2002	0.1696
提升/%	8.43%	8.64%	16.27%	12.72%	7.0%	6.73%

After comprehensive testing on the three public datasets, the proposed model demonstrates significant advantages in all performance indicators, especially in the key indicator of Recall@20, compared with the best baseline model, the proposed model achieves 8.43%, 16.27%, and 7.0% performance improvement on the three datasets, respectively. This significant performance improvement fully verifies the validity and superiority of the proposed model in this paper, which is mainly due to the following aspects: (1) MGCF adopts a multivariate sampling method, which can generate difficult-to-negative samples with higher quality and lay a good foundation for subsequent improvement. (2) MGCF adopts the idea of comparative learning, which can excavate the hidden features of the samples, maximize the distance between the difficult-to-negative samples and the positive samples in the feature space, and further improve the quality of the difficult-to-negative samples.

3.1.1. Ablation Experiment

In order to deeply investigate the practical effects of the multivariate negative sampling module and contrast learning module proposed in this paper, three ablation models, MGCF-R, MGCF-P, and MGCF, are designed for experiments, where MGCF-R denotes the removal of the contrast learning module and the use of traditional random sampling for model training, MGCF-P denotes the substitution of the traditional random sampling into the multivariate sampling mechanism for model training, and MGCF denotes adding the contrast learning module to MGCF-P for model training. The experimental results on the three public datasets are shown in Figure 3:

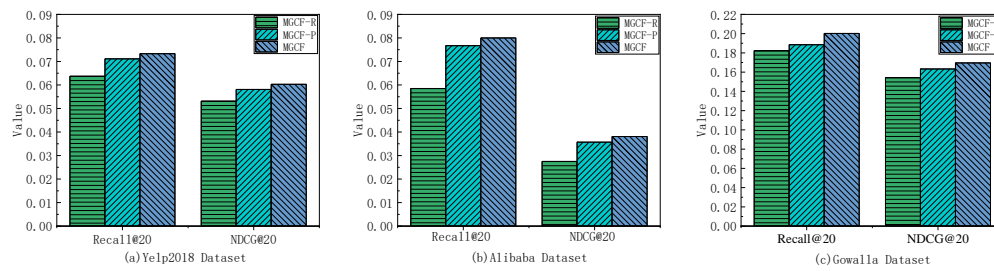


Figure 3. Experimental results of MGCF and ablation models on three datasets.

From the experimental results, it can be seen that both the multivariate sampling module and the contrast learning module in this model play an optimization role for the model, taking the Yelp 2018 dataset as an example, the MGCF-P improves by 11.6% on the basis of the MGCF-R, and the MGCF-F improves by 3.5% on the basis of the MGCF-P. The enhancement of the experimental results is mainly due to the fact that the multivariate negative sampling mechanism proposed in this paper is able to utilize the graphical convolutional network structure to generate difficult negative samples with higher quality, and then on the basis of this, the contrast learning idea is used to maximize the similarity between difficult negative samples and positive samples to further improve the model's ability to identify the boundaries of positive and negative samples.

The Materials and Methods should be described with sufficient details to allow others to replicate and build on the published results. Please note that the publication of your manuscript implicates that you must make all materials, data, computer code, and protocols associated with the publication available to readers. Please disclose at the submission stage any restrictions on the availability of materials or information. New methods and protocols should be described in detail while well-established methods can be briefly described and appropriately cited.

Research manuscripts reporting large datasets that are deposited in a publicly available database should specify where the data have been deposited and provide the relevant accession numbers. If the accession numbers have not yet been obtained at the time of submission, please state that they will be provided during review. They must be provided prior to publication.

Interventionary studies involving animals or humans, and other studies that require ethical approval, must list the authority that provided approval and the corresponding ethical approval code.

4.4.2. Hyperparametric Analysis

In order to achieve better performance, the parameters L and hyperparameter q , which affect the effect of the model, are tuned. In the process of parameter tuning, a stepwise optimization strategy is used, i.e., one of the parameters (e.g., L) is first fixed, and the optimal value of the other parameter (e.g., q) is determined through experiments. Then, based on the optimal q value that has been found, q is fixed again and the search for the optimal setting of parameter L continues. For the parameter L , three possible values of 1, 2 and 3 were considered, while for the parameter q , five different values between 0.1 and 0.5 were evaluated. In order to assess the impact of these parameters on the model performance, a sensitivity parameter analysis was performed on the Yelp 2018 dataset. The experimental results are shown in Tables 3 and 4.

Table 3. Effect of number of layers L and hyperparameters q of graphical convolutional network on Recall@20.

q/L	1	2	3
0.1	0.0706	0.0710	0.0716
0.2	0.0709	0.0712	0.0720
0.3	0.0710	0.0715	0.0723
0.4	0.0712	0.0716	0.0727
0.5	0.0713	0.0718	0.0733

Table 4. Number of layers and hyperparameters of graphical convolutional network Effect on NDCG@20.

q/L	1	2	3
0.1	0.0581	0.0583	0.0596
0.2	0.0582	0.0585	0.0598
0.3	0.0585	0.0586	0.0600
0.4	0.0587	0.0588	0.0602
0.5	0.0588	0.0589	0.0603

From Tables 3 and 4, it can be obtained that the number of graph convolutional network layers L and hyperparameters q have a certain impact on the performance of the recommendation model. In general, by increasing the number of layers of the GCN, the model can explore the association between nodes in the graph structure as well as the relationship between users and items more deeply, so as to better capture the patterns and information hidden in the graph. The hyperparameter q mainly controls the degree of attention to positive and negative samples in the learning process, when the hyperparameter is larger, the model will pay more attention to those difficult-to-distinguish pairs of samples, i.e., those pairs of samples with similarity scores close to each other, which will help the model to improve the accuracy of distinguishing between positive and negative samples in the noisy environment. Recall@20 and NDCG@20 achieve the best results when L=3 and q=0.5.

5. Conclusions

In this paper, we propose a multivariate sampling-based graph convolutional network recommendation algorithm (MGCF), which firstly uses multivariate sampling to inject positive sample information into the negative samples to generate a candidate set of difficult-to-negative samples, secondly, uses the inner-product method to select difficult-to-negative samples with higher quality, and finally, uses comparative learning to maximize the similarity between difficult-to-negative samples and positive samples, to further optimize the feature learning effect of the model. Experiments on three publicly available datasets verify the effectiveness of the MGCF recommendation algorithm and its superiority over other methods. In order to further improve the accuracy and personalization level of recommendation, the introduction of auxiliary information as a means of enhancement is considered in the next step of the work so as to provide more accurate and personalized recommendation services.

Author Contributions: H.Y. conceived the research idea, designed the experiments, and provided guidance throughout the research process. Z.J. conceived the idea, designed the experiments, conducted the experiments, provided data, analyzed the data, created figures and tables and wrote the manuscript. L.Z conceived Project management; The first draft review proposes relevant revisions. W.D conceived data management and wrote the manuscript. D.C conceived formal analysis and Writing assistance. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China, grant number 62273290.

Acknowledgments: This project is supported by the National Natural Science Foundation of China.

Conflicts of Interest: The authors declare no competing interests.

References

1.

Chai, W.G.; Zhang, Z.J. A recommendation system based on graph attention convolutional neural networks [J]. Computer Applications and Software,2023,40(08):201-206+273.

2.

Li W , Ning B .Autonomous Recommendation of Fault Detection Algorithms for Spacecraft [J].IEEE/CAA Journal of Automatica Sinica, 2024, 11(1):273-275.DOI:10.1109/JAS.2023.123423.

3.

Hemaraju S , Kaloor P M , Arasu K .Yourcare: A diet and fitness recommendation system using machine learning algorithms [J].AIP Conference Proceedings, 2023.DOI:10.1063/5.0134382.

4. Maazouzi Q E , Retbi A , Bennani S .Enhancing online learning: sentiment analysis and collaborative filtering from Twitter social network for personalized recommendations [J].international journal of electrical and computer engineering, 2024, 14(3):3266-3276.
5. Li D , Esquivel J A .Lightweight and personalised e-commerce recommendation based on collaborative filtering and LSH [J].International journal of ad hoc and ubiquitous computing: IJAHUC, 2024, 45(2):82-91.
6. H, D.P; Spark-based collaborative filtering parallel recommendation algorithm design [J]. Information Record Material,2023,24(06):212-214.DOI:10.16009/j.cnki.cn13-1295/tq.2023.06.050.
7. Zhang, J.R; Application of new media short video platform in education industry [J]. Television Technology,2023,47(07):175-178.DOI:10.16280/j.videoe.2023.07.047.
8. Li M , Ma W , Chu Z .KGIE: Knowledge graph convolutional network for recommender system with interactive embedding [J].Knowledge-Based Systems, 2024, 295.DOI:10.1016/j.knosys.2024.111813.
9. Wang, C.; Research on recommendation algorithm based on graph neural network and its privacy protection [D]. Southeast University,2024.DOI:10.27014/d.cnki.gdnau.2022.002497.
10. Laroussi C , Ayachi R .A deep meta-level spatio-categorical POI recommender system [J].International Journal of Data Science and Analytics, 2023, 16(2):285-299.
11. Chen L , Bi X , Fan G ,et al.Amultitask recommendation algorithm based on DeepFM and Graph Convolutional Network [J].Concurrency and computation: practice and experience, 2023.
12. Tong G , Li D , Liu X .An improved model combining knowledge graph and GCN for PLM knowledge recommendation [J].Soft computing: A fusion of foundations, methodologies and applications, 2024, 28(6):5557-5575.
13. Zhang, A.Q.; Li, R; Tian, X.X; A socially aware sequential recommendation model based on graph neural networks [J]. Computer Applications and Software,2024,41(03):246-252+282.
14. Boughareb R , Seridi H , Beldjoudi S ,et al.Explainable Recommendation Based on Weighted Knowledge Graphs and Graph Convolutional Networks [J].Journal of information & knowledge management, 2023.DOI:10.1142/S0219649222500988.
15. Patel R , Thakkar P , Ukani V .CNNRec: Convolutional Neural Network based recommender systems - A survey [J].Engineering Applications of Artificial Intelligence, 2024, 133.DOI:10.1016/j.engappai.2024.108062.
16. Wang Z .Research on Sports Marketing and Personalized Recommendation Algorithms for Precise Targeting and Promotion Strategies for Target Groups [J].Applied Mathematics and Nonlinear Sciences, 2024, 9(1).DOI:10.2478/amns-2024-1694.
17. Yang P , Xifeng M A , Wei M ,et al.Research on Recommendation Algorithms Based on Cloud Models in Probabilistic Linguistic Environments [J].Journal of Systems Science and Information, 2024(1).
18. Iglesias G , Talavera E ,Jesús Troya,et al.Artificial intelligence model for tumoral clinical decision support systems [J].Computer Methods and Programs in Biomedicine, 2024, 253.DOI:10.1016/j.cmpb.2024.108228.
19. Krishnamoorthi S , Shyam G K .DESIGN OF RECOMMENDATION SYSTEMS USING DEEP REINFORCEMENT LEARNING – RECENT ADVANCEMENTS AND APPLICATIONS [J].journal of theoretical and applied information technology, 2024, 102(7):2908-2923.
20. Mahendra Y , Bolla B .Unveiling the power of knowledge graph embedding in knowledge aware deep recommender systems for e-commerce: A comparative study [J].Procedia Computer Science, 2024, 235:1364-1375.DOI:10.1016/j.procs.2024.04.128.
21. Wang, X, He X, Wang M; Neural Graph Collaborative Filtering [J].ACM, 2019.DOI:10.1145/3331184.3331267.
22. He, X; Deng; K; Wang, X; LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation [J].ACM,2020.DOI:10.1145/3397271.3401063.
23. Ying, R; He, R;Chen, K; Graph convolutional neural net-works for web-scale recommender systems [C]//Proceedings ofthe 24th ACM SIGKDD International Conference on Knowledge Discovery &Data Mining.2018:974-983.
24. Huang, Y.; Mu, C.; Fang, Y.; Graph Convolutional Neural Network Recommendation Algorithm with Graph Negative Sampling. Journal of Xidian University, 2024, 51(01): 86-99. DOI: 10.19665/j.issn1001-2400.20230214.
25. Cheng, L.; Chenhuan, Y.; Ning, G.; SimGCL: Graph Contrastive Learning by Finding Homophily in Heterophily. Knowledge and Information Systems, 2024, 3: 66.

26. Yang, Z.; Ding, M.; Zhou, C.; Understanding Negative Sampling in Graph Representation Learning. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 1666-1676.
27. Zhang, W.; Chen, T.; Wang, J.; et al. Optimizing Top-N Collaborative Filtering via Dynamic Negative Item Sampling. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 785-788.
28. Gu, J.J.; Yang, D.; Nie, T.; A Recommendation Algorithm Based on Multi-View Fusion and Cross-Layer Contrastive Learning. *Journal of Computer Engineering*, 2024, 50(01): 120-128. DOI: 10.19678/j.issn.1000-3428.0066906.
29. Li, X.; Liu, X.; Li, W.C.; A Comprehensive Review of Contrastive Learning. *Journal of Miniaturization for Microcomputers*, 2023, 44(04): 787-797. DOI: 10.20009/j.cnki.21-1106/TP.2022-0538.
30. Chuang, C.Y.; Hjelm, R.D.; Wang, X.; et al. Robust Contrastive Learning against Noisy Views. [J]. 2022. DOI: 10.48550/arXiv.2201.04309.
31. Zhen, Y.; Ding, M.; Zhou, C.; Yang, H.; Zhou, J.; Tang, J. Understanding Negative Sampling in Graph Representation Learning. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2020. DOI: 10.1145/3394486.3403154. [CrossRef]
32. Fan, W.; Liu, X.; Jin, W., et al. Graph Trend Networks for Recommendations. [J]. 2021. DOI: 10.48550/arXiv.2108.05552.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.