Pruned Graph Convolutional Network for Collaborative Recommendation

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University
in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: July 16th, 2021

Wei Wang
(5119FG16)
Advisor: Prof. Hiroshi Watanabe
Research guidance: Research on Audio-visual Information Processing

Acknowledgements

Firstly, I would like to thank Prof.Hiroshi Watanabe from the bottom of my heart, who create a wonderful research circumstance for us.

Secondly, I want to thank Mr. Ishikawa who provide many professional research advices and academic inspirations for my research.

At last, I would like to express my gratitude to my parents, who always encourage me and support me to study abroad.

Abstract

The emergence of personalized recommendation systems solves the problem of information overload. As a bridge between users and information, it can help users find the information they need in the vast network resources. At the same time, it can also actively guide customers to discover some useful resources that they are interested in but do not know about their existence.

Challenges faced by recommender systems include data sparsity issues, cold start issues, privacy protection issues, and scalability issues. As the most widely used technology in recommendation systems, collaborative filtering recommendation technology largely solves these difficulties.

With the development of deep learning technology, Graph Convolutional Networks(GCN) has become an important part of collaborative filtering recommendation technology, which uses GCN's message transfer mechanism to stack multiple graph convolutional layers, and realizes the implicit transmission of cooperative signals from higher-order neighbors on the bipartite graph, making up the shortcomings of the transitional network can only utilize the low-order cooperative signal.

To make the GCN network more concise and appropriate for recommendation, this paper conduct the pruning experiment to simplify the design of GCN and test the result in Movielens dataset. Furthermore, this paper also add the side-information at product-side to utilize the label information of product. The experiment result shows the great improvement and potential of pruned GCN-based model.

Key words: Collaborative-filtering recommendation, Graph Convolutional Networks, recommendation system

List of Content

Chapter 1 Introduction

1.1 The background of the recommendation system

With the popularization of the Internet and the rapid development of information technology, network information resources are showing an explosive growth trend. Network information surrounds us in various forms, such as: social networks, advertising, online education, Internet finance, and e-commerce websites. Especially represented by e-commerce websites, various movie websites, music websites, news websites and shopping websites have penetrated every corner of our daily lives.

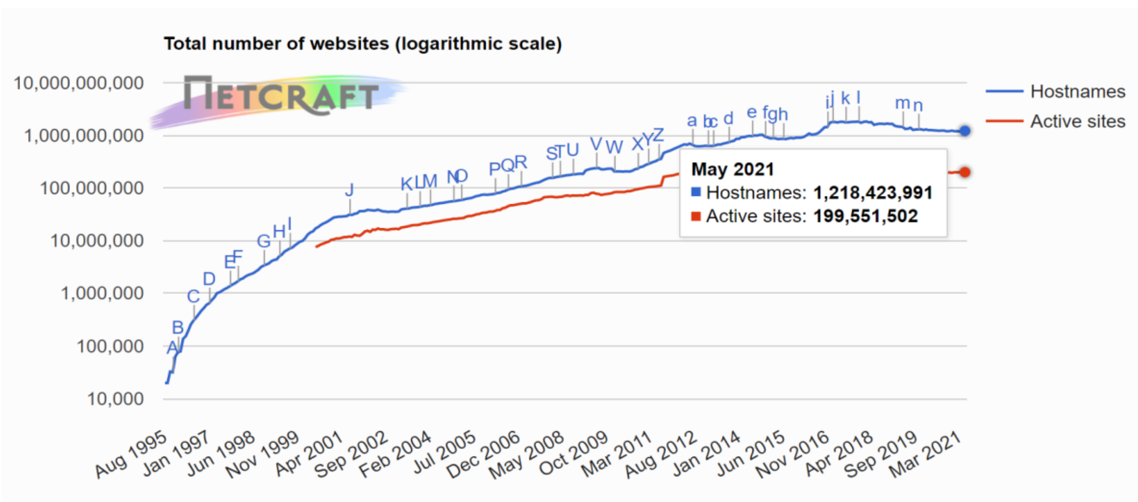

Figure 1.1 Internet Monitoring report in 2021 May

Although massive amounts of information provide people with unprecedented convenience, it can be said that there is mass information that users want on the Internet, but how to accurately locate the information they need has become the primary problem that people must face in the Internet era. According to the monthly monitoring report[1] of Netcraft, a US Internet testing company, as of May 2021, the number of global Internet sites exceeded 1,218,423,991.

[1]http://news.netcraft.com/archives/category/web-server-survey/

Figure 1.1 shows the change curve of the number of Internet sites in the report. It can be seen from Figure 1.1 that the number of global Internet sites has been growing rapidly since 1995, and has remained at the order of 1 billion after 2016. The massive amount of network information provides a guarantee to meet the diverse information needs of users and brings great convenience to people. However, the diversity and variability of network information has led to the excessive expansion of information and brought about the problem of information overload (Information Overload [1]), making it difficult for people to quickly and accurately find the information they need from the vast information resources. In the era of "information overload", how producers can accurately sell their products to users, and how consumers can accurately find the products they need, have become the primary problem that everyone faces in the information age.

The emergence of personalized recommendation systems [2, 3] solves the problem of information overload. As a bridge between users and information, it can help users find the information they need in the vast network resources. At the same time, it can also actively guide customers to discover some useful resources that they are interested in but do not know about their existence. The personalized recommendation system can intelligently recommend items that they are really interested in based on the user's historical behavior information (such as browsing products, scoring products, etc.). To some extent, the recommendation system can be regarded as a supplement to the website classification navigation and search engine tools.

At the same time, the extensive application of personalized recommendation technology in the field of e-commerce enables e-commerce websites to accurately grasp the interests and hobbies of users, so as to actively recommend products they like to users. This shift from user active search to website active recommendation not only saves the time of information search, but also greatly improves the user's Internet experience and makes users extremely dependent on the website; at the same time, when the user is only browsing electronic When a business website does not intend to shop, if the website can properly recommend the product they are interested in, then the user is likely to buy the product. This way of actively guiding consumers greatly increases the sales of the website Amount [4].

1.2 Application areas of the recommendation system

Since the 1990s, the research of recommender systems has made considerable progress. In academia, recommendation systems have become a specialized research field. ACM has hosted the ACM Conference on Recommender System since 2008.

In addition, some actual recommendation systems are already providing services for people's daily lives. For example, the recommendation system of the famous e-commerce website Amazon recommends products that may be of interest to users, providing Amazon with at least 35% of sales[2]; famous DVD The rental site Netflix uses a recommendation system to assist users in finding movies that may be of interest, providing Netflix with approximately 60% of rental records[3]. These recommendation systems use massive amounts of user behavior data through data analysis and mining to improve the user experience and effectively enhance the company's sales performance. Nowadays, the research and application of recommender systems have involved various fields in people's lives, including books, news, e-commerce, etc. Table 1.1 lists well-known recommendation systems in various fields.

[2] http://glinden.blogspot.com/2006/12/35-of-sales-from-recommendations.html

[3] http://www.wired.com/wired/archive/12.10/tail.html

Table 1.1 Examples of well-known recommendation systems

Application areas	Recommended system	URL
News	GoogleNews	http://news.google.com
	Genieo	http://www.genieo.com
Movies	Jinni	http://www.jinni.com
	MovieLens	http:// movielens.org
E-commerce	Amazon	http://www.amazon.com
	Alibaba	http://www.taobao.com
	Rakuten	http://www.rakuten.com
Music	Ringo	http://ringomo.com
	Lastfm	http://www.lastfm.com
	Pandora	http://www.pandora.com
Videos	Hulu	http://www.hulu.com
	Youtube	http://www.youtube.com
	Clicker	http://www.clicker.com
Articles	StumbleUpon	http://www.stumbleupon.com
	CiteULike	http://www.citeulike.com
Travels	TripAdvisor	http://www.tripadvisor.com
	NileGuide	http://www.nileguide.com
	YourTour	http://www.yourtour.com
Social Networks	Facebook	http://www.facebook.com
	Twitter	http://twitter.com
图书	Amazon	http://www.amazon.com
	Douban	http://book.douban.com
	Openlibrary	https://openlibrary.org/

Three parts make up a complete recommendation system: 1. Information collection model for obtaining the under-item interaction information. 2. Information analysis model for analyzing the user behavior. 3. The recommendation core, algorithm module

Among them, the information collection module is responsible for recording user behavior information, including browsing, downloading, scoring, purchasing, collecting, etc.; the information analysis model uses the collected user information to analyze the user's preference for different products, and combines them with models suitable for the current application Analyze user interests; finally, the recommendation algorithm combines user interests and product features to recommend products to users. The goal of the recommendation system is to help users choose some products that users may be interested in.

Personalized recommendation technology has a history of more than 20 years since its inception. With its intelligence and personalization, recommendation systems have been widely used in various fields of the Internet. Related recommendation algorithms specifically for personalized recommendation systems emerge in an endless stream. The recommendation algorithm is the most significant part of the recommendation system. The performance of the recommendation algorithm is related to the recommendation efficiency and recommendation quality of the entire recommendation system. This is also the research goal of this article.

1.3 The difficulties of the recommendation system

The most severe challenges currently faced by recommender systems usually include data sparsity issues [5], cold start issues [6], privacy protection issues [7], and scalability issues [8]. The following will highlight these issues. The problem is briefly introduced:

(1) Data sparsity issues

In the era of filtering data, the number of products on e-commerce websites is almost at the million level, and the user-item score matrix established by any e-commerce website is the highest dimensional. The key step of the algorithm is the calculation of similarity, and the calculation of similarity can be accurately estimated. If there is no social user rating information, there will be a high porter situation of the user-item rating matrix. Finding the neighbors of the user's item can not guarantee the accuracy of the recommendation result, so it can become the most important problem in the personalized recommendation technology.

(2) Cold start issues

The cold start problem recommendation system can accurately grasp the user's historical behavior information. The current e-commerce website dynamically generates millions of pieces of product information, and users usually can only perform historical behavior information on these extremely detailed users. Product recommendation evaluation. In the absence of a large amount of user history behavior information or even user history behavior information, the recommendation system cannot provide users with accurate personalized recommendation services, which is a cold start problem. The cold start problem is the most difficult problem to solve, especially for a newly launched website. Since there is no user behavior record as a reference and own item information, in this case, the synchronization filtering algorithm has no identity proofing power.

(3) Scalability issues

Scalability issues are mainly for the relationship model between the original users and items in the system products when new users and new users join the recommendation system when the system is updated, such as users' browsing, purchasing, and commenting on items, etc. The behavior information will be in the next continuously updated state. At the same time, the addition of more and more new users and new items will also increase the user-item rating matrix. At this time, it is recommended that traditional algorithms deal with big data. It takes too long to calculate user experience and long time. The problem is to calculate the recommendation effect of the system and make users feel disgusted with the recommendation system.

(4) Privacy protection issues

In the era of big data, people's privacy protection needs to be solved, but the user's personal information and behavioral information are required by the historical record system. When there is none, people will not protect and trust it again, no matter how well its recommendation results meet the user's needs, users will not have a good impression of the system. Therefore, the recommendation system must protect the age safety protection mechanism to protect the user's personal privacy. This is a virtuous cycle. The recommendation system expects users to provide detailed personal information. If users find that a recommendation system can protect their personal privacy information well, they will worry about personal information again. In this way, when a recommendation system user provides personal information, the user is willing to provide more personal feedback information, and the system uses the feedback user information to optimize the user recommendation model in order to provide users with better recommendation services.

1.4 Outline of Thesis

The outline of thesis is shown below:

Chapter 1: The background of recommendation system is described in this chapter, the application areas of the recommendation system and the difficulties that faced in the recommendation system field. At last, we briefly introduce the research content and structure arrangement of this article.

Chapter 2: We first introduced the classification of recommendation systems, and then introduced collaborative filtering algorithms, focusing on the deep learning methods applied in recommendation systems in recent years.

Chapter 3: We demonstrate the frameworks of proposed pruned GCN-base model and the date flow process in three parts: constructing the user-item interaction matrix, processing by pruned GCN-base network, and recommend by predict top-k recommended list. Besides, the design detail of product-side embedding, and loss function also discussed in this part.

Chapter 4: We firstly introduce the dataset and evaluation method, and then give the experiment result and the corresponding analysis of four experiments includes: 1. Analyzing the impact of the number of layers on the performance of pruned GCN model. 2. Analyzing the effect of each component of initial GCN-based model. 3.Analyzing the average consumption of training time of initial GCN-based model and pruned GCN-based mode. 4. Compare the overall performance among pruned GCN-based and other competing methods.

Chapter 5: Chapter 5 concludes this thesis.

Chapter 2 Previous work

Recommendation algorithm is the core module of recommendation system, which directly affects recommendation efficiency and quality. Since the first recommendation system appeared in the year, researchers have proposed various recommendation algorithms based on the characteristics of recommendation systems in different fields, among which there are many The methods are combined with research results in the field of data mining. At present, there are many classification methods for personalized recommendation technologies according to different classification standards.

Traditional recommendation methods include content-based recommendation methods, collaborative filtering recommendation methods, and hybrid recommendation methods. Although these methods can achieve the recommendation task, and the auxiliary information containing user behavior information and personalized needs can solve cold start, The problem of sparse matrix, complex feature problems such as multi-modality, data sparseness , uneven distribution , large-scale, data heterogeneity, and still exist in auxiliary information.

With the successful application of deep learning in data mining, natural language processing, image recognition and other related fields, more and more researchers have also introduced the idea of deep learning into recommendation algorithms. The recommendation system based on deep learning overcomes the obstacles of traditional models and achieves high-quality recommendations. Compared with traditional recommendation methods, methods based on deep learning have the advantages of exploring the nonlinear relationship between users and projects effectively, encoding more complex abstractions into higher-level data representations and capturing the complex relationships within the data itself from a large number of accessible data sources and so on.

2.1 Traditional recommendation method

Traditional recommendation methods include content-based recommendation methods, collaborative filtering recommendation methods, and hybrid recommendation methods.

2.2 Content-based recommendation algorithm

Content-based recommendation algorithm is the combination and application of information filtering and information retrieval technology in the field of personalized information services. This type of algorithm usually focuses on analyzing the content of the item, extracting its main attribute features, and then by calculating the similarity between these features and the user's preference, finding the items that users may be interested in and recommending them. The core idea is to use the similarity between the project and the user template to filter information to achieve the purpose of recommendation. Content-based recommendation algorithms often use vector space models to characterize and abstract items, and need to use relevant algorithms in the field of pattern recognition, such as decision trees [9], Bayesian classification [10], etc. Obtain and adjust the user's interest model.

Its advantages are: the system does not depend on the user's data, which overcomes the sparsity problem and cold-start problem; at the same time, the introduction of pattern recognition, machine learning and other tools reduces the system's dependence on domain knowledge.

The disadvantages are :First, the characterization of project content is limited by the vector space model, which makes it difficult to characterize multimedia information such as sound, image, and video; secondly, due to the limitations of user templates, it is difficult for the system to find the user’s New interest; Finally, because the algorithm relies on similarity calculations to filter information, it is difficult for the algorithm to pick out suitable items when the characteristics of the items are very similar.

2.2.1 Recommendations based on collaborative filtering

Collaborative filtering is currently the most popular type of recommendation algorithm. It was first proposed by Goldberg et al., and was used in a research mail recommendation system called Tapestry. So far, many well-known recommendation systems that have been put into commercial practice, such as WebWatcher, Let's Broswer, LikeMinds, etc., have used this type of algorithm. The core idea of this type of algorithm is that if the behavior or characteristics of some users are similar, then their interests also have a certain similarity. They use the collaborative relationship between users and users or projects and projects to filter out users' interests.

Recommendation methods based on collaborative filtering includes item-based collaborative filtering and user-based collaborative filtering.

User-based collaborative filtering

There are three steps for user-based collaborative filtering model to do the recommendation task: Firstly, the similarity between users is calculated based on the user preferences. Secondly, based on the similarity, the target user needs to be found. Thirdly, to select the item from the highest scores recommend items.

A large number of experiments have shown that the system using this strategy makes good use of people’s social attributes and herd psychology and can generate recommended items that are more in line with user interests. At the same time, it can also discover the potential interests of users and expand the interest coverage of recommended items. However, as the number of system users continues to increase, the data that the recommendation system based on user collaborative filtering needs to process also increases rapidly, which will lead to excessive equipment overhead, which restricts the application scope of the algorithm. At the same time, because users usually only become interested in a few items in the system, this will make the user rating matrix very sparse, affect the accuracy of user rating prediction and similarity calculation, and reduce the accuracy of recommendation.

Item-based collaborative filtering

Similar to user-based collaborative filtering, recommendations of item-based collaborative filtering are made by calculating the similarity between items.

In an actual system, on the one hand, the number of items is usually relatively stable and much smaller than the number of users. On the other hand, the release and management authority of project information is often controlled by the recommendation system, so for the system, the characteristics of the project are usually easier to obtain. The amount of data that needs to be processed using a recommendation system based on project system filtering is determined by the number of projects in the system, and the system design and maintenance personnel can accurately predict the cost of equipment based on the number of projects.

2.2.3 Mixed recommendation

Every recommendation method has its own unique advantages, but when one method is used alone in some scenarios, the recommendation result may not ideal. Combining multiple recommendation methods for mixed recommendation can learn from each other and improve the performance of the recommendation system. Most of the recommendation methods used in actual scenarios are obtained by mixing two or more recommendation methods.

There are many hybrid recommendation methods, such as weighted type, switching type, cross type, feature combination type, waterfall type, feature increasing type, meta-level type.

2.3 Deep recommendation method

In recent years, deep learning has developed rapidly in the field of recommendation systems. In the recommendation system, deep learning technology extracts the potential features of users and items. Based on these potential features, users generate recommendation items and complete recommendation tasks. The neural network in deep learning technology can not only learn the potential feature representation of users or items, but also learn the complex nonlinear interaction characteristics between users and items, analyze user preferences in depth, and solve some problems in traditional recommendation methods.

2.3.1 Deep Recommendations based on DNN

Deep neural networks (DNN) can be regarded as neural networks with many hidden layers, known as multi-layer perceptron (MLP), deep feedforward networks (DFN),

COVINGTON et al. [11] considered the large data size, high data freshness, and high noise of video data on the YouTube website, and proposed the use of DNN to implement efficient recommendations. The system is divided into two stages: candidate set generation and sorting, each of which uses a DNN model. Candidate Generation Mode is used in the candidate set generation stage to retrieve the most relevant resources from the massive video library (million level) based on user portraits and scene data as the candidate set. In the candidate set generation stage, the author transforms the recommendation task into a super multi-classification problem. That is, the video $\omega_t$watched by user U in scene C and time t belongs to category i in video library V, and each video i can be regarded as a category, and its classification model formula is shown in formula :

$$P(\omega_t = i \mid U, C) = \frac{\mathrm{e}^{v_i u}}{\sum_{j \in \mathrm{V}} \mathrm{e}^{v_j u}} \tag{2.1}$$

ZHANG et al. [12] proposed a model combining collaborative filtering recommendation algorithm and DNN. The model is composed of a feature representation module and a score prediction module. The experiments prove that the model is model can effectively improve the recommended performance.

2.3.2 Deep Recommendations based on CNN

KIM et al. [13] proposed a novel context-aware recommendation model that integrates Convolutional Neural Network (CNN) into Probabilistic Matrix Factorization (PMF) Convolutional Matrix Factorization (ConvMF). ConvMF captures the context information of the document and further improves theaccuracy of scoring prediction. The experimental results in this paper prove that the ConvMF model can deal with the sparsity of contextual information well.

TUAN et al. [14] proposed a three-dimensional convolutional neural network to model different types and properties of data, and perform character-level encoding on all input data, using session clicks and content functions (such as item descriptions and item categories) for prediction.

OORD et al. [14] proposed using the latent factor model to realize automatic music recommendation. The model predicts these latent factors from music audio when the latent factors cannot be obtained from the use data. A deep convolutional neural network is used to compare traditional methods of the audio signal's pocket representation, and quantitative and qualitative evaluations are made on the prediction of a one million song data set.

2.3.3 Deep Recommendations based on GCN

Embedding the user and item is the core of the current recommendation system, but now many algorithms only use the characteristics of the user or item itself for embedding. Wang et al. proposed the NGCF model [43], which represented the user and item as a two-part graph, and embedding the user and item through the algorithm propagated on the graph, thus using the interactive information between item and user, showing the relationship between user and item The high-order connectivity can better perform collaborative filtering.

Chapter 3 Proposed method

As shown in Figure 4.1, the whole model consists of three parts. First part is to build a user-item interaction matrix based on original dataset and the relationship can be regarded as structure of user-item interaction graph.

The second part is the pruned GCN-based network, in this stage, the information of graph that obtained in part 1 and the initial random embedding of each node are enter into the network. Based on the propagation rule of pruned GCN-based model, the embedding of each node in the layer i will be calculated by the embedding of layer $i-1$. After the training is ended, the final predict score can be calculated by the inner product of the user embedding and item embedding.

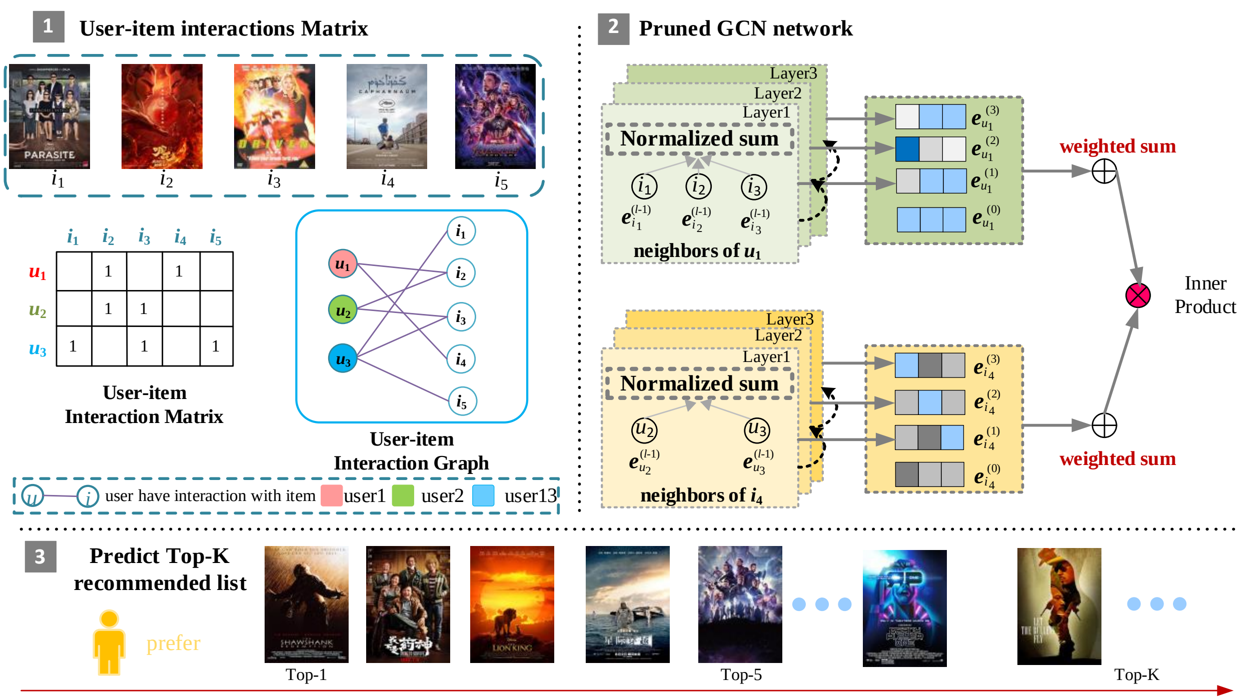

Figure 4.1 Framework of Proposed Model

The final part is to make the recommendation based on the score that calculated in the step 2, and the details of the whole model are given in following graphs.

3.1 Designing the product-side embedding

The user's preference for a certain product can not only be reflected by the user's evaluation for the product, but also the similarity with other user information such as gender, age, occupation, nationality, education level, income level, etc. At the same time, starting from the category of the product, products with the same label are often liked by the same group of users. For example, in the field of movie recommendation, when two movies can belong to the same genre, or starred by the same actor, or have the same director. The similarity between a product and a product can also be considered in the recommendation.

Table 3.1 Movie feature information in Movielens dataset

Movie type	Production year	Country ISO 3166
Action 000	1960	China 156
Comedy 001	1961	Japan 392
Drama 010		
Fantasy 011	……	……
Horror 100		
Mystery 101		
Romance 110	2020	America 840
Thriller 111	2021	Korea 410
……	……	……

However, in the real recommendation task, only real industrial datasets will have real user information, and in the experiment, the information on the product side is easier to obtain, so we start from the product side to construct the embedding. The specific product feature information is shown in the table 3.1 (take the movie dataset Movielens as an example):

In our experiment, we utilize the 64-bit embedding to represent the feature, and in the production side, we design the product-side embedding by splicing the product feature information and add 0 to the remaining bits.

3.2 Selecting negative sample and designing loss function

The following introduces several loss functions in the field of recommendation systems to solve the sorting problem. The recommendation problem generally has only explicit positive samples, but no explicit negative samples, thus we need to define negative samples through rule. The following loss functions are based on the idea of pairwise loss.

3.2.1 Top1 Loss

This is a heuristic combined loss function that consists of two parts: The first part aims to increase the target score above the sample score, and the second part reduces the score of the negative sample to zero. The second part is actually a regular term, but it does not directly constrain the weight. It penalizes the score of the negative sample. Because all items may be a negative sample of a certain user. The specific formula is as follows:

$$L_{\mathrm{top1}} = \frac{1}{N_S}\sum_{j=1}^{N_S} \sigma\left(r_j - r_i\right) + \sigma\left(r_j^2\right) \tag{3.1}$$

Here j corresponds to a negative sample (unobserved), and i corresponds to a positive sample.

3.2.2 BPR Loss

BPR Loss is a kind of raking loss that is more used. It is based on Bayesian Personalized Ranking. The idea of BPR Loss is very simple, which is to make the scores of users and interacted products are greater than the scores of users and non-interactive products. In another word, to maximize the difference between the scores of the positive sample and the negative sample as much as possible. The specific formula is as follows:

$$L_{\text{bpr}} = -\frac{1}{N_S}\sum_{j=1}^{N_S} \log \sigma\left(r_i - r_j\right) \tag{3.2}$$

3.3 Pruning Convolutional Neural Network model

Graph Convolutional network is a kind of neural network which calculates on graphs. The input of GCN can be written as $G = (V, E)$. The general propagation rule of GCN is shown below:

$$f\left(H^i, A\right) = \sigma\left(\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{\frac{1}{2}}H^iW^i\right) \tag{3.3}$$

The description of each components is shown as follows:

H^i is the feature (or embedding) of Graph represented at i-th layer.

A is the Adjacency Matrix of graph, and $\hat{A}$ means the node connect with itself.

σ is a non-linear activation function like the ReLU function which is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

D is the Degree Matrix of Graph A, and $\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{\frac{1}{2}}$ is used for normalization.

W^i is a weight matrix for the layer i.

Gain the inspiration from the Light GCN [18], we conduct the pruning experiment based on the formula 3.4 to compare the effect of each component, and the final propagation rule after simplified is shown below:

$$f\left(H^i, A\right) = \widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{\frac{1}{2}}H^i \tag{3.4}$$

Chapter 4 Experiment Result

4.1 Dataset

The experimental data used the open-source Movielens dataset provided by the Group Lens project team of the University of Minnesota in the United States. It contains more than 100,000 ratings of 1,682 movies by 943 users, and each user has rated at least 20 movies, with a score ranging from 1 to 5. The higher the score, the higher the user’s evaluation of the movie, less than 20 The users who rated the reviews were cleared out of the data set. The data set consists of three data tables: User (user table), Item (movie table), Rate (user rating table).

The user rating information table for the movie includes the user id, the movie id, the user's rating value of the movie, and the rating date. The number in the rating date column is the number of seconds, which is the total number of seconds that have elapsed since January 1, 1970, when the user rated the movie. The general form of the rating information table is shown in Table 4.2.

Table 4.2 Movie information table

Movie id	Movie name	Publish time	IMDb url	Movie type
56	Pulp Fiction	1994	http://us.imdb.com/M/title-exact?Priest%20	Action/Cri
236	Citizen Ruth	1996	http://us.imdb.com/M/titleexact?Citizen%20Ruth%20	Comedy
339	Mad City	1997	http://us.imdb.com/M/title?exact?Mad+City+	Thriller
……	……	……	……	……

Table 4.3 Movie rating table

user id	Item id	rating	timestamp
378	15	4	880044312
6	538	2	883268483
……	……	……	……

4.2 Evaluation method

Score prediction accuracy compares the score predicted by the system with the user's actual score data and evaluates the accuracy of the recommendation algorithm by measuring their errors. In response to this problem, Mean Absolute Error (MAE for short) and Root Mean Squared Error (RMSE for short) are the two most used evaluation methods. The calculation methods are as shown in formula (1.15) and formula (1.16) respectively. Show:

The smaller the value of MAE or RMSE, the more accurate the recommendation algorithm is to predict the user's rating. Compared with MAE, RMSE magnifies the penalty for inaccurate scoring results in the form of square penalty and has more stringent requirements on the prediction accuracy of the recommendation system. Early research mainly used MAE to evaluate the accuracy of the recommendation algorithm's score prediction [94, 106], but with the Netflix competition using RMSE as an evaluation indicator, RMSE has gradually become the main evaluation indicator of the prediction accuracy in the scoring prediction problem.

$$MAE = \frac{\sum_{<u,i>\in T} |r_{ui} - \hat{r}(u,i)|}{|T|} \tag{4.1}$$

$$RMSE = \sqrt{\frac{\sum_{\leq u,i> \in T} (r_{ui} - \hat{r}(u,i))^2}{|T|}} \quad (4.2)$$

Among them, T is the test set, $\hat{r}(u,i)$ is the rating value of user u on product i predicted by the recommendation algorithm, and $r_{ui}$is the actual rating of user u on product i in the test set.

The display result is a recommendation system in the form of a list, and its performance evaluation indicators are generally represented by recall and precision. Let $R(u)$ represent the prediction result given by the recommendation algorithm, and $T(u)$ represent the list of items scored, then the recall rate can be defined as follows:

$$\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (4.3)$$

The accuracy of the recommended results can be defined as follows:

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (4.4)$$

4.3 Experiment Result and Analysis

Several sets of experimental schemes have been designed in this paper and we have compared the pruned GCN-based model(the model base on the propagation rule of formula 3.4) with the initial GCN-based model(the model base on the propagation rule of formula 3.3) to contrast the recommended result.

Experiment 1: Analyzing the impact of the number of layers on the performance of pruned GCN model in the Movielens dataset.

The result of impact of the number of layers on the performance of pruned GCN-based model is shown above. The figure shows that the performance of model improved gradually with the increasing of layers. However, the performance begins to decline when the number of layers reaches 4, which indicates that the model performs the best when the propagation layer comes to 4.

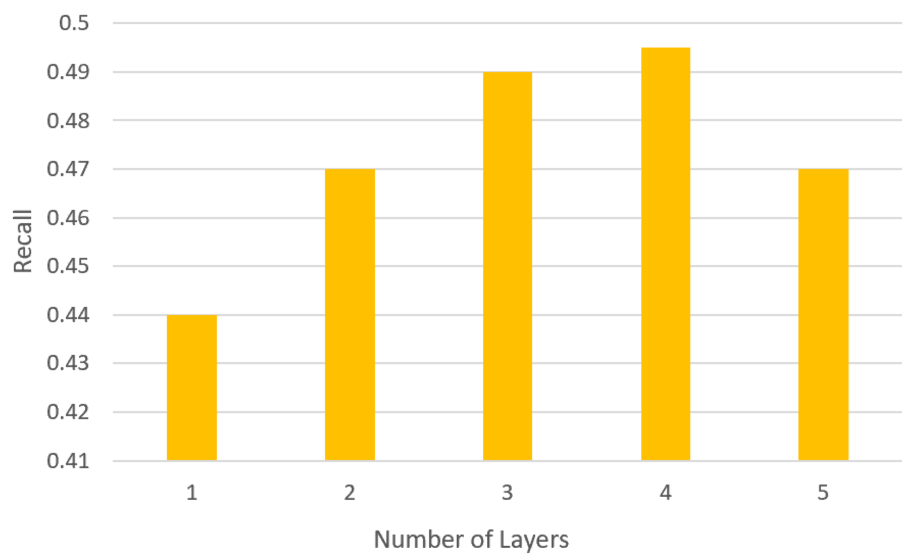

Figure 4.1 Result at Different Layers on Movielens

Experiment 2: Analyzing the effect of each component of initial GCN-based model in the Movielens dataset.

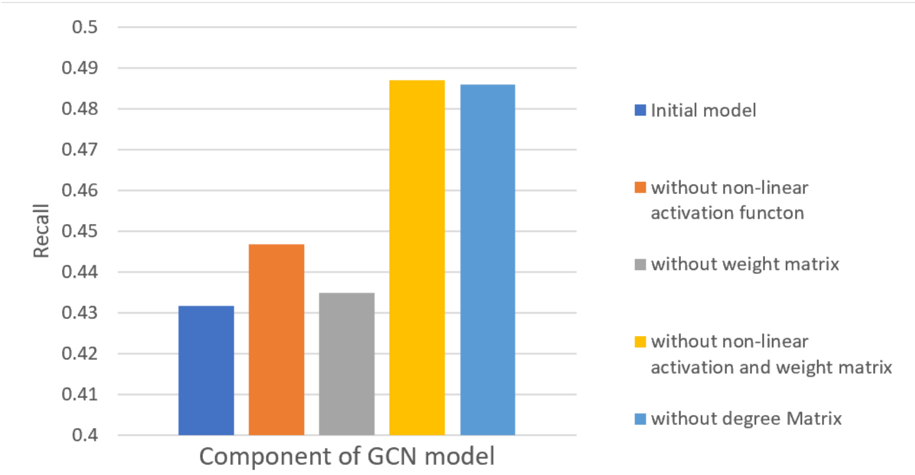

Figure 4.2 Pruning Experiment Result of Four Variants

Transformed from the Initial GCN-based model, there exist four variants including model A: Initial GCN-based model that without non-linear activation function, model B: Initial GCN-based model that without weight matrix, model C: Initial GCN-based model that without non-linear activation function and weight matrix, and model D: Initial GCN-based model that without degree matrix.

For these four variants, we hold all the hyper-parameters same and observe the experiment result on the 4- layer setting on Movielens dataset. As the bar chat shows in the Figure 4.2, discarding the non-linear activation and weight matrix part leads to the significant improvement of the initial GCN-based while removing degree matrix nearly doesn't affect the performance, thus, it can be concluded that removing the non-linear activation and weight matrix part can make the model be more concise and light.

Experiment 3: Analyzing the average consumption of training time of initial GCN-based model and pruned GCN-based model in the Movielens dataset.

Table 4.4 The average consumption of time

Number of rating samples	Initial GCN	Pruned GCN
D1:29936	6781.92(ms)	5561.32(ms)
D2:65645	9953.54(ms)	7935.68(ms)
D3:98635	14106.32(ms)	11147.39(ms)

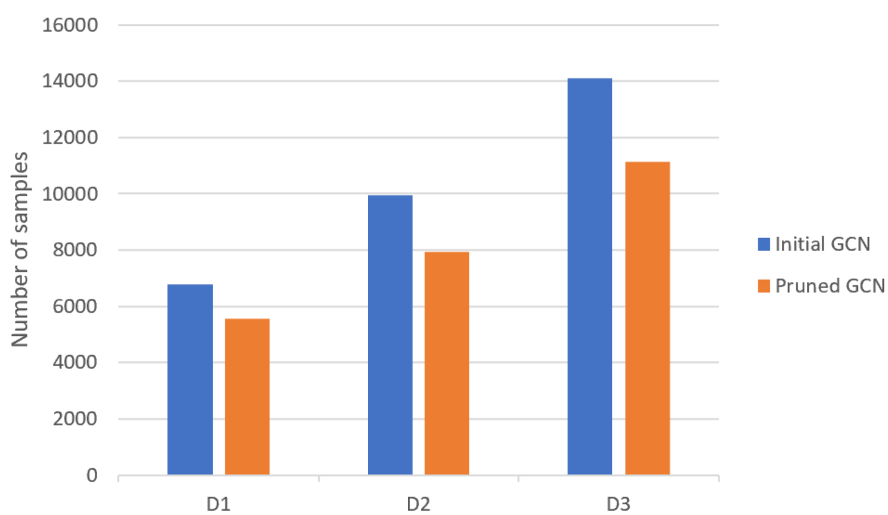

Figure 4.3 The average consumption of time

In order to evaluate the performance of the network in terms of time consumption of recommendation, we randomly select the rating data of 300 users, 600 users, and 900 users from the Movie Lens data set to form the experimental data set of this article, denoted as D1, D2, D3, and the corresponding number of rating samples is shown in table 4.4. As the Figure 4.3 shows that the performance of training time has been decreased greatly after we prune the initial GCN-based model.

Experiment 4: Compare the overall performance among pruned GCN-based and other competing methods.

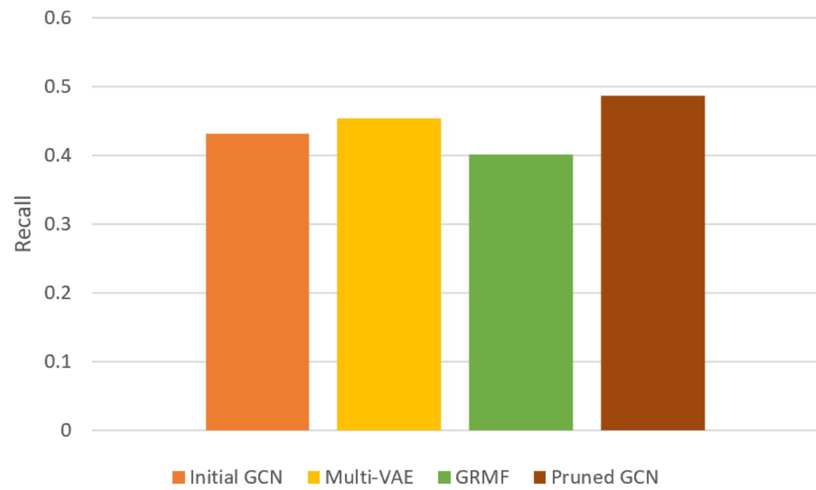

Figure 4.4 The comparison of overall performance among pruned GCN-based and other methods

The Figure 4.4 shows the performance of pruned GCN-based comparison with competing methods. For each method, the best score is selected, and the experiment shows that the performance improved about 12.8% over the initial GCN-based model on Movielens dataset, from the score of 0.4316 improve to 0.4869, shows the great improvement and potential of pruned GCN-based model.

Chapter 5 Conclusion

In order to make the GCN network more concise and appropriate for recommendation, we conduct the pruning experiment to simplify the design of GCN and test the result in Movielens dataset. Moreover, we add the side-information at product-side to utilize the label information of product. The experiment result shows the great improvement and potential of pruned GCN-based model, exhibiting substantial improvements (about 12.8% relative improvement on average) over the initial GCN-based model.

Chapter 6 Appendix

6.1 List of Academic Achievements

Wei Wang, Takaaki Ishikawa and Hiroshi Watanabe: “Facial Age Estimation by Curriculum Learning,” IEEE Global Conference on Consumer Electronics (GCCE) 2020, pp.203-204, Sep. 2020.

Wei Wang, Takaaki Ishikawa, and Hiroshi Watanabe: “Expression Recognition Using Curriculum Learning Method“, IEICE General Conference D-12-49, Mar. 2020

References

[1] A. Rutkowski and C. Saunders, "Growing Pains with Information Overload", Computer, vol. 43, no. 6, pp. 96-95, 2010.

[2] R. DeRubeis, Z. Cohen, N. Forand, J. Fournier, L. Gelfand and L. Lorenzo-Luaces, "The Personalized Advantage Index: Translating Research on Prediction into Individualized Treatment Recommendations. A Demonstration", PLoS ONE, vol. 9, no. 1, p. e83875, 2014.

[3] A. Ughade, "Personalized Location Recommendation System Personalized Location Recommendation System", International Journal of Applied Evolutionary Computation, vol. 10, no. 1, pp. 49-58, 2019.

[4] A. Pathak, K. Singh, "A Survey on Personalized Recommendation System for Web Services", International Journal of Science and Research (IJSR), vol. 5, no. 3, pp. 1569-1573, 2016.

[5] N. Sano, N. Machino, K. Yada and T. Suzuki, "Recommendation System for Grocery Store Considering Data Sparsity", Procedia Computer Science, vol. 60, pp. 1406-1413, 2015.

[6] P. Nguyen, J. Wang and A. Kalousis, "Factorizing LambdaMART for cold start recommendations", Machine Learning, vol. 104, no. 2-3, pp. 223-242, 2016.

[7] S. Desale, "Personalized Web Search with Privacy Protection", International Journal Of Engineering And Computer Science, 2016.

[8] A. Milicevic, A. Nanopoulos and M. Ivanovic, "Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions", Artificial Intelligence Review, vol. 33, no. 3, pp. 187-209, 2010.

[9] G. Zhu and T. Zhao, "Deep-gKnock: Nonlinear group-feature selection with deep neural networks", Neural Networks, vol. 135, pp. 139-147, 2021

[10] L. Zhang, T. Luo, F. Zhang and Y. Wu, "A Recommendation Model Based on Deep Neural Network", IEEE Access, vol. 6, pp. 9454-9463, 2018.

[11] J. Benedetto, P. Sanabria, A. Neyem, J. Navon, C. Poellabauer and B. Xia, "Deep Neural Networks on Mobile Healthcare Applications: Practical Recommendations", Proceedings, vol. 2, no. 19, pp. 550, 2018.

[12] L. Zhang, T. Luo, and F. Zhang, "A recommendation model based on deep neural network",IEEE Access, p.6, 2018

[13] D. Kim, C. Park, and J. Oh, "Convolutional matrix factorization for document context-aware recommendation", Proceedings of the 10th ACM conference on recommender systems. pp.233-240, 2016.

[14] N. Bach, D. Long and T. Phuong, "Recurrent convolutional networks for session-based recommendations", Neurocomputing, vol. 411, pp. 247-258, 2020.

[15] H. Kai, "Automatic Recommendation Algorithm for Video Background Music Based on Deep Learning", Complexity, vol. 2021, pp. 1-11, 2021.

[16] X. Wang, X. He, and M. Wang, “Neural graph collaborative filtering”, Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, pp.165-174, 2019.

[17] D. Tran et al., "HeteGraph: graph learning in recommender systems via graph convolutional networks", Neural Computing and Applications, 2021.

[18] X. He, K. Deng, and X. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation", Proceedings of the 43rd International ACM SIGIR conference, pp.639-648, 2020.