Movie Recommendation System by using FunkSVD and neural network

Di Li
Research School of Computer Science
The Australian National University
Canberra ACT 2601
Team name: Eleven benevolent elephants
u7039884@anu.edu.au

Yingjia Cai
Research School of Computer Science
The Australian National University
Canberra ACT 2601
Team name: Eleven benevolent elephants
u6592822@anu.edu.au

Zeji Hui
Research School of Computer Science
The Australian National University
Canberra ACT 2601
Team name: Eleven benevolent elephants
u6987679@anu.edu.au

Abstract

With the rapid popularity of the Internet and mobile Internet, the number of movies on the Internet is quite large, and people's demand for obtaining movies they are interested is increasing, and personalized movie recommendation systems have become a hot topic. This paper describes the use of FunkSVD and neural network to train two movie recommendation systems and compares the performance of the two recommendation systems. Finally, we found that the performance of FunkSVD is better than the trained neural network. In addition, the paper also uses PCA and Kmeans as data visualization methods to observe the similarity of movies.

1 Introduction

The Internet has experienced explosive growth in the last two decades. With the rapid increase in the number of users, behind it is the exponential increase in the amount of user data. In the face of massive amounts of information, users often feel powerless. Therefore, how to help users obtain the information that users are most interested in from the massive amount of information has gradually become a popular research work. At this place, recommender systems come into the picture and help the user to find the right item by minimizing the options [11].

In recent years, recommendation systems have been widely used in real-time information, news, social media, movie ratings, music, and e-commerce. Through the recommendation system, it can effectively solve the problem of information overload, analyze user ratings and historical data, and build user interest models for users. There is no need for users to fill out a large number of interest questionnaires, which greatly reduces the burden on users and increases users' satisfaction.

On the other hand, the massive amount of data is also a huge burden on the company, and a large number of items are updated and replaced every day. Common recommendation algorithms are based on user ratings. However, since new items do not have any ratings, it is difficult to recommend them to users who are interested in them. These new items may face the problem of not having a chance to be visited. Therefore, the recommendation system needs to predict the ratings of these new items, and predict whether the user is interested in the new items by training the user's historical records. Therefore, the recommendation system enables new excellent movies to be quickly noticed by users, increases the exposure of new movies, and can also charge a certain amount of advertising costs to increase the company's profit.

The summarise our main contributions in this works as:

- Proposed to use FunkSVD and neural network to train two movie recommendation systems by user ratings.

- Complete the two movie recommendation systems by FunkSVD and neural network, then compare their performances.

- Project the 100 movies on a 2D plane by using PCA and Kmeans, then explain their similarities.

2 Literature Survey

2.1 Collaborative filtering

For recommendation algorithms, content-based and collaborative filtering are the current mainstream recommendation algorithms. Collaborative filtering is a recommendation algorithm based on similarity, which is mainly divided into user-based collaborative filtering algorithm and item-based collaborative filtering algorithm. Collaboration means predicting items that users may like based on previous data. Filtering means to filter out some data that is not worth recommending.

User-based collaborative filtering uses statistical calculation methods to search for similar users of the target user (calculate the set of nearest neighbors through the similarity measurement method), and predict the target user's rating for certain items based on the similar users' rating on the items, and then find out the items with the highest predictions and recommend them to the user. Item-based collaborative filtering is similar to user-based collaborative filtering, but at this time we turn to the similarity between items. If the target user has rated some items, we can find the items with high similarity to these items and predict the target user's rating on them, and recommend several similar items with the highest rating to the user. This paper will use the item-based collaborative filtering algorithm to complete the movie recommendation system [12].

2.2 FunkSVD

Funk SVD is the original algorithm proposed by Simon Funk. He factorized the user-item rating matrix as the product of two lower dimensional matrices – the first one has a row for each user while the second one has a column for each item. The row or column associated to a specific user or item is referred to as latent factors. Despite its name, in Funk SVD, no singular value decomposition is applied [1].

2.3 Neural network

The neural network consists of three layers: input layer, hidden layer and output layer. The units of each layer are connected to all the units of the adjacent layer, and there is no connection between the units of the same layer. When a pair of learning samples are provided to the network, the activation value of the neuron propagates from the input layer to the output layer through each hidden layer, and each neuron in the output layer obtains the input response of the network. Next, according to reducing the error between the target output and the actual value, the connection weights are corrected layer by layer from the output layer through the hidden layers, and finally back to the input layer [13].

3 Problem

The problem of this paper is how to recommend a new movie to users who may be interested in it. We assume that given a labeled data set, the data set includes user information, movie information, and user ratings of movies. Our task is to predict the movies that the user may be interested in based on the user's ratings of the movies they have watched in the past, and recommend these predicted movies to the user

4 Method

This paper will use two methods to complete our movie recommendation system. One of the methods is to use the FunkSVD collaborative filtering algorithm, and the other is to use a 3-layer neural network to predict the movies that users may be interested in.

4.1 FunkSVD

FunkSVD was proposed when the traditional SVD faced computational efficiency problems. Because the traditional SVD decomposed the matrix into the product of three matrices, it was time-consuming and also faced the problem of sparseness. Therefore, the purpose of FunkSVD is to transform the matrix M is decomposed into the product of 2 matrices [2].

$$M_{m \times n} = P^T_{m \times k} Q_{k \times n}$$

But how does FunkSVD decompose matrix M into P and Q? The idea of linear regression is used here. Our goal is to make the error between user's rating and the rating obtained by the matrix product as small as possible, that is, the mean square error can be used as the loss function to find the final P and Q.
For a certain user rating m_{ij}, if FunkSVD is used to decomposed it, the corresponding is expressed as $q_j^T p_i$, and the mean square error is used as the loss function, then we expect $(m_{ij} - q_j^T p_i)^2$ to be as small as possible, if all items and samples are considered , Then we expect to minimize the following formula [3]:

$$\sum_{ij} (m_{ij} - q_j^T p_i)^2$$

As long as we can minimize the above formula and find the p_i and q_j corresponding to the extreme values, we can finally get the matrices P and Q, then for any blank rating position in matrix M, we can predict the rating by $q_j^T p_i$.
In addition, in order to avoid over-fitting, a regularization term of L_2 will be added, so the formal FunkSVD optimization objective function $J(p, q)$ is like this:

$$argmin \sum_{ij} (m_{ij} - q_j^T p_i)^2 + \lambda(\|p_i\|_2^2 + \|p_j\|_2^2)$$

where λ is the regularization coefficient, which needs to be adjusted. For this optimization problem, we can use the gradient descent method to optimize the results. Differentiating the above formula with respect to p_i and q_j respectively, we get:

$$\frac{\partial J}{\partial p_i} = -2(m_{ij} - q_j^T p_i) q_j + 2\lambda p_i$$
$$\frac{\partial J}{\partial q_j} = -2(m_{ij} - q_j^T p_i) p_i + 2\lambda q_j$$

The iterative formula of p_i and q_j is:

$$p_i = p_i + \alpha((m_{ij} - q_j^T p_i) q_j - \lambda p_i)$$
$$\mathrm{q}_j = q_j + \alpha((m_{ij} - q_j^T p_i) p_i + 2\lambda q_j)$$

Through iteration, we can get final P and Q, which can then be used for recommendation

4.2 3-layer neural network

The composition of the three-layer neural network is mainly divided into three parts: input layer, hidden layer, and output layer. The connection between the layers is full connection. The mathematical principle is mainly divided into three parts: 1. the forward propagation process; 2. the error back propagation process [14].

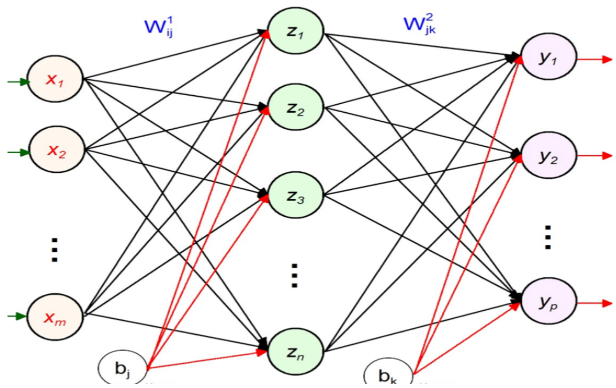

Figure 1: 3-layer neural network

In the forward propagation process:

$$z_{input} = \sum_{i=1}^{m} w_{ij}x_i + b_i$$
$$\mathrm{z}_{output} = \theta(z_{input})$$
$$\theta(z) = \frac{1}{1+e^{-z}}$$

where $\theta(z)$ is activation function, and then

$$y_{input} = \sum_{j=1}^{n} w_{jk}\theta(z_{input}) + b_k$$
$$\mathrm{y}_{output} = \theta(y_{input})$$

where b is bias and w is weight.

For the error back propagation, the stochastic gradient descent method is used to adjust the weight to minimize the error. The error can be obtained with the loss function.

$$L = \frac{1}{2}\sum_{k=1}^{p}(y_{target} - y_{predict})^2$$

5 Experiment

5.1 Benchmark Result

We implement our collaborative filtering system on the Funk SVD and Multilayer Perceptron (MLP). and the hyperparameter for the Funk SVD is the recommendation by Netflix Prize [5], with latent-variable=8, learning-rate=0.01, maximum-epoches=25, regularization=0.005. For MLP the hyperparameters are well tuned with one hidden layer and 40 hidden neurons, And we use Stochastic gradient descent (SGD) optimizer along with MSE loss to train our model. To evaluate the result of the experiment, we apply MAE (mean absolute error)and RMSEroot mean squared error) to test the accuracy of the model. Since the data set splits into 5 groups, we can evaluate the model in K-fold Cross Validation to derive a more accurate estimate of model prediction performance. The result shows in the Table 1 and Table 2below:

By observing the result of the experiment, we can find that both the Funk SVD and MLP could get a relatively good performance and the Funk SVD perform better than MLP model. However, compared to the benchmark result Table 2, there is still room for improvement.

Train set	u1	u2	u3	u4	u5	cross-valid
Funk SVD	0.95	0.95	0.94	0.93	0.94	0.94
MLP	1.15	1.13	1.11	1.11	1.12	1.12

Table 1: RMSE of the result

Train set	u1	u2	u3	u4	u5	cross-valid
Funk SVD	0.75	0.74	0.74	0.74	0.74	0.74
MLP	0.97	0.94	0.93	0.93	0.94	0.94

Table 2: MAE of the result

5.2 Benchmark Result

Visualization of Data After we implement the movie recommendation model, we design a data visualization method to display a small subset of the 1682 movies on a 2-dimensional plane to visualize their similarity. After the dimensional reduction by PCA and clustering by K-mean++, we visualized the 100 movie ratings randomly selected from the prediction matrix. Since the rating for a movie ranges from 1 to 5, we just simply set the clustering group into 5, and the result show below.

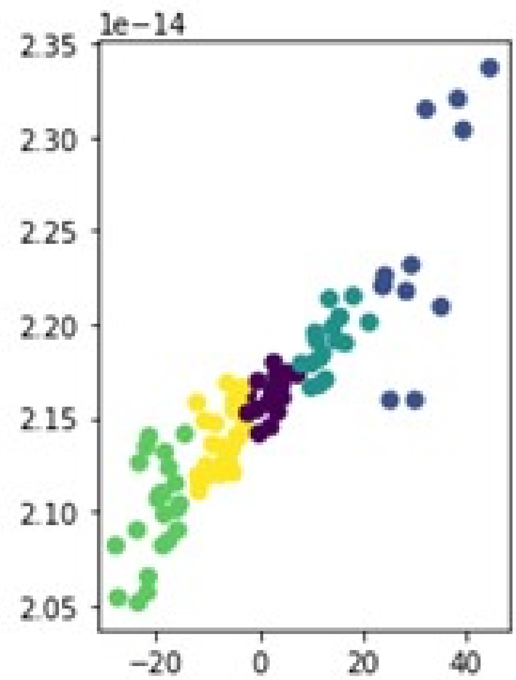

Figure 2: Visualization of 100 movie ratings

Obviously, we can find that there is a grouping phenomenon, in order to very that the grouping is valid, we calculate the mean rating for each of the movie show in table 3. We can find that in each of the group the mean rating for a movie is very similar there is an obvious stratification. With this phenomenon, we can evaluate the overview of popularity and public praise for a movie.

Method	RMSE	Comment
GraphRec	0.904	result from[5]
IGMC	0.905	result from[6]
MLP	0.91	result from[7]
Self-Supervised Exchangeable Model	0.91	result from[8]
Factorized EAE	0.92	result from[9]

Table 3: Benchmark of the reuslt

Group	Mean rating
1	2.8093897164163435
2	3.2229840335404067
3	3.503141915780966
4	3.7855726457000136
5	4.179602458975345

Table 4: group of movies in rating

6 Conclusion and Future Work

In this project, we propose and implement two collaborative filtering techniques for the movie recommendation system. We tuned the hyper-parameter for these two algorithms and get a relatively good result. We could predict the complete rating information for all of the movies which enables us to recommend a movie for a user how has not watched this movie. On the other hand, to evaluate this result, we design the visualization method to view the dimensional reduced matrix by clustering it, and we find that the data could cluster into several groups according to the ratings. With this finding, we can evaluate the public praise for a movie. However, compared to the baseline results, our model, the result has relative slow accuracy, which can be improved in the future work.

Funk SVD makes it easy for us to find a way to evaluate our recommendation engine and create good U and transform of T matrices, so that we can make recommendations even if we have a very sparse matrix. But funk SVD should not be used alone, because we may face a common problem in recommendation System, which is called "cold start problem". This problem means that we can't recommend for new users or new movies. A good way is to combine funk SVD with less advanced methods, such as ranking based algorithm [7] or content-based algorithm, This can be the future work. Besides that, there are some more details feature for movie and the users, those features may also connect with the rating for a movie by users, these will be the extra training data, along with this data we can get a better result up to 0.88 in RMSE [8], which could perform better than our model.

7 References

[1] Xue, W. (1995) FunkSVD: math, code, prediction, and validation.https://annie-wangliu.medium.com/funksvd-math-code-prediction-and-validation-4842bfaa219e

[2] Diven, S. Movie Recommender System: Part 1 https://towardsdatascience.com/movie-recommender-system-part-1-7f126d2f90e2

[3] Paul, S. Personalised restaurant recommendations using FunkSVD https://towardsdatascience.com/personalised-restaurant-recommendations-using-funksvd-3beff200b01c

[4] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets:History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages.DOI=http://dx.doi.org/10.1145/2827872

[5] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22, 1 (2004), 143–177.

[6] Muhan Zhang, Yixin Chen. 2019 Inductive Matrix Completion Based on Graph Neural Networks,CLR-2020

[7] Rianne van den Berg, Thomas N. Kipf, Max Welling. 2017 Graph Convolutional Matrix Completion. https://github.com/riannevdberg/gc-mc

[8] Jason Hartford, Devon R Graham, Kevin Leyton-Brown, Siamak Ravanbakhsh, 2017 Deep Models of Interactions Across Sets https://github.com/mravanba/deep-exchangeable-tensors

[9] Jason Hartford, Devon R Graham, Kevin Leyton-Brown, Siamak Ravanbakhsh 2018 Deep Models of Interactions Across Sets. ICML 2018

[10] Steffen Rendle and Li Zhang. 2019the Difficulty of Evaluating Baselines A Study on Recommender Systems.

[11] Prashant, B. Recommender Systems in Python `https://www.kaggle.com/prashant111/recommender-systems-in-python`

[12] Abhinav, A. Build a Recommendation Engine With Collaborative Filtering `https://realpython.com/build-recommendation-engine-collaborative-filtering/`

[13] JAMES, C. Neural Network `https://www.investopedia.com/terms/n/neuralnetwork.asp#:~:text=A%20neural%20network%20is%20a,organic%20or%20artificial%20in%20nature.`

[14] A.I. Wiki. A Beginner's Guide to Neural Networks and Deep Learning`https://wiki.pathmind.com/neural-network`

8 Personal reflection

9 Confidential peer review