Design and Implementation of Movie Recommender System Based on Graph Database

Ningning Yi
School of Computer Science Communication University of China
Beijing, China
1196372923@qq.com

Chunfang Li
School of Computer Science Communication University of China
Beijing, China

Xin Feng
School of Computer Science Communication University of China
Beijing, China

Minyong Shi
School of Computer Science Communication University of China
Beijing, China

***Abstract*—with the continuous development of Internet technology, information overload is becoming more and more serious. It's getting harder to get useful information from the network. Although the search engine can help users find information they need from the vast amounts of information in a certain extent, but cannot completely solve the problem of information overload, when users cannot accurately describe the information they need, you need to recommend system to help users find valuable information for users. So recommender systems are becoming more and more important. The movie recommender system implemented in this paper is based on the traditional user-based collaborative filtering algorithm, and the user project scoring matrix is pre filled. At the same time, database technology of this system uses graph database which is good at dealing with complex relations. In data visualization, the degree of recommendation of a movie is expressed by the size of the node and the thickness of the edge, so as to improve the user experience.**

Keywords—information overload; movie recommender systems; UserCF; graph database

I. INTRODUCTION

With the rapid development of Internet and Web2.0 technology, more and more people obtain information by Internet, and people gradually enter the era of information overload from the era of information scarcity. We are faced with a lot of complex and changeable information in the Internet with rich content, and it is very difficult for us to quickly find useful or interesting content. With the continuous development of the Internet, there are two main solutions: classification directory and search engine. The two solutions of the representative companies are Yahoo and Google[1].

But with the growing scale of the Internet, the categories are becoming more and more. It's impossible to display directories that cover almost all categories on a web page. So, gradually, classified directory as a solution to information overload is not adapted to the development of the Internet, and fade out the stage of history. To some extent, the search engine can solve the problem that users can find information quickly in the Internet environment with information overload. However, this method is still passive. It requires users to know some key words in the information they want to find. Sometimes they need to search, revise and search again to find the desired result. The recommender system changed the passive way. It establishes a link between the user and the information, that is to say, according to the user's behavior to infer what the user may need information. Help users find information that is useful or interesting to them, and it can provide users with unexpected, novel and interesting information. Thus, recommender systems are able to enrich the user's Internet experience [2].

Recommendation system is widely used in many fields such as electronic commerce, film and video, music, social networking and other fields, such as Facebook, Amazon. And Amazon is the earliest practitioner and promoter of personalized recommendation system. In China's film industry, the most successful movie recommender system is douban. We may have a habit of seeing the movie reviews in douban before we want to see a movie. Or see if it recommends a similar movie that is higher than the current score, and if the current film reviews are too bad or too low ratings, there's a good reason to recommend other good movies to watch.

II. INTRODUCTION OF RECOMMENDATION ALGORITHM

A. Introduction of Recommender Systems

Recommender systems use special information filtering techniques to recommend different items or content to users who may be interested in them. The figure below gives the working diagram of the recommendation engine[3].

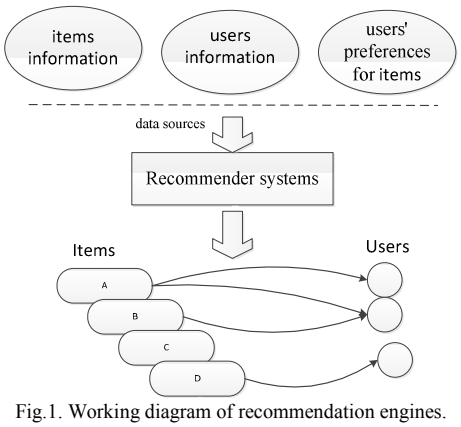

Fig.1. Working diagram of recommendation engines.

We consider the recommendation engine as a black box. The accepted input is the recommended data source. Generally, the data sources needed by the recommendation engine include:

978-1-5386-4806-3/17 $31.00 © 2017 IEEE
DOI 10.1109/WISA.2017.34

metadata of goods or content, such as keywords; user basic information, such as gender, age; users' preferences for items. According to the application itself, the user's preference information may include rating of the items, records of viewing items, purchase records. The preference information of these users can be divided into explicit user feedback and implicit user feedback.

Explicit user feedback is that users naturally browse or use web sites to explicitly provide feedback information, such as user ratings of items or comments on objects. Implicit user feedback is the data generated by the user using the site, implicitly reflects the user's preferences for items, such as the user to buy an article, the user to view the information of a commodity and so on.

Explicit user feedback can accurately reflect the user's true preferences for items, but requires users to pay extra costs. However, through some analysis and processing, implicit user behavior can also reflect the user's preferences. But the data is not very accurate, some behavior analysis exists big noise. But as long as the correct behavior characteristics are chosen, implicit user feedback can also get good results. The choice of behavioral characteristics may be very different in different applications[4]. For example, on e-commerce websites, buying behavior is actually an implicit feedback that can well reflect user preferences.

Most of the recommendation engines work on the basis of a similarity set of items or users. According to different data sources to find data correlation method can be divided into the following:

1) Demographic-based Recommendation: According to the basic information of system users, the correlation degree of users is found.
2) Content-based Recommendation: According to the metadata of goods, the relevance of objects or contents is found.
3) Collaborative Filtering-based Recommendation: According to the preferences of users for items, the correlation between items or content is found, or the relevance of users is found.

Collaborative filtering algorithm is one of the most widely used algorithms in Recommendation Algorithms. Its core idea is to choose based on collective intelligence. It finds similar neighbors of target users, and predicts and recommends target users according to similar neighbors. The algorithm can be divided into memory-based collaborative filtering recommendation algorithm, model-based collaborative filtering recommendation algorithm, and hybrid collaborative filtering recommendation algorithm. Among them, the memory-based collaborative filtering is divided into user-based collaborative filtering algorithm (UserCF) and item-based collaborative filtering algorithm (ItemCF) [5].

B. Introduction of User-based Collaborative Filtering

Collaborative filtering algorithm based on user method is the oldest algorithm in recommendation system. The birth of the algorithm marks the birth of the recommender system. The basic principle is that when a user needs personalized recommendation, you can find other users who have similar interests with the user, then, the objects, which are similar users favorite items, and that the target user has not bought, are recommended to the user. The diagram below vividly shows the principle of UserCF. And the target user is A.

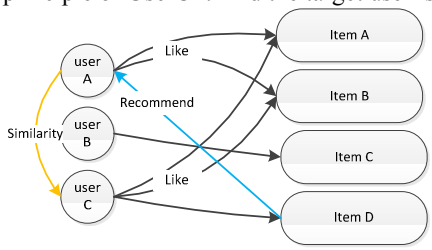

Fig.2. Working diagram of UserCF.

The collaborative filtering algorithm based on user method mainly includes two steps:

1) Find a set of users with similar interests to the target user.
2) Find the items that users like and haven't heard of in the collection and recommend them to the target user.

The key of step one is to calculate the similarity of interest between two users. The similarity calculation method used in the movie recommender system is Euclidean distance - the distance between two points x= (x_1, x_2, x_3… x_n) and y= (y_1, y_2, y_3 …y_n) in Euclidean space [6].

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^{\wedge}2} \quad (1)$$

When the Euclidean distance is calculated, if the two are completely similar, the two points are coincident, and the distance is 0. If the distance between the two is very close, the similarity is very low. However, we usually use [0,1] to express similarity when we use it. 0 means almost no similarity, and 1 means very similar, so we can use 1/ (1+d (x, y)).

Formula 2 calculates the degree of interest of user m to project u:

$$p(m,n) = \sum_{m \in S(m,k) \cap C(i)} w_{nm} r_{nu} \quad (2)$$

$S(m,k)$ contains k users closest to the user m interest, and $C(i)$ is the set of users who score the project I. w_{nm} represents the interest similarity between the user n and the user m, and r_{nu} represents the user n interest in the project u[7].

III. The Whole Inmplementation of The Movie Recommender System

A. Introduction of Graph Database

Database technology is divided into two categories: relational database (SQL) and non-relational database (NoSQL). Relational databases continue to provide theoretical bases for non-relational databases. That is to say, the development of NoSQL databases is based on relational databases, and many ideas and wisdom are derived from relational databases. NoSQL database is mainly divided into key-value database, document type database, column storage database and graph database.

Graph Database is a new NoSQL database based on graph theory. Its data storage structure and data query methods are based on graph theory. In graph theory, the basic elements of a

graph are nodes and edges. In graph database, nodes and relationships correspond to each other.

An obvious difference between a graph database and a relational database is the use of relationships to connect each node data rather than the foreign key. In a relational database, the relations between two tables or multiple tables are mutually referenced by using foreign key constraints. Searching for the matching primary key record in the main table to search and match the calculation operation is also through the foreign key. However, these operations will be the exponential level recorded in the table and consume a large amount of system resources. If using many to many relationships, you must add an intermediate table to save the foreign key correspondence of the two participating tables, which further increases the cost of the connection operation [8].

In graph databases, through relationships, we can associate nodes together to build complex models that are closely related to the problem domain. Each node in the graph database model directly contains a relational list, which stores the relationship records between the node and other nodes in the relational list. These relational records are organized by type and direction, and additional attributes can be attached. Whenever you run a relational database join operation, the graph database will use this list to directly access the nodes of the connection, without the need to record the search, match calculation operations.

The data model obtained by using graph database to organize data is simpler and more expressive than using traditional relational database or other NoSQL database. It can be used to model and manage data applications in a simple and intuitive way, and it can also make data units smaller and more standardized. At the same time, it can realize rich relational links [9].

Based on the above advantages of the database, the database technology of the movie recommender system is Neo4j. Neo4j is an open source NoSQL graph database implemented by Java, which realizes the storage of graph data model at the professional database level. Compared with common graphics processing and memory level databases, neo4j provides complete database features, including ACID transaction support, cluster support, backup and failover, and so on. In addition, the use of Neo4j only needs to understand the Cypher similar to the SQL language, and does not need to delve into the profound theoretical knowledge of graph theory. What's more, using neo4j can be developed with many existing class libraries. This movie recommender system is developed with Python language.

B. Data Set and Data Processing

The film recommender system uses Movielens100K data sets, including 100,000 ratings from 1000 users on 1700 movies. The score range is 1-5, and the greater the value, the higher the evaluation. The data formats of u.user, u.item, and u.data are shown in the following table.

TABLE I. DATA FORMATS

tables	field1	field2	field3	field4
u.user	user_id	age	sex	occupation
u.item	movie_id	name	release date	website
u.data	user_id	movie_id	score	timestamp

Design the following graph database schema based on the fields and relationship between tables.

1) (:User{id: user_id, age: age, sex: sex,})
2) (:Movie{id: movie_id, name: name, avgScore: avgScore })
3) (:User)-[r:RATE_IN{score:score}]->(:Movie)
4) (:User)-[r:SIM{ similarity: similarity }]-(:User)

"()" said node, "[]" said relationship. ":User" and ":Movie" represents type of nodes. :RATE_IN and :SIM represents type of edges. "{}" show attributes of nodes. For example, nodes of type User have ID, age, sex, occupation attributes. "()-[]-() "and "()-[]->()" indicates a simple graph model. The former is an undirected relation; the latter is a directed relation [10].

The above 1-3 modes can be directly converted from the original data source. First, we need to manually convert data into CSV format, and then use neo4j's LOAD CSV command to import data from the CSV file. The commands used in this system are as follows.

```
LOAD CSV WITH HEADERS FROM "file:///user.csv" AS line
MERGE(p:User{id:line.id,age:line.age,sex:line.sex,occupation:
line.occupation})
```

Fig.3. Command of import user.csv.

```
LOAD CSV WITH HEADERS FROM "file:///movie.csv" AS line
MERGE(p:Movie{id:line.id,name:line.name})
```

Fig.4. Command of import movie.csv.

```
LOAD CSV WITH HEADERS FROM "file:///rate.csv" AS line
match(from:User{id:line.START_ID}),(to:Movie{id:line.END _ I
D}) merge (from)-[r:RATE_IN{score:line.rating}]->(to)
```

Fig.5. Command of import rate.csv.

In the process of importing data, we need to pay attention to the following two points. First, we need to pay attention to the import path of csv file. The default path is <NEO4J_HOME>/import/my.csv. NEO4J_HOME is the path to the storage database. In my computer, this path is "C:\Users\Think\Documents\Neo4j\default.graphdb". Then I suggest using MERGE instead of CREATE. MERGE ensures that there is a specific schema in the graph database, and if that pattern does not exist, create it. In other words, there is no duplication pattern [11].

The average score attribute of the Movie tag node cannot be directly obtained from the source data. This attribute needs to be calculated with the cypher statement. The figure below shows the cypher statement of the average score of the movie with ID 1.

```
match(m:Movie{id:'1'})<-[r:RATE_IN]-(u:User{occupation:'educator'})
with round(avg(toFloat(r.score))) as  avgScore
SET m.avgScore= avgScore
return m
```

Fig.6. Cypher statement of calculating average score.

The similarity between users in the fourth models is based on the core algorithm of this paper – user-based collaborative filtering algorithm. However, due to the sparsity of the data, the user item scoring matrix is pre filled in the movie recommender system. The average score is used as the default score for movies that users don't judge. And this part of the work needs to be completed before calculating the user's recommended results; otherwise, if you put it in the back, then the system will respond slowly.

C. *The Whole Structure*

The background language of the movie recommender system is python. The program mainly uses the py2neo. It is a library working with Neo4j [12]. Through this library, you can build, query, modify, and delete nodes and relationships, no need to use the cypher language. And flask framework is used to render the front pages. Ajax encapsulated in jQuery implements the transfer of parameters between front and back. The front end uses Bootstrap framework. And the graph is drawn using echarts.

D. *Function of the Movie Recommender System*

The system default shows to movies with an average score of the top 50. The larger the radius of the node, the higher the recommendation. For movies with different averages, different colors are used to distinguish them.

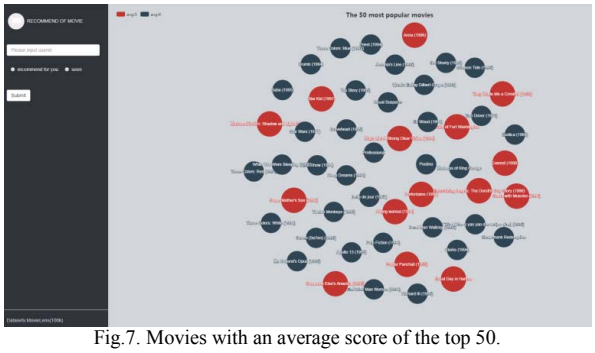

Fig.7. Movies with an average score of the top 50.

The figure below shows the recommended results for the user with ID 1. For the highly recommended movie, its node adds a yellow edge. And the radius of the node is also large. At the same time, the thickness of the edge also represents the recommendation of the film.

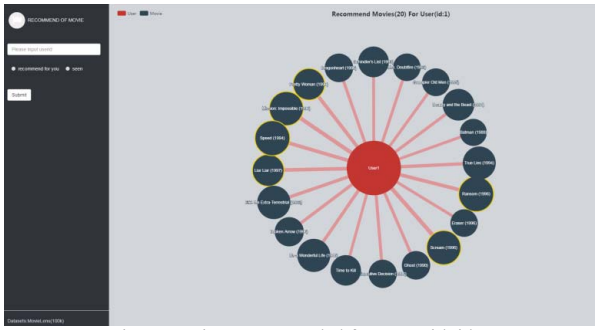

Fig.8. Movies recommended for user with id 1.

The figure below shows the seen movies for the user with ID 1. Because of the huge amount of data, only 30 movies are shown. Averages of different movies are distinguished by different colors. Moreover, the larger the radius of nodes and the thicker edges, the higher the score of movies.

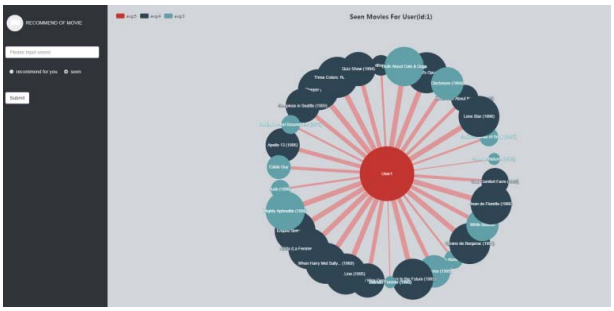

Fig.9.Seen movies for user with id 1.

IV. CONCULSION

On the basis of the traditional user collaborative filtering recommendation algorithm, taking into account the sparse matrix of the algorithm, the pre filling of the matrix is processed. Moreover, the system chooses the graph database which is good at dealing with the relation, it is helpful for the realization of the algorithm. Finally, the degree of recommendation of films is distinguished by color and size, allowing users to have a better experience.

ACKNOWLEDGMENT

This paper is supported by National Science Foundation of China (Grants No.61502437).

REFERENCES

[1] Donghua Liu, “Design and implementation of movie recommender system based on graph database,” Yuannan University, 2015.

[2] Liang Xiang, Recommendation system actual combat, Posts and Telecom Press, 2012.

[3] Shi Peng, “Research on collaborative filtering algorithm based on user interest and project characteristics,”Certral South University, 2015.

[4] Peng Lu, “Research and application of large data organization retrieval based on Neo4j,” Southeast University, 2015.

[5] Qi Qi, “User-based collaborative filtering on tag data,” Nanjing University, 2012.

[6] Xiaoqi Wang, “Research on collaborative filtering recommendation algorithm based on user,” Xidian University, 2015.

[7] Linyu Sua and Xuebin Chen, “Improvement of user-based collaborative filtering algorithm,” J.Computer engineering and software, vol.38, pp. 127-132, 2017.

[8] Fengqin Zi, Niu Jin, Lanzhu Bi and Jiamin Shen, “Design of movie recommender system based on graph database,” J. Software Guide, vol.15, pp. 144 -146, 2016.

[9] Minghao Han, “Survey of graph database system,” J. Computer CD software and application, vol.17, pp. 14 -15, 2014.

[10] Zhi Zhang, Guoming Pang and Jiahui Hu, Neo4j authority Guide, Tsinghua University Press, 2017, pp.100-150.

[11] http://neo4j.com/docs/developer-manual/current/.

[12] http://py2neo.org/v3/.