Research on Personalized Recommendation System for Graph Database

Yanjie Liang [a)]

College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

[a)] Corresponding author: 825129730@qq.com

Abstract. In the rapid development of the Internet, the abuse of Over Loading has become increasingly prominent in the production and life. Facing these challenges, recommender systems emerge as the times require. This paper first introduces the classic collaborative filtering recommendation algorithm and introduces a popular evaluation function to design a personalized recommendation algorithm based on graph database Neo4j and compares it with traditional relational database.

Key words: Collaborative filtering, graph database, hot evaluation function.

INTRODUCTION

Data surveys show that 90% of information on the Internet is useless to 90% of users, and users need to “filter” this information by themselves, which undoubtedly results in a significant reduction in information utilization [1].The emergence of search engines has greatly eased this contradiction. It is based on user needs to locate, such as keywords, can provide users with more accurate information services, but for the user's personalized needs, the traditional search engine seems powerless. The emergence of a recommendation system made it possible for personalized needs.

Until now, personalized recommendation systems have shown deeper and deeper development potentials and have penetrated into all aspects of the online world, affecting every aspect of our actual production and life. For example, e-commerce can provide buyers with more product choices; social networks can push more messages and discover potential friends. Personalized recommendation is one of the indispensable technologies in the current intelligent life [2].

INTRODUCTION TO NEO4J GRAPH DATABASE

The Graph Database organizes data stored by graphs and is one of the closest to high-performance data structures. It has better efficiency than traditional relational data in the processing of complex relational data. The secret is that the locality of the graph structure itself has a very fast traversal speed. At the same time, the traversal performance is not affected by the size of the graph database itself.

Neo4j is a member of the graph database family and has features such as scalability, high performance, and complete ACID transaction support. At the same time, Neo4j has unstructured data storage, has great flexibility in database design, and can provide more and more excellent algorithm designs.

We use the MovieLens part of the dataset as an example. The traditional database is usually stored in a table structure, a user table, a movie table, a score table, this relationship is achieved through the connection between the table and the table, and for more complex Relational performance, traditional databases often appear more redundant. In the graph database, it is more intuitive. As shown below:

Copyright © 2018, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

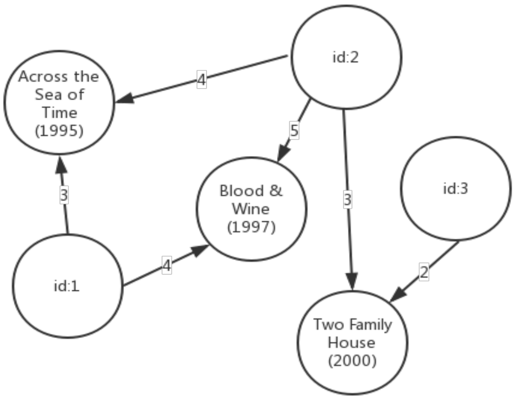

FIGURE 1. As shown below

User-based collaborative filtering algorithm

The user-based collaborative filtering algorithm is a classical algorithm in the field of recommended algorithms. It originates from the idea of "thinking together and grouping people". all users can benefit from the feedback evaluation of neighboring users. As long as each user contributes a piece of power to the system, the performance of the system will be improved. Then, user-based collaborative filtering has its drawbacks. Sparseness: Less than 1% of items in a large e-commerce recommendation system that users may buy. The overlap between items purchased by different users is low, causing the algorithm to find a user's neighbor. Cold start problem: When a new item first appears, no user has evaluated it, and user-based collaborative filtering cannot predict and grade it.

Finding Similar User Sets

The similarity calculation method between users mainly includes Jaccard formula, Pearson correlation coefficient or cosine similarity formula. This paper uses cosine similarity calculation method, as follows:

$$\mathrm{sim}(u,v)=\frac{\sum_{i\in I} r_{ui}\times r_{vi}}{\sqrt{\sum_{i\in I} r_{ui}^2}\times\sqrt{\sum_{i\in I} r_{vi}^2}} \tag{1}$$

Among them, u , v indicates that the users , r_{ui} , r_{vi} respectively indicate that the users u and v score the same item. I represents a collection of items.

However, for items that are popular or have a high rate of popularity in the project, the cosine similarity calculation method will have drawbacks. The more popular item scores cannot reflect the user's behavioral similarity. In order to reduce the "interference" of this type of data, we introduce a popular evaluation function:

$$\mathrm{pop}(i)=\frac{n_\mathrm{e}}{n_s}\times\frac{\sum_{u\in U} r_{ui}}{n_\mathrm{e} s_{\max}}=\frac{\sum_{u\in U} r_{ui}}{n_\mathrm{s} s_{\max}} \tag{2}$$

Among them, n_e , n_s represent the number of user nodes that generate relationships for item i , and the total number of user nodes, respectively, and $s_{\max}$ represents the maximum value given by the user, which is usually 5 points.

It is not difficult to see that this is a way of evaluating whether the project is a hot topic. It is divided into two parts. The first half represents the popularity of the project, that is, how many users of the project have scored it; The half represents the popularity of the project, that is, the proportion of users who have scored relationships with it. The search for the relationship in the traditional database often requires a lot of resources, and in the Neo4j graph database, through a node will quickly traverse all the nodes related to him, this feature is also mainly from the Special data structure.

Next we introduce the popular evaluation function in the calculation of similarity, as follows:

$$p\mathrm{sim}(u,v,i)=\frac{\sum_{i\in I}(1-pop(i))\times r_{ui}\times r_{vi}}{\sqrt{\sum_{i\in I}r_{ui}^2}\times\sqrt{\sum_{i\in I}r_{vi}^2}} \tag{3}$$

It can be seen that in the process of computing similarity, popular projects reduce its similarity. In this way, the similarity between users is more referenced.

Recommend items that are not directly related to users

After obtaining the similarity between users, for the specified target user u, we select the k user sets whose similarity is the closest, and use the set $S(u,k)$ to extract all the user-related items in the S and remove the target user u. A project with a rating relationship. The remaining items are scored and similarity weighted, and the results obtained are sorted. Finally, the target user u is recommended by the sorting result. Among them, for each possible recommended item i, the degree of interest of user u to it can be calculated by the following formula:

$$p(u,i)=\sum_{v\in S(u,k)\cap N(i)}p_{uv}\times r_{vi} \tag{4}$$

Among them, p_{uv} represents the degree of similarity between the user u and v, and r_{vi} represents the score that the user v generates for the item i.

In the next personalized recommendation process, based on the graph database, that is, the graph-based data organization will show advantages, the following structure:

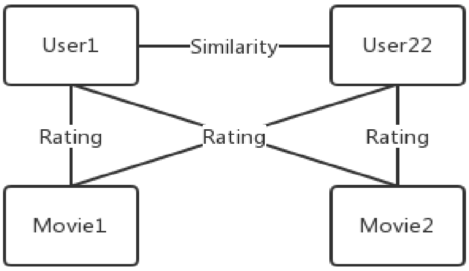

FIGURE 2. The following structure

In the graph traversal process, the search speed for the depth of 2 nodes is more efficient than the traditional database. The locality of the graph makes it not traverse edges or nodes that are not related to the starting node. This greatly improves the speed of recommendations.

FILM RECOMMENDATION SYSTEM BASED ON GRAPH DATABASE

Nowadays, recommendation system algorithms are widely used in social media, film and music portals, e-commerce platforms and other applications. In the recommendation system area, we will use the classic MovieLens dataset as experimental data and import it into the Neo4j graph database. Among them, two pictures are constructed with the film and the user as the nodes and the score relationship as the edge. Due to the large amount of data in the MovieLens dataset, we collected some data as a sample for testing.

FIG.1 User information form

userID	Sex	Age	Job	Timestamp
1	F	1	10	48067
2	M	56	16	70072
3	M	25	15	55117
4	M	45	7	02460

FIG.2 Movie information form

movieID	Name	Type
1	Toy Story (1995)	Animation|Children's|Comedy
2	Jumanji (1995)	Adventure|Children's|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama

FIG.3 Rating information form

userID	MovieID	Rating	Timestamp
1	1193	5	978300760
1	661	3	978302109
1	914	3	978301968
1	3408	4	978300275

FIG.4 The list of recommendations obtained is as follows:

Before the introduction	After the introduction
Assassins (1995)	An Unforgettable Summer (1994)
Slumber Party Massacre The (1982)	Slumber Party Massacre The (1982)
Slumber Party Massacre II The (1987)	Iron Eagle II (1988)
Sorority House Massacre (1986)	Bamboozled (2000)
Sorority House Massacre II (1990)	Adventures of Milo and Otis The (1986)
Bamboozled (2000)	Caligula (1980)

The approximate process of the algorithm is as follows:

1. According to the user's score recording of the movie, similarity is calculated for related users (users who have seen the same movie), and the popularity evaluation function is introduced in the process of calculating the similarity degree to increase the similarity to obtain accuracy, and between the users. Similarity is represented by a new edge.

2. Based on the calculated similarity user, select a user that is closest to the target user, and use the movie that the user has watched but the target user has not watched as a recommended movie candidate set.

3. Calculate the recommendation index of the candidate movie for the target user as the recommendation degree of the movie.

4. Sort the recommended value of the movie to get a list of recommendations.

CONCLUSION

A recommendation system based on a graph database has its own advantages. This advantage is based on an efficient data structure such as graphs. However, relying solely on structural advantages is not enough to fully exploit the advantages of graph databases. Taking this case as an example, what we have built is a simple modeling method of user->score->movie. No matter whether it is a node or an edge, there are many attributes, such as labels, categories, etc. Whether it is possible to extract one of the attribute values as a node separately to form a multi-partite model and build on this modelling basis., Combining the current classic recommendation algorithm to achieve a more efficient and stable algorithm, will also be the main research direction of the recommended system in the future map database.

REFERENCES

1. Recommended System Practice [M]. People's Posts and Telecommunications Press, Xiang Liang, 2012
2. Research on User Similarity Measurement Method in Collaborative Filtering Algorithm [J]. Ren Zhijian, Qian Xuezhong. Computer Program.2015(08)
3. Collaborative Filtering Recommendation in E-Commerce Recommendation System [J]. You Wen, Ye Shuisheng. Computer Technology and Development.2006(09)
4. Comparative research on internet recommendation system [J]. Xu Hailing, Wu Yi, Li Xiaodong, Yan Baoping. Journal of Software. 2009(02)
5. Wang Yulan. Research on embedded application of graphic database Neo4j[J]. Modern Electronic Technique,2012,35(22):36-38.
6. A collaborative filtering similarity measure based on singularities[J]. Jesús Bobadilla, Fernando Ortega, Antonio Hernando. Information Processing and Management.2011 (2)
7. Recommender system survey[J]. J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez. Knowledge-Based Systems.2013
8. A new similarity function for neighbors for each target item in collaborative filtering[J]. Keunho Choi, Yongmoo Suh.Knowledge-Based Systems. 2013