Collaborative filtering algorithm based on big data mining

Jianhao Li
1023041104
Nanjing University of Posts and Telecommunications
Jiangsu, Nanjing China

Abstract—**With the advent of the information age, people are faced with a large number of information choices. In this case, recommender system has become one of the effective means to solve the problem of information overload. As an important algorithm in the recommendation system, collaborative filtering can provide users with personalized recommendation content by analyzing users' historical behaviors and preferences. It is unique in that it does not require a clear categorization of items in advance, but rather enables accurate recommendations by mining the correlation between user behaviors. Collaborative filtering algorithms are mainly divided into two types: user collaborative filtering and item collaborative filtering. User collaborative filtering is based on the similarity between users, while item collaborative filtering is based on the similarity between items. Both methods rely on user historical behavior data to find the most similar users or items by calculating the similarity between users or items, and then make recommendations based on their behavior. Collaborative filtering algorithms have a wide range of applications in recommender systems, covering multiple fields, including e-commerce, social media, music, movies, etc. Although collaborative filtering has made significant achievements in recommender systems, there are still some challenges. One of them is the cold start problem, that is, for new users or new items, it is difficult to make effective recommendations due to the lack of historical data; Another challenge is data sparsity, where there is relatively little data on the user's interaction with the item. In this case, traditional collaborative filtering algorithms may fail. In addition, collaborative filtering algorithms also face problems such as user privacy and data security.**

Keywords—collaborative filtering algorithm,recommendation systems

I. INTRODUCTION

The recommendation algorithm used in the film recommendation system is based on collaborative filtering algorithm Collaborative filtering recommendation. Collaborative filtering algorithms have demonstrated superiority over traditional content-based filtering because collaborative filtering relies on user behavior and preferences, leveraging the collective wisdom of a user community to make personalized recommendations. This approach is dynamic and adaptable, as it continuously updates based on real-time user interactions. In contrast, traditional content filtering often relies on fixed rules or predefined criteria, which may not capture the evolving nature of user preferences.

The movie recommendation system references taste.Taste is a pivotal component within collaborative filtering algorithms, serving as a metric to gauge user preferences and similarities. In the context of recommendation systems, taste reflects a user's inclination towards certain items based on their historical interactions. The algorithm analyzes these tastes across a user community to identify patterns and correlations, comprehensive approach not only enhances the accuracy of music classification but also delivers a more enriched and enabling it to recommend items that align with a user's preferences. By leveraging taste, collaborative filtering enhances the accuracy and personalization of recommendations, ensuring that users receive suggestions that resonate with their unique interests and behaviors. This tool plays a crucial role in optimizing the user experience by tailoring content suggestions to individual tastes.

Collaborative filtering algorithms play a key role in movie recommendation systems by analyzing user behavior and preferences to provide personalized movie suggestions. There are two main types of collaborative filtering: user-based and item-based.In user-based collaborative filtering, the system recommends movies based on the preferences of users with similar tastes. It identifies users who have historically liked or disliked similar movies and suggests films that those like-minded users have enjoyed. This method leverages the collective wisdom of the user community to make personalized recommendations.On the other hand, item-based collaborative filtering recommends movies by identifying similarities between items themselves. The system analyzes the historical interactions of users with specific movies and suggests similar films based on shared characteristics or themes.

II. RELATED WORK

The concept of collaborative filtering was initially proposed by David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry in a paper titled "Using Collaborative Filtering to Weave an Information Tapestry" published in 1992. The early development of collaborative filtering focused on user-based approaches, where recommendations were based on the preferences of similar users. Over time, the field expanded to include item-based collaborative filtering and incorporated advanced techniques such as matrix factorization and machine learning models. This evolution aimed to improve recommendation accuracy and address challenges associated with scalability and the cold start problem, marking collaborative filtering's continuous refinement.

Recommendation algorithms can be roughly divided intothree categories: content-based recommendation algorithm, collaborative filtering recommendation algorithm and Hybrid Methods.

Content-based filtering recommends items by analyzing the intrinsic characteristics of the items and matching them with user preferences. This approach relies on item features, such as keywords, genres, or attributes, to create a user profile and recommend items that match the user's preferences. Content-based methods are effective in addressing the cold start problem, where there is limited user interaction data for new items.

Collaborative filtering is based on the idea that users who agreed in the past will agree in the future. It can be further divided into user-based and item-based approaches. User-based collaborative filtering recommends items based on the preferences of users with similar tastes. Item-based collaborative filtering recommends items by identifying similarities between items themselves. Both methods rely on user-item interaction data, forming a user-item matrix, and leverage this matrix to make personalized recommendations.

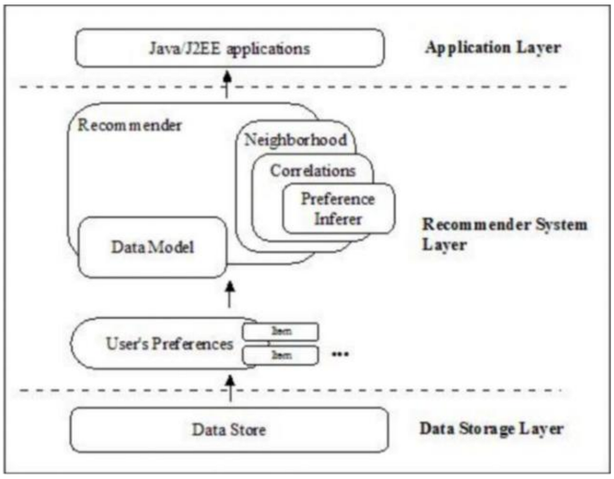
Collaborative filtering is based on the idea that users who agreed in the past will agree in the future. It can be further divided into user-based and item-based approaches. User-based collaborative filtering recommends items based on the preferences of users with similar tastes. Item-based collaborative filtering recommends items by identifying similarities between items themselves. Both methods rely on user-item interaction data, forming a user-item matrix, and leverage this matrix to make personalized recommendations.

Taste is an efficient implementation of a personalized recommendation engine provided by Apache mahout. The engine is implemented based on Java and has strong scalability. At the same time, some recommendation algorithms are transformed into MapReduce programming mode in mahout, so that the performance of recommendation algorithms can beimproved by using the distributed architecture of Hadoop.

At mahout 0 Taste in version 5 implements a variety of recommendation algorithms, including the most basic user based and content-based recommendation algorithms, the moreefficient slopeone algorithm, and the algorithm based on SVD and linear interpolation in the research stage. At the same time, taste also provides an extension interface for customized development of content-based or model-based personalized recommendation algorithms.

Taste is not only applicable to Java applications, but alsocan be used as a component of the internal server to provide recommended logic to the outside world in the form of HTTP and web service. Taste's design enables it to meet the requirements of enterprises for recommendation engine in terms of performance, flexibility and scalability.

The following figure shows the core components that make up taste:



As can be seen from the above figure, taste consists of the following main components:

Data Model: In the context of filtering algorithms, the Data Model plays a crucial role in representing and organizing user-item interactions. The Data Model serves as the foundation for collaborative filtering systems, encapsulating the user preferences and feedback on items. It typically involves constructing a user-item matrix, where each entry represents a user's preference or interaction with a specific item. This matrix forms the basis for similarity computations and recommendation generation. The Data Model essentially structures the input data, allowing algorithms to analyze and extract meaningful patterns from user behavior. It acts as a pivotal component in collaborative filtering by providing a structured representation of the user-item relationships essential for accurate and personalized recommendations.

Usersimilarity and itemsimilarity: User similarity and item similarity are crucial concepts in filtering algorithms, particularly in collaborative filtering methods. User similarity measures quantify the likeness between users based on their preferences and interactions with items. This similarity is utilized to identify users with comparable tastes, enabling personalized recommendations by suggesting items liked by similar users. On the other hand, item similarity gauges the resemblance between items themselves. It helps recommend items similar to those a user has shown interest in, enhancing the system's ability to uncover patterns and offer diverse yet relevant suggestions. Both user similarity and item similarity play key roles in shaping the effectiveness and accuracy of collaborative filtering algorithms.

Userneighborhood: User neighborhood is a pivotal concept in collaborative filtering algorithms, particularly user-based collaborative filtering. It involves identifying a user's neighborhood, which consists of other users with similar preferences. By analyzing the behavior of users within this neighborhood, the algorithm can make personalized recommendations for the target user. User neighborhood methods leverage the principle that users with comparable tastes tend to like similar items. This approach enhances recommendation accuracy by focusing on a subset of users whose preferences align closely, providing a more tailored and relevant set of suggestions based on the collective wisdom of the user community within the neighborhood.

Recommender: A Recommender in filtering algorithms serves as the core component responsible for generating personalized recommendations for users. It leverages various techniques, such as collaborative filtering or content-based filtering, to analyze user preferences and historical interactions with items. The Recommender uses this information to predict and suggest items that users are likely to be interested in. Whether employing user-based, item-based, or hybrid approaches, the Recommender aims to enhance user experience by delivering tailored and relevant content. It plays a crucial role in recommendation systems, providing users with suggestions that align with their preferences, ultimately optimizing user engagement and satisfaction.

III. PROBLEM STATEMENT

The method of similarity statistics is used to obtain adjacent users with similar hobbies or interests, so it is calleduser based collaborative filtering or neighbor based collaborative filtering. Taking movies as an example, this paper compares the preference of a specific user for movies with the information of other users. If a user has high information similarity with us, I will be called a similar user with the user, and recommend some movies that the user has seen or loved and that we have not seen for the specific user.

Algorithm steps:

1)Collect user preference information;

2)Looking for similar goods or users;

3)Generate recommendations.

IV. SOLUTIONS AND ALGORITHMS

This chapter introduces several similarity measurement functions used in the system. Each similarity measurement class has been implemented in taste. Both user CF and item CF depend on the calculation of similarity, because only by measuring the similarity between users or items can we find the user's "neighbors" and complete the recommendation. Common similarity calculation methods are described indetail below:

a) Pearson correlation-based similarity

Pearson correlation-based similarity is a method used in collaborative filtering algorithms to quantify the similarity between users or items based on their preferences or interactions. The Pearson correlation coefficient measures the linear relationship between two sets of data points and ranges from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.In collaborative filtering, Pearson correlation-based similarity is often applied to user-based filtering. For user similarity, the algorithm calculates the Pearson correlation coefficient between the preferences of two users for items they have both interacted with. This similarity measure is crucial for identifying users with similar tastes, forming the user neighborhood, and making personalized recommendations.

Expressed by mathematical formula, Pearson correlation coefficient is equal to the covariance of two variables divided by the standard deviation of two variables.

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y} = \frac{E(XY)-E(X)E(Y)}{\sqrt{E(X^2)-E^2(X)}\sqrt{E(Y^2)-E^2(Y)}}$$

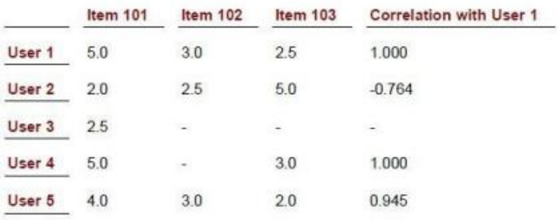
$$\rho_{X,Y} = \frac{N\sum XY - \sum X \sum Y}{\sqrt{N\sum X^2 - (\sum X)^2}\sqrt{N\sum Y^2 - (\sum Y)^2}}$$

$$\rho_{X,Y} = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}}$$

The similarity based on Pearson correlation coefficient has two disadvantages:

1)it assumes a linear relationship between user preferences, which may not always hold in real-world scenarios. Users might exhibit nonlinear preferences, and Pearson correlation may not capture such complex patterns accurately.

2)Pearson correlation is sensitive to outliers, meaning that a single highly-rated or low-rated item by one user can disproportionately influence the similarity measure. This sensitivity can lead to skewed similarity values, impacting the accuracy of the user neighborhood and, subsequently, the quality of recommendations. Outliers can distort the collaborative filtering process, making it less robust in the presence of extreme ratings.



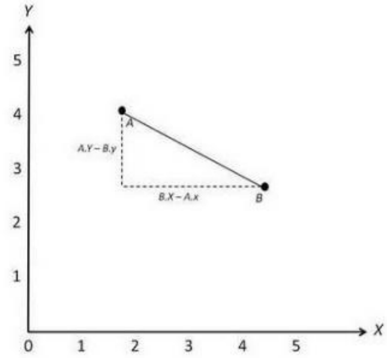
	Item 101	Item 102	Item 103	Correlation with User 1
User 1	5.0	3.0	2.5	1.000
User 2	2.0	2.5	5.0	-0.764
User 3	2.5	-	-	-
User 4	5.0	-	3.0	1.000
User 5	4.0	3.0	2.0	0.945

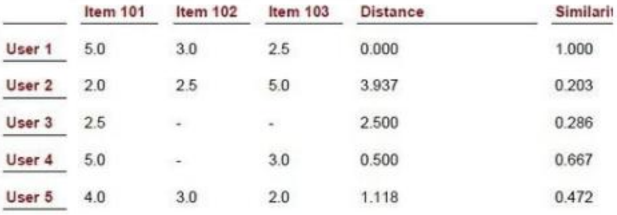
In the above table, rows represent some scoring values of users (1 ~ 5) for items (101 ~ 103). Intuitively, user1 and user5 use three common scoring items, and the score difference is not large. Logically, the similarity between them should be higher than that between user1 and user4, but user1 and user4have a higher similarity 1.

The same scenes often occur in real life. For example, two users watched 200 movies together. Although they do not necessarily give the same or completely similar scores, the similarity between them should be higher than that of the otheruser who only watched two identical movies! But this is not the case. If the similarity given by the two users is the same or very similar for the two films, the similarity calculated by Pearson correlation will be significantly greater than thatbetween users who have watched the same 200 films.

b) Euclidean Distance-based Similarity

Euclidean Distance-based Similarity is a metric used in collaborative filtering algorithms to measure the similarity between users or items. It calculates the Euclidean distance between the preference vectors of two users or items in a multidimensional space. The closer the vectors, the higher the similarity. It takes the items unanimously evaluated by people as the coordinate axis, then draws the people participating in the evaluation on the coordinate system, and calculates the lineardistance between them.





	Item 101	Item 102	Item 103	Distance	Similarit
User 1	5.0	3.0	2.5	0.000	1.000
User 2	2.0	2.5	5.0	3.937	0.203
User 3	2.5	-	-	2.500	0.286
User 4	5.0	-	3.0	0.500	0.667
User 5	4.0	3.0	2.0	1.118	0.472

In the figure, user a and user B scored items X and Y respectively. User a scores item x as 2 and item y as 4, indicating that it is coordinate point a (1.8, 4) in the coordinatesystem; Similarly, the score of user B on items X and Y is expressed as coordinate point B (4.5, 2.5), so the Euclidean distance (straight-line distance) between them is: sqrt ((b.x - a.x) ^ 2 + (a.y - b.y) ^ 2).

$$d(x,y)=\sqrt{\left(\sum (x_i-y_i)^2\right)}$$

The calculated Euclidean distance is a number greater than 0. In order to better reflect the similarity between users, itcan be regulated to (0, 1]. The specific method is: 1 / (1 + d).

$$sim(x,y)=\frac{1}{1+d(x,y)}$$

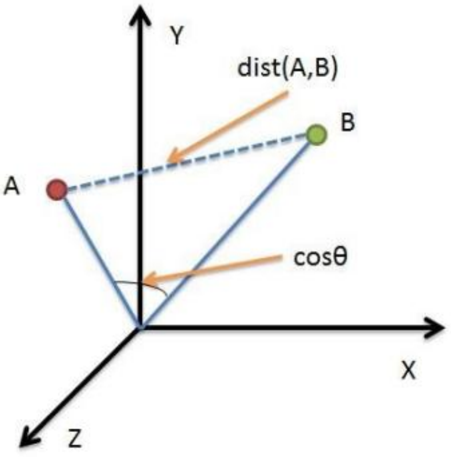
As long as there is at least one common scoring item, the similarity can be calculated using Euclidean distance; If there is no common scoring item, the Euclidean distance loses its effect. In fact, as a matter of common sense, if there is no common rating item, it means that the two users or rating items are not similar at all.

c) Cosine Similarity

Cosine Similarity is a metric widely used in collaborative filtering algorithms to measure the similarity between users or items based on their preferences. It calculates the cosine of the angle between two preference vectors, providing a measure of similarity irrespective of the vectors' magnitudes. This method is advantageous because it is scale-invariant, making it resilient to variations in rating scales among users. Cosine Similarity ranges from -1 to 1, where 1 indicates perfect similarity, 0 denotes no similarity, and -1 signifies perfect dissimilarity. It is a popular choice in collaborative filtering for its simplicity, effectiveness, and robustness in handling sparse and high-dimensional data.

$$sim(X,Y)=cos\theta=\frac{\vec{x}\cdot\vec{y}}{||x||\cdot||y||}$$

The difference between Euclidean distance and cosine similarity in terms of three-dimensional coordinate system:



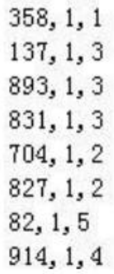
It can be seen from the figure that distance measurement measures the absolute distance between points in space, It is directly related to the position coordinates of each point (i.e. the value of the individual feature dimension); while the cosine similarity measures the included angle of the spatial vector, which is more reflected in the difference in direction than the position. If the position of point a remains unchanged and point B is away from the origin of the coordinate axis in the original direction, the cosine similarity cos at this time θ It remains unchanged, because the included angle remains unchanged, and the distance between a and B is obviously changing, which is the difference between Euclidean distance and cosine similarity.

According to the calculation methods and measurement characteristics of Euclidean distance and cosine similarity, they are applicable to different data analysis models: Euclidean distance can reflect the absolute difference of individual numerical characteristics, so it is more used for the analysis that needs to reflect the difference from the numerical size of the dimension, such as using user behavior indicators to analyze the similarity or difference of user value; Cosine similarity is more used to distinguish differences in direction than absolute values. It is more used to distinguish the similarity and differences of users' interests by users' content scores, At the same time, it corrects the possible inconsistency of measurement standards among users (because cosine similarity is not sensitive to absolute values). Mahout does not specifically give an implementation based on cosine similarity.

V. EVALUATION

A. Data Characteristics

Go to the GroupLens website(http://www.grouplens.org/node/12)Download data sets. In this movie system, we used nearly 900 users to evaluate nearly 100000 rows of data sets for 1683 movies. Ml data to be downloaded_ Take out the scoring data and movie information data in 0.zip. Convert the scoring file into a text file similar to CSV file format. CSV is a comma separated value file. It is a plain text file format used to store data. The file name is rating TXT.



```
358, 1, 1
137, 1, 3
893, 1, 3
831, 1, 3
704, 1, 2
827, 1, 2
82, 1, 5
914, 1, 4
```

Figure 1:Part of the data set

Then import the data into MySQL database, convert the movie data file into CSV format file, and then import it into the database. The database in MySQL is movie recommendation, and the corresponding tables of the above two files are movies and rating respectively. Because the taste engine needs frequent database operations, you can generally tune the MySQL database in my INI file to speed up the running time of database operations.

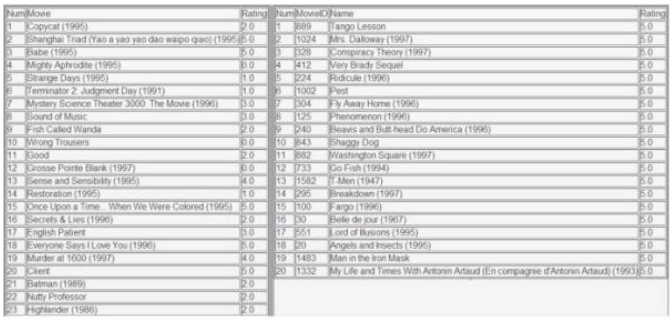
B. Experimental Results

The front navigation bar of the movie recommendation system has three menus: home page, recommended movie and parameter setting.

Homepage:The top 20 movies with the highest comprehensive score are displayed on the home page. The comprehensive score refers to the expected score of all users watching the modified movie for a movie. Implemented in index The database interface is called in the JSP page to query the database, and the results are displayed.

Parameter setting page:Since the collaborative filtering algorithm needs to set recommended parameters, the number of neighbors and similarity measurement function are mainly considered in this system.

Recommended movie page: on the recommended movie page, the user first needs to enter the user ID and the number of recommendations. In the JSP page, JavaScript is used to verify the legitimacy of the user's input data. The range of user ID is 1 to 990. After entering legal parameters, the system will display the recommendation results:



Num	Movie	Rating
1	Copycat (1995)	2.0
2	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)	5.0
3	Babe (1995)	5.0
4	Mighty Aphrodite (1995)	0.0
5	Strange Days (1995)	1.0
6	Terminator 2: Judgment Day (1991)	1.0
7	Mystery Science Theater 3000: The Movie (1996)	3.0
8	Sound of Music	3.0
9	Fish Called Wanda	2.0
10	Wrong Trousers	0.0
11	Good	2.0
12	Grosse Pointe Blank (1997)	0.0
13	Sense and Sensibility (1995)	4.0
14	Restoration (1995)	1.0
15	Once Upon a Time... When We Were Colored (1995)	5.0
16	Secrets & Lies (1996)	2.0
17	English Patient	3.0
18	Everyone Says I Love You (1996)	5.0
19	Murder at 1600 (1997)	4.0
20	Client	5.0
21	Batman (1989)	2.0
22	Nutty Professor	2.0
23	Highlander (1986)	2.0

Num	MovieID	Name	Rating
1	889	Tango Lesson	5.0
2	1024	Mrs. Dalloway (1997)	5.0
3	328	Conspiracy Theory (1997)	5.0
4	412	Very Brady Sequel	5.0
5	224	Ridicule (1996)	5.0
6	1002	Pest	5.0
7	304	Fly Away Home (1996)	5.0
8	125	Phenomenon (1996)	5.0
9	240	Beavis and Butt-head Do America (1996)	5.0
10	843	Shaggy Dog	5.0
11	882	Washington Square (1997)	5.0
12	733	Go Fish (1994)	5.0
13	1582	T-Men (1947)	5.0
14	295	Breakdown (1997)	5.0
15	100	Fargo (1996)	5.0
16	30	Belle de jour (1967)	5.0
17	551	Lord of Illusions (1995)	5.0
18	20	Angels and Insects (1995)	5.0
19	1483	Man in the Iron Mask	5.0
20	1332	My Life and Times With Antonin Artaud (En compagnie d'Antonin Artaud) (1993)	5.0

VI. CONCLUSIONS

This system is a movie recommendation system based on Apache mahout's taste. The data used are 990 users, 1618 movies and nearly 100000 comment data sets of GroupLens website. And use MySQL database as the data source. Provide user parameter selection. Users can set the number of neighbors and similarity measurement class. The final system outputs the movies scored by the user and the movies recommended to the user.

The advantages of collaborative filtering algorithm are as follows: Firstly, these algorithms enhance user experience by providing personalized and relevant content. By analyzing user preferences and interactions, filtering algorithms can suggest items that align with individual tastes, leading to increased user satisfaction and engagement.Secondly, filtering algorithms address the information overload problem by assisting users in discovering relevant items from a vast pool of options. In e-commerce, for example, these algorithms streamline the decision-making process by presenting users with tailored product recommendations based on their browsing or purchase history.Moreover, filtering algorithms contribute to increased user retention and loyalty. Personalized recommendations create a more enjoyable and efficient user journey, fostering a sense of connection with the platform or service. This, in turn, encourages users to return and explore additional content or products.

REFERENCES

[1] 郑策,尤佳莉.电影推荐系统中基于图的协同过滤算法[J].计算机与现代化,2019,(11):38-43+48.

[2] 李容.协同过滤推荐系统中稀疏性数据的算法研究[D].电子科技大学,2016.

[3] 闫燕.基于协同过滤算法的电影推荐应用研究[D].河北大学,2014.