RECOMMENDATIONS USING COLLABORATIVE FILTERING

Pratyush Kumar Mangalam[1], Rohan Raina[2], Saurabh Choudhary[3], Shikeb Ali[4], Anitha C[5]

[1234]BE, DEPARTMENT OF CSE, NIE, MYSORE, KARNATAKA, INDIA
[5]ASSISTANT PROFESSOR, DEPARTMENT OF CSE, NIE MYSORE, KARNATAKA, INDIA

Abstract - *Recommender systems are being increasing used to improve user experience in many areas. The most common technique used for recommendations is collaborative filtering. Recommender systems based on collaborative filtering work on past user-item relationships from a group of user who share similar taste and using these predictions are generated. In this paper we have explored how collaborative filtering can be used to recommend fitness plans to users. We have shown how ratings are collected from the user to rate items and how the similarity is computed to give personalised recommendations to users. Working of the proposed system as an android app has also been discussed.*

Key Words: recommender systems, collaborative filtering, ratings, personalised recommendations, fitness plans, android.

1. INTRODUCTION

People look for efficient, easy to use solutions that decrease the time required for completing recurrent activities. Additionally, they look for new and exciting things, compare alternatives, are interested in what similar or close people buy or achieve and hope to find a place / system / website that satisfies most of their interconnected interests. To sum up, people want applications that provide them with a solution that is genuine as well as efficient without having to search for information from various sources. Current websites regarding fitness only solve half the problem as information may be available but they are not personalised. In this domain, most of the existing websites present expert opinions by providing articles from which users only learn general information about fitness issues. There is no way to figure out users' needs and preferences and systems are inflexible by not being able to give a personalized solution to the user, tailored to their interests particular interests. On the other hand, websites that give recommendation based on other users' profiles by means of social similarity aren't reliable due to the uncertainty and, sometimes, lack of experience from other members.

Users expect from the recommender system to mix these two aspects social and expert-based and to provide them a solution they can confidently rely on. Collaborative filtering (CF) is a popular recommendation technique that bases its predictions and recommendations on the ratings or behaviour of other users in the system and can be used to provide a solution to the problem of meaningful recommendations. The fundamental assumption behind this method is that other users' opinions can be gathered in such a way as to provide a reasonable prediction of the active user's preference. Also they assume that, if users agree about the quality or relevance of some items, then they will likely agree about other items that are similar to those items in some way. The majority of collaborative filtering algorithms in service today, operate by first generating predictions of the user's preference and then produce their recommendations by ranking candidate items by predicted preferences.

1.1 Approaches for Collaborative Filtering

1. User-Based Collaborative Filtering

This approach is to calculate distances to quantify how closely two users match each other in respect with a certain common item. For example, if user1 and user2 put in same ratings in the same item, the distance will be 0. On the other hand, assuming they give different ratings, the distance will be farther depending on the difference.

2. Item-Based Collaborative Filtering:

Most recommender systems utilize an item-based collaborative filtering technique rather than a user-based one. For instance, when users who like item1 also like item2, the distance between two items is regarded as being close.

2. THE PROPOSED RECOMMENDER SYSTEM

The main contribution of this paper is to provide a practical implementation of a recommender system as an android application user to search fitness plant. However, collaborative filtering, that is, personalization based on a database of user preferences, is more difficult because we need to infer unknown preferences. Recommended structure give suggestion based on user inputs and rating. The measure we're going to be using to find the similarities in our data is called the Pearson Correlation Coefficient. Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations. The form of the definition involves a "product moment", that is, the mean (the first moment about the origin) of the product of the mean-adjusted random variables; hence the modifier product-moment in the name. It has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation. The reason for this choice is that it is effective and easy to implement and can be applied to discreet data such as user ratings.Because ratings given by real users are more neutral compared to expert/critic ratings. The formula for calculating the Pearson correlation coefficient is as follows:

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Fig -1: Formula Pearson Correlation Coefficient

Where 'x' belongs to one set of data and 'y' belongs to other set of data and 'n' is the number of pairs of such data. This can be used as a similarity measure in our model. Here, we have a collection of fitness plans, and each plan has a collection of reviews. Each review has a value from 1 to 5 describing how well they liked the product. This value can be based on what you are trying to profile. For example, you could use a value greater than or equal to 3 to suggest a user likes a particular fitness plan and a value less than 3 suggests that a user dislikes a plan. As long as we have a way to convert the data to numerical value, the data can be analyzed. The more the value of the coefficient is nearer to 1 the more two sets of data are similar and the more negative the value is the more opposite they are.

After ratings of fitness plans (items) have been obtained from users they are used to form a user-item matrix for calculation of similarities. For our model we are using item-item similarities to predict plans for users similar to those that they already like. Consider comparing any two plans, first we find users that rated both the plans all such pairs are considered, then Pearson score is calculated using the following algorithm:

i. First find the users that rated both fitness plans and form a user-item matrix consisting of all user ratings.
ii. Then compute the sums and the squared sums of the ratings, the sum of products of the ratings of the plans.
iii. Use these results to calculate Pearson's coefficient using the formula.
iv. Based on the calculations we find out similar fitness plans on the basis of Pearson's coefficient
v. These similar plans are then recommended to the users.

The figures below clearly show the working of the proposed system.

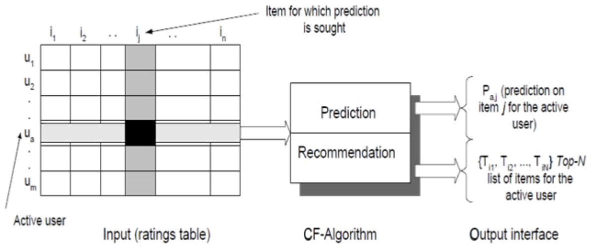

Fig 2: user-item matrix containing ratings for items (plans).

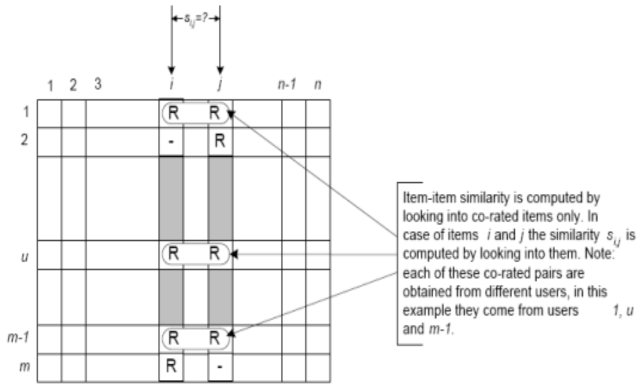

Fig 3: Item-Item similarities calculated between two items based on users who rated both items using the Pearson Correlation Coefficient.

The figures 2 and 3 clearly describe the way the proposed model works. After ratings of fitness plans(items) have been obtained from users they are used to form a user-item matrix for calculation of similarities. For our model we are using item-item similarities to predict plans for users similar to those that they already like. Consider comparing any two plans, first we find users that rated both the plans all such pairs are considered, then Pearson score is calculated to see if plans are similar. For every plan this process is carried out to find plans that are more similar to it. Users are then recommended plans based on similar plans the liked.

2.1 Android Application

Implementing the system as an android app makes the system more dynamic and easy to use. The flow chart for the application demonstrates the working of the application.

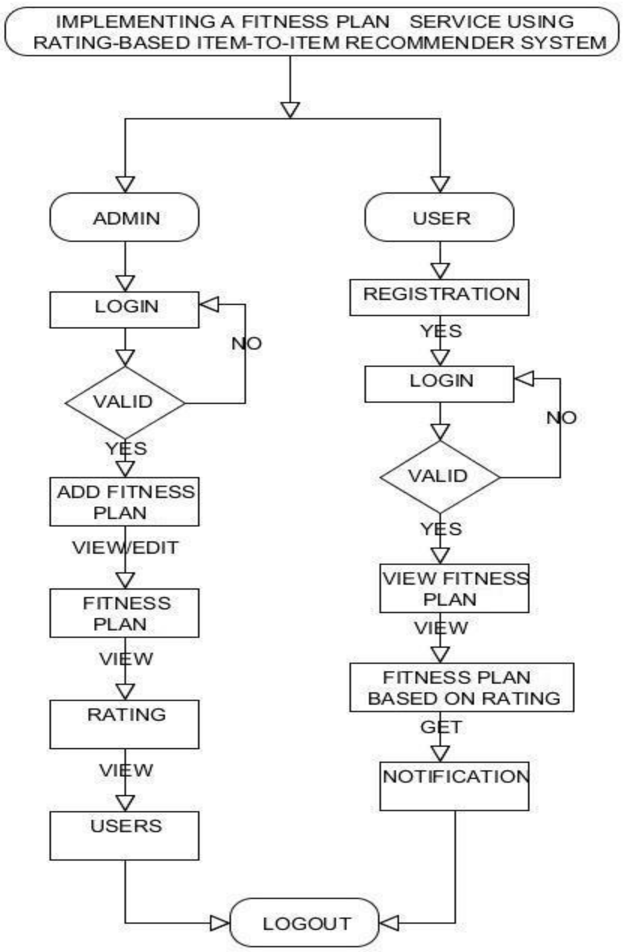

Fig 4: Flowchart for the proposed application.

By using the application users will be able to keep track of the plans they are following as all data related to the recommended plans is present in the application. Also an administrator module has been added to add/modify fitness plans. Administrators are the experts who have the knowledge in designing and maintaining fitness plans thus ensuring the users get access to carefully designed and genuine fitness plans.

2.2 Addressing Cold Start Problem

In the collaborative filtering approach, the recommender system would identify users who share the same preferences (e.g. rating patterns) with the active user, and propose items which the like-minded users liked (and the active user has not yet seen). But if a new user registers into the system , then there are no prior ratings for that particular user available with the database. So recommendations cannot be given based on ratings. So some other kinds of metrics are required to do the filtering initially. As we are rating fitness plans it is only obvious to use fitness related metrics to do the initial filtering of data. For our model we are using BMI (body mass index) to do the required filtering. Based on user height and weight the BMI can be easily calculated using the following formula:

BMI=M/H^2

Where M is the weight in kilograms and H is the height in meters. Based on the BMI users can be categorized as follows:

1. Less than 18.5: Underweight

2.18.5 to 25: Normal-weight

3. Greater than 25: Obese

So fitness plans can be categorized based on the BMI and initially when a user has not rated any plans those plans which fall under the user's BMI type can be recommended to the users. Similarly for plans that have not been rated yet, they can be recommended to users along with the plans being recommended based on ratings on the basis of BMI.

3. CONCLUSIONS

This adaptive recommending system, designed for people who want to achieve their goals and performances, combines expert knowledge with the social dimension. Dynamic adaptation to social changes makes our approach up to date and relevant. Data is built from multiple users' activities, with all the profile characteristics, purposes, trainings stored in our database. Users will have the possibility to see workout exercises and publicly available programs, make physical improvements, score better results and finally they will have all the required means of achieving their purposes. The model helps users get recommendations easily even if they don't have much knowledge about fitness related issues. Collaborative filtering implemented in such manner can help solve many such problems and also ensure user satisfaction.

REFERENCES

[1] J.S.Breese , D.Heckerman and C.Kadie. Empirical analysis of predictive algorithms for collaborative filtering. Fourteenth Conference on Uncertainty in AI. Morgan Kaufmann, July 1998.

[2] A Survey of Collaborative Filtering Techniques by Xiaoyuan Su and Taghi M. Khoshgoftaar.Department of Computer Science and Engineering, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA, August 2009.

[3] R. Zhang, Q. d. Liu, Chun-Gui, J. X. Wei and Huiyi-Ma, "Collaborative Filtering for Recommender Systems," *2014* Second International Conference on Advanced Cloud and Big Data,Huangshan,2014,pp.301-308,doi:10.1109/CBD.2014.47

[4] Y. Dou, H. Yang and X. Deng, "A Survey of Collaborative Filtering Algorithms for Social Recommender Systems," 2016 12th International Conference on Semantics,

Knowledge and Grids (SKG), Beijing, 2016, pp. 40-46. DOI: 10.1109/SKG.2016.014

[5] L. Sheugh and S. H. Alizadeh, "A note on Pearson correlation coefficient as a metric of similarity in recommender system", AI & Robotics (IRANOPEN), Qazvin, 2015, pp. 1-6.doi: 10.1109/RIOS.2015.7270736

[6] https://en.wikipedia.org/wiki/Collaborative_filtering