Recommendation System Using Hybrid Filtering

Aayush Ladda[1], Meet Shah[2], Ruchi Davda[3], Ms. Dhanashree Toradmalle[4], Mr. Harish Motekar[5]

[1]*Dept. of Information Technology, Shah & Anchor Kutchhi Engineering College, Mumbai (M.S.), India*
[2]*Dept. of Information Technology, Shah & Anchor Kutchhi Engineering College, Mumbai (M.S.), India*
[3]*Dept. of Information Technology, Shah & Anchor Kutchhi Engineering College, Mumbai (M.S.), India*
[4]*Associate Professor, Shah & Anchor Kutchhi Engineering College, Mumbai (M.S.), India*
[5]*Assistant Professor, Shah & Anchor Kutchhi Engineering College, Mumbai (M.S.), India*

---***---

Abstract - *The purpose of recommendation system is to apply multiple data processing techniques and prediction algorithms aimed to predict users' interest on services, information and products. Today, they have become an unavoidable service in our day-to-day lives. The constant increase in the number of users of the system possesses few challenges like managing and handling a large amount of user data and producing accurate recommendations. Therefore, the new recommendation system aims at implementing technology able to deliver fast and accurate recommendations even for large information sets. This paper aims at developing a recommender system using hybrid of content-based and collaborative filtering which yields highly accurate recommendations.*

Key Words: Recommendation system, Content-based filtering, collaborative filtering, Hybrid system, LightFM, Predictions.

1. INTRODUCTION

In the last few decades, with the rise of YouTube, Netflix, Spotify and many such web services, providing relevant recommendations to the users has gained a lot of importance. From providing relevant digital content, personalized advertisements, products and services as per the interests of each and every individual, recommendation systems have already become a significant part of our everyday lives.

Recommendation systems have a very wide area of applications and have become a critical component of some industries. Also, they have emerged to be a service generating huge amount of income by helping companies stand out from their competitors. They help improve the reach of the content to such users who would be interested in that service/product and potentially help them in generating revenues. So, in a very general way we can say that recommendation system are algorithms intended to deliver relevant data to the users with an intention to increase the reach of the service provided.

In today's day and age, accuracy plays a very important role in any technology. Collaborative filtering and content-based filtering being the newest technologies in building any recommendation system possesses some of their own advantages and disadvantages. Keeping all these things into consideration, we intent to develop a recommender engine which would help overcome these disadvantages with the individual algorithms being used at present. In the proposed system, we'd use a hybrid of content-based filtering and collaborative filtering algorithms to develop a recommendation engine with high accuracy and precision.[8]

2. TYPES OF RECOMMENDATION SYSTEMS

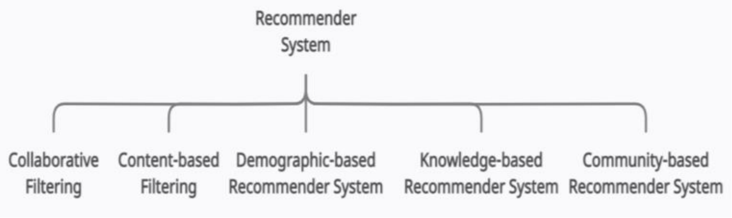
Recommender System
Collaborative Filtering
Content-based Filtering
Demographic-based Recommender System
Knowledge-based Recommender System
Community-based Recommender System

Fig -1: Types of Recommender System

2.1 Knowledge Based Recommendation System

In knowledge-based recommendation system recommendations are made specific queries made by the user. Users' needs and preferences are identified and system suggest objects based on that. It works on functional knowledge i.e. the how an item meets a user's needs. This recommendation system is applied when content and collaborative approaches cannot be applied. These recommendations are not based on explicit data such as user ratings.[4]

2.2 Demographic Based Recommendation System

In demographic based recommendation systems users are classified based on personal attributes into demographic classes and recommendations are made based on the demographic classes. This model is not that easy to implement as it requires a thorough understanding and market research. It does not require any explicit data such as ratings. The costumer provides details via surveys.[1]

2.3 Community Based Recommendation System

This recommendation model predicts and recommends items to a particular user based on explicit feedback such as ratings and implicit feedback such as items purchased, item liked by user, Wishlist products etc. based on community of users. People tend to rely on groups of their specific communities. Problems of cold start are not encountered because even if the user is new but similar users may exist.

2.4 Content–Based Recommendation System

We begin with a user and establish set of item user likes both explicit and implicit data is considered (user profile is built) and item profile (description of the item). Profile may be a set of features for instance for movies (author, director, actor… etc). Item profiles are vectors. Vectors are often real valued or Boolean. Profile refers vital words within the item (document). The technique used for locating vital words in a document is TF-IDF (term frequency * inverse document frequency).

f_{ij} = frequency of term(feature) i in document(item) j
$TF_{ij} = f_{ij}/(\max_k * f_{kj})$

Note that we have a tendency to normalize TF to discount for larger documents
n_i = range of documents that mention term i
N=total range of docs
$IDF_i = \log N/n_i$ more common the term larger the $n_{i,}$ lower the IDF_i.

So higher rate of n_i to fewer common words and lower rate to a lot of common words.
TF-IDF score: $w_{ij} = TF_{ij} * IDF_i$

Set of words with highest TF-IDF score is named document profile.

For a given document the TF-IDF scores are computed for every and each term (word) within the document and then sort all the terms within the document consistent with their TF-IDF scores then we choose words with highest TF-IDF scores that will be the top profile(feature) of that particular document here top profile is real valued vector.

Now item profiles are constructed, next we do is build user profiles
User profiles
Users has rated items i_1, ……………………, i_n
i_1, ………… ,i_n are vectors of entries

Simplest way to create user profile is to create average of item profiles
$(i_1+……………+i_n)/N$
N here is total number of item profiles.

This doesn't take into account whether a user has liked a certain item more than others so we can use weighted average where the weight is equal to rating given by the user for an item then we will have weighted average item profile.

Another approach can be normalizing the ratings using average ratings of the user.

Many other sophisticated approaches are also possible.

Now when we have user profiles and item profiles the next task is to recommend certain items to the user.

The key step in this is to take a pair of user profile and item profile and find out what the rating for that user and item pair is likely to be. Both user profiles sand item profiles are vectors in high dimensional space. In reality we have shown in 2-dimensional space in reality they are embedded in a much high dimensional space.

Making predictions,
Take user as x and item as i.
Estimate $U(x,i)=\cos(\theta)=(x.i) / (|x|.|i|)$
Technically the cosine distance is the angle θ and the cosine similarity is 180- θ

For convenience we use cos(θ) as similarity measure and call it the "cosine similarity" in this context.

As θ becomes larger, cos(θ) smaller and smaller, more smaller the θ, more similarity between x and i and more likely that x will give higher rating to i.

For prediction of a user compute cos(θ) for all items in the catalogue and then pick the items with highest cosine similarity and recommend these to our user.[6]

3. CONTENT-BASED RECOMMENDATION MODEL

3.1 Advantages of Content-Based Recommendation System

- No need for data from other users.
- Can recommend niche items.

3.2 Content–Based Recommendation System

- Predicting ratings and providing future recommendations results may not be as accurate due to irrelevant item description.
- Not very accurate as there may be more relevant features of an item than just the one's listed.
- Sometimes text provided by the user may be null and may lead to system failure.
- Only safe recommendations generated (recommendations stay within user's safe bubble of the embeddings space)
- Need domain knowledge- a human has to label the features. Example: - movie genres, text description for the items, collect audio, video attached to that item

4. COLLABORATIVE FILTERING RECOMMENDATION SYSTEM

This filtering is probably the most widely used algorithm in recommendation systems Collaborative systems are based on collecting and analyzing a large amount of data both explicit and implicit and provide recommendations based on inter-user comparisons activities and predicting what users will like based on their similarity to other users' technologies Suppose we consider user x. We are going to find a group of other users whose likes and dislikes are similar to user x. In our cases group of users. Suppose a group of user's likes a same item as user x then we call this set of users the neighborhood of user x. Once we find the set N of users similar to user x then we find other items that are liked by a lot of users in the set N and recommend those items to user x. This is the basic idea behind collaborative filtering.[3]

Idea here is- you would probably like things people with similar viewing habits ,preferences as of yours would like .Instead of dreaming about the features used to connect movies and people here we use user preference data to generate the features for example we would have incomplete set of preferences data and instead of learning and discovering the relevant features., we use patterns in the data and this is done by simply reversing the problem here we perform approximate factorization into two matrices and we can do this using machine learning approach. The job of machine learning algorithm is to generate values for those matrices that will match the existing data in the preference data as closely as possible. Once our estimation is done, we multiple matrix X as before to fill in all the missing values and we name the discovered features as latent features because they arise as a result of underlying patterns and data. One can think of that as an average of weighted sum of the patterns in the data they are not based on human defined features such as comedy.[2]

The method that the machine learning algorithm uses to find missing ratings is centred cosine similarity (Pearson Correlation). Suppose consider A, B, C D as users and itemPQ, itemRS, itemTU, itemVW, itemXY, itemYZ, itemQP as movies.

	ItemPQ	ItemRS	ItemTU	ItemVW	ItemXY	ItemYZ	ItemQP
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Table -1: Grouping Similar Users and depicting respective ratings

Here, we normalize the ratings by subtracting the row mean.

	ItemPQ	ItemRS	ItemTU	ItemVW	ItemXY	ItemYZ	ItemQP
A	2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

Table -2: Normalized Ratings

sim(A,B)=cos(r_A,r_B)=0.09
sim(A,C)= -0.56
sim(A,B)> sim(A,C)

- This method captures intuitions better.
- Handles tough raters and easy raters.

Predict ratings
$r_{xi} = (\sum_{y} \in_{N}\ S_{xy}.\ R_{yi}\ ) \div (\sum_{y} \in_{N}\ S_{xy})$
where $S_{x,y}$=sim(x,y)

4.1 Advantages of Collaborative Recommendation Model

- No domain knowledge required.
- Data is generated by users interacting with the items which can be used to harness information related to other users.
- Can solve problems of safe recommendations because not only can collaborative filter see the user of interest points in embedding space it can also tune in to users points in embedding space and find similarities between them.
- Collaborative filtering is a great staring point. Using user-item interaction data we can create a quick baseline model that we can then check against other models to find gaps to fill by using other recommendation systems as content-based makeup the lack of data just like the rest of the machine learning is important to find out what works the best.

4.2 Problems with Collaborative Filtering

- User and Item cold-start problem: Not enough known about new user to decide who is similar (and perhaps no other users yet). Similarly, when an item is interacted with a lot of users will have very good idea about what type of users will like the item but with new items there is little to no interaction data for that item so we really don't have a great idea because the user sample size is so small or non-existent
- Sparsity: Even the items with high number of purchases with have very few ratings. When recommending from an outsized item set, users can have rated just few of the items so it is difficult to find out similar users. Matrix multiplication in collaborative filtering takes our user-item interaction matrix A and factorize it into two

smaller matrices U for users and V for items each with a dimension of number of latent factors. It is not easy to tell looking at this toy problems with very few users and items but as these both increases the number of interactions between them becomes very sparse imagine millions of users and thousands or millions of items even the most active user will interact with only a small sample of items and even the most popular item will usually be interacted with by a small set of users. This can lead to scalability problems later on as well.

- Scalability: With large number of users and products, computations become slow and huge amount of power is necessary to calculate recommendations.
- No context feature-no domain knowledge needed in our model can be pretty useful. This lack of context feature can reduce performance of our collaborative filtering models and this leads to combining our model with others like content based.[5]

5. WHY HYBRID?

The idea is that independent errors within each model will cancel out and we will have much better recommendations. Using a hybrid system of content based as well as collaborative filtering will lead to solving the problem of cold start of the system from collaborative filtering and safe recommendations from content-based recommendations and will also help us to provide accurate results as it will be a result of combination of two algorithms.[7]

6. WHY LIGHTFM?

In our hybrid system if we use LightFM it will help us to a great extent because it determines which filtering technique should be used first, collaborative or content based. For example, if there is no item data present it will use content-based filtering and from the data retrieved from the user while he creates his/her profile, and then item will be recommended and vice versa.

And if already data is present regarding items and a new user joins without entering any specific data about his/her preferences, then he/she will be matched with other similar users using collaborative filtering, the above procedure will be performed by LightFM and hence is best for our recommendation System.it also implements WARP : Weighted Approximate Rank Pairwise loss for implicit data which will be used from our dataset. It is tested and used by many developers for different recommendation systems, API is user friendly, and is very interactive, it is very fast; it is a highly reputed python model.

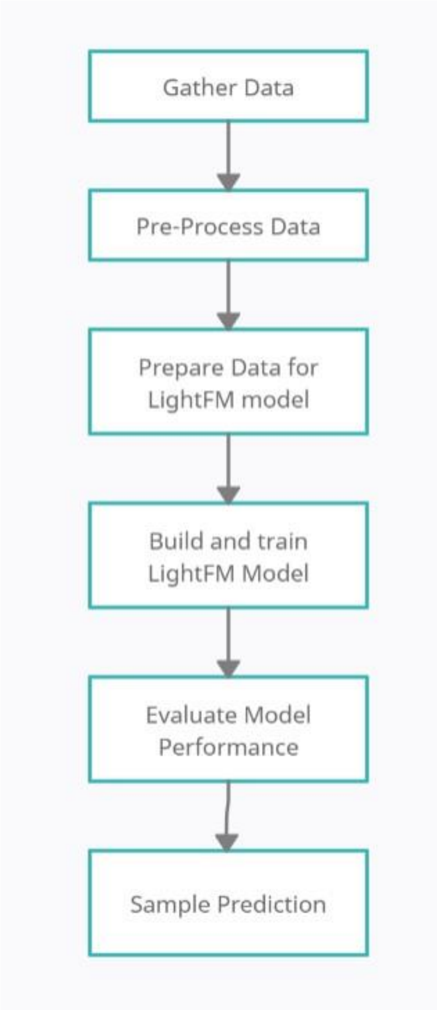
Gather Data
Pre-Process Data
Prepare Data for LightFM model
Build and train LightFM Model
Evaluate Model Performance
Sample Prediction

Fig -2: Flow of our Recommendation System

7. CONCLUSIONS

In this paper we have discussed about various algorithms used in recommendation systems and their various advantages and disadvantages. Taking all these into consideration, we chose a hybrid of content-based filtering and collaborative filtering for the recommendation system with the help of LightFM model. Our proposed system would help overcome the drawbacks faced by the individual methods, hence, improving the accuracy of the system.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our Professors Ms. Dhanashree Toradmalle and Mr. Harish Motekar for their constant help and guidance throughout the research. The research would not have been possible without their constant and exceptional support. Their expertise and generosity helped us improve this study.

REFERENCES

[1] Pazzani, M.J. "A Framework for Collaborative, Content-Based and Demographic Filtering" (1999) Artificial Intelligence Review 13, 393–408

[2] Bixi Wang, Wenbin Wu, Wenfeng Zheng, Yan Liu and Lirong Yin "Recommendation Algorithm of Crowdfunding Platform Based on Collaborative Filtering" (2020) Journal of Physics: Annual International Conference on Computer Science and Applications

[3] Z. Zhang et al., "A Novel Social Situation Analytics-Based Recommendation Algorithm for Multimedia Social Networks," in IEEE Access, vol. 7, pp. 117749-117760, 2019

[4] Iateilang Ryngksai and L. Chameikho "Recommender Systems: Types of Filtering Techniques" International Journal of Engineering Research & Technology (IJERT) Vol. 3 Issue 11, November-2014

[5] Mohammadsadegh Vahidi, Farashah, Akbar Etebarian, Reza Azmi & Reza Ebrahimzadeh Dastjerdi "A hybrid recommender system based-on link prediction for movie baskets analysis" J Big Data 8, 32 (2021)

[6] Geetha G, Safa M, Fancy C, Saranya D "A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System" National Conference on Mathematical Techniques and its Applications (NCMTA 18)

[7] Maciej Kula "Metadata Embeddings for User and Item Cold-start Recommendations" arXiv:1507.08439v1 [cs.IR] 30 Jul 2015

[8] Pooja V. Chavan, Prof. P.M. Chawan "Recommender System for volunteers in connection with NGO" International Journal of advanced studies in Computer Science and Engineering IJASCSE, Volume 4, Issue 7, 2015