A Hybrid Approach towards Content Boosted Recommender System

Prof. Rohini Nair[#1], Bhushan Mehta[#2], Bhavesh Gor[#3], Jeet Bhanushali[#4]

[#]*Computer Department, KJ Somaiya College of Engineering, Vidyavihar- Mumbai University*
KJ Somaiya College of Engineering, Vidyavihar, Mumbai, India

[1]rohininair@somaiya.edu
[2]bhushan.m@somaiya.edu
[3]bhavesh.gor@somaiya.edu
[4]jeet.b@somaiya.edu

Abstract— With the exponential increase in data over the web the users face the problem in retrieving relevant knowledge. For eliminating this problem recommenders are used. They are based on one of the traditional recommendation approaches – content based approach and collaborative based approach. Recommendation can be provided to users using past user activities with help of data mining concepts and the market trend can be merged with it to provide optimized results from recommender.

The user profile similarity for personalization, the hit based approach for new movies, history based approach all tackle one problem or the other faced by the traditional recommender systems. The paper proposes a new hybrid approach which combines the effect and positive functionality of all the above methods and tries to tackle major problems faced by recommender systems. The approach can be used to develop web based applications in other domains as well. The approach can be further refined by considering additional parameters based on the system's need.

Keywords— *Movie Recommendation, User Similarity, Bayesian Rating, Data mining, Group Cosine, Hybrid Approach, Hit Counts.*

*****

I. Introduction

There has been an exponential increase in the volume of available data, electronic information, and e-services in recent years. This information overload has created a potential problem of filtering and efficiently delivering relevant information to a user. To tackle this problem there is a need for backend systems that can use all the unseen data information and predict the users liking and provide relevant information accordingly. Such systems are called recommender systems.

In electronic commerce applications, prospective buyers may be interested in receiving recommendations to assist with their purchasing decisions. Research on this topic has described two main models for recommender systems – collaborative filtering and content based approach. The amount of data available to be processed is so vast that it led to development of recommendation system. A recommender system customizes its responses to a particular user and provides a degree of personalization. To improve performance and eliminate drawbacks of each model, these methods have sometimes been combined into Hybrid recommenders. Recommender system represents user preferences for the purpose of suggesting items to purchase or use. They have become fundamental applications in electronic commerce and information access like search engines, providing suggestions that effectively prune large information spaces so that the user extracts those items that best meets the preferences and needs of the user.

Let $U = \{u_1, u_2, \cdots, u_x\}$ be the set of all users,
$M = \{m_1, m_2, \cdots, m_y\}$ be the set of all possible movies that can be recommended, and $r_{ui,mj}$ be the rating of user u_i on movie m_j. Let f be an evaluation function that measures the probability of user u_i to like movie m_j.

$$\text{i.e. } f : U \times M \rightarrow R,$$

where R is a totally ordered set. Now for each user $u_i \in U$, the aim of a recommender system is to choose that movie
$m_j \in M$ which maximizes the user's probability of liking the movie.

Currently most recommender systems are designed based on content-based filtering or collaborative filtering. In this paper, we propose a hybrid approach which produces accurate and practical recommendation and can be used in cold-start scenarios. Our proposed scheme is based on a hybrid recommendation technique that considers item's rating, feature, and hit counts as well as user's preferences, profile similarity; and generates more accurate prediction than available state of the art recommender algorithms. We evaluate our algorithm on Movie Lens datasets. The data set contains 100000 entries.

Abbreviations	
DB	Database
CF	Collaborative Filtering
ML	Movie Lens
API	Application Program Interface

II. Problems in Existing Systems

There are two potential problems with the recommender systems. First is the scalability, which determines how quickly a recommender system can generate recommendation, and the second is to provide the most accurate quality of recommendation to the user. Collaborative Filtering recommender systems produces high accuracy recommendation as compared to purely content based recommender systems. However, due to the sparsity, they cannot use rating correlation to find find similar items or users. This results in poor quality predictions and reduced coverage. Most systems fail in cold-start problems as well.

ISSN: 2321-8169
2533 - 2536

It is known that the content-based filtering technique suffers from the weaknesses of content limitation and overspecialization. Another noted problem is as recommender systems are being increasingly adopted by commercial web sites they have started to play significant role in affecting the profitability of seller. There are attempts is to inflate the perceived desirability of their own product (push attack) or lower the ratings of their competitors (nuke attack). Such attacks usually involve setting up dummy profiles and assume different amounts of knowledge of the system.

III. SYSTEM ARCHITECTURE

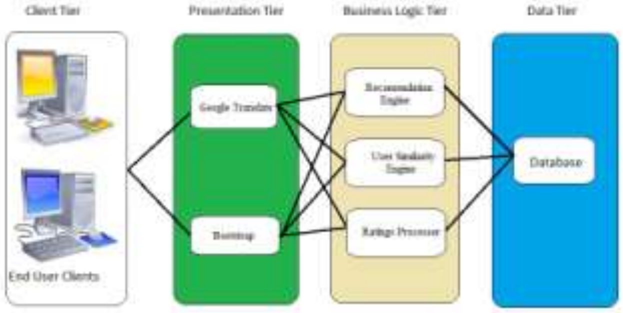

Fig. 1.0 Architecture of Movie Recommendation System.

The Architecture of the system is Multitier/N-Tier which is a client–server architecture. In this architecture presentation, application processing, and data management functions are physically separated. The Data Tier consists of databases which consists of data of past users activity, Movie ratings, information etc. The Business Tier consists of PHP, logic modules which consist of all the business logic for the system which are hosted on a separate application server. The Presentation Tier consists of view oriented API's like Google Plugins and Bootstrap for presentation to users and the Client Tier consists of users with browser clients for system access.

A. *Dataset*

We have used Movie Lens (ML) datasets for implementing our proposed method. The dataset has 100000 ratings on the scale of 1 to 5. 1 indicates movie is rated as bad and 5 indicates movie is excellent. The dataset contains 943 users and 1682 movies.Each user has rated at least 20 movies.

The sparsity of this dataset is 93.7%.

Sparsity = (1− non zero entries/all possible entries)

=1− (100000/943×1682) =0 .937.

B. *Group Cosine Similarity*

The cosine of two vectors can be denoted as the dot product between them given by the formula:

$$a.b = |a||b| \cos\theta$$

Therefore given two vectors of attributes, *a* and *b*,the cosine similarity can be calculated as

$$\text{Similarity}= \cos(\theta) = (a.b)/(|a||b|)$$

The resulting similarity ranges from −1 to 1 because cos function range is -1 to 1.-1 meaning exactly opposite (dissimilar), to 1 meaning exactly the same(similar), and values in between indicating intermediate similarity or dissimilarity.

This is used to identify different users having similar profile & provide personalization.

Utility matrix for users and clusters of items.

	X	Y	Z
A	4	5	1
B	4.67		
C		2	4.5
D	3		3

We cluster the movies released in category X into one cluster, released in category Y into one cluster and released in category Z into one cluster. Greater the Cosine between 2 users, more is the similarity between them. We would then recommend with the items purchased by the user having maximum similarity. Using Grouped Cosine removes unnecessary calculations for finding cosine between every two movies. In that place overall category based cosine is much efficient and easy to implement.

C. *Bayesian Rating Method:*

This method is used when the user provides rating to a movie & that rating is to be incorporated into the existing system.

$$\text{Rating (new)} = C*m + \sum x_i / (C + n)$$

Where

m = current mean rating of that movie

n = total number of ratings

$\sum x_i$ = sum of all the ratings including new rating

C = constant,

The bigger C is, the higher the number of reviews required to deviate substantially from m.

Eg. A movie is rated by 10 users,the ratings are as follows

User 1 – 9
User 2 – 9
User 3 – 8
User 4 – 9
User 5 – 7
User 6 – 9
User 7 – 8
User 8 – 8
User 9 – 9
User 10 – 7
Mean m = 8.3

Suppose the 11th user enters false rating as 3 so the new rating using Bayesian rating formula is calculated as

$$\begin{aligned}\text{Rating (new)} &= (10*8.3 + 86) /(10+11)\\ &= 169/21\\ &= 8.04\end{aligned}$$

This mean is not much deviated from original mean.

Whereas the

$$\begin{aligned}\text{Simple mean} &=((10*8.3)+3)/11\\ &=86/11\\ &=7.81.\end{aligned}$$

Thus Bayesian rating solves the problem of false rating if undertaken by a few number of users with malicious intent.

D. Preference Based Recommendation

Users register their preferences of genres while registering on movie recommendation site. The website keeps their preferences and used it to process the recommendation for users. This method is useful especially when a new user registers and logs in and he hasn't rated or viewed any movie as yet. This is a case of cold start problem. This can be tackled using this approach.

Similar problem is faced by a new movie released which does not get recommended since it hasn't been rated by considerable number of user's. This is called sparsity problem. Until the system automatically starts recommending the movie based on mass popularity the movie may have already gone from the theatres. In order to avoid this problem the hit count of a movie is maintained. Thus even if a movie isn't been rated still it may be recommended based on its hit count.

Consider following table which represents record of newly released movies in user's favourite genres

Id	Hits
1	250
2	470
3	520
4	100
5	43

This type of recommendation will simply consider hit count to suggest new top picks for users.

E. History Based Recommendation

Many a times users specify different preferences but their views history or rating history suggest they like some other genre. In such cases the system should identify his actual interests and use it to process the recommendation. The history tables keep updating based on his rated movies and the genre that they belong to, which suggests that genre which the user has been watching most of the times. Top 3 such genres are considered in descending and the movies of these top 3 genres are recommended to the user. This provides a way in which the user can get a better recommendation without explicitly mentioning his recent choices.

F. User Similarity Based Recommendation

User similarity in the system is calculated by using grouped cosine similarity. For calculating this similarity the past ratings of all the users are considered. The similarity between two users helps in finding out the most matching choices among users and getting the effect of personalization. The movies watched by similar user and the ones not watched by the user are the target movies. The filtering can be done on them and the top rated movies by the similar user and which are not seen by the user are recommended.

Parameters considered while calculating recommendation

- The user profile similarity
- The past ratings of the users
- The data of movies seen by the user.

This method gives the user the feel of personalization since he is mapped to the most resembling other user. The main advantage of it is that it does not strictly calculate similarity based on any one parameter like geographical location, age, occupation but based on choices of movies. This means users in totally different demography may be most similar to each other as well.

IV. Proposed Approach

In our proposed hybrid approach we combine the results of all three of the above approaches to get hybrid recommendation.

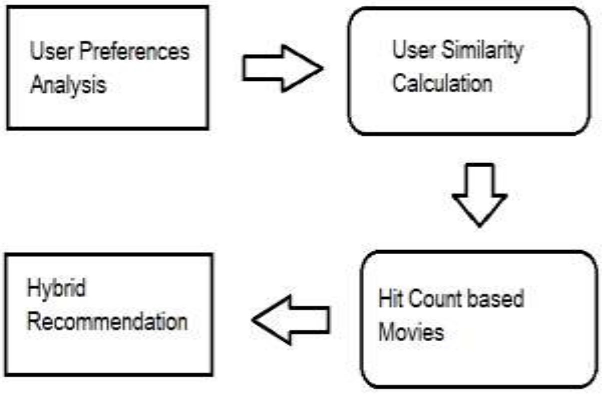

Fig 2.0 Workflow diagram for Movie Recommendation system

Our approach solves majority of problems faced by recommendation engines. Initially the user preferences are taken either from default preferences or based on history based modified preferences. The preferences are used to narrow down the recommendation to top 3 genres of user's preference. This selection of genres in the beginning considerably reduces the processing load of engine since the number of movies are cut down to a fraction of actual data set available.

Next the similarity between user profiles provides a level of personalization to the system. The user gets the set of target movies data set which is obtained by considering the top rated movies by the similar user and the ones not seen by the user. This target data set provides accurate and better suited movies for the user. The hit count of each of the movies from target data sets is considered to eliminate sparsity and to give certain level of collaborative approach to these selected movies. Top 3 movies from each of these top 3 genres are selected and recommended to the user as hybrid recommendations.

The advantage of this approach is that it solves the problem of cold start, sparsity, false ratings and also tries to tackle scalability problem. Here the system can be scalable since at each step of the flow diagram and implementation the number of movies considered is reduced by a major amount. Thus even if the number of movies goes on increasing the system still can handle it since at a time it has to consider only 3 genres of user's preference. This hybrid approach can be further developed by considering more parameters like movie to movie based similarity as well so that it can provide more number of similar movies once the data set increases.

V. Conclusions

The paper proposes the use of data mining to provide recommendations to users based on user similarity, user preferences, hit ratio and user ratings history. The paper has tried to solve the problems such as cold start, scalability, false rating and sparsity that are common in recommender systems these days. The parameters are studied upon and considered keeping in mind the fact that how the user feels, expects from an online recommendation system environment. The results from the recommendation system are optimized with respect to parameter consideration. In future work we will be focusing to go in more micro level of parameter consideration for recommendation which will result in increase in accuracy of the system for e.g. movie to movie based recommendation, actors, information consideration etc. Also we have planned to turn this web application into portal where all information about movies will be available in one single place.

ACKNOWLEDGMENT

This work is a part of B.E project on “Content Boosted Approach towards Recommendation System” under guidance of K.J Somaiya College of Engineering-Computer Department, Vidyavihar-Mumbai.

REFERENCES

[1] Mustansar Ali Ghazanfar and Adam Prugel-Bennett, “A Scalable,Accurate Hybrid Recommender System”, University of Southampton, 2010.

[2] Yolanda Blanco-Fernandez, Jose J. Pazos-Arias, Alberto Gil-Solla, Manuel Ramos-Cabrer, and Martin Lopez-Nores, “Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender System”, University of Vigo, 2008.

[3] KonstantinosChristidis, GregorisMentzas, “A topic-based recommender system for electronic marketplace platforms”, National University of Athens, 2013

[4] Adomavicius, G., & Tushilin, A. (2001), “Extending Recommender Systems: A Multidimensional Approach.”, IJCAI-01 Workshop on Intelligent Techniques for Web Personalization(ITWP’2001).

[5] Breese, J., Heckerman, D., &Kadie, C. (1998), “Empirical Analysis of Predictive Algorithms for Collaborative Filtering.”, In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), pp 43-52.

[6] Geyer-Schulz, A., &Hahsler, M. (2002). “Evaluation of Recommender Algorithms for an Internet Information Broker based on Simple Association Rules and on the Repeat-Buying Theory.” *Fourth WEBKDD Workshop: Web Mining for Usage Patterns & User Profiles*, pp. 100-114.