Automated Recommendation System

Niraj Dedhia
njd4185@rit.edu

Siddharth Bidwalkar
ssb9012@rit.edu

Sandeep Chaudhary
ssc4763@rit.edu

1. INTRODUCTION:

Abstract:

On the internet, there are large number of options from various domains available for any type of selection. So, there must exist a filter to present best of all options to make best selection out of it. Recommendation system solves this problem by filtering out the items from large set of items based on the user's previous selections. The aim of recommendation engine is to navigate through large data collections and predict user preferences based on user's navigation history. Different prediction techniques are available which are implemented with the help of machine learning. These techniques are Collaborative Filtering, Content-based filtering and hybrid technique.

Problem Definition:

Obtaining recommendations from trusted sources about a product is a tedious job. You may or may not like the product recommended. Additionally, searching that product online in an inventory makes it a bit difficult if you are not able to remember the correct name of the product. To overcome this issue, Recommender Systems have gained lot of popularity. It basically deals with prediction of options to user based on user's previous responses on available options. It is a way of suggesting items to users based on their past records which matches with people with similar taste. The aim is to generate useful recommendations to users which might interest them. There are various applications of Automated Recommendation Systems such as online recommendation system used by Facebook to suggest people you may know or Product or Movie recommendation used by many E-commerce sites such as Amazon, E-Bay and BestBuy.

Motivation:

Motivated by emergence of Web applications and Recommendation systems for different domains all around the world, we are creating an Automated Recommendation System for restaurants. We have built a recommendation system which will provide restaurant recommendation to users by applying learning algorithm. We will be implementing collaborative filtering algorithm on the Restaurant Dataset and context based filtering algorithm on the user dataset. We will also implement Hybrid Recommendation system which is a combination of both the algorithms which will provide more accurate recommendation by combining users with similar tastes and restaurants with similar features.

2. Proposed methods:

We have explored the following techniques to build the Recommendation Engine:

(A) Collaborative Filtering: This filter out the items and creates the group of items having similar features.

(B) Content based Filtering: The result is calculated based on user's history.

(C) Hybrid Technique: Combining the result of above two filtering methods.

(A) Collaborative Filtering:

As a name suggests collaborative filtering approach is a method of filtering information on the collaboration of users, or the similarity between items. Collaborative filtering is one of the most popular and widely used recommendation system. This algorithm has the property of learning the features which tells an algorithm, what features to use. Here we are using the dataset of restaurants and using this method we have found the list of similar restaurants (having same features). Basically, it has two steps as stated below.

1) User-Based Collaborative Filtering:
   (a) The approach first searches the similar users (similar users are the users who share the common ratings in other terms user having the similar tastes for that item).
   (b) Use these ratings from similar users found in above steps for calculating the prediction for active current.
2) Item-Based Collaborative Filtering:
   (a) This approach first builds the item Vs item matrix and value for which are the total number of features matches between the compared items.
   (b) It examines the user taste based on which it creates the list of similar items of user's like.

Algorithm for finding similar items:

1) Read the items dataset. (It contains the list of all the items and their features).
2) Created Item Vs Item matrix of size [n][n] where n is the total number of items.
3) Looped through each cell of matrix. (A cell stores the values of total number of matched features between a and b where a and b are compared items).
4) Generated the list of all the similar items for each individual item. (If there are 10 features then we can those are similar items for which cell value is >=8 for item a).
5) Now we got all the similar items for each single item.

Algorithm for finding the list of items based on user's taste:

1) From the content based filtering we have already calculated the parameter for every user and we can use them as the taste for each user.
2) Now calculated the similar users by applying Euclidian distance. (Apply the Euclidian distance formula on the calculated parameter values).
3) Generated the list of similar users for each user. (Similar users are the users having Euclidian distance close to 0 i.e users share similar ratings).

(B) Content-Based Filtering:

Here we know the set of users and set of items and the sets of ratings given by user for item. Content based filtering looks at the user's ratings for rated items and predicts what ratings user would give to a non-rated item. If prediction is high then user will like the item. How to predict the rating value? So, every item has certain features and by exploring the features it generates the feature vector for each item. Taste of user can be calculated by gradient decent or linear regression depends on the datasets.

Terms:

R(i , j) = 1 if user j has rated for item i (else its value will be 0).
Y(i , j) = Ratings given by user j for item i.
θ (j) = Parameter vector for user j (user j taste).
X(i) = Feature vector for item i (item feature)
m(j) = Number of items rated by user j.
Prediction for i^{th} item by j^{th} user = $[\ \theta\ (\ j\ )\ ]^{T}\ [(\ i\ )\ ]$ (T is transpose)

Algorithm:

1) Read the rating dataset and generate the Y(i , j) matrix. Y(i , j) = V, where V is rating given by user i for item j else -1
2) Generated the matrix R(i , j) (1 user i has given rating for item j and 0 for rest of them)

3) X(i) is calculated by the taking an average of total received rating for the item and this will be the vector of feature's rating (if there are 10 features then length of X will be 11 -> 10 for features and 1 for bias)
4) Calculated the m(j) by summing over total number of item rated by user j.
5) θ (j) is calculated with the help pf gradient descent as follow. Minimized the value of θ

$$\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(n_u)}:$$

$$\min_{\theta^{(1)},\dots,\theta^{(n_u)}} \frac{1}{2}\sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2$$

6) Now for prediction we applied following formula.

Prediction = $[\ \theta\ (\ j\ )\ ]^{T}\ [(\ i\ )\ ]$

(C) Hybrid Technique:

Hybrid recommendation system combines the results of both collaborative and context based filtering which results in higher accuracy in some cases.

Algorithm:

1) Ran the content based to get the user's parameter vector (i.e. to understand the user's taste)
2) Ran the collaborative filtering to classify the datasets of items in to group of similar items and to generate the list of similar users.
3) Now reading user ID and based on the given user first filtered out similar users list.
4) Calculated and merged all the items rated by those similar users and not rated by active user.
5) Calculated prediction with the help of content based method for all the items with respect to active user's taste.
6) Suggesting all the items from the above list for which prediction value is higher and best.

3. Experimental result and analysis:

We have tested our engine on the available dataset of restaurants and user rating for restaurants. Here we are using common user whose ID is 'U1082' and we will be using this user as an example to compare all the results.

Content-Based Filtering:

Initially we implemented content based filtering system. We could see the results but this was based on active user tastes. Following is the list of restaurants we got on testing content based filtering system.

User: U1082 (User with history)
Length of the List: 19 (This result was based on active user testing

Here since this system only consider the active user tastes. There is possibility that user might like restaurants having low ratings but good food ratings.

Collaborative Filtering:
To resolve the above problem, we implemented Collaborative Filtering and this worked well as this system first finds all the similar users having similar taste and then whatever the restaurants the other has visited that is also being considered.

User: U1082 (User with history)
Length of the List: 30 (This was the results based on the similar user's and similar places as visited liked places)

Here we finally got some restaurants suggested. However, it was still not so accurate as it was just comparing similar user's profile and returned all the Restaurants.

Hybrid Technique:
To make the results more accurate, we merged all the restaurants from above two methods and then we calculated the prediction using parameter vector of active user and feature vector of all the places. Displayed all the restaurants for which prediction is >=75% of highest ratings.

User: U1082 (User with history)
Length of the List of Recommendations: 23

4. Future Work:

Each of the above discussed filtering technique is best suitable for a specific data. When the user is new, the user will have to dedicate some efforts on to the system by searching a product or just browsing which can help in generating a user profile. This technique is also known as cold start and can be overcome by actively learning. Additionally, all these algorithms can break free from the Filter keyword at the start. This means basically when you start as a new user, you don't need to filter on what you are searching; the system can show the results based on the latest trending products or else highly rated product by different users.

5. Conclusions:

We have implemented all the three Filtering algorithms on the restaurant dataset. In Context based filtering the system only consideres active user's taste. However, it could also like different restaurant which has a lower rating. In Collaborative based filtering the system recommends all the restaurants based on similar user profile. We observed that highly accurate and precise results were found out when we ran Hybrid Recommendation System compared to the above two techniques.

6. References:

1. https://en.wikipedia.org/wiki/Recommender_system
2. http://recommender-systems.org/content-based-filtering/
3. http://recommender-systems.org/collaborative-filtering/
4. https://www.youtube.com/watch?v=saXRzxgFN0o
5. https://www.youtube.com/watch?v=i6u5ykEHSP8
6. https://www.youtube.com/watch?v=KkMAgWlYCAQ
7. http://archive.ics.uci.edu/ml/datasets/Restaurant+%26+consumer+data#