FeedMeRight:Comparison of Recipe Recommendation Systems

Pallavi Amod Chavan

April 18, 2020

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The internet is huge network of machines that connects large number of computers together worldwide allowing them to communicate with any other computer. The World Wide Web is an information sharing model that is built on top of the internet in which information can be accessed or manipulated easily hence experiencing dramatic growth in increased usage of internet which results in BigData.
BigData is an exponentially increasing data with high volume, high velocity with variety. This huge amount of data has intrinsic value but it's of no use until it's discovered [5]. One of the ways of finding value in BigData is analyzing it with its interrelated features such as new products, corresponding reviews, ratings and user preferences. Forming information from raw data is an entire discovery process that requires insightful analysis that would recognize patterns to predict user behaviors to recommend products.
Handling BigData by manual process is very inefficient. More efficient way of processing such huge amount of data is automating the process of classifying, filtering data of user's opinions, features, and preferences in order to understand and predict new set of related products.
Recommender system can be defined as a tool designed to interact with large and complex information spaces to provide information or items that are relevant to the user [**recommeder˙overview**].
Nowadays recommender systems are widely used in variety of applications. Initially it applied for commercial use to analyze data. Amazon is a good

example of such one of E-commerce websites. However, it is now present in several different domains including entertainment, news, books, social tags and some more sophisticated products where personalization is critical such as recipes domain. This paper would further discuss the different approaches for recipe domain to recommend healthy recipes based on user's profile.

Chapter 2

Background

2.1 Recommender System

A recommender system is an Information Filtering (IF) system that provides or suggests relevant items to user based on the user profile and preferences. Basic idea of general recommender model is given in : Figure 2.1

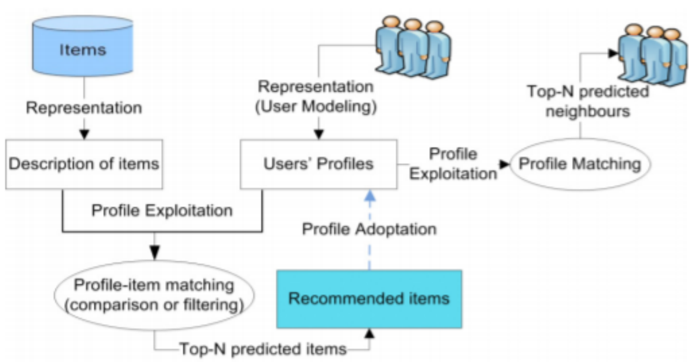

Figure 2.1: General Recommender Model [2]

Traditionally there are two basic models of recommender systems.

- Content based filtering
- Collaborative filtering

2.1.1 Content Based Filtering

In Content based method algorithm, user preference is considered based on item description. The rating and buying behavior of users are combined with content information available in the items. The main aim of content based filtering is to create profile for each item and each user to find similar items the user is looking for [8].

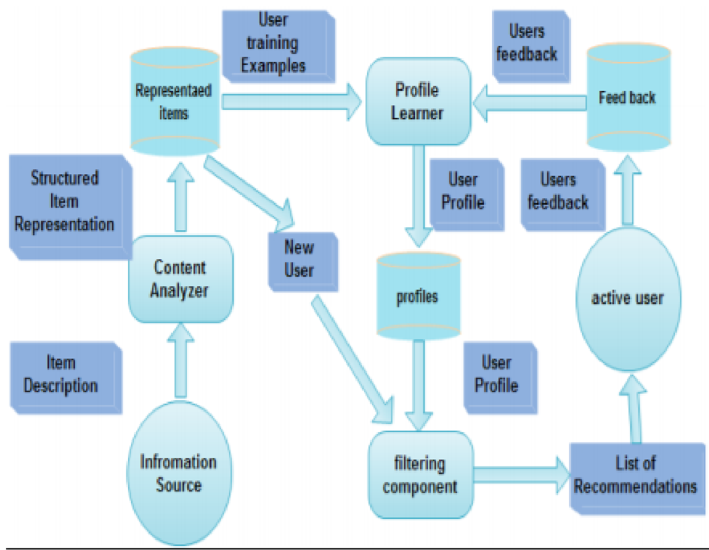

Figure 2.2: Content Based filtering Architecture [4]

In this algorithm each user's information can be stored in vector form which contains past behavior of the user. This vector is known as profile vector or user profile. All the information about item is stored in item vector / item profile which contains all the details about item specific attributes. Based on similarity score between user profile and item profile most relevant items are recommended to user.
Advantages of content-based recommenders are –
Content-based recommender systems are heavily reliable on the contents of the items that have been rated by the user. So, while making recommendations, this approach would consider user's taste and accordingly recommend an item that matches user's preferences. Generally, most popular items dominate less popular items. But this approach will not miss less popular item

if it matches the user's unique taste [8].
Disadvantages of content-based recommenders
User profiles are generated based on rated items. But for any new user who has not rated any items yet, user profile will be empty. In that case, recommending perfect item that matches to user's taste is difficult as system does not have user taste information. This problem is known as cold start. Also, to understand each items feature, system needs to examine content of every item. Therefore if number of items rises quickly, performance of the system decreases [8].

2.1.2 Collaborative Filtering

Collaborative filtering uses other users' behavior in the system to predict and recommend items. It depends on user's contribution such as ratings, reviews which considered as filter for user preference information. The fundamental idea of collaborative filtering is it selects other users' opinions and aggregate in such way that it provides prediction for active user based on his preferences [3].
The main source of input for this algorithm is in the form of matrix of collected user-item ratings. Based on this input it provides recommendations as an output. The first step of output is to predict ratings for items that user may like. Second step is to recommend a list of top rated items as top-N items.

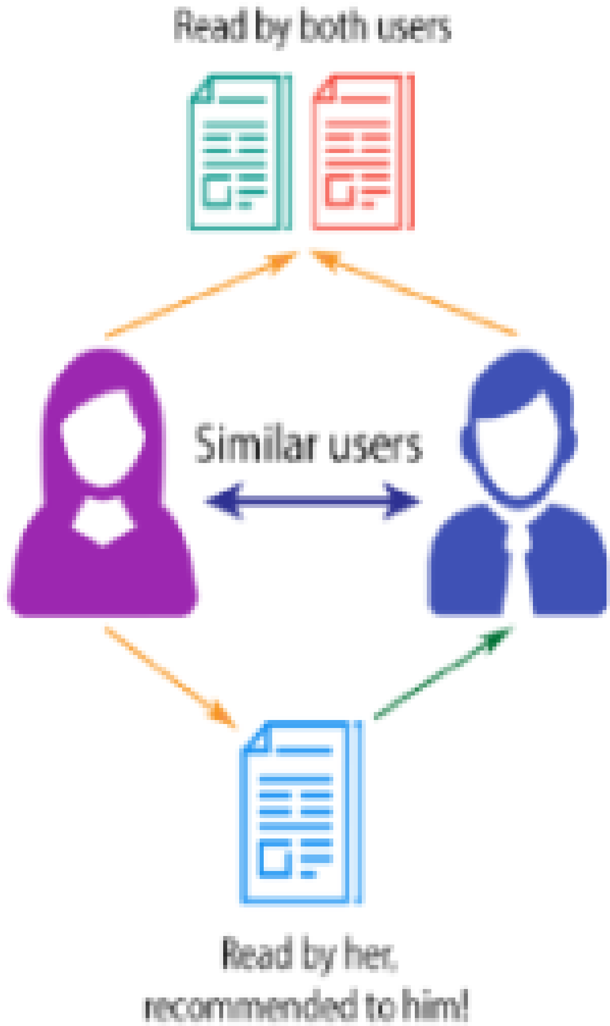

Figure 2.3: Collaborative Filtering [**cf˜figures**]

Collaborative Filtering is broadly divided into 2 categories.

Memory-Based (user based)

A memory-based collaborative filtering approach predicts item ratings based on ratings given by different users for an item. There are primary two forms of memory-based collaborative filtering.

User-User CF: Similarity between users is calculated based on how similarly they rate several items. It finds other users whose ratings are similar to active user and use their ratings on other items to predict what active user may like. Thus it recommends items to the users that are most preferred by similar users.
Consider example of user and ratings given by users to different recipes. This algorithm will find similarity between each user based on the ratings they have given to the recipes in the past. The prediction of a recipe for a user u is calculated by computing weighted sum of the user ratings given by other users to recipe i. The prediction for recipe I is given as below:

$$P_{u,v} = \frac{\sum_v (r_{v,i} * S_{u,v})}{\sum_v S_{u,v}} \tag{2.1}$$

Where,
$P_{u,i}$ = prediction of recipe i
$R_{v,i}$ = rating given by user v to recipe i
$S_{u,v}$ =similarity between users.

To predict the ratings for other user we need to calculate similarity score. The similarity between users can be calculated with the help of several methods described in the section 2.4.3 Prior to that we need to find items rated by both users and its rating. Based on that rating, if we opt to calculate similarities with the Pearson correlation then we will get correlation score between users. Higher correlation implied higher similarity. Recommendations are made based on these prediction values.
This algorithm is quite expensive in terms of time as it involves calculating similarity score between each user and from that score calculating predictions. This could be very useful when we have less number of users and more items.

Model-Based (item based)

2.2 Similarity Methods

There are several methods available to calculate similarity score.

2.2.1 Cosine Similarity

In this method, cosine of the angle between profile vector and item vector is calculated. Consider A and B are profile vector and item vector respectively, the similarity between them can be calculated as per below formula:

$$sim(A,B) = cos(\theta) = \frac{A.B}{\parallel A \parallel \parallel B \parallel} \tag{2.2}$$

The value of cosine angle ranges between -1 to 1. Lesser the angle, less distance hence more similarity as cos(0) = 1. Then items are arranged in descending order and recommended to user

2.2.2 Euclidean Distance

If we plot similar items in n-dimensional space, then they will fall under close proximity. In that case, we can calculate distance between items with Euclidean distance formula which is given by:

$$EuclideanDistance = \sqrt{(x_1 - y_1)^2 + ... + (x_n - y_n)^2} \tag{2.3}$$

2.2.3 Pearson's Correlation

Person's correlation helps in finding correlation between similar items. Correlation on higher side implies more similarity. It can be calculated as below:

$$sim(u,v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{(\sum(r_{ui} - \bar{r}_u))^2}\sqrt{(\sum(r_{vi} - \bar{r}_v)^2}} \tag{2.4}$$

2.3 Conclusion and future work

Your work goes here [7] something [3] [5] [2] [1] [4] [8] [3] [6]

Bibliography

[1] Robin Burke, A. Felfernig, and Mehmet Göker. "Recommender Systems: An Overview". In: *Ai Magazine* 32 (Sept. 2011), pp. 13–18. DOI: 10.1609/aimag.v32i3.2361.

[2] Ullah I. Khusro S. Ali Z. "(2016) Recommender Systems: Issues, Challenges, and Research Opportunities". In: (2016).

[3] John T. Riedland Joseph A. Konstan Michael D. Ekstrand. *Collaborative Filtering Recommender Systems*. 2011.

[4] Marwa Mohamed, Mohamed Khafagy, and Mohamed Ibrahim. "Recommender Systems Challenges and Solutions Survey". In: Feb. 2019. DOI: 10.1109/ITCE.2019.8646645.

[5] T. L. Nguyen. "A Framework for Five Big V's of Big Data and Organizational Culture in Firms". In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 5411–5413.

[6] Kali Pradeep and M Bhaskar. "Comparative analysis of recommender systems and its enhancements". In: *International Journal of Engineering and Technology* 7 (Jan. 2018), pp. 304–310.

[7] Paul Resnick and Guest Editors Hal R. Varian. "RecommenderSystems". In: (1997).

[8] Harry Zisopoulos et al. "Content-Based Recommendation Systems". In: (Nov. 2008).