Comparing Content Based and Collaborative Filtering in Recommender Systems

Parul Aggarwal, Vishal Tomar, Aditya Kathuria

Abstract— **In daily life we need many things to be searched over the internet, for search purpose there are many search engines available. Whenever we search something we try to get the most relevant results, and this can be achieved using Recommender systems.In a world where the number of choices can be overwhelming, recommender systems help users find and evaluate items of interest. They connect users with items to "consume" (purchase, view, listen to, etc.) by associating the content of recommended items or the opinions of other individuals with the consuming user's actions or opinions. The paper presents an overview of the field of recommender systems and describes the difference between two of the most used approaches in recommender systems, i.e. Collaborative Filtering and Content based Filtering Techniques.**

Index Terms— **Recommender Systems, Content-Based Filtering, Collaborative Filtering.**

I. Introduction

The goal of a Recommender System is to generate meaningful recommendations to a collection of users for items or products that might interest them. Suggestions for books on Amazon, or movies on Netflix, are real world examples of the operation of industry-strength recommender systems. The design of such recommendation engines depends on the domain and the particular characteristics of the data available. Recommender systems were developed to help close the gap between information collection and analysis by filtering all of the available information to present what is most valuable to the user.

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet. Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information according to user's preferences, interest, or observed behavior about item. Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user's profile.

Recommender systems are beneficial to both service providers and users. They reduce transaction costs of finding and selecting items in an online shopping environment. Recommendation systems have also proved to improve decision making process and quality. In e-commerce setting, recommender systems enhance revenues, for the fact that they are effective means of selling more products. In scientific libraries, recommender systems support users by allowing them to move beyond catalog searches. Therefore, the need to use efficient and accurate recommendation techniques within a system that will provide relevant and dependable recommendations for users cannot be over-emphasized.

Most recommender systems take either of two basic approaches: collaborative filtering or content-based filtering. Collaborative filtering arrives at a recommendation that's based on a model of prior user behavior. The model can be constructed solely from a single user's behavior or also from the behavior of other users who have similar traits. When it takes other users' behavior into account, collaborative filtering uses group knowledge to form a recommendation based on like users. In essence, recommendations are based on an automatic collaboration of multiple users and filtered on those who exhibit similar preferences or behaviors. Content-based filtering constructs a recommendation on the basis of a user's behavior. For example, this approach might use historical browsing information, such as which blogs the user reads and the characteristics of those blogs. If a user likely to leave comments on blogs about software engineering, content-based filteringcan use this history to identify and recommend similar content (articles on Linux or other blogs about software engineering). This content can be manually defined or automatically extracted based on other similarity methods.

II. Comparing the 2 approaches

A. Content-based Filtering:

Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user. Items that are mostly related to the positively rated items are recommended to the user. CBF uses different types of models to find similarity between documents in order to generate meaningful recommendations. Content-based filtering technique does not need the profile of other users since they do not influence recommendation. Also, if the user profile changes, CBF technique still has the potential to adjust its recommendations within a very short period of time.

Parul Aggarwal, Department of Computer Science, Atma Ram Sanatan Dharma College, University of Delhi, New Delhi
Vishal Tomar, Department of Computer Science, Atma Ram Sanatan Dharma College, University of Delhi, New Delhi
Aditya Kathuria, Department of Computer Science, Atma Ram Sanatan Dharma College, University of Delhi, New Delhi

For example, if a user likes a web page with the words “mobile”, “pen drive” and “RAM”, the CBF will recommend pages related to the electronics world. Item description and a profile of the user’s orientation play an important role in Content-based filtering. Content-based filtering algorithms try to recommend items based on similarity count. The best-matching items are recommended by comparing various candidate items with items previously rated by the user.

A content based recommender works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

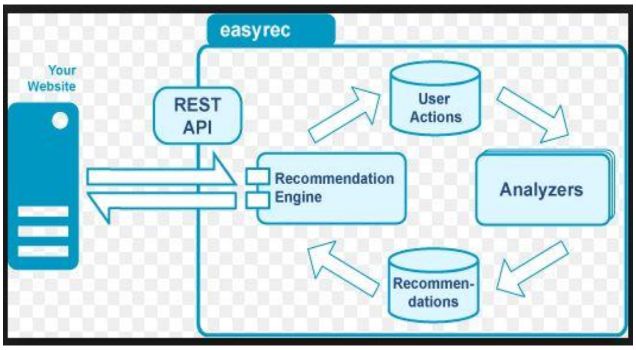


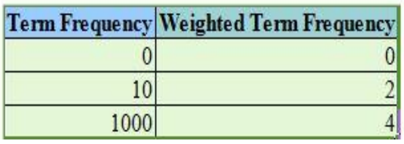
Fig 1:

The concepts of Term Frequency (*TF*) and Inverse Document Frequency (*IDF*) are used in information retrieval systems and also content based filtering mechanisms (such as a content based recommender). They are used to determine the relative importance of a document / article / news item / movie etc.

TF is the frequency of a word in a document. IDF is the inverse of the document frequency among the whole corpus of documents. TF-IDF weighting negates the effect of high frequency words in determining the importance of an item (document). But while calculating TF-IDF, log is used to dampen the effect of high frequency words. For example: TF = 3 vs TF = 4 is vastly different from TF = 10 vs TF = 1000. In other words the relevance of a word in a document cannot be measured as a simple raw count and hence the equation below:

Equation:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$



Term Frequency	Weighted Term Frequency
0	0
10	2
1000	4

It can be seen that the effect of high frequency words is dampened and these values are more comparable to each other as opposed to the original raw term frequency. After calculating TF-IDF scores we determine which items are closer to each other by using the Vector Space Model which computes the proximity based on the angle between the vectors. In this model, each item is stored as a vector of its attributes (which are also vectors) in an **n-dimensional space** and the angles between the vectors are calculated to **determine the similarity between the vectors**. Next, the user profile vectors are also created based on his actions on previous attributes of items and the similarity between an item and a user is also determined in a similar way.

IDF is calculated by taking the logarithmic inverse of the document frequency among the whole corpus of documents. So, if there are a total of 1 million documents returned by our search query and amongst those documents, ‘smart’ appears in 0.5 million documents. Thus, it’s IDF score will be: **Log10 (10^6/500000) = 0.30.** The length of the vectors are calculated as the **square root of sum of the squared values** of each attribute in the vector. After finding the TF-IDF weights and the length of the vectors, the vectors are normalized by dividing each vector by the document vector length. After finding the normalized vector the cosine values are calculated to find out the similarities between articles. Cosine values are calculated as –

Given 2 articles pj and pk represented as vectors of weight, their similarity is measured by –

$$\mathrm{Sim}(p_j, p_k) = \frac{\overline{p_j} \cdot \overline{p_k}}{|\overline{p_j}| \cdot |\overline{p_k}|} = \frac{\sum_{i=1}^{n} w_{i,j} \cdot w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\,\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

Where:
W i, j = Weight of term iin article j
W i, k = Weight of term iin article k

This concept can be applied to ‘n’ articles and we can find out which article a user will like the most. Therefore, along with new articles in a week, a separate recommendation can be made to a particular user based on the articles which he hasn’t read already.

Content based recommenders have their own limitations. They are not good at capturing inter-dependencies or complex behaviors. For example: I might like articles on Machine Learning, only when they include practical application along with the theory, and not just theory. This type of information cannot be captured by these recommenders.

B. *Collaborative Filtering:*

Collaborative recommender systems (or collaborative filtering systems) try to predict the utility of items for a particular user based on the items previously rated by other users. Collaborative filtering filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future.

Collaborative Filtering uses either a User-Based approach or an item-based approach. In the user-based approach, the users perform the main role. If certain majority of the customers has the same taste then they join into one group.

Recommendations are given to user based on evaluation of items by other users form the same group, with whom he/she shares common preferences. Item-based collaborative filtering is a Model-based algorithm for making recommendations. In the algorithm, the similarities between different items in the dataset are calculated by using one of a number of similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset.

Instead of just relying on the most similar person, a prediction is normally based on the weighted average of the recommendations of several people. The weight given to a person's ratings is determined by the correlation between that person and the person for whom to make a prediction. As a measure of correlation the Pearson correlation coefficient can be used.

III. Pearson (Correlation)-Based Similarity

This similarity measure is based on how much the rating by common users for a pair of items deviate from average ratings for those items:

$$sim(i,j) = \frac{\sum_{u\in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j} - \bar{R}_j)^2}}$$

The ratings of person X and Y of the item k are written as X_k and Y_k, while $\overline{X}$ and $\overline{Y}$ are the mean values of their ratings. The correlation between X and Y is then given by:

$$r(X,Y) = \frac{\sum_k (X_k - \overline{X})(Y_k - \overline{Y})}{\sqrt{\sum_k (X_k - \overline{X})^2 \sum_k (Y_k - \overline{Y})^2}}$$

In this formula k is an element of all the items that both X and Y have rated. A prediction for the rating of person X of the item i based on the ratings of people who have rated item i is computed as follows:

$$p(X_i) = \frac{\sum_Y Y_i \cdot r(X,Y)}{n}$$

Where Y consists of all the n people who have rated item i. Note that a negative correlation can also be used as a weight. For example, because Amy and Jef have a negative correlation and Amy did not like "Farg" could be used as an indication that Jef will enjoy "Fargo".

In GroupLens a prediction is made by computing the weighted average of deviations from the neighbor's mean:

$$p(X_i) = \overline{X} + \frac{\sum_Y (X_i - \overline{X}) \cdot r(X,Y)}{\sum_Y r(X,Y)}$$

Note that if no one has rated item i the prediction is equal to the average of all the ratings person X has made. Ringo recommends music albums and artists a person might be interested in. Someone considers several collaborative algorithms and reports that a constrained Pearson r algorithm performs best for Ringo's information domain. The constrained Pearson measure is similar to the normal Pearson measure but uses the mean value of possible rating values (in this case the average is 4) instead of the mean values of the ratings of person X and Y.

Collaborative Filtering algorithm has the limitations of Cold-start problem where a recommender does not have the adequate information about a user or an item to make relevant predictions. Data Sparsityis the problem that occurs as a result of lack of enough information, that is, when only a few of the total number of items available in a database are rated by users. This leads to a sparse user-item matrix, inability to locate successful neighbors resulting in the generation of weak recommendations.

IV. Conclusion

Collaborative algorithm uses "User Behavior" for recommending items. They exploit behavior of other users and items in terms of transaction history, ratings, selection and purchase information. Other user's behavior and preferences over the items are used to recommend items to the new users.

In content-based filtering we have to know the content of both user and item. Usually we construct user-profile and item-profile using the content of shared attribute space. For example, for a movie, you represent it with the movie stars in it and the genres. For user profile, you can do the same thing based on the users likes some movie stars/genres etc. To calculate how good a movie is to a user, we use cosine similarity. Here, you have product attributes like image (Size, dimension, color etc.) and text description about the product then it is Content Based Recommendation.

Acknowledgment

We would like to thank our honorable principal Dr. Gyantosh K Jha, ARSD College, Head of Department of Computer Science, Dr. V.S. Dixit and our mentor Ms. Shalini Gupta for giving us the support, facilities and providing us with a propitious environment for working in college.

References

[1] May, R.M. 1997. "The Scientific Wealth of Nations".

[2] Torres, R. McNee, S.M. Abel, M. Konstan, J.A. and Riedl, J. 2004. "Enhancing Digital Libraries with TechLens".

[3] Pennock, D.M. Horvitz, E. Lawrence, S. and Giles, L.C. 2000. "Collaborative Filtering by Personality Diagnosis:

[4] A Hybrid Memory- and Model-Based Approach", in Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (San Francisco).

[5] Middleton, S.E. Shadbolt, N.R. and De Roure, D.C. 2004.

[6] "Ontological User Profiling in Recommender Systems".

[7] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms.

[8] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization.

[9] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems.

[10] F. Ricci , L. Rokach, B. Shapira; Introduction to Recommender Systems Handbook,, Springer, 2011.

[11] https://www.hindawi.com/journals/aai/2009/421425/

[12] www.fxpal.com/publications/FXPAL-PR-06-383.pdf

[13] Research Paper Recommender Systems: A Subspace Clustering Approach http://www.public.asu.edu/~ huanliu/papers/waim05.pdf