# Book Recommendation System using Item based Collaborative Filtering

**Kaivan Shah**[1]

[1]*Bachelor Student, Dept. of Computer Engineering*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract -** *Nowadays, the recommendation system plays a very important role in any person's day-to-day life. It makes life easy for everyone. Basically, recommendation systems are those systems which suggest users, items based on their past choices or also suggest by looking into items of another user with similar taste by generating patterns and finding similarities. Considering, E-Commerce it is one of the best tools of personal service on any website. There are several techniques to solve the recommendation system problem and that are Collaborative Filtering, Content-based Filtering and Hybrid ones. This paper provides a brief of the techniques described and the working of Item-based Collaborative Filtering approach which can be enhanced in further research.*

***Key Words***: Item-based collaborative filtering, Recommendation System, Content-based filtering, Memory-based filtering approach, Model-based filtering approach.

## 1. Introduction

### 1.1 Overview

Recommendation System helps e-commerce companies to boost their business by gaining profit when they urge the user to get the thing they are most interested in by providing a recommendation on various items on a website. While improving the business for E-commerce it is beneficial in terms of user perspective also, because they are suggested the things they like instantly. Considering the book recommendation system it allows user to maintain a personal read-list. In recommendation system predictions are made on rating, it can also be done using natural language processing by binary outcomes i.e. when a user comments on a book, using natural language processing we can identify the nature of the statement i.e. whether the statement/review is positive or negative[12,16].

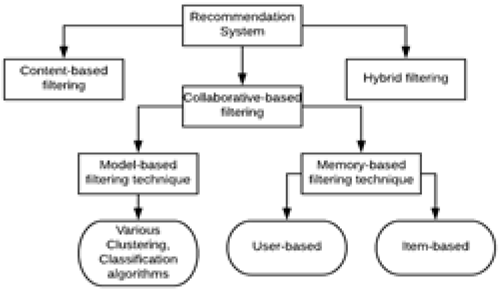

Recommendation System
Content-based filtering
Collaborative-based filtering
Hybrid filtering
Model-based filtering technique
Memory-based filtering technique
Various Clustering, Classification algorithms
User-based
Item-based

**Fig 1: Types of Recommendation system**

### 1.2 Classification of Recommendation Systems

Some of the recommendation systems commonly used are Collaborative-based Filtering, Content-based Filtering techniques. A recommendation system combining both the techniques can also be designed and it is called to be a hybrid recommendation system. Some of the types of recommendation systems mentioned above are explained as follows[2,16].

### 1.2.1 Collaborative-based filtering approach

Collaborative filtering is based on giving present predictions based on other user's past experience. There are further 2 categories in collaborative-based filtering approach which are termed as follows[16]

### 1.2.1.1 Memory-based filtering technique

In this method, entire user-prediction database is used. Some statistical methods are used to find the nearest neighbors. This method is divided into 2 forms and is also called a neighborhood based technique[16].

### 1.2.1.1.1 User-based

This approach is based on users. Consider user A who like books *a1, a2.* There is another user B who likes book *a1.* So here books read by A i.e *a2,* can be recommended to user B as they (users) have kind of similar interests[16,19]

### 1.2.1.1.2 Item-based

This approach calculates similarity among the items. In this approach, the main focus is on what items the target user enjoys the most out of all the available items[16,19]

### 1.2.1.2. Model-based filtering technique

This technique provides a recommendation by first building models of user ratings. This model can be built using many machine learning models such as Clustering, Classification[16,19]

### 1.2.2 Content-based filtering approach

This type of recommendation system works with the data that is being provided by the user either by rating given to a product or by determining the nature of the sentence by using natural language processing. By maintaining a user profile when the user provides more and more input to the system it, gets more accurate in giving recommendations. Here inputs are considered as ratings or comments on a given product[15]

### 1.2.3 Hybrid filtering technique

When implemented individually both the systems i.e. Content-based filtering and Collaborative-based filtering technique have some pros and cons. So to overcome some of the cons of the system a new system was established by combining both the systems and the new system is called a hybrid filtering technique. Here the prediction is made by weighted-average of content-based filtering technique and collaborative based filtering technique. The rank of the item will be the value of the weight and by this way, most precise prediction can be given on the basis of highest weights [21]

## 1.3 Problems related to Recommendation System

### 1.3.1 Sparsity

It occurs many times that most of the users do not rate an item/s or they forget to rate it/them, due to this rating matrix becomes sparse. Hence, it occurs that due to the lack of information and sparsity sometimes we are not able to produce desirable accuracy for the output[12].

### 1.3.2 Scalability

This issue is related to large dataset maintained by the company based on user's rating, comments. Sometimes it occurs that an algorithm functions perfectly and gives a good result on small dataset but when exposed to an enormous amount of data it's quality may deteriorate[12]

### 1.3.3 Cold-Start problem

This problem is related to the user's who are new to the system or haven't interacted with the system in the past. When a new user registers to a recommendation system, he/she has no previous data on which the recommendations are to be made. So, a new user may experience futile results for some time in the beginning[12]

### 1.3.4 Security

When providing recommendations the companies store a large amount of user data. Due to this if intended these data can be used to exploit the user. Although companies claim for the security of the data, users always have a fear of data being misused[12,13]

### 1.3.5 Veracity of profiles

Sometimes in order to ameliorate or deteriorate ratings of a particular product, some users create fake profiles to rate and comment on certain items. These kinds of ratings indeed affect the accuracy of the system, providing spurious results[12]

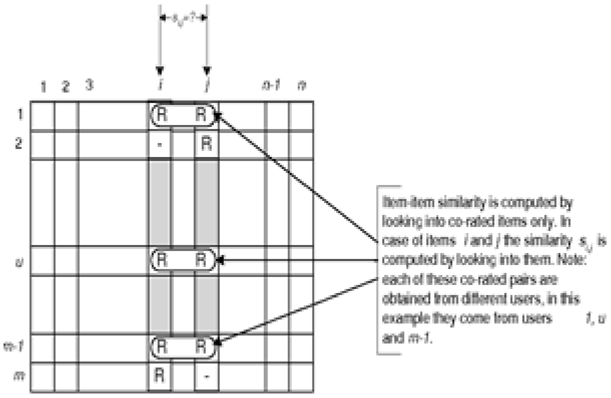
## 2. Item-based Collaborative Filtering

### 2.1 Overview

There were many problems in user based collaborative filtering approach such as.

**1.** The computation of similarity between each and every pair of user very costly.
**2.** Behavior of user changes very often. So for better efficiency of the model we need to re-evaluate the whole model.
**3.** The algorithm compromises it's efficiency when there are many items but fewer ratings.
The item-based filtering approach solves the above problems as in here the concept of rating distribution per items is used instead of rating distribution per user. Here each item tends to have more ratings than each user, so average rating does not change very often. Thus, this increases the stability of rating distribution and the model does not need to re-evaluate very often. The item-based filtering approach looks into the data that the target user has rated and then computes similarity with target item $i$ to select the most favorable $k$ items $\{i1, i2, \ldots, ik\}$. Meanwhile, their corresponding similarities are also being calculated i.e. $\{si1, si2, \ldots, sik\}$. After finding the similar items we need to find the weighted sum average of the target user's rating on these similar items for prediction[11]

### 2.2 Item-similarity evaluation



Item-item similarity is computed by looking into co-rated items only. In case of items *i* and *j* the similarity $s_{i,j}$ is computed by looking into them. Note: each of these co-rated pairs are obtained from different users, in this example they come from users *1, u* and *m-1*.

**Fig 2: Visualization of user with similar items**

This is first needed for predicting the results. Here we will evaluate the similarity between the items and then will select the most similar items. Here to find similar items, we are first needed to separate the users who have rated the same kind of items. For example, Considering, the figure the rows represent the users and the columns represent the items. So, we can tell that the items *i, j* are rated by users *1, u, m-1.* One method for calculating the similarity between the items can be *cosine similarity* which is measured by computing the cosine of the angle of the 2 vectors. Mathematically the formula for *cosine similarity* for 2 items *A, B* is given as follows[12]

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

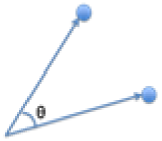


**Fig 3: Formula for cosine function**

Here we are using *adjusted cosine similarity* because we need to solve the limitation caused by *cosine similarity* i.e. difference in rating scale of the users*.* The solution for the problem is provided by *adjusted cosine similarity* i.e. by deducting the corresponding user average from each co-rated pair. In item-based collaborative filtering, similarity is computed along the columns i.e. each pair in the co-rated set corresponds to the different user. The formula for adjusted-cosine similarity is stated as follows[12,13]

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

**Fig 4: Formula for adjusted cosine function**

This formula is known as the *Pearson Coefficient.* In the formula given *U* is the set of correlated users who have rated books *i* and *j. $R_{u,i}$* is the rating of book *i* for user *u.* Similarly for book *j.* $\bar{R}_u$ is the average of all ratings of the user. This will be used for finding the similarity among the books.

### 2.3 Evaluating weighted sum

Now the next step is to generate predictions based on the similarity discovered by looking at the target user's ratings. There are several techniques which can be used for making predictions. Here we are using the weighted sum method. The formula for calculating the weighted sum is given as follows.

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}}(s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}}(|s_{i,N}|)}$$

**Fig 5: Formula for weighted sum**

This formula helps us to catch how the active users rate the homogenous items. To confirm that the predictions remain in the specified range, the weighted sum is scaled by sum of similarity terms.

## 3. Experimental Analysis

### 3.1 Dataset

We are going to use "***goodbooks10k***" [13] dataset for the books recommendation system. This dataset contains ratings of 10,000 popular books. Approximately there are 100 reviews, ratings for each book. The scale of rating goes from 1-5. 1 is the worse and 5 is the best. For books, there are 1-10000 books and for users, there are 1-53424 users. This "***goodbooks10k"*** module uses different datasets such as books, ratings. The "books" dataset contains the metadata of all the books present in the dataset. Some of the columns are{*"book_id","id","book_count","average_rating","ratings_count","original_title"*}. The ratings dataset contains information about the "*user_id*", "*book_id*" and "*average_rating"*[15]

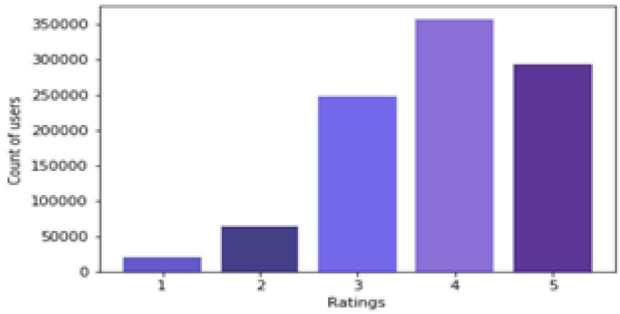


**Fig 6: Ratings(1-5) vs Count of user**

This bar graph shows the ratings vs the number of users on the x-axis and y-axis respectively i.e. It shows total how many users have rated for 1-rating, 2-rating, 3-rating, 4-rating, 5-rating. Now, let's take a look at the languages of the books, as it is present in the books dataset. The following is the representation of a number of books in that language vs the language itself in the form of a graph.

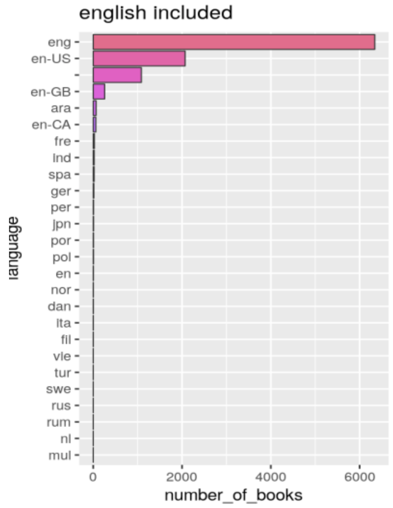


**Fig 7: number of books vs language (English Included)**

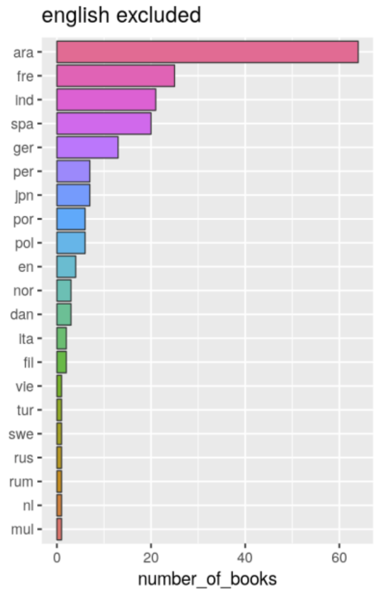


**Fig 8: number of books vs language (English Excluded)**

### 3.2 Evaluation Metrics

Evaluation metrics are used to determine the quality of a recommendation system. Here we are using *MAE* i.e. Mean Absolute Error to determine the quality. This method is a type of *statistical accuracy metric.* In statistical based approach numerical recommendation scores is compared with the actual user rating. *MAE* is widely used in case of *statistical accuracy metric*. The basic approach for *MAE* is as follows.

#### 3.2.1 Mean Absolute Error

The calculation of *MAE* is explicitly based on mathematics/statistics but there are some libraries available in various programming languages which helps us to achieve the *MAE.* In *MAE* prediction error for each record of test-data is calculated. If negative, then we need to convert each of them to positive, which is the absolute value of each observation. Finally, the last step is to calculate the mean of all the errors. The formula for *MAE* is as follows[14].

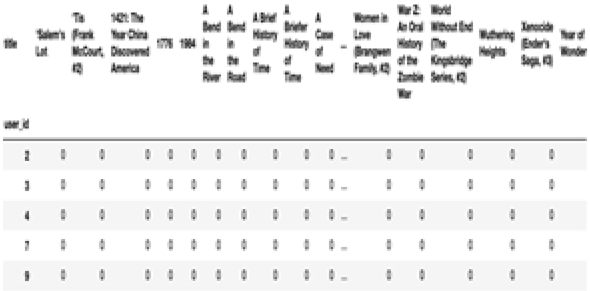
$$\frac{1}{n}\sum_{i=1}^{n} abs(y_i - x_i)$$

**Fig 9: Formula for *MAE***

As we can see in the formula for each pair of <$y_i$, $x_i$> we calculate the absolute difference and then the mean of all errors is the result. Lower the *MAE* better is quality of the recommendation system. We will be using this in our experiment for finding the quality of our book recommendation system[14]

### 3.3 Implementation

This book recommendation system using item-based collaborative filtering is experimented in python and compiled in Jupyter Notebook. All the experiments run on MacOS based PC with Intel i5 processor and 8GB Ram. The First step is to import the datasets which are mentioned in the previous section. For this, we are using '*pandas 0.20.3*' library of python. Our dataframe of books is a very sparse matrix of 28906 rows × 812 columns. We will fill in 0's for better efficiency. Now, after filling the matrix, the next step is to create a correlation matrix. For this, we are using the "*numpy 1.13.3*" library of python. Numpy is the fundamental package for statistical and scientific computation with python.



| title | 'Salem's Lot | 'Tis (Frank McCourt, #2) | 1421: The Year China Discovered America | 1776 | 1984 | A Bend in the River | A Bend in the Road | A Brief History of Time | A Briefer History of Time | A Case of Need | ... | Women in Love (Brangwen Family, #2) | World War Z: An Oral History of the Zombie War | World Without End (The Kingsbridge Series, #2) | Wuthering Heights | Xenocide (Ender's Saga, #3) | Year of Wonder |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| user_id | | | | | | | | | | | | | | | | | |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | |

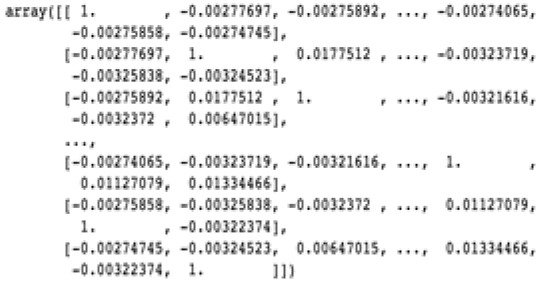
**Fig 10: Matrix filled with zeros whose rating is not specified**

The basic terminology for correlation matrix[14] is that, it is a matrix or a table showing correlation coefficients between any two variables. Each cell in the matrix represents the relation between the value in the corresponding column and the corresponding row. Formally, the relation between the coefficients is determined by the following formula.

$$M_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}}$$

**Fig 11: Formula for Correlation Matrix**

Here M is considered the matrix. $C_{ij}$ is the covariance of $x_i$ and $x_j$. The value $C_{ii}$ is the variance of $x_i$. The value of $C_{jj}$ is the variance of $x_j$. The correlation matrix generated looks like the following.



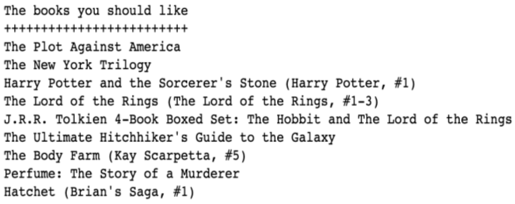
```
array([[ 1.        , -0.00277697, -0.00275892, ..., -0.00274065,
        -0.00275858, -0.00274745],
       [-0.00277697,  1.        ,  0.0177512 , ..., -0.00323719,
        -0.00325838, -0.00324523],
       [-0.00275892,  0.0177512 ,  1.        , ..., -0.00321616,
        -0.0032372 ,  0.00647015],
       ...,
       [-0.00274065, -0.00323719, -0.00321616, ...,  1.        ,
         0.01127079,  0.01334466],
       [-0.00275858, -0.00325838, -0.0032372 , ...,  0.01127079,
         1.        , -0.00322374],
       [-0.00274745, -0.00324523,  0.00647015, ...,  0.01334466,
        -0.00322374,  1.        ]])
```

**Fig 12: Correlation Matrix**

In Correlation matrix each cell has values [-1, 1]. This was the demonstration of how the correlation matrix is formed. Now for prediction we are using the weighted sum method as described in *fig 5* and the final results are shown in the next section.

### 3.4 Results

We tested our algorithm on the book *"The Alchemist"* and the result obtained after running the above simulations we got the following results.



```
The books you should like
+++++++++++++++++++++++++
The Plot Against America
The New York Trilogy
Harry Potter and the Sorcerer's Stone (Harry Potter, #1)
The Lord of the Rings (The Lord of the Rings, #1-3)
J.R.R. Tolkien 4-Book Boxed Set: The Hobbit and The Lord of the Rings
The Ultimate Hitchhiker's Guide to the Galaxy
The Body Farm (Kay Scarpetta, #5)
Perfume: The Story of a Murderer
Hatchet (Brian's Saga, #1)
```

**Fig 13: Final Result**

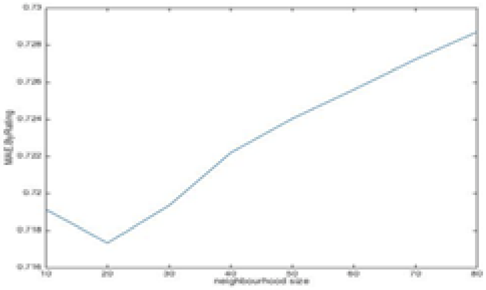


**Fig 14: Evaluating MAE VS number of neighbors**

### 4. Conclusion

Over a last few years recommendation systems are used widely in almost every business in the market. Best examples of recommendation systems are given by Amazon, Ebay etc. This paper discusses various methods that can be used to build a recommender system but implemented an item-based collaborative filtering approach on ***"goodbooks10k"*** dataset found on kaggle. Also the implementation of the experiment and the results are presented in the paper.

### 5. References

[1] Cureton, E. E., and D'Agostino, R. B. (1983). Factor Analysis: An Applied Approach. *Lawrence Erlbaum associates pubs.* Hillsdale, NJ.

[2] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science, 41(6)*, pp. 391-407.

[3] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*. December.

[4] Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J. (1999). Combining Collaborative Filtering With Personal Agents for Better Recommendations. In *Proceedings of the AAAI-'99 conference*, pp 439-446.

[5] Herlocker, J., Konstan, J., Borchers, A., and Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of ACM SIGIR'99*. ACM press.

[6] Herlocker, J. (2000). Understanding and Improving Automated Collaborative Filtering Systems. *Ph.D. Thesis, Computer Science Dept., University of Minnesota.*

[7] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and Evaluating Choices in a

Virtual Community of Use. In *Proceedings of CHI '95*.

[8] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM,* 40(3), pp. 77-87.

[9] Peppers, D., and Rogers, M. (1997). The One to One Future : Building Relationships One Customer at a Time. *Bantam Doubleday Dell Publishing*.

[10] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of CSCW '94*, Chapel Hill, NC.

[11] Sarwar, B., M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J. (1998). Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In *Proceedings of CSCW '98*, Seattle, WA.

[12] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. In *Proceedings of the ACM EC'00 Conference*. Minneapolis, MN. pp. 158-167

[13] https://www.kaggle.com

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

[15] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining Content-Based and Collaborative Filters in an Online Newspaper," Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation, Aug. 1999.

[16] https://www.bluepiit.com

[17] Michael Hashler, "Recommender Lab: A Framework for Developing and Testing Recommendation Algorithms" Nov. 2011.

[18] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems" KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2007, ACM.

[19] D.M. Pennock and E. Horvitz, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory And Model-Based Approach," Proc. Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering, Aug. 1999.

[20] Joonseok Lee, Mingxuan Sun, Guy Lebanon: A Comparative Study of Collaborative Filtering Algorithms (2012)

[21] Robin Burke, "Hybrid Recommender Systems: Survey and Experiments", California State University, Fullerton Department of Information Systems and Decision Sciences.

[22] Dharmendra Pathak, Sandeep Matharia and C. N. S. Murthy, "NOVA: Hybrid Book Recommendation Engine", IEEE, 2012.