

Suggestive Approaches to Create a Recommender System for GitHub

Surbhi Sharma

Shri Mata Vaishno Devi University, Katra-182320, India

Anuj Mahajan*

Shri Mata Vaishno Devi University, Katra-182320, India
E-mail: anuj.mahajan@smvdu.ac.in; anuj.mahajan.iiitb@gmail.com

Received: 08 February 2017; Accepted: 09 May 2017; Published: 08 August 2017

Abstract—Recommender system suggests users with options that may be of use to them or may be of their interest or liking. These days recommender systems are used widely on most systems and especially on those which are connected to World Wide Web, it may be a mobile app, a desktop application, or a website. Most advertisements on these systems are focused on targeting a specific group. Recommender systems provide a solution to such a scenario where the recommendations need to be targeted based on a user profile. Almost all commercial, collaborative or even social networking websites rely on recommender systems. In this paper, we specifically focus on GitHub, a source code hosting site and one of the most popular platforms for online collaborative coding and sharing. GitHub offers an opportunity for researchers to perform analysis by providing REST-based APIs for downloading its data. GitHub hosts a vast amount of user repositories so it is quite difficult for a GitHub user to decide to which repository she should contribute on GitHub. So, our paper aims to review different approaches that can be used for creating a recommender system for GitHub, to provide personalized suggestions to GitHub users to which repositories they should contribute. In this paper, we have discussed collaborative filtering, content-based filtering, and hybrid filtering, knowledge-based and utility-based approaches of a recommender system.

Index Terms—Recommender Systems, GitHub, Collaborative filtering, Content-based filtering, hybrid filtering, Knowledge-based approach, Utility-based approach.

I. INTRODUCTION

GitHub has gained a lot of popularity in the recent years due to its functionality. GitHub offers a well-defined user interface, one can easily create an account on GitHub and can create a repository to work upon, fork any repository. It also integrates social features, e.g. anybody can subscribe to other's information by 'following' that user and can 'watch' her updates related to repository [1]. Various open source projects like Ruby

on Rails have started migrating their code to GitHub. Social networking sites like Facebook, Twitter etc. also use GitHub. GitHub's backbone is Git which is a distributed version control system (VCS). Version control systems make sure that nobody can overwrite each other's modification while working collaboratively. Git takes a snapshot of every update by each user, so one can revert back any time to its previous version as it maintains a history of each event [2]. GHTorrent is a service which gathers all GitHub's data and stores this vast amount of data in the form of MongoDB database dumps. GHTorrent provides a simple interface to download this data so researchers can download GitHub's data from GHTorrent efficiently [3]. As GitHub have millions of users and repositories so it might be confusing for any contributor who wants to contribute on GitHub that to which repository he/should contribute. We here discuss the different approaches to recommender system which can suggest to which repository contributor should contribute based on her interest. Recommender systems play an important role nowadays in every field. It provides filtered information to users based on their interest. For Example, Facebook uses recommender system to suggest friends to people with an option 'People you may know', various e-commerce sites recommend items which one can purchase based on the previous history rather than exploring whole information. So, it saves a lot of time of

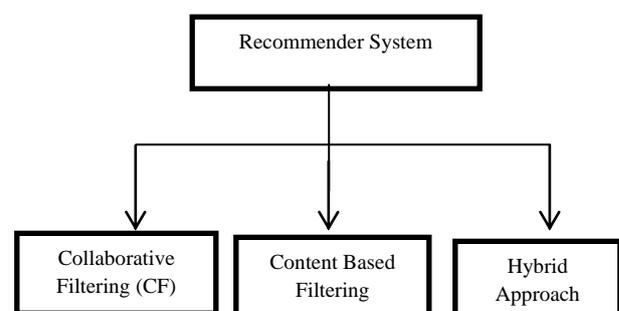


Fig.1. Different Approaches to creating a Recommender System.

users, due to this recommender system has gained a lot of popularity. Fig.1 describes different approaches to

creating a Recommender System. Approaches of Recommender systems are mainly divided into three categories- Collaborative Filtering (CF), Content-based and Hybrid approach.

In the next sections, we discuss each approach one by one and draw a comparison of all these approaches.

II. COLLABORATIVE FILTERING APPROACH

It is one of the most widely used approaches in a recommender system. Collaborative filtering (CF) offers suggestions/recommendations to users based on other users having similar tastes. It takes into account user's feedback in the form of ratings and then based on that similar users are determined using various correlation measures.

Considering an example of Collaborative Filtering (CF) in Table 1-Suppose user 1 has earlier read Book 1 and Book 3 and we have to predict the rating for Book 2 of the same user that whether the same user has interest in reading Book 2 or not then using this approach, first task is to determine users who are similar to user 1 based on correlation measures like Pearson Correlation Coefficient (PCC), Cosine Similarity etc. Secondly, we will get the user who is most similar to user 1, as it's clear in below table that user 3 is most similar to user 1 then if User 3 has read Book 2 then this approach will recommend user 1 to read book 2. This is the basic concept of Collaborative Filtering i.e. recommending items based on the likelihood of other similar users.

Table 1. Example of Collaborative Filtering

Users	Book1	Book2	Book3	Book4
User 1	5	??	4	-
User 2	4	5	-	4
User 3	5	5	4	3
User 4	4	-	-	4

A. Advantages of Collaborative Filtering (CF) [4] [15]-

(a) Collaborative Filtering can be applied in domains where less information is available about the content of items as it does not depend on the profile of items.

(b) It is considered to be faster and accurate than content-based approach.

B. Challenges of Collaborative Filtering [5] [6]-

Collaborative filtering has following 3 challenges- Cold-start, Sparsity, and Scalability.

(a) Cold-Start- To make recommendations for an item, that item need to be rated by other users. If any item is not yet rated by any user then this approach will not suggest that item to any user. Due to this, it is a challenge for CF.

(b) Scalability – Recommender systems should provide suggestions to users accurately and timely. So, with the increasing amount of data over e-commerce sites, GitHub etc. recommender system needs to scale up their

computation power to offer timely recommendations.

(c) Sparsity – If the existing data about ratings is sparse then it's difficult to determine the similarity between users and thus it will affect the quality of recommendations.

Collaborative Filtering is further divided into two categories as shown in Fig 2-

(a) Memory-based approach

(b) Model-based approach

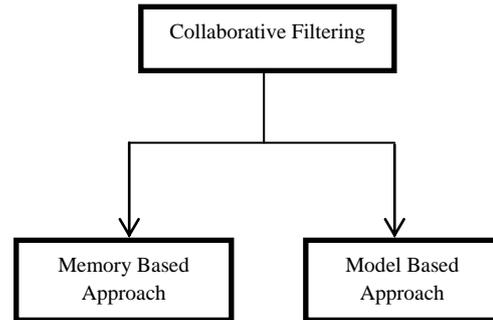


Fig.2. Approaches of Collaborative Filtering (CF)

C. Memory- Based Approach-

Memory based approach operates on user-item rating data to make a prediction for the target user. The memory-based approach generally employs neighborhood algorithms to determine the neighbors who are most similar to the desired user and ultimately based on the preferences of neighbors, preferences of the target user is predicted. To calculate the similarity between users various correlation measures can be used i.e. Pearson Correlation Coefficient (PCC), Cosine Similarity, and Spearman Coefficient etc. PCC calculates the value between -1 and 1 whereas Cosine Similarity calculates the value between 0 and 1. [7]. It is widely used in few e-commerce sites like Amazon .Fig 3 describes categories of Memory Based Collaborative Filtering.

1) User-based Collaborative Filtering:

In user-based collaborative filtering, firstly users similar to target user are determined using correlation measures, then items used by similar users are selected for recommendation to the active user. [8]

Pearson Correlation Coefficient, Spearman Coefficient, and Cosine Vector Similarity etc. are few of the measures used for determining similarity.

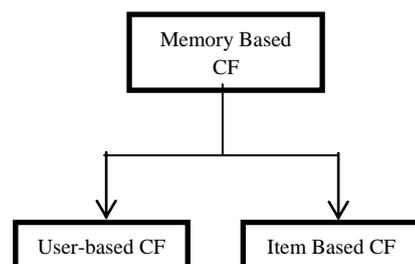


Fig.3. Categories of Memory-Based CF

Whereas $rank(r_{ui_1}) - rank(r_{ui_2})$ is Difference between two ranks.

n- Number of observations

Cosine Based Similarity [9] [6] –

In this similarity measure, two users/items are considered as two vectors in m- dimensional space. The similarity between them is measured as the cosine of the angle between two vectors. It is generally used to determine the similarity between two documents and then gradually used in collaborative filtering to determine the similarity between two items/users rather than documents. It is generally used in positive space so it calculates the similarity in the value of 0 to 1. It is efficient to use in case of sparse vectors. Equation 3 shows Vector Cosine Similarity between two documents i and j-

2) *Item-based Collaborative Filtering:*

In item-based collaborative filtering, instead of determining the similar users, here similarity is calculated between items which test user have rated and items which are not yet rated by the active user. So, here the profile of item is taken into consideration. Based on the similar items recommendation is provided to target user [9].

3) *Different Correlation Measures-*

Pearson Correlation Coefficient-

It is commonly used correlation measure, also called as Pearson Product Moment Correlation. It is used to determine the similarity between users/items. It calculates the similarity in the value of -1 to 1. If the value is 1 then it shows positive correlation and items are closely similar to each other, 0 value indicates no correlation exists between attributes and -1 value indicates attributes are negatively correlated. It is calculated by using below formula-

For user-based similarity [10] -

$$Sim_{a,b}^{PCC} = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}} \quad (1)$$

Sim (a, b) - Similarity between users a and b.

$r_{a,p}$ – Rating of the user a for product p.

$r_{b,p}$ – Rating of the user b for product p.

\bar{r}_a, \bar{r}_b – User's average ratings

P - Set of all items

Spearman Correlation Similarity [5] -

Spearman gives better results than Pearson if the dataset has not a normalized distribution. It also calculates the strength of the relationship between two variables.

$$r(U_1, U_2) = \frac{6 \sum (rank(r_{ui_1}) - rank(r_{ui_2}))^2}{n(n^2 - 1)} \quad (2)$$

$$w_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| * \|\vec{j}\|} \quad (3)$$

If vector $\vec{A} = \{x_1, y_2\}$, vector $\vec{B} = \{x_2, y_2\}$ then vector cosine similarity between \vec{A} and \vec{B} is calculated as-

$$w_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| * \|\vec{j}\|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}} \quad (4)$$

Tanimoto Similarity Coefficient [5]-

This is mostly used for sparse datasets. It denotes the ratio of the intersection of two datasets. Its value is between 0 and 1. Tanimoto Coefficient is calculated by using following formula-

$$T(X, Y) = \frac{X \cap Y}{(X + Y) - (X \cap Y)} \quad (5)$$

X, Y – Elements in datasets

Euclidean Distance Similarity [5]-

This measure computes the Euclidean distance between two vectors. Shorter the distance, higher the similarity between vectors. Firstly, Euclidean distance is determined between vectors and then based on that Euclidean Similarity is computed. Its value is also between 0 and 1.

Formula for calculating the Euclidean Distance is given as-

$$d_{i,j} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (6)$$

Euclidean Distance Similarity is calculated by using above value of $d_{i,j}$ as-

$$\frac{1}{1 - d_{i,j}} \quad (7)$$

4) *Advantages of Memory-based Approach [6]-*

(a) In this approach, new data can be added easily to existing one.

(b) It doesn't deal with the item description.

(c) It takes into account co-rated items.

5) *Flaws of Memory-based Approach [6] –*

(a) It doesn't deal well with large datasets.

(b) It is dependent on ratings

(c) It doesn't tackle with sparsity.

(d) Cold start problem exists i.e. new user and new items can never be recommended.

D. Model- Based Approach –

The model-based approach uses the existing rating data to learn a model using various data mining and machine learning algorithms like Bayesian networks, rule-based and clustering approaches [9], and then it uses this model to provide recommendations. Bayesian

network is one of the probabilistic models for the model-based approach of collaborative filtering. The probabilistic approach aims to predict the value of a vote for items which is not yet been rated by a test user. In a Bayesian network, nodes correspond to items in a domain and states of the nodes correspond to possible vote values for unrated items. The situation, where values for missing data cannot be predicted, are considered as 'no-vote' state [11]. Rule based approach applies association rules between co-rated items to determine the association, support and confidence are the two measures to determine the strength of association rules. Based on the values of support and confidence recommendations are provided to users [12]. Association Rule Mining is fast to implement and deals well with large sets of data [14]. Clustering aims to group the people in clusters with similar interests and determining the optimal clusters is an easy task because in a particular cluster choice of users will be similarly related to an item so due to clusters, recommendations can be provided to users more accurately than other methods. Clustering based methods are the best choice in case of sparse data. [13]

One of the most commonly used algorithms for the model-based approach is matrix factorization method. This approach helps to learn complex patterns based on the training model and then it efficiently performs recommendations [8].

1) Advantages of Model-Based Approach-

- (a) It provides more accurate recommendations than memory-based approach.
- (b) It can handle scalability and sparsity efficiently.
- (c) Model-based algorithms such as association rule mining are more robust to profile injection attacks than memory-based approach. [14]

2) Flaws of Model-Based Approach [6] -

- (a) It is time-consuming approach as it requires first to learn a model, then it performs predictions
- (b) It is an expensive approach.
- (c) It covers less diverse user range than memory-based approach.

III. CONTENT BASED FILTERING APPROACH –

The content-based approach provides suggestions to users based on the metadata description of an item and user's preferences i.e. items which an active user have liked in the past, based on that new recommendations are provided to the user rather than determining the correlation between users having similar interests as in collaborative filtering. Content-based approach fetches the properties of an item from the textual description of an item, based on that it builds a model or profile of users describing the characteristics of items previously rated in the past. Based on the above structural information of items recommendations are predicted for unrated items. This approach considers both user's preferences and attributes of the items liked by the user. This approach is widely used in various domains like email, news and web

search [16].

Content-based filtering is performed in three steps as various methods are needed to represent item description in a structured manner and user profile. Along with it, various techniques needed to compare user profile with the item description. Each step is performed by a different component. Different components are - Content Analyzer, Profile Learner and Filtering Component [17].

(a) Content Analyzer- This component works on the unstructured data about items retrieved from multiple sources to make it in a structured format. This component converts the information of items in a form suitable for further steps.

(b) Profile Learner-This component tries to construct user profile based on user preferences using machine learning algorithms. Items liked or disliked by the user in past is taken into consideration

(c) Filtering Component – This component suggests items by comparing the user profile with the item description.

A. Classification Based Learning Algorithms:

Content-based filtering depends on classification learning algorithms to predict user's interests in new items. These algorithms estimate the numeric value by creating a function to estimate user's interests. Few of them are machine learning algorithms also. A few algorithms used in this approach are-

(1) Decision Trees-

Decision Trees deals well with structured data. Decision Trees works by partitioning the data into subclasses until a single instance of each class is included in subgroups. Based on the feedback of users for different items which have been liked in the past, decision trees can learn and model the profile of users as it is one of the main steps in content-based filtering approach.

(2) Nearest Neighbor Methods-

Nearest Neighbor methods operate over data stored in memory. Description of all items and the user profile is stored in memory, and then different measures are used to determine the similar item. Choice of correlation measure depends on the type of data. For vector space model vector cosine similarity is used. For structured data, Euclidean distance is used. Formulas of Euclidean distance and Vector Cosine Similarity have already discussed in Section 2.

(3) Naïve Bayes Classifier [19] –

Naïve Bayes Classifier is widely used for information retrieval and classification purposes. It uses simple Bayes' theorem for classifying the items into a particular class. Equation 8 and 9 shows formula of bayes classifier-

$$P(C = c_k | X = x) = P(C = c_k) \times \frac{P(X=x | C=c_k)}{P(x)} \quad (8)$$

Where,

$$P(X = x) = \sum_{k=1}^{e_c} P(X = x | C = c_k) \times P(C = c_k) \quad (9)$$

$e_c = e_1, \dots, e_k$ – All items fall into one of these classes.

C- Random variable

X- Vector random variable whose values are vectors of feature values ($x = x_1, \dots, x_j$), one vector for each item.

$P(C = c_k | X = x)$ – Conditional Probability that an item belongs to a particular class.

This method is very effective in classifying the items.

B. Advantages of Content-Based Approach:

(a) In Content-based filtering recommendations are based on individual’s user profile and item description liked in the past rather than depending on users with similar interests as in collaborative filtering

(b) Content-based approach solves the problem of rating an item for the first time as it is present in collaborative filtering. It can also recommend new items that have not been yet rated.

(c) Content-based approach exhibits transparency as it defines clearly that this item is recommended based on preferences of users and item descriptions.

C. Flaws of Content-Based Filtering Approach:

(a) As this approach is dependent on item description so if less information is available then content-based approach cannot provide accurate recommendations.

(b) Another drawback is ‘new user’ problem i.e. the new user should rate few items before it is being recommended.

(c) Serendipity problem also exists as it always provides expected outcomes; it never recommends something interesting which user have never rated.

IV. HYBRID BASED APPROACH

Hybrid based Approach is a combination of both collaborative filtering and content-based filtering. Fig 4 shows hybrid approach.

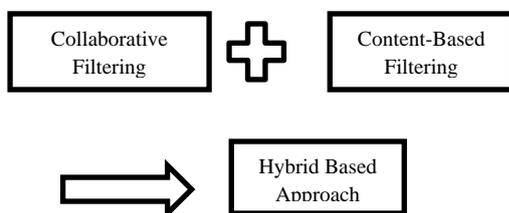


Fig.4. Hybrid Based Filtering Approach

As Collaborative filtering approach and Content-based approach both have their own advantages and limitations. To improve the quality of recommendations, a hybrid approach is used to overcome the limitations of both collaborative and content-based approach.

Different approaches used in hybrid recommender systems are –Weighted, Mixed, Switching, Cascade, Meta-level, Feature Combination and Feature

Augmentation [20] [21]

A. Weighted:

Weighted based hybrid approach computes the score of recommendation by combining the scores of individual recommendation techniques exists in the system. Table 2 describes an example of the weighted based approach.

Table 2. Example of Weighted Based Approach

Items	Item 1	Item 2	Item 3
Collaborative Filtering Scores	5	4	1
Content based score	-	3	2
Scores based on Weighted approach	5	7	3
Rank	2	1	3

In above table, this approach computed the score by summing up the scores of collaborative and content-based approach and based on that, the ranking is provided to recommend items. Item 2 is given Rank 1 as it has maximum score 7, Rank 2 is given to item 2 and Rank 3 is given to item 3 as it has a minimum score of 3.

B. Mixed:

Table 3 shows an example of Mixed Approach. In Mixed Approach, recommendations from different approaches are presented simultaneously to overcome the ‘new item’ and ‘new user’ problems encountered in a collaborative and content-based approach.

Table 3. Example of Mixed Approach

Items	Item 1	Item 2	Item 3	Item 4
Collaborative Filtering Score	5	4	1	-
Content based score	-	2	3	4
Scores based on Mixed approach	10	8	7	9
Ranking	1	3	4	2

Considering above example, here the mixed approach is used so recommendations based on collaborative filtering and content-based are provided at the same time.

Zipper principle is used here, an item with highest collaborative filtering score is assigned value 10 and item with highest content-based filtering score is assigned a value 9 and so on.

So in above example highest collaborative score is 5 so it is assigned value 10 and content have the highest score of 4, hence 9 value is assigned to it, further next highest score of collaborative is 4, it is assigned 8 value, then next highest score of content is 3, it is assigned 7 value. In this way, the mixed approach works.

C. Switching:

In this approach, the system switches between different techniques of recommender system based on the current situation. For Example, if data is less sparse in a

particular situation then this approach will choose collaborative filtering scores; in case data is sparse then it will switch to content-based approach. Switching approach increases the complexity as switching criteria needs to be clearly defined.

D. Cascade:

Cascade based approach performs its task in two steps. In the first step, any recommendation technique is applied to produce selective items and then in the second stage, another recommendation technique is applied on that data to refine the recommendations. If the content-based approach is applied at first stage then the output of this stage will have filtered content rather than whole data, so now collaborative filtering approach will refine this selective data rather than working on whole data.

E. Feature Augmentation:

This approach also consists of 2 steps- Output of one stage is given as input to another stage. Augmentation increases the efficiency as it aims to improve the efficiency of the technique used in the first step by adding the extra functionalities using another recommendation approach.

F. Feature Combination:

In Feature Combination Approach, different data sources are utilized by a single recommender system.

G. Meta- Level:

The meta-level approach also performs its task in 2 stages. In the first stage, the model is generated using a particular recommendation technique. In the second stage, this approach utilizes the entire model generated in the first stage. One of the benefits of meta-level approach can be observed easily in a situation where in the first stage, the content-based approach is used then it will generate ratings in a compressed form and then in the second stage, collaborative filtering can easily operate on dense data. In this way, efficiency can be greatly increased.

V. OTHER APPROACHES OF RECOMMENDER SYSTEMS

Collaborative filtering, Content based filtering and Hybrid based approach are the three most commonly used approaches for recommender systems which we have discussed above. In this section, other approaches of a recommender system are also discussed. i.e. Knowledge-based Recommender system and Utility-based Recommender System.

A. Knowledge-Based Recommender Systems [23] [24]-

Knowledge-based recommender systems are the best choice in case of customer buying complex products such as financial services etc. In such scenarios on the basis of similar users or on the basis of the item, description recommendation is not possible. So, in such cases,

knowledge-based recommender systems are used as it focuses on the deep knowledge of product domain so it has a more clear idea of choice of users and can provide more accurate recommendations. It consists of two categories-Case Based and Constraint Based. Along with this, knowledge-based recommender systems can determine the relationship that how a particular item can meet user's choices. It also provides an explanation for all recommendations that why this particular item will suit their needs. Explanation feature about suggestions attracts a lot of customers as it also increases the trust of customers.

B. Utility-Based Recommender Systems [25]-

Utility based recommender systems compute the utility of each item to provide recommendations to users. Description of items is used as background data for this approach and to describe the user's preferences over different items a function is used. This function is then utilized to rank different items for the user. Sparsity, 'new user' and 'new item' issues which exist in collaborative and content-based approaches are resolved in utility based recommender systems. However, the main issue in this approach is the creation of utility function for each user.

VI. CONCLUSION

We have discussed the need of a recommender system for GitHub. Such a system shall suggest the contributors to which repositories they should contribute based on their interests. Providing the filtered information in the form of recommendations shall enhance the overall user experience on the GitHub and which in turn may lead to increase in the number of contributors on the GitHub. We have also discussed the approaches which may be used to create such a recommender system. The following approaches have been elaborated in detail in this paper: Collaborative Filtering, Content based Filtering, Hybrid Filtering, Knowledge-Based and Utility based approach. We conclude our study in the form of a detailed table discussing the merits and demerits of each approach in Table 4.

VII. FUTURE WORK

Based on our research, we shall be working on implementing these approaches to create a recommender system for the GitHub. We shall be using the GitHub dump [26] available on the internet. We shall be experimenting with the Collaborative filtering approach along with a few correlation measures like Cosine similarity and Pearson Correlation Coefficient (PCC) to make recommendations to the GitHub users regarding which repositories to work on.

Table 4. Overview of Different Approaches of a Recommender System

Approaches	Categories	Description	Representative Techniques	Advantages	Limitations
1. Collaborative Filtering (CF)	Memory-Based CF	User-Item rating database is used as input to find similar users. Based on preferences of similar users, recommendations are provided to the target user.	<ul style="list-style-type: none"> Neighborhood Methods User-Based CF (Similarity Measures) Item-Based CF (Similarity Measures) 	<ul style="list-style-type: none"> Independent of item description Easy to use. High performance 	<ul style="list-style-type: none"> Cold start problem Slow and less accurate approach Doesn't deal with sparsity
	Model-Based CF	Uses existing rating data to first learn a model by using various machine-learning techniques, based on that it provides recommendations	<ul style="list-style-type: none"> Rule-Based Approaches Bayesian Networks Clustering Methods 	<ul style="list-style-type: none"> More Scalable Faster Approach More Robust 	<ul style="list-style-type: none"> Inflexible Resource Consuming Expensive Approach
2. Content-Based Filtering	NA	It recommends items based on the past history of a user. It takes into consideration item description and user's preferences.	<ul style="list-style-type: none"> Decision Trees Nearest Neighbor Methods Naïve Bayes Classifier 	<ul style="list-style-type: none"> New item problem is resolved Independent of Determining Similar Users. Transparency 	<ul style="list-style-type: none"> New User problem Dependency on Item Description Serendipity problem
3. Hybrid Filtering	<ul style="list-style-type: none"> Weighted Mixed Switching Cascade Feature Augmentation Feature Combination Meta-level 	It combines two or more recommendation techniques to improve the quality of recommendations	<ul style="list-style-type: none"> Latent Semantic Indexing (LSI) Markov Chain Monte Carlo Bayesian Mixed Effect Regression Model 	<ul style="list-style-type: none"> Improves Quality of Recommendations Overcomes sparsity issue Resolves new user and new rating problems 	<ul style="list-style-type: none"> High Complexity Dependent on external information Expensive Approach
4. Knowledge Based	<ul style="list-style-type: none"> Constraint-Based Case Based 	Recommendations are provided based on inferences of user's needs and preferences	<ul style="list-style-type: none"> Multi-Attribute Utility Theory Correlation Measures 	<ul style="list-style-type: none"> No cold start problem Improved Quality Explanation of Predictions 	<ul style="list-style-type: none"> Need of Knowledge Expert
5. Utility-Based	NA	Computes the utility of each item to provide recommendations to users	<ul style="list-style-type: none"> Multi-Attribute Utility Theory (MAUT) 	<ul style="list-style-type: none"> Resolves new user and new item issues Overcomes sparsity 	<ul style="list-style-type: none"> Creation of utility function

NA – Not Applicable

REFERENCES

[1] Hu, Jiangtang. "The Hitchhiker's Guide to GitHub: SAS Programming Goes Social."

[2] Bruno, Rodrigues. "Version Control Systems to Facilitate Research Collaboration in Economics." *Computational Economics* (2015): 1-7.

[3] Gousios, Georgios, and Diomidis Spinellis. "GHTorrent: GitHub's data from a firehose." In *Mining software repositories (msr)*, 2012 9th IEEE working conference on, pp. 12-21. IEEE, 2012.

[4] Aberger, Christopher R. "Recommender: An Analysis of Collaborative Filtering Techniques."

[5] Tapucu, Dilek, Seda Kasap, and Fatih Tekbacak. "Performance comparison of combined collaborative filtering algorithms for recommender systems." In *Computer Software and Applications Conference Workshops (COMPSACW)*, 2012 IEEE 36th Annual, pp. 284-289. IEEE, 2012

[6] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." *Advances in artificial intelligence 2009* (2009): 4.

[7] Gong, SongJie, HongWu Ye, and HengSong Tan. "Combining memory-based and model-based collaborative filtering in a recommender system." In *Circuits, Communications and Systems, 2009. PACCS'09. Pacific-Asia Conference on*, pp. 690-693. IEEE, 2009

[8] Bogers, Toine, and Antal Van den Bosch. "Collaborative and content-based filtering for item recommendation on social bookmarking websites." *Submitted to CIKM 9* (2009).

[9] Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. "Item-based collaborative filtering recommendation algorithms." In *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295. ACM, 2001.

[10] Polatidis, Nikolaos, and Christos K. Georgiadis. "A multi-level collaborative filtering method that improves recommendations." *Expert Systems with Applications* 48

- (2016): 100-110.
- [11] CarlKadie, JohnS Breese DavidHeckerman. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering." *Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA 98052* (1998).
- [12] Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. "Analysis of recommendation algorithms for e-commerce." In *Proceedings of the 2nd ACM conference on Electronic commerce*, pp. 158-167. ACM, 2000.
- [13] Ungar, Lyle H., and Dean P. Foster. "Clustering methods for collaborative filtering." In *AAAI workshop on recommendation systems*, vol. 1, pp. 114-129. 1998.
- [14] Lemdani, Roza, Nac'ra Bennacer, Géraldine Polailon, and Yolaine Bourda. "A collaborative and semantic-based approach for recommender systems." In *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, pp. 469-476. IEEE, 2010.
- [15] Melville, Prem, Raymond J. Mooney, and Ramadass Nagarajan. "Content-boosted collaborative filtering for improved recommendations." In *AAAI/IAAI*, pp. 187-192. 2002.
- [16] De Pessemier, Toon, Kris Vanhecke, Simon Doooms, and Luc Martens. "Content-based recommendation algorithms on the Hadoop MapReduce framework." In *7th International Conference on Web Information Systems and Technologies (WEBIST-2011)*, pp. 237-240. Ghent University, Department of Information technology, 2011.
- [17] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends." In *Recommender systems handbook*, pp. 73-105. Springer US, 2011.
- [18] Pazzani, Michael J., and Daniel Billsus. "Content-based recommendation systems." In *The adaptive web*, pp. 325-341. Springer Berlin Heidelberg, 2007.
- [19] Lewis, David D. "Naive (Bayes) at forty: The independence assumption in information retrieval." In *Machine learning: ECML-98*, pp. 4-15. Springer Berlin Heidelberg, 1998.
- [20] Felfernig, Alexander, Michael Jeran, Gerald Ninaus, Florian Reinfrank, Stefan Reiterer, and Martin Stettinger. "Basic approaches in recommendation systems." In *Recommendation Systems in Software Engineering*, pp. 15-37. Springer Berlin Heidelberg, 2014.
- [21] Burke, Robin. "Hybrid recommender systems: Survey and experiments." *User modeling and user-adapted interaction* 12, no. 4 (2002): 331-370.
- [22] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *Knowledge and Data Engineering, IEEE Transactions on* 17, no. 6 (2005): 734-749.
- [23] Felfernig, Alexander, Gerhard Friedrich, Bartosz Gula, Martin Hitz, Thomas Kruggel, Gerhard Leitner, Rudolf Melcher et al. "Persuasive recommendation: serial position effects in knowledge-based recommender systems." In *Persuasive technology*, pp. 283-294. Springer Berlin Heidelberg, 2007.
- [24] Felfernig, Alexander, and Kostyantyn Shchekotykhin. "Debugging user interface descriptions of knowledge-based recommender applications." In *Proceedings of the 11th international conference on Intelligent user interfaces*, pp. 234-241. ACM, 2006.
- [25] Huang, Shiu-Li. "Designing utility-based recommender systems for e-commerce: Evaluation of preference elicitation methods." *Electronic Commerce Research and Applications* 10, no. 4 (2011): 398-407
- [26] Chatziasimidis, Fragkiskos, and Ioannis Stamelos. "Data

collection and analysis of GitHub repositories and users." In *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on*, pp. 1-6. IEEE, 2015.

Authors' Profiles



Surbhi Sharma was a PG Scholar in the Department of Computer Science & Engineering at Shri Mata Vaishno Devi University, Katra.



Anuj Mahajan* is working as Assistant Professor in the Department of Computer Science & Engineering at Shri Mata Vaishno Devi University, Katra.

How to cite this paper: Surbhi Sharma, Anuj Mahajan*, "Suggestive Approaches to Create a Recommender System for GitHub", *International Journal of Information Technology and Computer Science(IJITCS)*, Vol.9, No.8, pp.48-55, 2017. DOI: 10.5815/ijitcs.2017.08.06