PERSONALIZATION USING RECOMMENDATION SYSTEMS: STATE-OF-THE-ART REVIEW

V Lakshmi Chetana*, Dr. SP Syed Ibrahim, Dr. B Rama, Dr. P KiranSree

Asst. Prof., DVR & Dr.HS MIC College of Technology, Kanchikacherla
Professor, School of Computing Science and Engineering, VIT University-Chennai Campus, Chennai
Asst. Prof., Computer Science Department, Kakatiya University, Hanamkonda
Professor, Dept. of CSE, Shri Vishnu Engineering College for Women, Bhimavaram
*Corresponding author: mailtochetana@gmail.com

ABSTRACT:

In the era of internet, the proliferation of services will be a great challenge to the users in selecting the best and optimal services that suit the user requirement. So there is a need to filter, prioritize and efficiently deliver relevant information according to the user requirement. Recommendation systems rescue the users by selecting the required content after searching through large volumes of services. Recommendation system predicts user responses to options using diverse prediction techniques like content-based system, collaborative filtering system and hybrid system. This paper, in detail, explores different characteristics, approaches, evaluation metrics and potentials of different prediction techniques in recommendation systems that endorse the quality of services for the users.

Keywords: Content-based Recommendation system, Collaborative Recommendation system, Hybrid Recommendation system, Evaluating Recommendation system, Evaluation metrics, Similarity measures, Recommendation systems.
[1] INTRODUCTION

Increase in the growth of available information online has created the problems of information overload and paradox of choice. Most of the e-commerce sites uses personalization and prioritization which provides content and services tailored to individuals based on knowledge about their preferences and behavior. Recommendation systems are the important tools to solve the challenges of information overload and paradox of choice.

The objective of the recommendation system is to create meaningful predictions to the users for products or items of their interest. There are multitude ways to find the interests of the users like tags, reviews, likes and number of visits for a page or items or blogs. It is the process of finding what the customer needs, either data or an item [1]. Some of the example recommender systems are Ringo, Pandora, Last.fm for music, Netflix, Sceneami.com for movies, YouTube, IMDb for videos, LinkedIn for jobs, Facebook, Twitter for friends, Amazon for books, OKCUPID for dating and ACM, Springer, IEEE, ScienceDirect etc for published papers.

The point that differentiates the recommender system with search engines is, recommender systems try to present relevant content that the user did not necessarily search for and which they have not even heard about. The goal of the recommendation system is to make legitimate recommendations for the user which the user has not seen or to provide users with item lists ranked in accordance with the estimated preferences. The recommendation systems are mostly commonly used in two scenarios. First, when there are a very large number of items available for products from Amazon, Movies from Netflix, Music from Pandora, news items on Google news etc, it becomes difficult for the user to find what they want. Searching is helpful when the users know what they are looking for. In this situation, the recommender system recommends items that the user may not know about or hard-to-find or non-hit items. The total sale of such items is called long-tail. [Figure-1] represents the long-tail phenomenon.

![Figure: 1. Long-tail Phenomenon](image-url)
The y-axis represents the popularity i.e., the number of times a product is chosen and the x-axis represents products ordered according to their popularity. Online sites provide the entire range of products that are most popular and the rest of the tail. Second, as personal taste plays an important role in selecting an item, recommendation systems often utilize the interests of a group which helps in discovering items based on the similarities.

This paper reviews various techniques used in recommendation systems and is organized as follows - The second section describes the steps involved in recommender systems like information gathering, learning and recommendation. Section three explains about various filtering techniques used to produce recommendations followed by various evaluation metrics in section four.

[1.1] STEPS IN RECOMMENDER SYSTEMS

The steps that are involved in recommender systems are
1. Information gathering
2. Learning
3. Recommendations
[1.1.1] INFORMATION GATHERING STEP

In this phase, user preferences or interest is collected from different sources like user profiles, reviews, tags, shares, likes, newsgroup and so on. Reddit.com is a website which collects votes for an item or newsgroup. Recommender systems depend on different types of input extracted through.

**Explicit feedback**:
The system generates an interface for the user to provide ratings or feedback for an item to improvise itself. The accuracy of the recommender system relies on the quantity of the ratings given by the user. But the main problem with this method is, the user may not be everytime ready to give complete information and apt information.

**Implicit feedback**:
The system implicitly derives the user preferences or interests by monitoring the previous actions of users likes purchase history, navigation history, a number of visits to a web page, number of shares and likes in social networking sites, content of the email and so on.[2]. It reduces burden on the user but the problem with it is, it gives less accuracy.

**Hybrid feedback**:
To improve the performance of the system the pros of both explicit and implicit feedback is combined. Here the system takes explicit feedback only when he is interested in expressing.

[1.1.2] LEARNING STEP

It uses different machine learning algorithms to filter user interests collected in the first step to predict interests of user's choice. The different learning algorithms that are used are Bayesian, Decision Tree, Neighborhood-based , Neural Network, Rule learning, Ensemble, Gradient – Descent based, Kernel methods, Clustering, Associative classification, Lazy learning, Regularization methods, Topic Independent Scoring Algorithm[3].

[1.1.3] RECOMMENDATION STEP

It predicts what sort of items the user prefers. These predictions are done either by the system observed activities or by analyzing the data collected in the information collection step. Any of the learning algorithms may be applied to analyze the input data. [Figure- 2]exhibits the steps involved in the recommendation process.

![Figure: 2. Recommendation Process Steps](image-url)
[2] APPROACHES FOR RECOMMENDATION SYSTEMS

Recommendation systems should be efficient in predicting the items as per the user's interests accurately. The different techniques that are used in filtering the recommendations are shown in [Figure-3]

![Figure 3: Approaches for Recommender Systems](image)

[2.1] CONTENT-BASED APPROACH

Content is the foundation for building recommender applications. These recommendations are based on item description rather than similarity of other users. There are various contents available, such as wikis, polls, articles, blogs, classification terms, photos and videos. Metadata can be extracted by analyzing content. The procedure consists of creating tokens, normalizing the tokens, approving them, stemming them, and detecting phrases. [1]. The major idea of this approach is to recommend items to customer X similar to the user profile or previous items purchased or liked. [Figure-4] explains how the content-based recommendations are generated. The recommender system generates the user and item profiles from the web, where user profile consists of information regarding their ratings, likes etc and the item profile consists of a complete description of the items like actors, music director, a genre of the movie etc. The recommender systems take these as inputs and generate recommendations.

The advantages of content-based recommender systems are – information of other users is not required, ready to predict items to users with unique tastes. Ready to recommend new and hard to find items. Some of the disadvantages are – finding the appropriate features is hard, over specialization, cold start problem for new users, never recommends items outside user's content profile and scalability.
COLLABORATIVE FILTERING APPROACH

The word “Collaborative Filtering” was first coined by one of the developers of Tapestry[4]. In this approach, recommendations are drawn based on the interests of the crowd. The vital assumption of collaborative filtering is that if two users A and B rate ‘n’ items similarly or have similar behaviors then they rate or act on various items similarly[4]. This technique uses a data structure called “User-Item Matrix”. It consists of ratings given by the users for different items where these ratings usually scale from 1 through 5. Table 1 shows a simple movie database with some rows (called as records) that describes five movies and some columns (which are as attributes, fields, characteristics or variables) that describe the properties like Genre of the movie and its year of release.

Table 1 Movie Database

<table>
<thead>
<tr>
<th>ID</th>
<th>Movie Name</th>
<th>Genre</th>
<th>Year of release</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Terminator</td>
<td>Action, Science Fiction</td>
<td>1984</td>
</tr>
<tr>
<td>2</td>
<td>Matrix</td>
<td>Action, Science Fiction</td>
<td>1999</td>
</tr>
<tr>
<td>3</td>
<td>Final Destination</td>
<td>Horror, Thriller</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>Lord of the Rings</td>
<td>Adventure, Drama, Fantasy</td>
<td>2001</td>
</tr>
<tr>
<td>5</td>
<td>Avatar</td>
<td>Action, Adventure, Fantasy</td>
<td>2009</td>
</tr>
</tbody>
</table>

User-Item Matrix which consists of ratings given by different users for the movies mentioned in the movie database. Table 2 depicts the User-Item Matrix for the movie database.
Table 2 User-Item Matrix

<table>
<thead>
<tr>
<th>Movie</th>
<th>Terminator</th>
<th>Lord of Rings</th>
<th>Avatar</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U2</td>
<td>5</td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>U3</td>
<td>3</td>
<td></td>
<td>?</td>
<td>2</td>
</tr>
<tr>
<td>U4</td>
<td>2.5</td>
<td></td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>U5</td>
<td></td>
<td>4.5</td>
<td>4</td>
<td>3.5</td>
</tr>
<tr>
<td>U6</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Collaborative filtering is classified into two categories: Memory-based and Model-based.

[2.2.1] MEMORY-BASED COLLABORATIVE FILTERING

In this technique, complete or part of the user-item matrix is used to generate recommendations. This technique is also called as Neighborhood-based CF. Memory based CF is further classified into user-based CF and item-based CF. In user-based CF, the similarity is calculated between the users by differentiating their ratings on the same item. In the item-based CF, the similarity is calculated between items rather than users[2]. To generate recommendations, the neighbors of the active users are found by calculating the similarity measure and a weighted aggregate of those user’s ratings are computed. [Figure-5] describes the architecture of memory-based collaborative filtering.

![Figure:5. Architecture for memory based Collaborative filtering](image)
SIMILARITY MEASURES

There are many similarity computation methods like Jaccard coefficient, Kendall’s coefficient, Manhattan distance, Spearman’s correlation, Cosine, Adjusted Cosine, Euclidian distance, constrained Correlation, Mean Squared Difference etc among which Pearson’s correlation coefficient and cosine–based similarity are most commonly used.

Pearson’s correlation coefficient is a statistical approach which is used to measure to which extent the two variables are linearly related. For user-based CF, the Pearson’s correlation coefficient between user x and user y is given as (1)

$$w_{x,y} = \frac{\sum_{i \in U}(r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in U}(r_{x,i} - \bar{r}_x)^2 \sum_{i \in U}(r_{y,i} - \bar{r}_y)^2}}$$

where \(r_{x,i}\) is the rating given by user x on item i. \(r_{y,i}\) is the rating given by user y on item i. \(\bar{r}_x\) is the average ratings given by user x. \(\bar{r}_y\) is the average ratings given by user y and U is the set of all items in User-Item space.

For item–based CF, the Pearson’s correlation coefficient will be

$$w_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2 \sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}}$$

where \(r_{x,i}\) is the rating given by user x on item i. \(r_{y,i}\) is the rating given by user y on item i. \(\bar{r}_i\) is the average ratings on item i given by all users. \(\bar{r}_j\) is the average ratings on item j given by all users and U is the set of all users in User-Item space.

The other important similarity metric which is most commonly used is cosine similarity. It is a linear algebraic technique widely used in the fields of text mining and information retrieval. Documents are represented as vectors (i.e., instead of users or items, documents are used and instead of ratings, a frequency of words in the document bare used). It measures the similarity based on the angle formed between the vectors. For User–based CF, the similarity between users x and y is defined as

$$w_{x,y} = \cos(\hat{x},\hat{y}) = \frac{\hat{x} \cdot \hat{y}}{|x||y|} = \frac{\sum_{i \in I} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I} r_{x,i}^2} \sqrt{\sum_{i \in I} r_{y,i}^2}}$$

where \(r_{x,i}\) is the rating given by user x for item i, \(r_{y,i}\) is the rating given by user y for item i and I is the list of all items in the User-Item space.

For item–based CF, the similarity between users i and j is defined as

$$w_{i,j} = \cos(\hat{i},\hat{j}) = \frac{\hat{i} \cdot \hat{j}}{|i||j|} = \frac{\sum_{u \in U} r_{u,i} r_{u,j}}{\sqrt{\sum_{u \in U} r_{u,i}^2} \sqrt{\sum_{u \in U} r_{u,j}^2}}$$

where \(r_{u,i}\) is the rating given by user u for item i. \(r_{u,j}\) is the rating given by user u for item j and U is the list of all users in the User-Item space.

The important step in collaborative filtering is to obtain recommendations. So after calculating the similarity, nearest neighbors of the active user are chosen and a weighted aggregate of their ratings is used to generate predictions [4]. The weighted aggregate of other ratings is given as follows.

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U}(r_{x,i} - \bar{r}_i)w_{a,u}}{\sum_{u \in U}w_{a,u}}$$
where $P_{a,i}$ is the predicted rating for the active user an on item $i$, $r_{x,i}$ is the rating given by $x$ on item $i$, $w_{a,x}$ is the similarity coefficient between $a$ and $x$, $\bar{r}_x$ is the average of ratings given by user $x$.

For example, consider the user-item matrix in Table 2 and we want to find whether Avatar movie can be recommended to user 3. So first we calculate similarity between user 3 and other users and is as follows:

$w(1,2)=0.403072$, $w(1,3)=0.323575$, $w(2,3)=-0.41523$, $w(3,4)=-0.2325$, $w(3,5)=0.518545$, $w(3,6)=0.382336$ etc., From the above values, the highest similarity is between user 3 and user 5. Now we calculate the predicted rating for user 3 using weighted aggregate of other ratings and is given as follows:

$$P(3,3) = \bar{r}_3 + \frac{(r_{2,3} - \bar{r}_2)w_{3,2} + (r_{4,3} - \bar{r}_4)w_{3,4} + (r_{5,3} - \bar{r}_5)w_{3,5}}{|w_{3,2}| + |w_{3,4}| + |w_{3,5}|}$$

$$= 2 + \frac{(3.6667 - 3.5325 + 3.5325 - 3.5325 - 0.41523)}{|0.41523| + |-0.2325| + |0.518545|}$$

$$= 2 + \frac{0.679116}{1.166275} = 2.058229$$

### 2.2.2 MODEL BASED COLLABORATIVE FILTERING

The performance of the memory-based technique decreases when the user-item matrix becomes extremely sparse, which is very common on e-commerce sites. In model-based collaborative filtering, the model is generated from a sample of larger databases using machine learning and data mining algorithms. This pre-computed model is used to generate recommendations similar to memory-based CF. The model based collaborative filtering resolves the major problem of recommender systems called sparsity by employing dimensionality reduction techniques. Some of the dimensionality reduction techniques are Singular Value Decomposition (SVD) [44-45], Matrix Factorization (MF) [46-48], Principal Component Analysis (PCA), Latent Semantic Indexing (LSI), among which MF is rapidly used. Dimensionality reduction helps in improving the performance of the recommender systems. Widely used model-based techniques are Association rule mining [11], Bayesian classifiers [5-13], Decision trees [9,11,12], Clustering [8][21], Latent factor models (MF using SVD, ALS, Regularization) [27]. Some of the model-based CF techniques are based on the bio-inspired approaches like Neural networks [9,12,14-16], Fuzzy systems [17], Genetic algorithms [18-20], Slope One Method deals with sparsity and cold start problems. It is used to predict the vacant ratings wherever necessary [26]. Classification algorithms like Association rule mining, Decision trees, Naïve Bayesian classification, SVM etc are used if the user ratings are categorical. If the user-item matrix is of multi-class data then it is converted to binary class for simplicity. Regression models are used if the user ratings are numerical. [Figure-6] describes the architecture of Model-based Collaborative filtering.

### 2.2.3 HYBRID COLLABORATIVE FILTERING

This technique is used to improve the prediction accuracy by combining some of the advantages of both neighborhood based and model based collaborative filtering techniques. Neighbourhood based approach does not perform well with sparse data in the user-item matrix and it also suffers from scalability i.e., if the user-item matrix consists of more number of items or users, then the entire computation has to be done to generate a single recommendation. A
model-based approach is extremely complex to generate a model if the dataset consists of multiple parameters. The construction of the model takes a large amount of time and machine and user perspective are different, so if the assumptions of the model do not fit the data, results in wrong predictions. To avoid these problems hybrid collaborative filtering generates algorithms using models that are fast and easy to calculate[40].

The recommendations generated by this approach are more accurate compared to memory based and model based collaborative filtering algorithms. Paritosh Nagarnaik et al.[28] Uses a hybrid collaborative filtering approach to generate web page recommendations using pattern discover algorithms such as clustering and association rule mining. K. Yu, A. Schwaighofer et al. [29] combined memory based and model based techniques to generate predictions based on the user profiles and posterior distribution of user ratings. D. M. Pennock, E. Horvitz et al.[30] introduced Personality Diagnosis which selects the active user uniformly at random and calculates same personality type (neighbors for the active user) by adding Gaussian noise to his or her ratings.

![Architecture for Model-based Collaborative filtering](image)

**HYBRID RECOMMENDER SYSTEMS**

To overcome the individual disadvantages of Content-based RS and Collaborative RS, both are combined together to form a new Recommender system called Hybrid Recommender system. It uses the advantages of both the systems to optimize the predictions. There are multiple ways to build a hybrid recommender system. They are

1. Both the approaches are individually implemented and the results are combined in generating the recommendations.
2. Content-based techniques are used in Collaborative recommender systems.
3. Collaborative techniques are used in Content-based Recommender systems.
4. Both are combined together as a model.

[Table 3] depicts different combinations of to build a hybrid model.

The different hybridization methods are

**Weighted Hybrid**

It integrates the results of each technique and generates new recommendations. It linearly combines the predicted ratings. Initially, the predicted ratings are considered equal and gradually weights are added during the process of generating Top-N recommendations.
Tango[31] is an example of weighted hybrid recommender system which combines Content-based RS and Collaborative RS. For further examples refer (Pazzani1999), (Towle&Quinn 2000). The advantage of this techniques is that all the benefits of the recommendation systems are used during the process of generating recommendations.

**Cascade Hybrid**

It follows an iterative approach in generating the recommendations. Initially, one recommendation technique is used to generate a coarse list of predictions and again a second recommendation technique is used on this coarse list of predictions to generate a finer recommendations. The advantage of this technique is, it is optimal and tolerant to noise as low-priority recommendations are filtered at the first iteration and in the second iteration, the only high-level coarse list of predictions are processed to generate the final recommendations. Its disadvantage is, it adds complexity. EntreeC[13]-a restaurant recommender system, Fab are the examples of cascade hybridization.

<table>
<thead>
<tr>
<th>SNo</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CBRS → Recommendations → CFRS</td>
</tr>
<tr>
<td>2</td>
<td>CBRS / CFRS → Recommendations</td>
</tr>
<tr>
<td>3</td>
<td>CBRS / CFRS → Recommendations</td>
</tr>
<tr>
<td>4</td>
<td>CBRS / CFRS → Recommendations</td>
</tr>
</tbody>
</table>

**Switching Hybrid**

It uses the strategy of switching in between the recommendation techniques based on some criteria. DailyLearner is an example of switching technique which uses content and collaborative techniques where content-based recommendation technique is employed first and collaborative recommendation technique is implemented later when the result drawn from content based RS are with low confidence. The advantage of this approach is it uses all the benefits of recommendation techniques used. The disadvantage of this technique is the criterion has to be determined on which switching has to take place. It results in another level of parameterization.
**Mixed Hybrid**

It is a combination of more than one recommendation systems. The recommendations are drawn individually by each technique and the results are combined to generate final recommendations. Each item has multiple recommendations associated with it from different recommendation techniques. PTV – A television show recommender system, PickAFlick, ProfBuilder are some of the examples of mixed hybrid recommendation techniques.

**Feature combination Hybrid**

It is another hybrid approach which combines content and collaborative filters. The predicted rankings generated by the collaborative technique is taken as an extra feature and on this augmented dataset content based filter is employed to generate recommendations. Pipper is an example of feature combination hybrid technique which uses ratings of collaborative filters as a feature in the content-based system for movie recommendations. GroupLens is another example. The feature that is combined with the data set may not be always collaborative filters.

**Feature augmented Hybrid**

Feature augmented hybrids are superior to feature combination hybrids. It employs any technique to predict ratings or classify the dataset and this output is taken as input to the recommendation process. Libra is an example of feature augmentation which generates book recommendations using content-based system based on the data found in Amazon.com using a naive Bayes text classifier. GroupLens is another example. The feature that is combined with the data set may not be always collaborative filters.

**Meta–level Hybrid**

It is another type of combining the recommendation systems. The model generated by the first system is taken as input to the other. The difference between the feature augmentation and meta level is, in feature augmentation, the feature generated by the model is taken as input to the second whereas, in meta level hybrid, the entire model is taken as input to the second. Fab, LaboUr are the examples of meta-level hybrids.

[Table 4] describes the various approaches used by the recommendation systems.

**[3] EVALUATION METRICS**

To evaluate the performance of the recommendation systems and accuracy of the recommendations generated, a myriad evaluation metrics are used. Herlocker et al., broadly classified the evaluation metrics into predictive metrics and recommendation accuracy metrics.

**[3.1] Predictive metrics**

It measures the difference between the predicted ratings generated by the system and the real ratings. The most popular prediction metrics are Mean Absolute Error (MAE), Rooted Mean Squared Error (RMSE), Normalized Mean Absolute Error (NMAE) etc. Predictive metrics are also called as statistical accuracy metrics.

**MAE:** It is the most common and popularly used metric. It is the absolute difference between the predicted ratings and real ratings. It is defined as follows

\[
MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N}
\]

where N is the total number of ratings of all the users on the item i, p_i is the predicted rating and r_i is the real rating. Lower the MAE indicates that the recommendation engine is accurately
predicting the user ratings. \((p_i - r_i)\) represents the error in prediction.

**RMSE**: It has become more popular after its usage in Netflix Prize Competition. Willmott et al.\[38\] claimed that RMSE is more appropriate to represent model performance than the MAE when the error distribution is expected to be Gaussian\[39\]. It is defined as follows

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N}(p_i - r_i)^2}{N}}
\]  

(7)

Table 4 Classification of Recommendation Systems

<table>
<thead>
<tr>
<th>Recommendation approaches</th>
<th>Description</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content based RS</td>
<td>Recommendations are generated based on user and item features</td>
<td>ColdStart Problem Over Specialization</td>
</tr>
<tr>
<td>Memory-based Collaborative Filtering</td>
<td>It uses statistical methods to find the similar users and items. It is also called as Neighborhood-based CF. It is further categorized into User-based CF and Item-based CF</td>
<td>ColdStart Problem Sparsity Gray Sheep Shilling Attacks Overfitting Scalability Synonymity Trust</td>
</tr>
<tr>
<td>Model-based Collaborative Filtering</td>
<td>It uses machine learning and data mining algorithms to generate a model. This model is responsible for generating the recommendations.</td>
<td></td>
</tr>
<tr>
<td>Hybrid Collaborative Filtering</td>
<td>It combines both memory based and model based approaches.</td>
<td></td>
</tr>
<tr>
<td>Weighted Hybridization</td>
<td>The results of several recommendation techniques are integrated to generate a final recommendation. Weights are added linearly to individual results of each RS to draw a single recommendation.</td>
<td>Expensive Complex to implement</td>
</tr>
<tr>
<td>Switching Hybridization</td>
<td>The system switches between the recommendation techniques based on some criteria.</td>
<td></td>
</tr>
<tr>
<td>Cascade Hybridization</td>
<td>It is an iterative procedure where one recommendation technique refines the output of another recommendation technique.</td>
<td></td>
</tr>
<tr>
<td>Mixed Hybridization</td>
<td>Recommendations generated from individual techniques are mixed to generate the final recommendation.</td>
<td></td>
</tr>
</tbody>
</table>
where $N$ is the total number of ratings of all the users on item $i$. $p_i$ is the predicted rating and $r_i$ is the real rating. This metric puts more emphasis on larger absolute error because, on a 5-point scale, a 1-point error may not be perceptible by the user (items rated with either 4 or 5 points are good recommendations) whereas with a 4-point error, the algorithm could be recommending bad items[40] and the lower the RMSE is, the better the recommendation accuracy.

### [3.2] Recommendation accuracy metrics

These are further classified into classification accuracy metrics and rank accuracy metrics. These metrics are also called as decision support accuracy metrics.

#### [3.2.1] Classification accuracy metrics

Classification accuracy measures how well the recommendation system differentiates relevant (good) items from non-relevant (bad) items. Some of the classification accuracy metrics are precision, recall, F-measure.

**Precision:** It is the ratio of relevant recommended items to the total recommended items. It is defined as follows.

$$Precision = \frac{\text{Relevant Recommendations}}{\text{Total Recommendations}}$$

(8)

**Recall:** It is the ratio of relevant recommended items to the total relevant items. It is defined as follows

$$Recall = \frac{\text{Relevant Recommendations}}{\text{Total Relevant Items}}$$

(9)

It is preferable to have high precision and recall, but both the metrics are inversely proportional i.e., when precision increases, recall decreases and vice versa[40]. So most of the systems use another measure called f-measure.

**F-measure:** It is a combination (harmonic mean) of precision and recall and is defined as follows.

$$F - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

(10)

The best f-score is 1 and the worst is 0.

#### [3.2.2] Rank accuracy metrics

When the number of recommended items are large, then the users usually prefer the top-
N recommendations from the list. This creates more errors than to the items placed at the last of the recommendation list. Rank accuracy metrics considers this situation and it measures the ability of the recommendation algorithm to produce the recommendation list that matches with, how the user would have ordered the same list of items. For example, if the top ten list of items generated by the recommendation algorithm is relevant to the user and if the user does not feel that the top item in the list is good(relevant), then the rank accuracy metric score for that specific algorithm is low. Some of the rank accuracy metrics are ROC, half-life utility (HL) and discounted cumulative gain(DCG).

**ROC:** Swets[41,42] introduced ROC metric to the information retrieval community, which was very popular in signal detection theory. It is an alternative to precision and recall. It is a graphical representation of what extent the recommendation algorithm discriminates good(relevant) items with bad(irrelevant) items. It is also called as Relative Operating Characteristic. The area below the ROC curve is called as Swet’s A measure, which is used as a single metric of the system's ability to discriminate between good and bad items, independent of the search length[37].

**Half – lifeUtility:** It was proposed by Breese et al.[43]. It evaluates the utility of an ordered list of recommendations by the user. It is based on the fact that, the utility of the user is greater for the beginning list of items and it exponentially decreases going deeper into the list. It is defined as follows

\[
HF = \frac{\sum (r_{ai} - d, 0)}{2(\alpha - 1) - (i - 1)}
\]

where \(r_{ai}\) is the rating of the user a on item i of the recommended list, d is the default rating, and \(\alpha\) is the half-life. The half-life is the rank of the item on the list such that there is a 50% chance that the user will view that item[37].

**Discounted Cumulative gain(DCG):** It is defined as

\[
DCG = r_{ai} + \sum_{i=2}^{k} \frac{r_{ai}}{\log_{2}(i)}
\]

where \(r_{ai}\) represents the rating of the user a on item i of the recommended list, d is the default rating, and \(\alpha\) is the half-life. The half-life is the rank of the item on the list such that there is a 50% chance that the user will view that item[37] and k is the rank of the evaluated item.

[4] CONCLUSION

The review of various approaches like content-based, collaborative and hybrid alleviates the problem of information overload and opens new opportunities of retrieving personalized information to the users. This paper discussed the steps involved in recommender systems, the approaches for recommendation systems and highlighted their strengths and challenges with diverse kind of hybridization strategies used to improve their performances. Various learning algorithms used in generating recommendation models and evaluation metrics used in measuring the quality and performance of recommendation algorithms were discussed. This review will serve as a blue print to the researchers in improving the state of the art recommendation techniques besides developing new approaches.
REFERENCES


[26] Pu, WANG, Hong, Wu YE, A Personalized Recommendation Algorithm Combining Slope One Scheme and User Based Collaborative Filtering, In 2009 International Conference on Industrial and Information Systems (pp. 152-154)
[28] Paritosh Nagarnaik and Prof. A. Thomas: Survey on Recommendation System Methods. In IEEE SPONSORED 2ND INTERNATIONAL CONFERENCE ON ELECTRONICS AND COMMUNICATION SYSTEMS (ICECS 2015)
[39] T. Chai, R. R Draxler: Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature, Geo-scientific Model Development, 2014.
AUTHORS INTRODUCTION

V Lakshmi Chetana has obtained her MCA degree from Acharya Nagarjuna University and M.Tech in Computer Science and Engineering from JNTU Kakinada with distinctions. She is now working as Assistant Professor in the department of CSE at DVR & Dr. HS MIC College of Technology, Vijayawada. She has 8 yrs of experience in teaching. She has more than 10 international Journal and Conference papers. Her areas of interest in research are Big Data Analytics, Data Science, Artificial Intelligence and Machine Learning.

Dr. S.P Syed Ibrahim has received his B.E(CSE) degree from Bharathidasan University, M.E(CSE) degree from Anna University and PhD in CSE from Anna University. He is now working as a professor in the School of Computing Sciences and Engineering at VIT University, Chennai Campus and also coordinates Data Analytics Research Group. His research interests include Data Mining, Big Data Analytics. He has published more than 40 International Journal and Conference papers and completed five industrial consultancy works related to Big Data Analytics.

Dr. B Rama has obtained her PhD from Sri Padmavathi Mahila Viswa Vidyalayam, Tirupati. She has 5 years industrial and 8 years of teaching experience. She is now working as HoD in the Department of Informatics/Computer Science, University college Kakatiya University, Warangal. She is the recipient of best paper award for the paper “A Data mining perspective for forest fire prediction” presented in International Conference. She has more than 45 international journal and IEEE and Springer conference papers. She organized various workshops and delivered many guest lectures. Her research areas of interest are Data Mining, Artificial Intelligence and Big Data Analytics.

Dr P Kiran Sree has obtained his B.Tech degree in Computer Science & Engineering from JNTU Hyderabad and M.E in Computer Science & Engineering from Anna University with distinctions. He has obtained his Ph.D in Artificial Intelligence from Jawaharlal Nehru Technological University-Hyderabad. His interests include Parallel Algorithms, Artificial Intelligence, and Bioinformatics. His bibliography was listed in Marquis Who’s Who in the World, 28 Edition (2012), U.S.A. He is the recipient of Bharat Excellence Award from Dr GV Krishna Murthy, Former Election Commissioner of India in 2013. He worked as Principal I/C of the N.B.K.R. Institute of Science & Technology, the second oldest private engineering college of the state for two years (04/12/2009-02/12/2011). He is the editor in chief of international journal of Parallel and Cloud Computing Research (PCCR). He has published 52 research papers in international journals and conferences. He is recognized supervisor in the stream of Artificial Intelligence for guiding Ph.D scholars at VIT and KL universities.