Solution To Challenges In Recommendation Systems

Prof. Anagha Chaudhari[1], Arya Kale[2], Pranav Lad[3]
[1, 2, 3] Pimpri Chinchwad College of Engineering, Nigdi, Pune, Maharashtra

Abstract- *The WWW or Web has changed how we live and interact. Today, we use online search engines and other Internet technologies to find relevant information on almost any subject.*

One method to overcome these concerns is using a recommender system that suggests relevant things. This modification will cause information and cognitive overload for Web users seeking reliable information.

Videos, photos, items, articles, news, and books employ recommender systems. In daily life, recommendation systems should be built to deliver better suggestions with fewer defects, higher accuracy, and scalability. These systems search the Web for user recommendations based on their stated preferences or objective behavior. This research analyzes model-based and memory-based techniques to find the best Collaborative Filtering algorithm to improve recommendation system accuracy and scalability.

Keywords- Accuracy, Clustering, Comparative Analysis, Collaborative Filtering, FunkSVD, KNN, MAE, Matrix Factorization, Model-Based, Memory-based, Metrics Measures, Recommendation Systems, RMSE, Scalability, Singular Value Decomposition, SVD, SVD++,

I. INTRODUCTION

Recommender systems are valuable tools for filtering online content because of changing computer user behaviors, increasing personalization tendencies, and expanding internet availability.

The most advanced recommender systems are excellent at providing precise recommendations, but they also have a variety of flaws and issues, such as cold-start, scalability, accuracy, sparsity, etc. Due to the existing numerous techniques, selecting one for creating application-focused recommender systems becomes challenging.

Each strategy also possesses a distinct set of traits, advantages, and disadvantages, which raises new problems that must be addressed. Thanks to recent technological advancements and the popularity of online services, a significant amount of internet information may now be accessed more quickly.

The issue of online data overload is a result of recent developments in ubiquitous computing. This data avalanche makes it increasingly challenging to find relevant and useful stuff online. However, recent advancements in several techniques with lower computing requirements can now more effectively and quickly guide users to the necessary information.

A. Need for Recommendation Systems:

The growing volume of data makes common jobs and activities challenging. For instance, it is a routine and typical task to browse the web and look for interesting information or products. Huge amounts of data are increasing the noise on the internet, making it more difficult and time-consuming to select meaningful bits of information from all this noise.

The processing and management of this enormous amount of data also reveal the limitations of the systems, technologies, and tools that are now in use. New technologies are created, including Google's Map-Reduce and Yahoo's Hadoop.

In light of this, already-in-use systems have been modified to handle Big Data utilizing recently developed tools and technologies. The recommender system is one of these systems. Since the dawn of computing, there has been talked of using computers to recommend the best product to the user.

II. RECOMMENDATION APPROACHES

A. Types of Recommendation Systems:

1. Collaborative Filtering
2. Knowledge-based Systems
3. Hybrid Recommender Systems
4. Demographic Systems
5. Community-based Systems
6. Content based filtering Systems

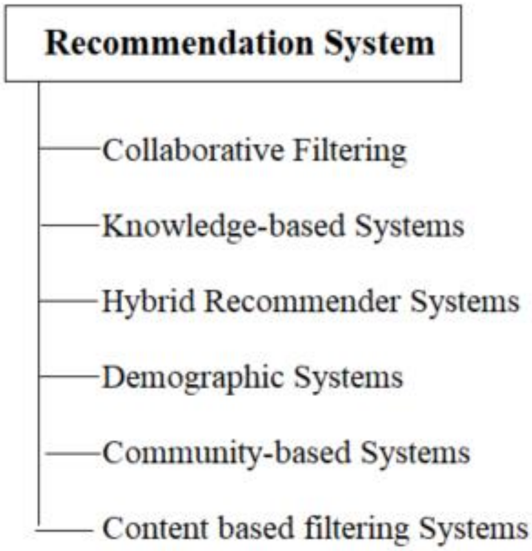

Fig. 1. Types of Recommendation System.

B. Collaborative Recommendation System (RS):

The most common and effective way to provide recommendations is through recommender systems, which provide suggestions for goods based on user collaboration. There are three types on which collaborative filtering is based on:

1. Latent factors
2. User to user similarity
3. Item to item collaborative filtering.

C. Challenges in Collaborative Recommendation system:

1. Limited Content Analysis
2. Over Specialization
3. Cold Start
4. Sparsity
5. Scalability
6. Accuracy

D. Introduction to Challenges:

a. Accuracy:

Collaborative filtering is one of the top recommenders commonly used in a variety of e-commerce platforms. In order to predict what will be recommended, CF-based recommendation systems consider the similarity value of the ratings offered by the top-n comparable neighbors. Similarity measures have a substantial impact on CF accuracy.

In CF-based recommendation systems, low prediction accuracy is brought on by an inaccurate top-n comparable neighbor of the target user. However, there are a number of issues with traditional similarity measures when it comes to calculating the top-n neighbors over different time frames.

Since the neighbors' tastes and interests are likely to change over time, The recommender system's accuracy tends to be different from that of traditional recommendation techniques when it uses neighborhood-based collaborative filtering. The calculated collection of neighbors does not necessarily represent the ideal neighborhood at any one time as a result.

b. Scalability:

When dealing with a high number of people and millions of unique objects, RSs get more difficult as they work on vast datasets.

High scalability goods are required since several systems must respond quickly to online requests and provide suggestions to every user based on their past ratings and purchases.

III. TYPES OF COLLABORATIVE RS

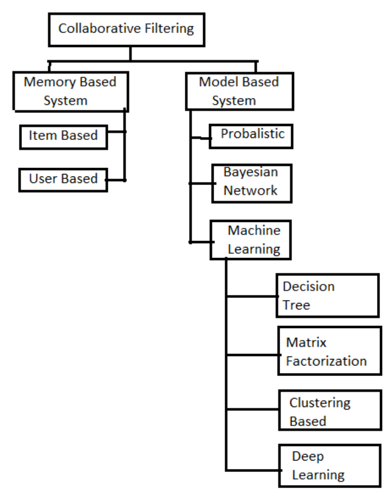

Fig. 2. Types of Collaborative Recommendation System

A. Memory-Based Collaborative Filtering:

Memory-based CF algorithms use all or part of a user factor database to generate predictions. Each user belongs to a community of people with similar interests. Predicting a new user's preferences for a new product can be done by determining the new user's (or active users) called neighbors.

Neighbor-based CF algorithms and memory-based CF are common techniques. Here are the steps: Create a prediction of active users using a simple weighted average or a weighted average of all ratings for a user or item. Receive a user or item. Calculate the similarity or weight (wi, j) between two users or two objects i and j reflecting distance, correlation or weight. To get the top N suggestions, first compute the similarity, then find the k closest users or items (neighborhoods), and finally aggregate the neighbors to find the top N most popular item should be offered as a recommendation.

1. Memory Based CF Types:

a. User based Collaborative Filtering:

Recommendations which are based on user-user similarity Collaborative Filtering. The system is based on a matrix with x users and y items, where x users preferences for y items are recorded. The closest neighbors in the system are found when a new user requests a suggestion and is referred to as the target user. A forecast is made for the target user on this item considering their previous ratings on the particular item. In other words, recommended products are ones that users who share the target user's taste in products are favored.

b. Item-Based Collaborative Filtering:

Here, we search for people who have given similar ratings to a variety of goods before using these individuals to determine the rating of the missing item. The initial stage in creating the model is to identify similarities between each pair of items. There are various methods for determining how similar two item pairs are. Utilizing cosine similarity is one of the most widely used techniques.

2. Memory Based CF:

a. Advantages:

i. Easy implementation
ii. Entry of new data can be done easily
iii. The content of the items being recommended is not considered
iv. Handle co-rated items easily

b. Disadvantages:

i. Reliant on ratings from people
ii. When data is sparse, performance suffers.
iii. Cannot suggest new users or products.
iv. Have a restricted ability to scale for large datasets

B. Model Based Collaborative Filtering:

Develop and build models (machine learning, data mining algorithms) to enable systems to recognize complex patterns in training data, and use these patterns to inform information in collaborative filtering tasks performed in the real world. You can make predictions based on and run test data. For this reason, model based CF algorithms are a possible solution to the problems faced by memory-based CF algorithms. Some examples are clustering models, dependency networks, and Bayesian Networks. If the user ratings are numeric, you can use regression models and SVD methods. If the user ratings in categories, the classification algorithm can be used as a CF model.

1. Type of Model Based:

a. Bayesian:

A compact, adaptable, and understandable representation of a joint probability distribution is a Bayesian Network. Given that directed acyclic networks allow for the representation of causal relationships between variables, it is also a useful tool in knowledge discovery. A Bayesian Network is often trained using data.

A probabilistic graphical model where a random variable is represented by each node and each edge represents the conditional probability for the related random variables is called Bayesian Network (BN). It is used to represent information about an uncertain domain. BNs are also known as Bayes nets or belief networks.

b. Probabilistic:

Because it gives a foundation for understanding what learning is, probabilistic modeling has become one of the most important theoretical and practical methods for creating machines that learn from experience-based data. Machine learning, Robotics, Cognitive science, and Artificial Intelligence, the probabilistic framework which are included in scientific analysis— explains how to represent and handle uncertainty regarding models and predictions—plays a key role.

c. Machine Learning:

i. Clustering Techniques:

Among a big collection of objects in RS, to create a group based on resemblance, structures, and cluster analysis, or unsupervised learning, is used to find patterns. Clustering can be used in recommender systems to improve the diversity, consistency, and reliability of suggestions, and to address issues such as user preference matrices and the sparseness of preference data that changes over time.

ii. Decision Tree:

The decision tree is a potent tool for selecting an option from a range of several different options. It is utilized in RS to determine and forecast the missing preferences of users.

iii. Matrix Factorization:

By multiplying two different types of entities, matrix factorization can produce latent characteristics. Matrix Factorization (MF) is used in collaborative filtering to determine the connection between the entities of items and users. We would like to forecast user ratings of store items using the input of user ratings so that users can receive recommendations based on the prediction.

iv. Deep Learning:

Data scientists are increasingly turning away from more conventional machine learning techniques and toward highly expressive deep learning models to improve the quality of their suggestions as the growth in the number of data available to fuel recommender systems accelerates rapidly. Deep learning for recommendations can be broadly divided into two phases: training and inference. During the training phase, the model is taught to forecast the likelihood of user-item interactions (generate a preference score) by being shown examples of historical interactions (or lack thereof) between users and things.

2. Model Based CF:

a. Advantages:

i. The accuracy and scalability is addressed properly
ii. Prediction performance is improved
iii. Give recommendations a logical justification.

b. Disadvantages:

i. Building models is expensive.
ii. Strive to balance scalability and prediction performance but better than memory based.
iii. Ignore information that would help dimensionality reduction approaches.

III. ADVANTAGES OF MODEL BASED OVER MEMORY BASED

The offline evaluation shows that, in terms of recommendation accuracy, model-based accuracy exceeds memory-based accuracy.

In terms of calculation time, model-based computations are typically 10 times faster than memory-based ones.

Model-based product suggestions are fast therefore compute more quickly than memory-based ones.

The respondents claimed that although memory-based computation time was slower than model-based within this set of data, it was still tolerable.

When making relevant recommendations based on the typical number of relevant products that the user perceives, model-based is more successful than memory-based.

A. Result Show Memory Based Over Model Based:

[1] MovieLens dataset 100k is used with Dataset 1 and Dataset 2. Dataset 1 has 45% trained dataset and 55% test dataset as shown in Fig.3, Fig.4 and Fig.5 to compare between model based and memory based CF.

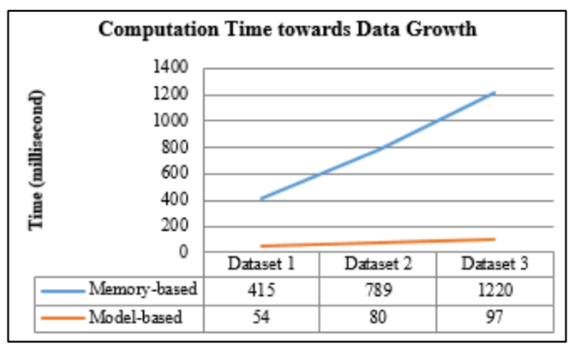

Fig. 3. Comparison of time towards data growth for Memory Based and Model-Based.

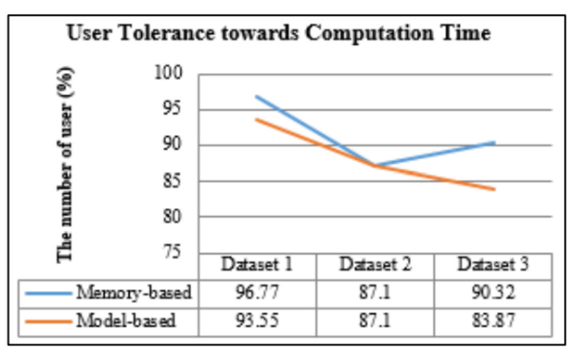

Fig.4. Comparison of user tolerance towards computation time for Memory Based and Model Based.

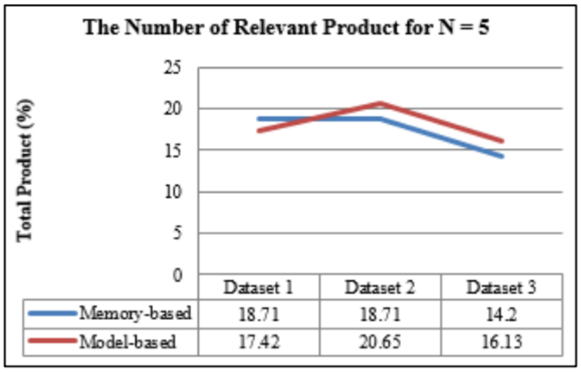

Fig. 5. Comparison of number of relevant products for Memory Based and Model Based.

B. Prediction Metrics MAE and RMSE :

MAE (mean absolute error) and RMSE (Root Mean Square Error) are the two most widely used metrics in this category as shown in equation (1) and equation (2) respectively. These measures are intended to gauge how closely your prediction matches your actual value numerically. While RMSE penalizes greater mistakes, MAE penalizes smaller errors equally.

1. RMSE:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(p_i - \hat{q}_i)^2}{N}} \quad (1)$$

2. MAE:

$$MAE = \frac{\sum_{i=1}^{N}|p_i - \hat{q}_i|}{N} \quad (2)$$

IV. LITERATURE REVIEW

A. Accuracy Literature Review:

In paper [1], This paper talks about the user being given a recommendation based on the tastes of their neighbors who have rated the same goods similarly and are comparable to them. Therefore, finding the clustering of neighbors accurately is essential for providing a customized and accurate recommendation. The typical CF has trouble locating the top-n neighbors of the target user due to the exponential growth of users.

Due to the target user's shifting interests over time, standard similarity metrics have difficulties determining the top-n comparable people. The accuracy of the prediction is impacted by this. This study solves the issue by computing the top n neighbors on an annual basis. This method ensures that the list of top-n neighbors is always updated to reflect any changes in the target user's neighbors' preferences. As a result, compared to the current traditional CF algorithm, the suggested CF offers a notable improvement in prediction accuracy.

In paper [2], the accuracy of the conventional Collaborative Filtering (CF) approach under the sparse data issue was improved by the new similarity method provided in this paper. Based on an analysis of user preferences that have a strong association to the user's choice, CF gives the user the things they need. However, the technique used to detect neighbors has an impact on the accuracy.

The most popular techniques for determining correlations between users focus on the rating of only co-rated items and include pearson correlation coefficient and Cosine measurements. As a result, these approaches are unable to deal with the scalability and sparsity.

In paper [3], the two specific recommender algorithms discussed in this study are item-based collaborative filtering, which makes use of item similarity, and FunkSVD, a matrix factorization approach. In this study, the prediction accuracy of the algorithms on small and large datasets will be compared.

In this study, cross-validation of the algorithms is used to gather data allegedly capable of resolving questions about the algorithms' correctness. Results from the testing suggested that the FunkSVD algorithm might be more accurate than the item-based collaborative filtering method, but further analysis is needed to draw a firm conclusion.

In paper [4], the author talked about certain implicit information processing techniques. Also recommend a new weighted similarity metric that takes the relative relevance of the items into account. To further enhance the conventional neighborhood models' ability to make accurate recommendations, a new grading technique is also suggested. Matrix Factorization (MF) models typically produce superior predictions for explicit Collaborative Filtering (CF) problems when compared to neighborhood models.

We add a few modifications to the basic MF models to assess implicit data. In a supermarket data set, our new MF models greatly outperform the neighborhood models in terms of suggestion accuracy. We also suggest a hybrid MF model that incorporates data on user or item features. We are hopeful that the hybrid approach will be able to address the inherent cold start issue with pure CF machines. However, more research is required to confirm its efficiency. In our trials, the hybrid model increases the suggestion accuracy.

In paper [5], this research compares a number of collaborative filtering algorithms, both traditional and state-of-the-art, in various experimental settings.

The most accurate techniques are typically those that use Matrix Factorization. With the exception of extremely scarce circumstances, regularized SVD, PMF, and its variations exhibit the best performance in terms of MAE and RMSE.

Better precision trades off against other elements like low volatility in accuracy, computing efficiency, memory usage, and fewer changeable parameters. That is, the more accurate algorithms have a bigger variance in accuracy, are less computationally efficient, have more changeable parameters, and depend more on the size and density of the dataset. Matrix Factorization techniques are the most suitable when computing efficiency is less critical.

In paper [6], it claims that Singular Value Decomposition methods outperform FunkSVD algorithms by a wide amount in terms of speed. Due to the lack of an optimization step, SVD is the fastest algorithm available, whereas FunkSVD has a computational delay.

B. Scalability Literature Review:

In paper [7], in this study, we look at the similarities and differences between memory based and model based collaborative filtering. Both of these models are used by the system being discussed to predict user preferences for specific items. The results show that, in terms of overall effectiveness, SVD based collaborative filtering outperforms KNN item based CF by a large margin. When applied to CFs based on KNN approach, the SVD based CF approach has proven effective in addressing the scalability and sparsity problems that arise. When compared to the output of the other approaches, the recommended method produced the lowest RMSE and MAE values for rating predictions.

In paper [8], it is clear that the FunkSVD algorithm is more adaptable because the Item based collaborative filtering algorithm was given a good opportunity to scale due to the dataset's characteristics and still underperformed it. We can draw the conclusion that when scaling from small to large datasets, the FunkSVD algorithm appears to be more accurate than the item-based collaborative filtering algorithm.

In paper [9], the primary goal of this research is to examine several collaborative filtering algorithms while taking the issue of scalability into account. Numerous algorithms, including cluster based, item-based, and context-based ones, are investigated. Study of four cluster-based CF algorithms reveals that, because it exhibits lower MAE than conventional CF, the collaborative filtering recommendation algorithm based on user-based clustering and item-based clustering is the best approach. Additionally, it has improved quality and is more accurate and scalable than conventional CF.

In paper [10], the Recommendation System is thoroughly examined in this research, including the various recommendation methodologies, related problems, and information retrieval strategies. In this paper, difficulties, methodologies, and algorithms are discussed in relation to a comparison of recommendation systems.

In paper [11], the merits and cons of each RS strategy are listed along with an overview of the various types of recommendation approaches based on user preferences, ratings, domain expertise, user demographics, and user context. We have suggested a method for recommending movies in this paper that uses collaborative filtering and SVD++. The suggested method is contrasted with well-known machine learning methods including co-clustering, singular value decomposition, and k closest neighbor (K-NN). In this research, we proposed a movie recommendation system based on the fusion of collaborative filtering and SVD++ method. The generated result has a lower RMSE (0.9201) and MAE(0.7219) error rate.

In paper [12], as all pairs of items are compared using typical CF methods, this becomes a bottleneck in such systems. All partners use a cooperative strategy rather than employing brute force.Algorithms of memory based CF, like the item-based Pearson Correlation CF algorithm, as well as model based CF algorithms, like clustering algorithms is examined in this paper which address the scalability issue by grouping users into a more manageable and highly similar cluster and using this cluster and nearby partitions for predictions, and can achieve satisfactory scalability. Scalability issues can be resolved and fast, high-quality recommendations can be produced using dimensionality reduction techniques like SVD, although doing so requires expensive matrix factorization operations.

In paper [13], This study improves upon previous work in the field of recommender systems by developing an advanced SVD-based recommender system using Apache Hadoop and Spark. This effort addresses the scalability issue by utilizing Hadoop and its helpful features. It also showed that utilizing a reliable imputation method as a preprocessing step before using SVD on the user-item matrix may produce excellent outcomes.

In paper [14], With an emphasis on a variety of applications, including books, movies, and products, this study seeks to conduct a thorough evaluation of numerous recent advances in the field of recommender systems. The different uses of each recommender system are initially examined. Then, a simulation platform is created, datasets are gathered, performance metrics specific to each contribution are assessed, and notes are made. Finally, an algorithmic analysis of various recommender systems is carried out, and a taxonomy that takes into account the different elements needed to develop an effective recommender system is framed.

IV. ALGORITHMIC SURVEY

A. Improving Accuracy

In paper [1], the following uses an optimized K Means clustering algorithm with thorough execution steps:

1. Groups data into conceptually sound and practical categories
2. The group's objects are comparable to one another.
3. Distinct from items in another group
4. In some circumstances, this serves as a springboard for further objectives like data summary.

In paper [2], to address the sparsity issue and enhance the accuracy of recommendations, a novel similarity method is developed that is based on user preferences globally.

Process stages:

1. To address this problem, a similarity algorithm was devised that took into account user preferences globally.

2. Based on user evaluations, the overall preference is deduced to represent their preferences.

3. These preferences are then entered as data into the suggested similarity measure. As a result, even when users do not share goods, the correlation between them is still determined.

4. The results also demonstrate that using two factors—fairness and the proportion of co-rated items—in the suggested similarity to increase the accuracy of the recommendations has a beneficial impact.

In paper [3], in this report, two distinct recommender algorithms— item based collaborative filtering, which makes use of item similarity, and FunkSVD, a matrix factorization algorithm—are studied.

Singular Value Decomposition (SVD), a matrix factorization technique, is used by the algorithm known as FunkSVD. One matrix can be split into two smaller ones via Singular Value Decomposition. The two matrices that are produced represent users and generalized things, respectively. Predictions are generated by computing the dot product of one or more element vectors with the user vector. The resulting value indicates the most relevant suggestions. An illustration of the reduction of a matrix into two other matrices via a Singular Value Decomposition. A factorization input variable determines the number of features. When it comes to singular value decomposition, characteristics are typically referred to as latent factors.

In paper [4], this report studies implementation of matrix factorization model based techniques to improve accuracy and metric measures like MAE, RMSE.

A new rating strategy is proposed to further improve the recommendation accuracy of conventional neighborhood models. When compared with neighborhood models, Hybrid Matrix Factorization (MF) models usually generate better predictions for explicit Collaborative Filtering (CF) problems.

In paper [6], it states the fact that Singular Value Decomposition algorithms are faster than FunkSVD algorithms by a significant margin. SVD is the fastest of all algorithms because there is no optimization process, whereas FunkSVD has computational delay.

B. Improving scalability

In paper [7,13], Singular Value Decomposition (SVD) algorithm:

SVD can be used to reduce the dimensionality of the feature space. The SVD formula for an x * y matrix A is given in equation (3)

$$\text{SVD (A)} = \text{U } \Sigma \text{VT} \tag{3}$$

U and V are matrices of size m and n respectively, from which we get a unique, non-negative, m by an orthogonal matrix.

A given matrix is called an orthogonal matrix if it has the same rank as the identity matrix. The diagonal elements of Σ (σ1, σ2, σ3, …σn) are called the singular values of the matrix A.

The single values are typically listed in decreasing order in. The left and right singular vectors, respectively, are the column vectors of U and V. SVD is used in several significant applications and has a number of beneficial characteristics. A low-rank approximation of matrix A is one such example. According to the truncated SVD of rank k, as in equation (4) value of SVD is:

$$\text{SVD (AK)} = \text{UK } \Sigma\text{KVKT} \quad (4)$$

In paper [8],

Funk SVD Algorithm:

Build 2 matrices U and V^T, respectively a matrix of users by the number of latent factors chosen and a matrix of these same latent factors by items and fill these matrices with random numbers.

Rating matrices are often sparse, making it difficult to apply SVD directly and necessitating preprocessing to fill in the missing values. Pre-processing is not required for Funk SVD.

$$\hat{R} = U_K {V_K}^T \quad (5)$$

$$\hat{R} = \{\hat{r}_{ci}\}_{n_C \times n_I} \quad (6)$$

where
is a matrix approximating R as in equation (6). $\hat{R}$ has no missing value.

$$UK = \hat{R} = \{\hat{r}_{ci}\}_{n_C \times n_I} \quad (7)$$

are factor matrices which cropped to K features.

Train each feature to converge using gradient descent with regularization.

The algorithm estimates R and determines the error at the start of each loop.

The factors are updated until the difference of RMSE between the current and previous loop is smaller than the stopping criterion.

$$eci = rci - \hat{r}ci$$
$$uck \leftarrow uck + \lambda(ecivik - \gamma uck)$$
$$vik \leftarrow vik + \lambda(eciuck - \gamma vik)$$

where λ is the learning rate, γ is the regularization term and $k \in \{1, . . ,K\}$.

A prediction of unknown ratings of target customers can be derived as a matrix operation.

$$\hat{R}T = U_{K^T} V_{K^T}$$

Where, $\hat{R}, T = \{\hat{r}ti\}nT \times nI$

Rating matrix of the target customers. Since VK is a matrix about items and there is no change in the list of the items even a new customer. Therefore, VK is fixed when making the prediction. UK is a matrix about customers and they have been changed to the target customers. Thus, optimization of

$UKT = \{\mathrm{u}tk\}nT \times K$ proceeds again.

$$eti = rti - \hat{r}ti$$
$$utk \leftarrow utk + \lambda(etivik - \gamma utk)$$

KNN (K Nearest Neighbour) algorithm:

1. Decide on the neighbors' K-numbers.
2. Calculate the Euclidean distance between K neighbors.
3. Based on the determined Euclidean distance, select the K closest neighbors.
4. Count the total number of records in each of these k adjacent groups.
5. Place the most recent data point in the cluster with the most neighbors.
6. Our model is complete.

In paper [9], K means Clustering Technique algorithm:

1. Site files, log files, and user information are the key sources of information access, which is then analyzed using the clustering technique.
2. A comparison of patterns' similarities are made.
3. The recommendation is created using a clustering method to the transaction mode cluster.

The clustering algorithm utilized by the researchers is K-means. The center-based approach is used to identify the neighbors

Proposed Approach to improve Accuracy and Scalability:

For Accuracy :

SVD Algorithm[17]

1. SVD will be used as the main algorithm to reduce accuracy problems in Recommendation Systems.

2. The maximum accuracy is typically found in matrix factorization-based techniques. In particular, regularized SVD outperforms MAE and RMSE, with the exception of extremely sparse conditions.

3. One way to solve the accuracy problem caused by CF is to use a latent factor model to account for similarities between users and things.The recommendation problem is essentially what we want to transform into an optimization problem. We can think of it as how well we forecast a user's rating of an item.

4. Root Mean Square Error is one such statistic. Performance improves when RMSE decreases. We will temporarily disregard the unseen goods because we are unsure of their ranking. More specifically, minimization of the root-mean-square error (RMSE) for known values of the utility matrix. The below equation (8) shows how to obtain a small RMSE using singular value decomposition (SVD).

$$X=USV^T \quad (8)$$

5. X represents the utility matrix and the left singular matrix L represents the connections between users and latent components. There is a right singular UxS matrix for similarities between elements and latent factors, and a diagonal V transposed matrix for the strengths of each latent factor. It's a broad term that can describe a wide variety of characteristics and ideas. The genre of music can be a hidden factor. SVD reduces the dimension of the utility matrix by removing latent elements. Basically, we create an r-dimensional latent space in which each person and object can be placed. You can now more easily see similarities between people and things, and your ability to understand their interactions has improved as shown in Fig. 6.

The figure below shows an example of this idea:

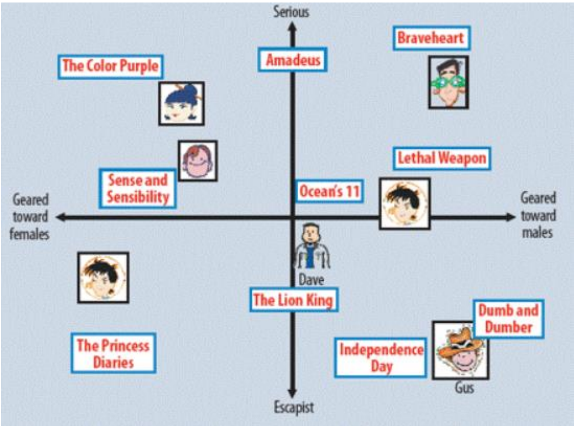

Fig.6. Single Value Decomposition

6. SVD is widely used in dimensionality reduction Sum Sum of Squared Error (SSE) due to its excellent property of having a least squared error reconstruction sum. In the given below equation (9), X is replaced by A and S by Σ.

$$\min_{U,V,\Sigma} \sum_{ij\in A} \left(A_{ij} - [U\Sigma V^{T}]_{ij}\right)^2 \quad (9)$$

It can be seen that RMSE and SSE are monotonic to each other. Therefore, the smaller the SSE, the smaller the RMSE. SVD is known to reduce.

RMSE because it also reduces SSE, another desirable property. SVD is therefore a powerful method for solving this optimization problem. Multiplying U, Σ, and T is sufficient to infer a user's reaction to an unknown object.

For Scalability :

SVD++ Algorithm

In its most basic form, a collaborative recommender system uses the following stages to recommend the top N items:

(1) Identify similarities between items and users;
(2) Create Neighborhoods.
(3) Score and evaluate matrix prediction.

The proposed model consists of the following five parts. More details are provided below.
One, data preparation.

(1) Data Preprocessing
(2) Cosine Similarity Evaluation
(3) Prediction Matrix Lookup
(4) SVD++ Approach Application

(5) Top N Film Recommendations

User ratings of various movies serve as the initial input for the rating matrix. Then we found that cosine similarity provides similarity between movies and users by applying formula (10). Evaluate the prediction matrix with the formula. (11).Top N candidates are recommended after applying the SVD++ method

$$Cosine = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sum_{i=1}^{n} A_i^2 B_i^2} \quad (10)$$

Here,

A, B : Features of movie / item
(.) : Dot Product
|| || : Length of matrix

$$P_{u,i} = \frac{\sum_{t \varepsilon N} \left(sim(i,t) \times R_{u,t} \right)}{\sum_{t \varepsilon N} (sim(i,t))} \quad (11)$$

IV. COMPARATIVE STUDY

A. Analysis of the Current and Proposed System (accuracy)

TABLE I. Comparison between existing and proposed system for accuracy

Existing System	**Proposed System**
Existing systems produce lower accuracy in Recommendation Systems.	Matrix Factorization method SVD has high accuracy than most of the methods in Collaborative filtering and neighborhood based algorithms
The response time is comparable to that of others involving processing and computation time.	SVD algorithm computer results in less time than many traditional and model based algorithms
Top N neighbors are calculated to computer similarity and provide recommendations. Over time, the accuracy of the advice based on the previous group of comparable users tends to diminish.	SVD can be used even if it is not a square matrix. It decomposes the rating matrix into three matrices to characterize both customers and items. Data is compressed and noise is removed according to dimension reduction.
K-Means is an unsupervised learning approach for machine learning that may be used to classify data into a variety of different categories.	By reducing the space dimension from N to K (where KN), SVD is a matrix factorization approach that can reduce the number of features in a dataset.

B. Analysis of the Current and Proposed System (Scalability):

Synonyms are often used, which gives model-based matrix factorization techniques a competitive advantage over memory-based CF systems.

TABLE II. Comparison between existing and proposed system for scalability

Existing System	**Proposed System**
KNN finds the neighbors of one user through computing the similarities with all the other users.	Matrix factorization approaches, in addition to lowering dimensionality, may aid in discovering latent characteristics that explain observed ratings.
The computing speed, simplicity, and clarity of the k-nearest neighbor's method are all advantages. Classification and regression issues benefit from their higher precision, which they provide well.	It is common practice for recommender systems to use a group of collaborative filtering methods known as matrix factorization. To function, matrix factorization methods reduce the user-item matrix to matrices with fewer dimensions.

It applies Singular Value Decomposition to the user-item matrix, decomposing it to 3 smaller matrices, intending to learn about the users' and things' characteristics. While SVD is scalable, it incurs high costs due to the need for matrix factorization at each stage.	As an attempt to combine the strengths of latent and neighborhood models, SVD++ is a model with many advantages. Using SVD++ trained with collaborative filtering, the RMS error, and MAE decrease to 0.92101 and 0.7219 respectively.

VI. RESULTS

Scalability comparisons:

[11] In Fig. 7 and Fig. 8 MovieLens dataset 100k is used to compare the MAE and RMSE values of different algorithms.

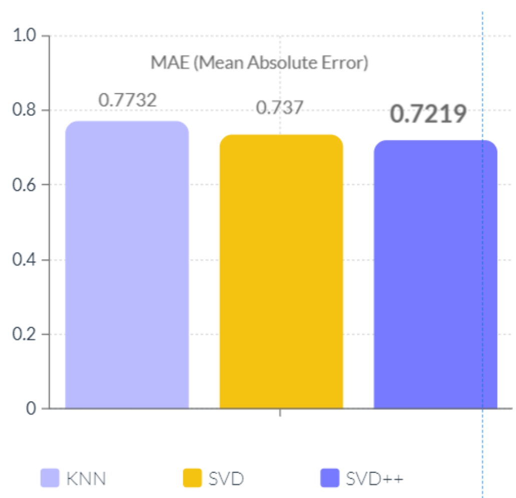

Fig. 7 (MAE) 100k Movie Lens Data Set

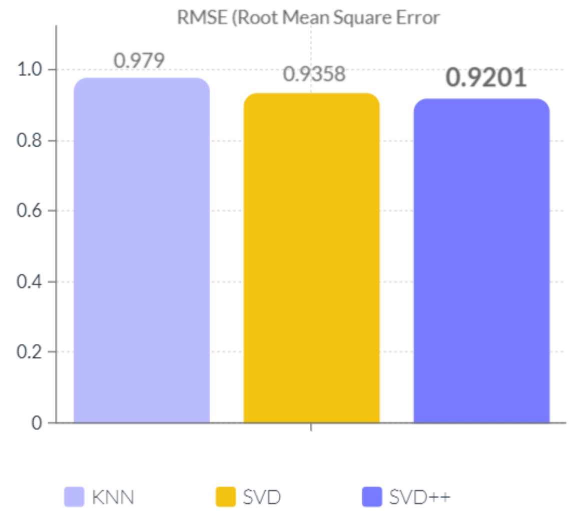

Fig.8 (RMSE) 100k Movie Lens Data Set

Accuracy Comparisons:

[6] Algorithms like SVD AND FunkSVD time is computed in seconds to compare the fastest algorithm

TABLE III. Computation time of algorithms in seconds

Algorithm	**SVD**	Funk SVD
Computation time	**0.045**	20.711

VII. CONCLUSION

In this paper we have studied the collaborative filtering recommendation system and its limitations like cold start, accuracy, sparsity, and scalability. This paper discusses the accuracy and scalability issues in collaborative filtering and model-based algorithms.

Table 3 For accuracy issues in Collaborative Recommendation system algorithms such as SVD and Funk SVD are compared for computation time which showed SVD (0.045s) is significantly faster than FunkSVD (20.711s).

From Fig.7 and Fig.8, SVD, SVD++, and KNN are analyzed for the dataset [15] and its metric measures such as RMSE and MAE are compared. From this analysis best suited to the matrix factorization algorithm, SVD++ is proven to reduce scalability issues in Collaborative Recommendation Systems to provide scalable recommendations. The RMSE (0.921) and MAE (0.7219) produced by SVD++ collaborative filtering are the lowest among known Recommendation Systems algorithms.

REFERENCES

[1] Pradeep Kumar Singh,Pijush Kanti Dutta Pramanik,Showmik Setta,Prasenjit Choudhury "Improving the Accuracy of Collaborative Filtering Based Recommendations by Considering the Temporal Variance of Top-N Neighbors"-2019

[2] Hael Al-bashiri, Mansoor Abdullateef Abdulgabber, Awanis Romli "A Developed Collaborative Filtering Similarity Method to Improve the Accuracy of Recommendations under Data Sparsity"-2018

[3] Safir Najafi, Ziad Salam "Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems"-2016

[4] Xinxin Bai, Jin Dong "Recommendation Accuracy "-2012

[5] Joonseok Lee, Mingxuan Sun, Guy Lebanon "A Comparative Study of Collaborative Filtering Algorithms"-2012

[6] "Comparison of Customer Personalized Item Recommendation Algorithms Based on E-Commerce Data" - 2020

[7] Rouhia M. Sallam,Mahmoud Hussein, Hamdy M. Mousa "An Enhanced Collaborative Filtering-based Approach for Recommender Systems"- 2020

[8] SAFIR NAJAFI ZIAD SALAM "Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems"-2016

[9] Reena Pagare,Shalmali .A. Patil "Study of Collaborative Filtering Recommendation Algorithm Scalability Issue"-2013

[10] Pradeep Kumar Singh,Pijush Kanti Dutta Pramanik, Avick Kumar Dey,Prasenjit Choudhury"Recommender Systems: An Overview, Research Trends, and Future Directions"-2021

[11] Taushif Anwar, Uma Vijayasundaram "Comparative study of recommender system approaches and movie recommendation using collaborative filtering"-2021

[12] András Péter Kolbert "A Scalable Recommendation system using Latent Topics and Alternating Least Squares Techniques"-2017

[13] Khadija A. Almohsen,Huda AI-Jobori "Recommender Systems in Light of Big Data"-2015

[14] Deepjyoti Roy, Mala Dutta "A systematic review and research perspective on recommender systems"-2022

[15] www.grouplens.org/datasets/movielens/100k

[16] www.grouplens.org/datasets/movielens/1m/

[17] www.easychair.org

[18] Medium.com blog, anime recommendation with matrix factorization