A Unified Solution To Challenges In Recommendation Systems - Cold Start & Data Sparsity

Prof. Anagha Chaudhari[1], Mugdha Kshirsagar[2], Rugved Kulkarni[3]
[1, 2, 3] Dept of Computer Engineering
[1, 2, 3] Pimpri Chinchwad College of Engineering, Nigdi, Pune, Maharashtra

Abstract- *By utilizing a subset of information filtering systems called recommender systems, it is feasible to prevent information overload, review a large amount of data, and choose data that is particular to each user. Several Internet juggernauts, such as Amazon.com, Facebook, and Google, have long utilized recommender systems. In order to suggest new things that the user would find interesting, these systems look at user profiles, online behavior, and, if relevant, purchase history. One of the most important sub-domains of information retrieval is collaborative filtering recommender systems. These systems' primary focus is on identifying users who share the target user's interests and gathering those users' recommendations.The two major issues that our research is focusing on are Data Sparsity and Cold Start. The aim is to identify a unified solution that solves the cold start and mitigates the data Sparsity issue.*

Keywords- (KNN), (SVD), - approach, Cold-Start, Collaborative Cosine Data Decomposition Factorization, Filtering(CF), K-Nearest Matrix Measures model-based Neighbours Recommendation Similarity similarity. Singular Sparsity, systems, Value

I. INTRODUCTION

Due to evolving computer user habits, rising inclinations toward personalization, and growing internet accessibility, recommender systems are one of the best tools for organizing online content[1].

The most sophisticated recommender systems excel at making accurate recommendations, but they also have several drawbacks and problems, including cold-start, sparsity, etc. It can be challenging to decide which technique to utilize when developing application-focused recommender systems because there are so many of them. Each technique also has a unique set of characteristics, benefits, and drawbacks, which creates new issues that need to be resolved.Because of the proliferation of online services and the consequent advances in technology, it is now possible to get a substantial amount of knowledge through the internet in a shorter amount of time[1]. Because of recent advancements in ubiquitous computing, the problem of online data overload has arisen[1]. Finding meaningful and helpful content online is becoming more and more difficult as a result of this data storm.

However, several contemporary methods that require less processing can now more easily and swiftly direct consumers to the required content. The two serious problems that CF inherently faces in recommendation systems are the focus of this study. The first major issue is “COLD-START” and the second is “DATA SPARSITY”.

The Cold-Start problem occurs when the system is unable to connect users and products for whom it lacks appropriate data.The term "cold start" describes the challenge of producing precise suggestions for customers who only evaluated a small number of things and are new to the system.The Recommendation Systems utilize quite huge datasets in real-world applications. Because of this, the matrix used for CF is very scanty, which affects how well CF systems perform when making predictions or recommendations. It also happens when a customer uses a certain product but chooses not to give it a rating. Other times, users won't rank things they don't recognize. **Data Sparsity** leads to generating unreasonable recommendations for those users who provide no ratings[2].Due to the fact that, on average, only a tiny percentage of things were rated by active users, data sparsity relates to the challenge of locating enough trustworthy similar users[2].Our objective is to evaluate, review, and contrast the solutions that have been implemented to mitigate the sparsity and cold-start issue in order to provide recommendations that are more specific and pertinent.[8]. Collaborative filtering techniques used two different approaches, they are memory-based and model-based approaches[6]. In this section, we will be concentrating on the many methods that make use of model-based collaborative filtering algorithms such as Singular Value Decomposition, or SVD, Singular Value Decomposition plus plus, or SVD++, and K-Nearest Neighbours.

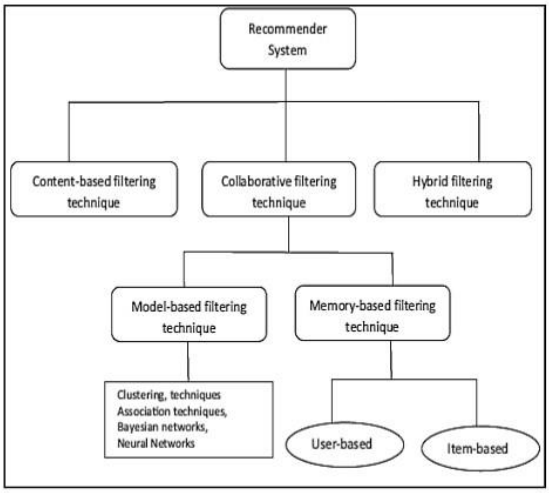
II. ALGORITHMIC SURVEY

By utilizing a subset of information filtering systems called recommender systems, it is feasible to prevent information overload, review a large amount of data, and choose data that is particular to each user. Several Internet

juggernauts, such as Amazon.com, Facebook, and Google, have long utilized recommender systems. In order to suggest new things that the user would find interesting, these systems look at user profiles, online behavior, and, if relevant, purchase history. As a result, recommender systems are widely used in a variety of commercial and academic domains[4]. Various approaches and model have been built so far to mitigate sparsity and cold-start issue. Like, The social community is segmented using relationship mining and analysis of previous user history data. Then, an ontology decision model built on the static data of users and the sub-community bases its recommendations for new users on their static ontology information.[11].In the absence of a good initial profile, the recommendations are analogous to random probes. However, if they are not chosen wisely, both poor recommendations and an excessive number of recommendations have the potential to discourage a user from continuing to use the service.[15].

A. Recommendation Approaches

1. Collaborative Filtering technique
2. Content-based Filtering method
3. Knowledge-based Systems
4. Hybrid Recommender Systems
5. Demographic Systems



One of the essential elements of information retrieval is the use of collaborative filtering recommender systems. By gathering preferences or taste data from numerous users, collaborative filtering is a technique for predicting a user's interests[14]. Collaborative filtering involves very large data sets and these sets are sparse because we can't expect users interacted with all the items that we have. Also, if they interacted with all of them, there would be nothing to recommend. All collaborative filtering methods are ways to analyze the rating matrix from different perspectives[12].

Collaborative filtering (CF) methods are classified into two categories:

1. Memory-based
2. Model-based

B. Memory-Based Collaborative Filtering:

Recommendations are generated using this neighbourhood based collaborative filtering by making use of the user and item data that is available to it. Statistical methods are utilised by the system in order to locate a group of individuals who have conducted transactions in the past that are analogous to those carried out by the person actively using the system.

C. Model-Based Collaborative Filtering:

Few models can enable the system to identify patterns based on data and their statistics. Hence based on this technique, wiser predictions can be generated for the recommendation engine. Model-based algorithms are a great solution to the drawbacks of memory-based algorithms. Machine learning(ML) and deep learning(DL) concepts are used in model-based approaches to make systems think smartly and give more accurate output.

Model-based collaborative filtering techniques are:

1. **Using Bayesian Networks**
2. **Machine learning algorithms**
3. **Matrix Factorisation**
4. **Deep Learning approaches**

Matrix factorization can be broken down into its most fundamental idea, which is to predict an individual user's ranking over a set of items based on similarities between the users and the items.[13]

D. KNN (K-Nearest Neighbours)

KNN is an abbreviation for the K closest neighbour algorithm, which describes what the user matrix is. The algorithm is also referred to by its name, which is KNN. Finding missing data can be done in a number of different methods, all of which can be accomplished with the help of this tactic. When there is a limited quantity of information contained in the matrix, it is helpful to use this method

throughout the process of data discovery since it can help you make the most of the information that is available. Those who are new to the organisation and do not have any prior experience working there can benefit from the suggestions made, which is another way in which this activity is useful. This is an example of the instructional strategy known as "directed learning," which you can read more about here. The supervised learning method is the building block that the machine learning algorithm known as K-Nearest Neighbor uses to construct its foundation. In addition to this, it is one of the machine learning algorithms that is the easiest to grasp due to its simplistic nature.

Using the K-NN approach, the newly discovered example is placed in the category that is most comparable to the pre-existing categories that are available for selection. This placement was decided upon on the basis of the premise that the newly added case and the data are comparable to the cases that are already available to us at our disposal.

Following the maintenance of all of the data that was acquired in the past, the K-NN method is then used to assign a category to a new data item based on the similarities that they have with the other data items. This occurs after all of the data that was previously gathered has been preserved. This is done on the basis of the similarities that they share with the other elements of the data. This demonstrates that utilising the K-NN method enables the new data to be sorted into an appropriate category in a timely manner while retaining a high level of accuracy. This is made possible by the utilisation of the K-NN technique. Using the K-NN method will allow you to achieve this goal successfully.

The K-NN method can be used to solve problems involving regression in addition to classification; nevertheless, the majority of the time, it is utilised to handle problems needing classification. K-NN can also be used to tackle problems requiring regression.

E. Singular Value Decomposition(SVD)

Singular Value Decomposition is a method that has its roots in linear algebra and is applied to problems in order to reduce the total number of dimensions that are involved in the matter at hand. The method is sometimes referred to by its abbreviated form, SVD, and is sometimes known by its full name, Singular Value Decomposition. The field of machine learning is only one of the many different domains where it has been successfully applied and utilised. In addition to these, there are a great many other fields. By changing the space dimension from N to K (where K is equal to N), the SVD matrix factorization strategy is able to cut down on the overall number of features that are contained within a dataset. This is accomplished by shifting the space dimension from N to K, which in turn modifies the value of N. The SVD can be found within the context of the recommender system. When it is put into operation, the SVD performs the function of collaborative filtering. It employs a matrix format, with each row representing a different person and each column representing a different object. The rows represent the users, while the columns represent the objects. The comments and scores that end users have provided on a wide range of products serve as the building blocks for this matrix's many cells and rows. When we use SVD, we look for a matrix that has fewer dimensions and takes up less space. The SVD of a matrix with the dimensions m by n takes the form:

$$SVD(A) = U\sum V^T$$

…Equation 1

The application of the SVD method is particularly widespread in the field of data science.[5].

F. Singular Value Decomposition plus plus (SVD++)

The most important piece of information for recommendation systems is the user-item matrix. It's also the most important thing to know. [Here's what I mean:] MF is a good way to predict ratings before they have been given in the context of collaborative filtering. Because MF uses what other users have said, this is the case. This is because MF looks at information from both the past and the present. The MF method is used to factorise a sparse matrix, which leads to the discovery of two latent factor matrices: the user matrix, which represents the features of the user (that is, the level of preference of a user for each factor), and the item matrix, which represents the properties of the object. "Factor matrices" is the name for these two tables. The user matrix and the item matrix are both called by their names. One of these matrices is called the "user matrix," and its job is to show what the user is like. The other matrix is called the "item matrix," and it shows what the qualities of an object are. The user matrix and the item matrix each have their own names: the user matrix and the item matrix, respectively. People talk about both of these matrices by their names. The user matrix shows what the person doing the looking at is like, and the item matrix shows what the thing being looked at is like. Both of the matrices are used together. The next step is to guess the missing ratings by taking the inner product of these two factor matrices. One way to make the model's predictions more accurate is to use data about how different users and goods have their own biases. Funk The style called SVD++ comes from the fact that you can add to what SVD has already done.

This model also goes by the name "SVD derivative model," which is another name for it. A "SVD derivative model" is another name for this model. By using implicit feedback data, the SVD++ model can make suggestions that are both more useful and more efficient. This makes it more likely that the suggestions will be taken into account. The SVD++ derivative model's accuracy as a forecasting tool is significantly improved with the addition of implicit feedback data [7].

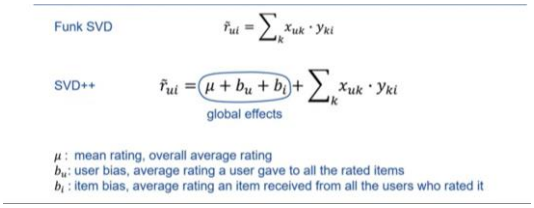


Fig-1 Funk SVD and SVD++

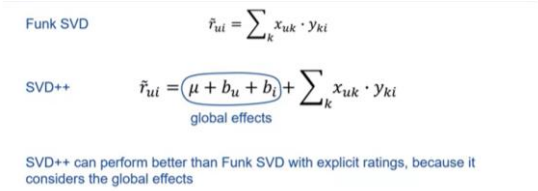


Fig-2 Funk SVD and SVD++ conclusion

III. EVALUATION METRICS USED IN THE COMPARATIVE ANALYSIS

A.Prediction accuracy metrics (MAE, RMSE):

Mean absolute Error and RMSE are the two most widely used metrics. These metrics are intended to gauge how closely your prediction matches your actual value numerically. While RMSE aims at greater mistakes, MAE aims at smaller errors equally. To determine the root-mean-square error, first determine the residual, which is the difference between the prediction and the actual value of each data point. Next, determine the norm, mean, and square root of each data point (RMSE). Because it requires and makes use of real measurements at each data point that is forecasted, RMSE is frequently utilized in applications that involve supervised learning. Root mean square error can be expressed as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

…Equation 2

The average of all absolute errors is **Mean Absolute Error**(MAE). The formula is:

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |x_i - x|$$

…Equation 3

IV. IMPLEMENTATION

A. Python Library Digram

Surprise library is the easiest way to use python Scikit for recommender engines. Python's Scikit Surprise is a tool for developing and analyzing recommender systems that work with explicit rating data[16].

Python package used: Surprise library algorithm

5. Base class = AlgoBase
6. Sub classes = SVD, KNN, SVD++

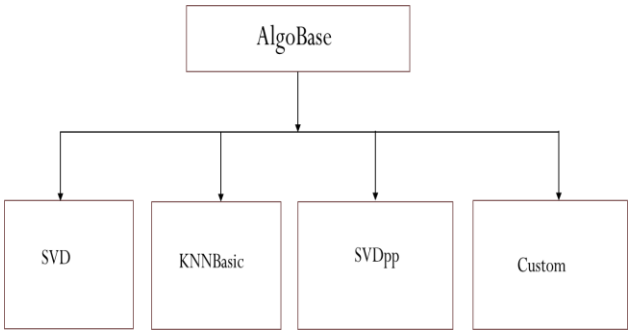


Fig 3. Surprise library class structure

Dataset used is Movielens. The MovieLens website, which is a movie recommendation service, is the source of the ratings that are included in this dataset. There are five different versions available: "25m", "latest-small," "100k," "1," and "20m"[10].

The most recent stable version of the MovieLens dataset is "25m." For research purposes, it is recommended.

"latest-small" refers to a small portion of the most recent MovieLens dataset. GroupLens continuously modifies and updates it.

The MovieLens datasets' earliest form is "100k." This little dataset contains demographic information.
"1m": This is the biggest MovieLens dataset with demographic information.

Alongside the "1m" dataset, the "20m" dataset is one of the MovieLens datasets that sees the highest use in scholarly publications.

Users have the option of viewing only the movie data for each version by appending the suffix "-movies" to the version number (for example, "25m-movies"), or they can view the ratings data coupled with the movie data (and users data in the 1m and 100k datasets) by appending the "-ratings" suffix to the version number (e.g. "25m-ratings").

Movielens dataset latestsmall[10] datasets are loaded using the surprise python library. The setup is installation. An installation of Python and R programming languages for scientific computing called Anaconda is installed. Its goal is to make package management and deployment easier. A Sypder environment is installed and the code is run on the same.

B. Work plan and methodology

1. Importing movielens latest small user dataset
2. Implementing a collaborative recommendation system on the dataset with existing issues present in the collaborative recommendation system.
3. Resolving and improving the following issues in collaborative recommendation:
    a. Accuracy
    b. Cold Start
    c. Scalability
    d. Sparsity
4. Combining solutions to these issues to generate an efficient, unified algorithm to tackle all the problems.
5. Calculating metrics of evaluating such as RMSE and MAE.

V. RESULTS

A. Results of the KNN vs SVD algorithm are stated below in the tabular format:

Table 1. Result of KNN vs SVD

Algorithms	**RMSE**	**MAE**
KNN	**0.9961**	**0.7711**
SVD	**0.9039**	**0.6984**

Results in Graphical format:

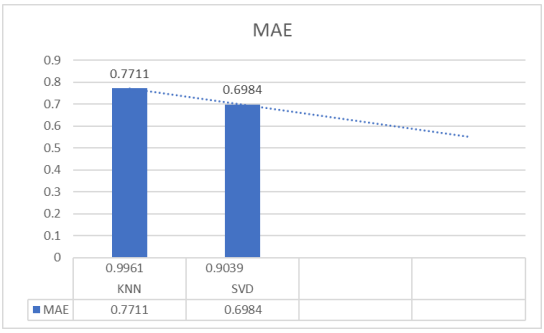


Fig 10. Graph 1

B.Results of the SVD and SVD++ algorithm are stated below in the tabular format:

Table 2. Result of SVD vs SVD++

Algorithms	**RMSE**	**MAE**
SVD	**0.9039**	**0.6984**
SVD++	**0.8943**	**0.6887**

Results in Graphical format:

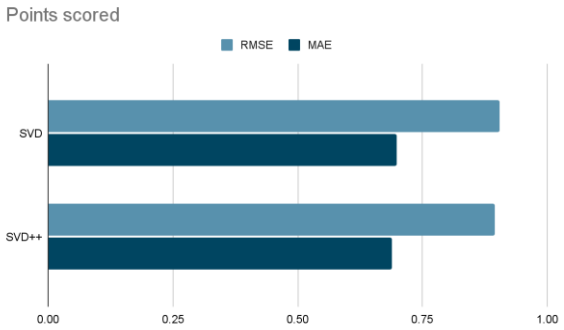


Fig 11. Graph 2

VI. CONCLUSION

Due to evolving computer user habits, rising inclinations toward personalization, and growing internet accessibility, recommender systems are efficient tool to categorise the online data. The most sophisticated recommender systems excel at making accurate recommendations, but they also have a number of drawbacks and problems, including scalability, cold-start, sparsity, etc. Although the problem of a recommender system's cold start has existed for some time, substantial advancements in artificial intelligence and data analytics have led to the creation of numerous remedies. We are utilising a model-based collaborative filtering technique that uses a description architecture drawn from the database to forecast user activity.

We have studied and applied the SVD (Singular Value Decomposition) and KNN model-based approaches in this section (K Nearest neighbors). Based on their RMSE measurements, we found that the optimal collaborative filtering approach is SVD because it offers the least amount of error. Similar to this, one of the key problems with collaborative filtering algorithms is data sparsity. Further implementation work revealed that the SVD++ algorithm gave the lowest RMSE and MAE values when compared to the SVD algorithm. Therefore, based on RMSE and MAE values, we can say that Singular Value Decomposition plus plus (SVD++) is the best strategy for somewhat resolving cold-start and data sparsity difficulties while producing more accurate forecasts.

REFERENCES

[1] Pradeep Kumar Singh, Pijush Kanti Dutta Pramanik, Avick Kumar Dey, Prasenjit Choudhury, "Recommender Systems: An Overview, Research Trends, and Future Directions", January 2021, International Journal of Business and Systems Research

[2] Guibing Guo, Nanyang Technological University, Singapore, "Improving the Performance of Recommender Systems by Alleviating the Data Sparsity and Cold Start Problems", Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.

[3] Vijaysinh Lendave, "Cold-Start Problem in Recommender Systems and its Mitigation Techniques",analyticsindiamag,published on september 26,2021.

[4] Fu Jie Tey, Tin-Yu Wu, Chiao-Ling Lin & Jiann-Liang Chen, "Accuracy improvements for cold-start recommendation problem using indirect relations in social networks", Journal of Big Data volume 8, Article number: 98 (2021).

[5] Zhengzheng Xian, Qiliang Li, Gai Li, and Lei Li, "New Collaborative Filtering Algorithms Based on SVD++ and Differential Privacy", Hindawi Mathematical Problems in Engineering Volume 2017, Article ID 1975719,1 School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, China 2 Guangdong University of Finance, Guangzhou, Guangdong, China 3 Shunde Polytechnic, Foshan, Guangdong, China.

[6] Nouhaila Idrissi, Ahmed Zellou, "A systematic literature review of sparsity issues in recommender systems", Social Network Analysis and Mining, Volume 10, Article number: 15 (2020).

[7] Yancheng Jia; Changhua Zhang; Qinghua Lu; Peng Wang. Users' brands' preferences are based on SVD++ in recommender systems. Published in: 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA).

[8] Atisha Sachan , Vineet Richhariya, "Reduction of Data Sparsity in Collaborative Filtering based on Fuzzy Inference Rules" International Journal of Advanced Computer Research (ISSN (print):2249-7277 ISSN (online):2277-7970) Volume-3 Number-2 Issue-10 June-2013.

[9] YiBo Chen, "Solving the Sparsity Problem in Recommender Systems Using Association Retrieval", Computer School of Wuhan University, Wuhan, Hubei, China chenyibo8224@yahoo.com.cn ChanLe Wu, Ming Xie and Xiaojun Guo Computer School of Wuhan University, Wuhan, Hubei, China National Engineering Research Center for Multimedia Software, Wuhan, China.

[10] https://www.tensorflow.org/datasets/catalog/movielens/

[11] Chen Meng, Yang Cheng, Chen Jiechao, Yi Peng, "A Method to Solve Cold-Start Problem in Recommendation System based on Social Network Sub-community and Ontology Decision Model",2013, the Authors. Published by Atlantis Press.

[12] Melih Kacaman, "Matrix Factorization For Recommendation Systems
" , Published in Towards Data Science.

[13] Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017, August). Deep matrix factorization models for recommender systems. In IJCAI (Vol. 17, pp. 3203–3209).

[14] En.wikipedia.org. 2022. Collaborative filtering — Wikipedia. Available at: https://en.wikipedia.org/wiki/Collaborative_filtering

[15] Laks Lakshmanan,Senjuti Basu Roy, "Combating the Cold Start User Problem in Model Based Collaborative Filtering", published in ResearchGate,February 2017.

[16] Surprise.readthedocs.org 2015. Surprise' documentation — Readthedocs. Available at: https://surprise.readthedocs.io/en/stable/