I.J. Information Technology and Computer Science, **2023, 1, 21-29**
Published Online on February 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2023.01.03

Evaluating the Scalability of Matrix Factorization and Neighborhood Based Recommender Systems

Nikita Taneja
MRU/CSE/Faridabad, Haryana, 121004
E-mail: nikita.taneja@gmail.com

Hardeo Kumar Thakur*
MRU/CSE/Faridabad, Haryana, 121004
E-mail: hardeokumar@gmail.com
ORCID iD: https://orcid.org/ 0000-0002-2954-1308
*Corresponding Author

Received: 29 July 2022; Revised: 07 November 2022; Accepted: 01 December 2022; Published: 08 February 2023

Abstract: Recommendation Systems are everywhere, from offline shopping malls to major e-commerce websites, all use recommendation systems to enhance customer experience and grow profit. With a growing customer base, the requirement to store their interest, behavior and respond accordingly requires plenty of scalability. Thus, it is very important for companies to select a scalable recommender system, which can provide the recommendations not just accurately but with low latency as well. This paper focuses on the comparison between the four methods KMeans, KNN, SVD, and SVD++ to find out the better algorithm in terms of scalability. We have analyzed the methods on different parameters i.e., Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Precision, Recall and Running Time (Scalability). Results are elaborated such that selection becomes quite easy depending upon the user requirements.

Index Terms: Recommendation, Singular Value Decomposition (SVD), SVD++ K- Nearest Neighbor (KNN), K-Means.

1. Introduction

Recently companies are showing increasing interest in recommender systems to enhance user experience as well as to increase sales, as customized recommendations provide an extra dimension to the user experience. So, to enhance the user experience, recommender systems focus on user interest patterns of products so as to give personalized recommendations. Consequently, major e-commerce platforms now-a-days have already incorporated recommender systems as a significant part of their websites. There are broadly two approaches to deploy recommender systems [1-4], Content filtering approach and Collaborative filtering approach.

To characterize the properties of each person or product, the content filtering technique develops a profile. A movie profile, for example, may include information about its type, featuring, rating, etc.

Instead of creating explicit profiles, there is another technique that depends only on historical user behavior, such as previous purchases or product reviews and this method is termed Collaborative filtering. Collaborative filtering finds new user-item relationships by analyzing user affiliations and product interdependencies. This method has the benefit of being domain doubter yet also being able to handle data items that are normally indescribable and hard to define using content filtering [2].

Collaborative filtering is primarily classified into two approaches known as the neighborhood and latent factor approach [3]. The computation of associations between things or, alternatively, between users, is the focus of neighborhood approaches. The item-oriented technique assesses a fondness of the user for an item based on the same user's assessments of "neighboring" items [2, 5]. Other items that receive comparable ratings when evaluated by the same user are referred to as a product's neighbors. E.g., If a user likes a comic genre, he or she will give similar ratings to other comic movies meaning the likelihood of that user to like other comic genres is higher than the rest. The Latent approach, on the other hand, defines ratings by defining both products and users based on characteristics deduced from rating patterns. The deduced features in movies could evaluate some characteristics like fiction against comedy, action, drama, animation. These factors evaluate a user experience through a score on the associated movie factor.

Latent factor models are best implemented on matrix factorization. Matrix factorization assigns vectors of factors to both items and users deduced from item rating patterns [2,5]. In this method the foundation of recommendation is a high

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

degree of association between item and user factors. These approaches have become famous due to their capacity to scale and anticipate accuracy. Furthermore, they provide a great deal of versatility when it comes to simulating various real-life scenarios. There is also a Hybrid approach to implement Recommendation Systems [4], which is the blend of above-stated approaches.

1.1. Need and Contribution of the Study

Recommender System work is to respond as per user's actions and interest. With growing user's gradually, the demand for a recommender system to keep each customer engaged is also rising. So, to keep every customer engaged and store the user behavior and interest is a challenge. To respond accordingly with ever growing customers within seconds, a recommender system needs to be scalable. As the user base grows the time required to provide recommendations to each new user will also rise. Thus, it's very important to select a scalable recommender system which can provide the recommendations not just as accurately but with low latency as well.

As there are two major ways to implement recommender systems, this study focuses on Matrix Factorization and Neighborhood methods. In this paper, we are addressing the concern identifying scalable yet accurate algorithms. The work is trying to find a method that is not only better in terms of providing speedy outcomes, but it should generate them with a reasonable level of accuracy as well. To do so, we are performing a comparison between four notable algorithms K-Means, KNN which are neighborhood search-based methods and SVD, SVD++ which are Matrix Factorization based Methods. The work will be comparing them in terms of the Precision, Recall MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and time. The work will additionally validate the algorithms using k-fold validations.

Rest of this paper is further organized in five Sections. In the first Section we have discussed the Literature Review. The next Section discusses the approaches which can be used to implement the recommendation systems. Then is the dataset and the experimental setup. Further Section discusses the results, and the last section is the conclusion of the paper.

2. Related Works

The paper in [2] explains a basic matrix factorization model. As per the model, user and objects are mapped to a joint latent space of the dimension of f and user-item relations are defined as a inner product of the given space. The members of vector p_u for a given user u- examine the level of significance given user has in the items. The consequential dot product, $q_i^t p_u$, gives the interaction between user$u$ and item i . In paper [6] the Deep Structure Semantic Model initially maps the query and the documents to a shared lower semantic space. The relevance of a query to each document is then computed using cosine similarity between the query and document's low-dimensional vectors for web search ranking. Because they are only interested in the standard topN recommendation problem, they solely employ observed ratings and comments. They presented a deep neural network architecture to project users and items into a latent structured space using this matrix Y as the input. In [7] They proposed a quick (semi)-positive MF technique, which approximates features by employing optimistic values for the users and the given items. A transductive version of Factorization Model is also offered, which improves prediction accuracy by using data from test instances (like the ratings which have been given for various products by the users). They have described an incremental variation of MF as well as a hybrid MF-neighbor-based technique, both of which are aimed at increasing efficiency. In [8] They've presented a Simultaneously Incremental MF approach, which combines multiple streams into a particular model, which is then incrementally updated to track data changes over time. The collaborative filtering and data fusion by MF are the foundations of this method. In [9] they have used SVM at various stages first for classification, then for pattern recognition, voice feature extraction and collectively for recognizing the six target emotions. They have used all these factors to form recommendation systems based on emotional features. In [10] they have performed a recommendation system for online bookstores in which they have used KNN algorithm to address the sparsity problem, to find missing values and finally KNN for recommendation. In [11] they have addressed the problem of varying taste and interest of users with time using the K-means algorithm. They proposed a new technique founded on time correlation coefficient and an improved K-means along with cuckoo search. In the next section we will discuss the methodology used.

3. Methodology Used

There are various ways to implement the Neighborhood Methods and the Matrix Factorization methods. In this manuscript, KNN and K-Means are chosen as neighborhood methods and for Matrix Factorization SVD and SVD++are chosen. Each one of them is explained in the next subsection.

3.1. SVD

A recommendation based on SVD is performed by SVD reconstructing the evaluation matrix. This reconstruction is achieved through factorizing the original evaluation matrix, sinking the matrix to a lower ranked matrix then re-constructing evaluation matrices as mentioned in [12]. This is the same as Probabilistic MF when no baselines are used. The prediction $\hat{r}_{ui}$ is given by the Equation. 1:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (1)$$

The factors p_u and the bias b_u are assumed to be zero, if user μ is unknown. The same is for i item with b_i and q_i. SVD minimize the following regularized squared error, to estimate all the unknown using Equation 2,

$$\sum_{r_{ui}}^{R_{train}}(r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2) \quad (2)$$

A very straightforward stochastic gradient descent is used to perform the minimization as in Equation 3:

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u) b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i) p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u) q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i) \quad (3)$$

in the Equation 3 s$e_{ui} = r_{ui} - \hat{r}_{ui}$, λ is learning rate andγ is regularization term. All these steps are repeated for n epochs times over all the training set ratings.

3.2. SVD++

The implicit feedback information is used by the SVD++ matrix factorization model as in [13]. Implicit feedback, on the other hand, refers to any information about a user's rental or purchase history that might help them determine their preferences.

The prediction $\hat{r}_{ui}$ is given by Equation 4:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T(p_u + |I_u|^{-\frac{1}{2}} \sum_j^{I_u} y_j) \quad (4)$$

In the above equations y_j refers to a new set of item factors that represent implicit ratings. Here a user u rating an item$j$, independent of the rated value, is referred to as an implicit rating.

The factors p_u and bias b_u, are assumed to be zero, if user u is unknown. The same is applicable on item ii with b_i, $q_i$and y_i

3.3. K-Nearest Neighbor

KNN as in [14] is a supervised learning technique and is used for classification and regression. To classify a new data point, this algorithm saves all the data points and classifies new points based on similarity index by putting that new point into the most similar saved category. It is also known as the lazy learner algorithm because instead of learning immediately from training data, it learns at the time of classification. The algorithm is as follows

- Store all the data points and choose a similarity measure as mentioned in [15].
- To classify a point D, we need to set a value of K which are the nearest data points.
- Now calculate the similarity measure between point D to K viz calculating the distance among all the selected nearest neighbors.
- Now it will choose the Top K where distance is minimum and classify the data point D to that category.

3.4. KMeans

KMeans an unsupervised approach is generally used in both data mining and pattern recognition. The foundation of this algorithm is to minimize square-error and error. To find out the optimized outcome, KMeans tries to find K divisions to fulfill a certain criterion. This algorithm in the first step, picks some points to signify the initial cluster centroid as in [16,17]; in the second step, it gathers the remaining sample points to their centroid points through the method of minimum distance (say euclidean distance), this way algorithm generates the initial classification. If the categorization is incorrect, the algorithm will change the centroid and iterate until the classification is correct.

4. Experimental Setup

To validate our analysis, we have conducted our experiment on Movielens dataset in Python 3.8. Movielens is a public dataset and is accessible online containing 100,000 user ratings provided by around 943 users for 1682 movie-items [17,18]. The Movie lens dataset is then divided in many files, out of which 80% is used for training and 20% is used for testing data and validation. Items are categorized into nineteen different genres like comedy, horror, romantic, animation, action etc. The user information is available in the "u.user" file and movie information in the u-item file. We have compared four algorithms SVD, SVD++, KNN and K-Means using the Prediction Recall, Precision MAE and RMSE and Scalability using K-Cross validation Method.

Cross-validation is a statistical approach for assessing as mentioned in [18] and comparing training algorithms that partition data into two sections: first partition is used for training a model and the second partition can be validate the data. In the work randomized K-fold validation was used where the data is initially divided into k sized folds for cross-validation. Now for training and the validation iterations size of k of is taken and every iteration can be fold of the data different from other validation sets and the rest k-1 folds are being utilized for learning.

For our algorithms we have set k=10, the value has been chosen to have a low bias and a reserved variance. The

compared results have been represented using graphs for better understanding.

5. Results Analysis

To find the accuracy of our recommender system, we need it to evaluate through different measures. Some of the famous measures are Precision, Recall, Mean Absolute Error, Root Mean Squared Error, and Scalability.

5.1. Mean Absolute Error (MAE)

MAE is a measure of errors between the predicted and actual observations as in [18]. It is expressed as Equation 5.

$$MAE = \frac{\sum|\tilde{P}_{i,j} - r_{i,j}|}{N} \tag{5}$$

In the Equation 5, N represents the total number of expected outcomes. $\tilde{P}_{i,j}$ is the predicted value for user i on item j and $r_{i,j}$ gives the true rating.

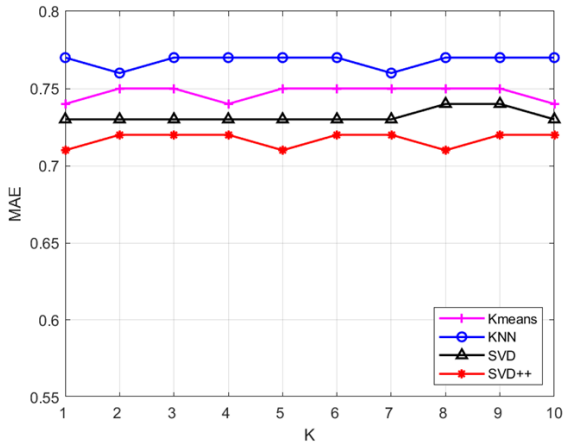

Fig.1. MAE of K-Means, KNN, SVD, SVD++. Error in case of SVD++ was Minimum and in Case of KNN was Maximum.

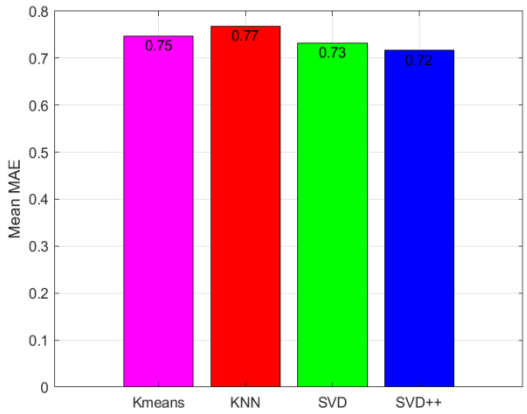

Fig.2. Mean MAE of K-Means, KNN, SVD, SVD++. Error in Case of SVD++ was Minimum and in Case of KNN was Maximum.

From Fig. 1, it is evident that both the matrix factorization methods SVD and SVD++ are performing better than the neighborhood methods K-Means and KNN. The error observed in SVD++ was minimum and maximum in case of KNN. On an average the MAE difference between SVD++ and KNN has been just 0.05 as presented in Fig. 2.

5.2. Root Mean Squared Error (RMSE)

It is a commonly used method to differentiate between the calculated values by a model and the observed values that are being modeled as in [18]. RMSE can be given by the square of subtraction with the predicted and identified results summation up to n times. Then dividing the result and then raised to power half. The RMSE is also defined as the square root of the mean squared error.

It is clear from Fig. 3 that yet again both the matrix factorization methods SVD and SVD++ are performing better than the neighborhood methods K-Means and KNN. The error in the case of SVD++ is much lower than the rest. It is also observed that with increasing numbers of K, the error is also increasing. On an average the MAE difference between SVD++ and KNN has been just 0.06 as shown in Fig. 4.

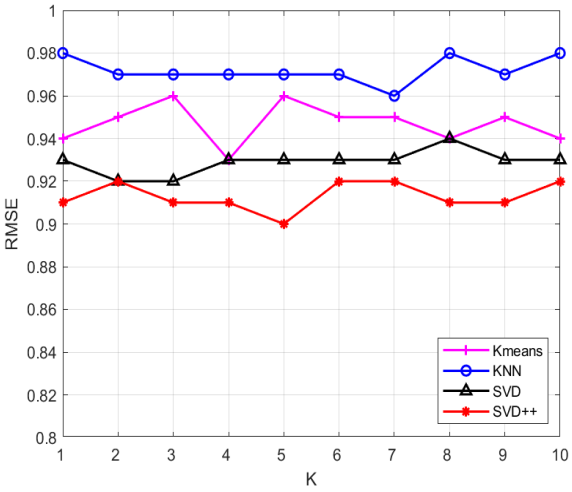

Fig.3. RMSE for all the Four Algorithms K Means, KNN, SVD, SVD++. Error in Case of Matrix Factorization Methods was Minimum.

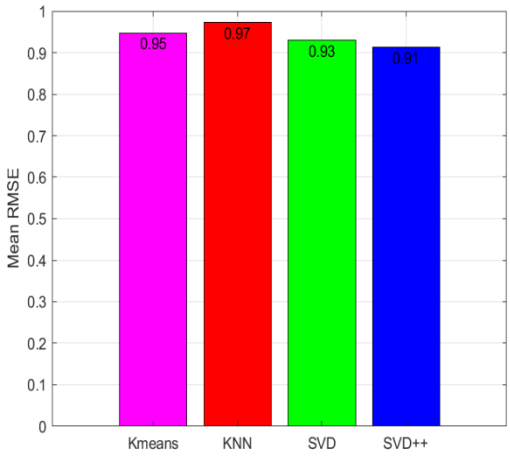

Fig.4. Mean RMSE for all the Four Algorithms K Means, KNN, SVD, SVD++.

5.3. Precision

It is the count of correct positive outcomes (recommended items liked by the user) divided by the classifier's predicted count of positive results (Total Recommended items) as in Equation 6.

$$Precision = \frac{TruePositives}{TruePositives+FalsePositives} \tag{6}$$

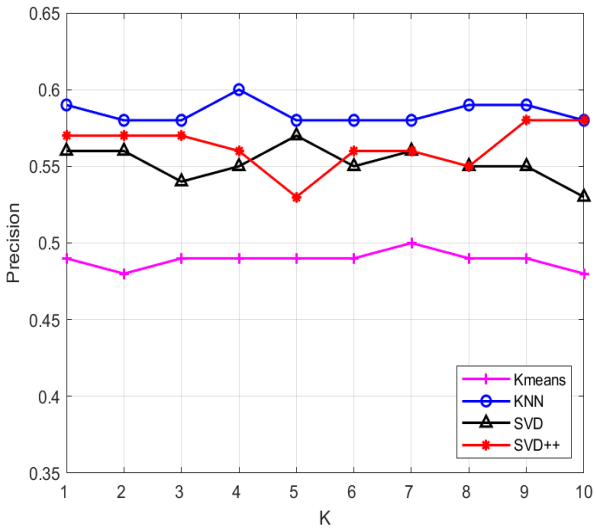

Fig.5. KNN has the Best Precision following SVD++ than SVD. KNN has the Lowest Precision.

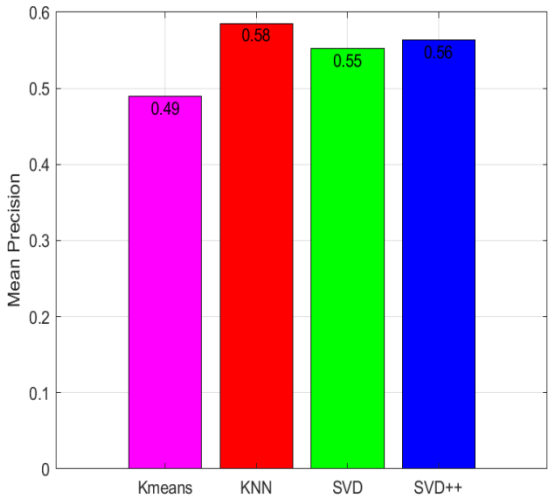

Fig.6. Mean Precision for all the Four Algorithms K Means, KNN, SVD, SVD++.

Precision in the recommender system is the percentage of relevant items recommended to the user divided by the total number of recommended items (liked or disliked both). So, in our case the value of precision is highest in KNN

Following there is not much difference in the values of SVD and SVD++. This inferred that the items recommended through KNN are most liked by user (Fig. 5). The lowest value of precision is achieved in K-Means. On an average the precision difference was just .02 more in case of KNN in contrast to SVD++ as shown in Fig. 6.

5.4. Recall

It is calculated by dividing the count of truly positive results (Recommended items liked) by the sum number of targeted samples (recommended liked items and also liked items which are not recommended) as in [19] presented in Equation 7.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{7}$$

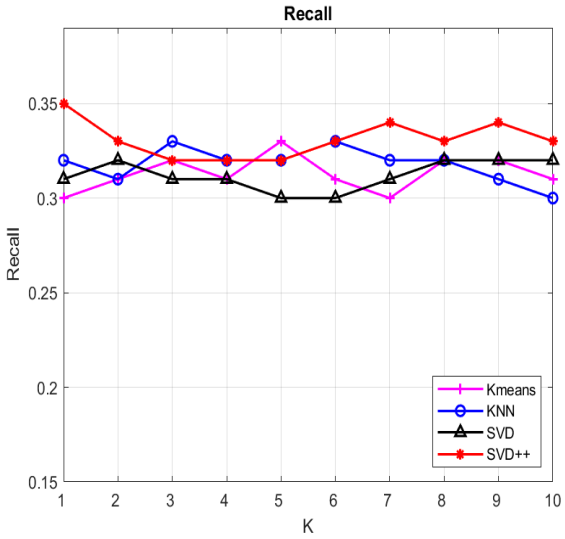

Fig.7. Recall of all Four Algorithms KNN, K Means, SVD++ and SVD.

Recall is the total number of recommended items liked by the user divided by the number of items which may be similar to the user likes (both recommended and non-recommended). Lower the recall the better is the algorithm. So, in our case the lowest recall is of SVD and K-Means as shown in Fig. 8. Finally, the highest Recall is of SVD++. Average difference between both the SVD++ and KMeans or SVD is just 0.02 as presented in Fig. 8.

5.5. Running Time (Scalability)

As more people and things are added, the amount of data utilized as input to the Recommendation System is rapidly increasing. The size of saved user behavior data on a popular website, for example, may quickly surpass TBs each day as in [20]. In order to keep users interested, most Recommendation Systems strive to reply interactively in less than a second,

despite the massive volume of data. It is evident from Fig. 9 and Fig. 10 that the time difference between the neighborhood methods and Matrix Factorization is significant. The time taken by neighborhood methods is much lower than Matrix Factorization methods. On average the difference between the KMeans and SVD++ is 1485.57 which is significantly high. So, it is difficult to scale Matrix Factorization methods than Neighborhood Methods.

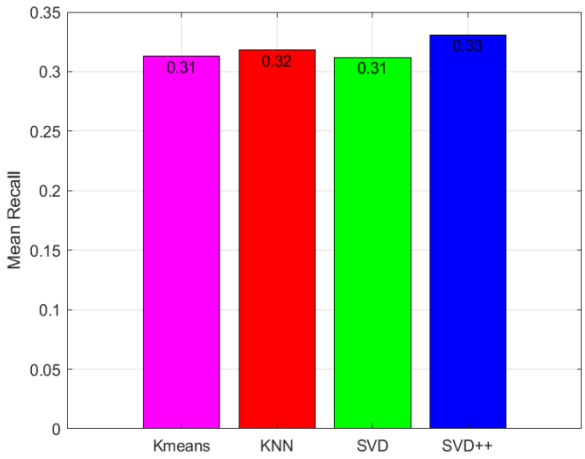

Fig.8. Mean Recall of all Four Algorithms KNN, K Means, SVD++ and SVD. Recall is Lowest in case of KMeans and SVD.

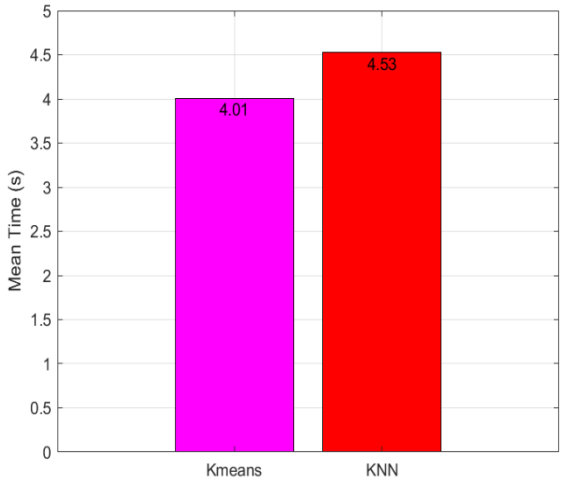

Fig.9. Mean time of Neighborhood Methods KMeans and KNN.

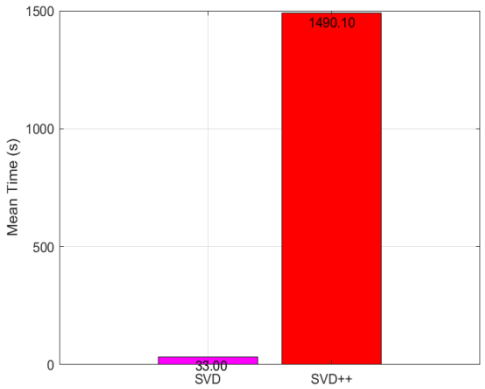

Fig.10. Mean time of Matrix Factorization methods SVD and SVD++.

6. Conclusion

Recommender systems are now the essential part of every website to keep the customer engaged with every single click. With growing customers, there is a need to store every customer interest and behavior to respond with an accurate recommendation. As there are two major ways to implement recommender systems Matrix Factorization and Neighborhood based methods. Our study has performed a comparison between the four methods KMeans, KNN, SVD,

and SVD++ to find out the better algorithm in terms of scalability. The work is trying to find a method that is not only better in terms of providing fast results, but it should generate them with a reasonable level of accuracy as well. We performed a comparison between four well known algorithms K-Means, KNN which are neighborhood search-based methods and SVD, SVD++ which are Matrix Factorization based Methods. The work compared them in terms of Precision, Recall Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and time. It is found out that in terms of MAE and RMSE Matrix Factorization methods are performing better, but in terms of scalability Neighborhood Methods perform much better. It was evident from results that the difference between the neighborhood methods and Matrix Factorization is significant in terms of time. The time taken by neighborhood methods is much lower than Matrix Factorization methods. On an average the difference between the neighborhood-based methods is 6x which is significantly higher. Even accounting for the precision and recall it does not make sense to select matrix factorization methods. So, it is more difficult to scale Matrix Factorization methods than Neighborhood Methods. Future work can focus on an efficient Matrix factorization method which has comparable time efficiency to Neighborhood based methods.

References

[1] Javari, A., &Jalili, M. (2015). A probabilistic model to resolve diversity–accuracy challenge of recommendation systems. Knowledge and Information Systems, 44(3), 609-627.

[2] Koren, Y., Bell, R., &Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

[3] De Vriendt, J., Degrande, N., &Verhoeyen, M. (2011). Video content recommendation: An overview and discussion on technologies and business models. Bell Labs Technical Journal, 16(2), 235-250.

[4] Ricci, F., Rokach, L., &Shapira, B. (2015). Recommender systems: introduction and challenges. In Recommender systems handbook (pp. 1-34). Springer, Boston, MA

[5] J. Jiao, X. Zhang, F. Li and Y. Wang, "A Novel Learning Rate Function and Its Application on the SVD++ Recommendation Algorithm," in IEEE Access, vol. 8, pp. 14112-14122, 2020, doi: 10.1109/ACCESS.2019.2960523.

[6] 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).

[7] Takács, G., Pilászy, I., Németh, B., &Tikk, D. (2008, December). Investigation of various matrix factorization methods for large recommender systems. In 2008 IEEE International Conference on Data Mining Workshops (pp. 553-562). IEEE.

[8] Jakomin, M., Bosnic, Z., &Curk, T. (2020). Simultaneous incremental matrix factorization for streaming recommender systems. Expert Systems with Applications, 160, 113685.

[9] Kim, T. Y., Ko, H., Kim, S. H., & Kim, H. D. (2021). Modeling of Recommendation System Based on Emotional Information and Collaborative Filtering. Sensors, 21(6), 1997.

[10] Singh, H., &Jain(2020), A. RECOMMENDATION OF BOOKS IN ONLINE BOOK STORE USING KNN. Journal of Analysis and Computation (JAC).

[11] Cui, Z., Xu, X., Fei, X. U. E., Cai, X., Cao, Y., Zhang, W., & Chen, J. (2020). Personalized recommendation system based on collaborative filtering for IoT scenarios. IEEE Transactions on Services Computing, 13(4), 685-695.

[12] Sano, N., Machino, N., Yada, K., & Suzuki, T. (2015). Recommendation system for grocery store considering data sparsity. Procedia Computer Science, 60, 1406-1413.

[13] Kumar, R., Verma, B. K.,&Rastogi, S. S. (2014). Social popularity based SVD++ recommender system. International Journal of Computer Applications, 87(14).

[14] Dehbozorgi, N., &Mohandoss, D. P. (2021, October). Aspect-Based Emotion Analysis on Speech for Predicting Performance in Collaborative Learning. In 2021 IEEE Frontiers in Education Conference (FIE) (pp. 1-7). IEEE.

[15] Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003, November). KNN model-based approach in classification. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 986-996). Springer, Berlin, Heidelberg.

[16] Li, Y., & Wu, H. (2012). A clustering method based on K-means algorithm. Physics Procedia, 25, 1104-1109.Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017, August).

[17] Ramzan, B., Bajwa, I. S., Jamil, N., Amin, R. U., Ramzan, S., Mirza, F., &Sarwar, N. (2019). An intelligent data analysis for recommendation systems using machine learning. Scientific Programming, 2019.

[18] Katarya, R., &Verma, O. P. (2017). An effective collaborative movie recommender system with cuckoo search. Egyptian Informatics Journal, 18(2), 105-112.

[19] Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. Encyclopedia of database systems, 5, 532-538.

[20] Xin, Y. (2015). Challenges in recommender systems: scalability, privacy, and structured recommendations (Doctoral dissertation, Massachusetts Institute of Technology).

Authors' Profiles

Ms. Nikita Taneja is currently working as a Research Professional at Siemens Technology and Services Private Limited. She has more than 13 years of teaching and industry experience. She is pursuing her Ph.D. (Computer Science and Engineering) from Manav Rachna University in the field of Cross Domain Recommender systems. Her areas of interest include Information Retrieval, Machine Learning, Artificial Intelligence, Data Mining.

Dr. Hardeo Kumar Thakur is working as Professor at department of Computer Science and Technology of Manav Rachna University (MRU), Faridabad. He has more than 134years of teaching and research experience in leading institutions of India. He has earned his Ph.D (Computer Engineering) from University of Delhi in 2017 in the field of data Analytics.

Dr. Thakur has published 22 research papers in international journal of repute, 13 papers in international conferences and 2 Edited books. His current research interests are Data Mining, Dynamic Graph Mining, Machine Learning and Big Data analytics. He is an active referee for many international Journals and Conferences.

How to cite this paper: Nikita Taneja, Hardeo Kumar Thakur, "Evaluating the Scalability of Matrix Factorization and Neighborhood Based Recommender Systems", International Journal of Information Technology and Computer Science(IJITCS), Vol.15, No.1, pp.21-29, 2023. DOI:10.5815/ijitcs.2023.01.03