Comparative Analysis of Recommendation System Algorithms for Building Recommendation System as a Service

Suyash Nehete
PCCOE, Pune

Riddhi Dhage
PCCOE, Pune

Sarvesh Hon
PCCOE, Pune

Tanuja Patankar
PCCOE, Pune

Laxmi Kale
PCCOE, Pune

ABSTRACT

In the modern world of online enterprises and e-commerce, there is a tremendous need to engage users by delivering relevant and interesting content. Recommendation systems play a crucial role in providing personalized content recommendations to users, which can improve customer interest, purchase rates, and, ultimately, company profits. This paper aims to categorize and explore different types of recommendation systems, such as collaborative and content-based filtering, and examine various methods and algorithms that can be used for implementing these systems. The accuracy of these algorithms is evaluated using an e-commerce dataset. Experimental results revealed that SVD++ achieved the best and lowest RMSE (Root Mean Square Error) with the parameters 'n epochs': 25, 'reg all': 0.4, and 'lr all': 0.01, while SVD had the second best RMSE with the parameters 'n epochs': 20, 'reg all': 0.2, and 'lr all': 0.005. Comparison between SVD and SVDpp revealed slight differences in RMSE and MAE values, but SVD had significantly shorter Fit Time and Test Time (12 times less) compared to SVDpp. Based on the research and experimental results, SVDpp performed the best in terms of RMSE among the Matrix Factorization Based Algorithms, and KNNWithMeans showed promising results in RMSE among the Collaborative Filtering Algorithms.

General Terms

Recommendation System, CF, CBF, MSE, KNN

Keywords

Recommendation Systems, Collaborative Filtering, Content-Based Filtering, K-means, K- Nearest Neighbor, Mean Square Error.

1. INTRODUCTION

In today's rapidly evolving business landscape, traditional enterprises are increasingly transitioning to become e-commerce businesses, conducting their operations online through electronic devices and the internet. This trend extends beyond just commercial enterprises, with other types of organizations, such as audio and video streaming providers, also relying heavily on virtual participation from their clients. The choice of business hosting platform plays a crucial role in enabling these online enterprises to expand and thrive.

In this highly competitive environment, it is essential to keep users engaged by providing them with relevant content that meets their expressed interests and information needs. This has led to the emergence of Recommendation Systems, also known as Recommendation Engines, as an effective solution to achieve personalized customization for users. These systems are designed to analyze user behavior, preferences, and other relevant data to provide tailored recommendations, enhancing the user experience and increasing engagement.

As e-commerce and online businesses continue to grow, Recommendation Systems have become a vital tool for businesses to stay competitive in the market. This paper will explore the key concepts of Recommendation Systems, including their types, evaluation metrics, and challenges, and discuss their significance in the current business landscape.

Recommendation Systems:

Information Filtering Systems can be subdivided into a subclass known as Recommendation Systems. These systems filter the data of the user based on the user's preferences, search histories, and browsing habits in order to determine the product or content that the user is most likely to enjoy and purchase. Businesses at the forefront of innovation in today's market, such as Amazon, Netflix, Medium, and Spotify, implement recommendation systems to give their customers a more individualized experience by proposing specific material based on their preferences.

Suggestions are created on the basis of the material and commodities that have received the most likes or views, as well as the user's personal search and purchase history. As a result, recommendation systems can be broken down into two primary categories:

1. Collaborative Filtering[6]
2. Content-Based Filtering[6]

1.1 Collaborative Filtering

This technique classifies users into clusters of similar types and recommendations based on the preferences of that cluster. This method considers the preferences and likings of other similar users. Thus, the collaborative filtering technique emphasizes user preferences[7].

It takes into account the user rating, and reviews feedback for the product to be recommended to other similar users[6].

A simple example to explain Collaborative Filtering would be recommending the *Harry Potter* series to the User A and B both because both users have their taste in Fantasy Literature.

This method lacks to provide recommendations to the newer users who don’t have user profiles or users without previous feedback and like. This is referred to as the Cold Start issue.

1.2 Content-Based Filtering

This strategy selects the contents depending on the user's previous queries and interests, in addition to the features of the items themselves[7]. The primary strategy is constructing the model on the basis of the characteristics that describe the user-item interactions. Content-based filtering, much like collaborative filtering, does not rely on the data contributed by other users in order to make recommendations to a single user[5].

For instance, there is a good chance that the customer will be advised to check out *Fantastic Beasts*. A's purchasing history reveals that he has a copy of *Harry Potter* and *The Ickabog*, both of which are works created by *J. K. Rowling*.

These recommendation systems can be implemented using any number of different machine-learning methods that are currently available. Let's take a look at each of them in turn.

2. BACKGROUND

2.1 K-Means

K-means is an unsupervised machine learning method used for data classification, which involves partitioning data into K distinct and non-overlapping subgroups. It follows an iterative approach that assigns data points to clusters based on the sum of their squared distances and the centroids of the clusters [1]. The Expectation-Maximization approach is utilized in k-means, where the E-step involves assigning data points to the geographically closest cluster, and the M-step involves determining the centroids of each cluster [1]. The goal of k-means is to minimise the target function over and over again. In math, this is shown as:

The goal of the k-means algorithm is to minimize the objective function for a given dataset X = {x1, x2, ..., xn}, where n represents the number of data points and each data point xi is a d-dimensional vector.

$$J = \sum_{i=1}^{m} \sum_{k=1}^{k} \omega_{ik} \parallel x^i - \mu_k \parallel^2 \qquad \text{......(1)}$$

In this context, w_{ik} represents an indicator variable that is set to 1 if data point xi is part of cluster k, and 0 otherwise. The centroid of cluster k is denoted as μ_k, and ||.|| denotes the Euclidean distance between two points.

The primary objective of k-means is to minimize the sum of squared distances between each data point and its respective cluster centroid[1]. This is accomplished through iterative updates of cluster assignments and centroids until convergence is reached.

2.2 Singular Value Decomposition (SVD)

SVD, which is also known as Singular Value Decomposition, is a method for reducing the number of dimensions in machine learning that is often used in joint filtering within recommender systems[1]. In this method, a utility matrix A is broken up into three matrices: U, S, and V^T. Each row in A represents a user, and each column represents a group or type of information. Here, U is a left-singular orthogonal matrix that shows how users and latent variables are connected, S is a diagonal matrix that shows how strong each latent component is, and V^T is a diagonal right-singular matrix that shows how similar items and latent factors are. By getting rid of its latent members, which represent people and things in an r-dimensional latent space, the size of the utility matrix A is decreased[1].

The goal of the SVD method, from a mathematical point of view, is to minimize the squared difference between the original utility matrix A and the product of U, S, and V^T. This is shown as:

Given a utility matrix A with the dimensions m x n (where m is the number of users and n is the number of things), the SVD can be found by factoring A into U, S, and V^T so that $A = USV^T$ and minimising the following objective function:

$$J = \parallel A - USV^T \parallel^2 \quad \text{...(2)}$$

where ||.|| stands for the Frobenius norm, which is the square root of the sum of all of a matrix's squared elements. The U matrix shows what the users like, the S matrix shows how strong each latent factor is, and the V^T matrix shows how similar things and latent factors are to each other.

2.3. K-Nearest Neighbor (KNN)

KNN is a non-parametric and lazy learner algorithm used for both classification and regression tasks[2]. During the training phase, KNN builds a reference database of labeled data points, referred to as the training dataset. The training dataset consists of input feature vectors (X) and their corresponding class labels (Y) for classification or target values for regression[2]. KNN does not explicitly define an objective function but follows a simple procedure during the prediction phase:

Prediction Formula:

For Classification: Prediction = Majority Class(Y) among K nearest neighbors

For Regression: Prediction = Average of Target Values(Y) among K nearest neighbors

where:

- Prediction: The estimated class label (for classification) or target value (for regression) of the test data point.
- Majority Class(Y): The most frequent class label among the K nearest neighbors, for classification task.
- Average of Target Values(Y): The average of the target values of the K nearest neighbors, for regression task.

In many cases, a distance measure such as Euclidean distance, Manhattan distance, or cosine similarity is utilized to assess the similarity between the test data point and the training data point. In the K-nearest neighbors (KNN) algorithm, the number of nearest neighbors to consider for predictions is determined by the parameter K. The algorithm identifies the K nearest neighbors of the test data point in the training dataset based on the chosen distance metric. It then predicts the category of the test data point by considering the majority class of its K nearest neighbors in the case of classification, or the average of their target values in the case of regression.

KNN is a simple and effective algorithm that utilizes the similarity between data points for making predictions. The provided formula represents the prediction step in KNN, which involves determining the majority class or averaging the target values of the K nearest neighbors to make predictions. KNN is well-known for its simplicity, ease of implementation, and versatility in handling both classification and regression tasks.

3. LITERATURE SURVEY

Numerous publications have proposed various approaches for building recommender systems, as detailed by several authors in their studies. Some of these strategies are highlighted below:

In their work, Hafed Zarzour, Ziad Al-Sharif, Mahmoud Al-Ayyoub, and Yaser Jararweh [1] investigated new strategies for constructing recommender systems. They utilized k-means clustering to categorize users based on their activities and interests. Additionally, they employed Singular Value Decomposition (SVD) not only for dimensionality reduction but also as an effective method within each cluster. The proposed method, known as k-means-SVD, achieved the best performance with a Root Mean Squared Error (RMSE) of 0.59.

The challenges associated with K-Nearest Neighbor (KNN) were addressed by Bin Li, Sailuo Wan, Hua Xia, and Fengshou Qian [2], who also proposed solutions to overcome the issues related to data sparsity and global effect variables. Furthermore, their work exclusively focused on content-based filtering without incorporating any user information. They enhanced the KNN algorithm and proposed an improved version called IKNN. The best-performing model based on IKNN achieved an RMSE of 0.88.

Babak Maleki and Shoja Nasseh Tabrizi [3] identified limitations in earlier recommendation systems and proposed a solution using Latent Dirichlet Allocation (LDA) to extract relevant characteristics from testimonials. They also employed collaborative filtering with matrix factorization to generate recommendations. The best-performing model for Accessories achieved an RMSE of 2.18, while the best-performing model for Office Items achieved an RMSE of 1.72.

Table 1. Comparison of Research Papers

Parameters	**Paper 1**	**Paper 2**	**Paper 3**
Authors	*Hafed Zarzour, Ziad Al-Sharif, Mahmoud Al-Ayyoub, Yaser Jararweh*	*Bin Li, Sailuo Wan, Hua Xia* and *Fengshou Qian*	*Babak Maleki Shoja* and *Nasseh Tabrizi*
Publication	ICICS 2018	IEEE 2020	IEEE ACCESS 2016
Algorithms Used	k-means, SVD	KNN (k-nearest-neighbour), IKNN	LDA
Use of algorithms	k-means: Clustering Users SVD: Matrix Factorization	KNN: For Recommendation IKNN: For Recommendation	LDA : Attribute Extraction PMF: Matrix Factorization
RMSE	k-means-SVD :- 0.59 k-means :- 0.64	KNN: 1.02 IKNN:- 0.88	Accessories: 2.18 Office Product: 1.72
Advantages	k-means is used to cluster users and SVD is used for dimensionality reduction and also as a powerful mechanism. And this techniques has the lowest RMSE.	Two characteristics of data sparsity and global effect factors were used to improve the KNN algorithm.	Each subject and topic distribution for each document are distributed using LDA.
Disadvantages	Results are dependent on the size of the dataset. Transformed data may be difficult to understand	Only the Content-based technique is used. User's own data is not used for generating recommendations	Reviews of other users are used. No data specific to user is used.

4. EXPERIMENTAL RESULTS

In the experimental study, several iterations of the SVD and KNN algorithms are performed using Amazon's Review Data from Kaggle, which consists of 7 million ratings provided by 4.2 million users for 470,000 unique products. The dataset was divided into a training set with 70% of the data and a testing set with 30% of the data for evaluation purposes. The Root Mean Square Error (RMSE) was employed as a metric to assess the predictive performance of the algorithms.

RMSE is a common way to measure how well a recommender system does its job[1]. It measures the average squared difference between the ratings that were expected and the ratings that users gave[1]. In math, RMSE is described as:

$$RMSE = \sqrt{\left(\frac{1}{N}\right)\sum_{i=0} (p_{u,i} - r_{u,i})^2} \quad ...(3)$$

where:

- N represents the total number of ratings in the dataset.
- u denotes a user in the dataset.
- i signifies an item (or product) in the dataset.
- $p_{u,i}$ represents the predicted rating that user u would assign to item i according to the recommender system.
- $r_{u,i}$ is the actual rating provided by user u for item i.

In equation (3), the RMSE metric calculates the square root of the average of the squared differences between the predicted ratings and the actual ratings for each user-item pair in the dataset. A smaller RMSE value indicates better performance as it signifies that the predicted ratings are more closely aligned with the actual ratings provided by users.

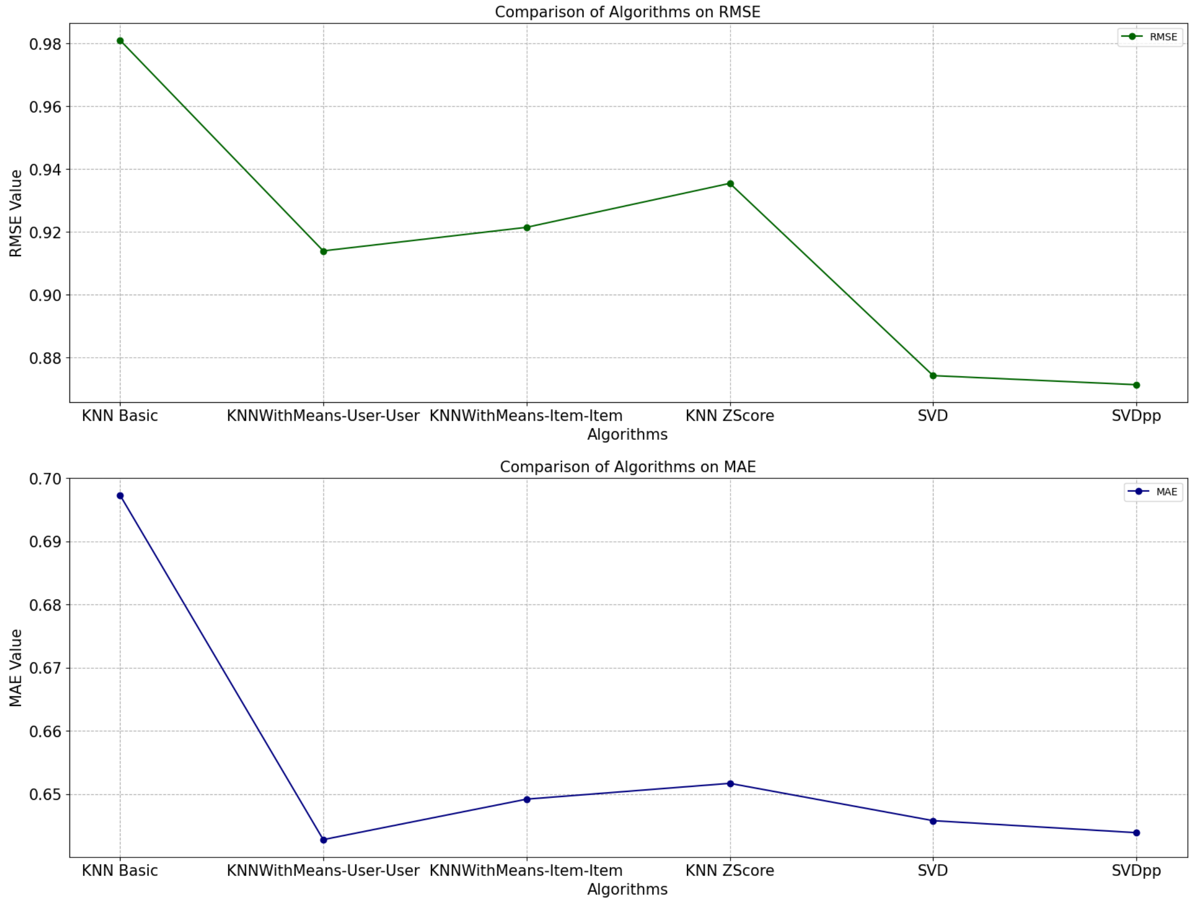

Fig 1:Comparison of Algorithms

Table 2. Comparison of Research Papers

Algorithm	**RSME**	**MAE**
KNN Basic	0.98	0.70
KNNWithMeans-User-User	0.91	0.64
KNNWithMeans-Item-Item	0.92	0.65
KNN ZScore	0.93	0.65
SVD	0.87	0.65
SVDpp	0.86	0.64

- The comparison of algorithms is shown in Figure 1. Based on the algorithm comparison plots, the following observations are made:

- RMSE: SVD++ achieved the best and lowest RMSE when using the parameters 'n epochs': 25, 'reg all': 0.4, and 'lr all': 0.01, while SVD had the second best RMSE with the parameters 'n epochs': 20, 'reg all': 0.2, and 'lr all': 0.005.

- MAE: Both SVDpp and KNNWithMeans achieved the best MAE values in this particular scenario.

- SVD++ showed the lowest RMSE among the Matrix Factorization Based Algorithms.

- KNNWithMeans performed the best in terms of RMSE among the Collaborative Filtering Algorithms.

- When comparing SVD and SVDpp, notice slight differences in RMSE and MAE values, but SVD had significantly shorter Fit Time and Test Time (12 times less) compared to SVDpp.

5. FUTURE SCOPE

In future, we’ll study more ways for building a recommender system. And the best-performing technique can be used for developing a recommender system. This recommendation system can be used in various business cases like e-commerce, media, etc. This can be used by small scales businesses that cannot afford or build their own recommendation system.

6. CONCLUSION

We have researched and conducted experiments on a variety of approaches to the construction of a recommender system utilising several algorithms, such as KNNBasic, KNNWithMeans, KNNZScore, SVD, and SVDpp.

We discovered that the SVDpp has the lowest RMSE when it comes to doing matrix factorization. In the case of collaborative filtering, KNNWithMeans provides the lowest RMSE values. The differences between SVD and SVDpp are not huge but they are there. Nonetheless, the amount of time needed for Fit Time and Test Time when using SVD is twelve times less than when using SVDpp. Thus, going on with SVD will be our strategy for our further research.

7. REFERENCES

[1] *Hafed Zarzour, Ziad Al-Sharif , Mahmoud Al-Ayyoub, Yaser Jararwe). “A New Collaborative Filtering Recommendation Algorithm Based on Dimensionality*

Reduction and Clustering Techniques.”. 2018 9th International Conference on Information and Communication Systems (ICICS)

[2] Bin Li, Sailuo Wan, Hua Xia and Fengshou Qian. “The Research for Recommendation System Based on Improved KNN Algorithm”. 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)

[3] *BABAK MALEKI SHOJA, AND NASSEH TABRIZI. “Customer Reviews Analysis with Deep Neural Networks for E-Commerce Recommender Systems”. Citation information: DOI 10.1109/ACCESS.2019.2937518, IEEE Access*

[4] Li H aihan, Qi Guanglei, He N ana, Dong X inri. “Shopping Recommendation System Design Based On Deep Learning”. 2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing (ICSP 2021)

[5] Kyle Ong, Su-Cheng Haw, Kok-Why Ng. “Deep Learning Based-Recommendation System: An Overview on Models, Datasets, Evaluation Metrics, and Future Trends”. CIIS’19, November, 2019, Bangkok, Thailand

[6] Rohit Dwivedi, Abhineet Anand, Prashant Johri. “ Product Based Recommendation System On Amazon Data”. Researchgate publication 352815845

[7] Hyeyoung Ko , Suyeon Lee, Yoonseo Park and Anna Choi. “A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields”. Electronics 2022, 11(1), 141; https://doi.org/10.3390/electronics11010141

[8] Beel, J., Gipp, B., Langer, S. *et al.* “Research-paper recommender systems: a literature survey”. *Int J Digit Libr* 17, 305–338 (2016). https://doi.org/10.1007/s00799-015-0156-0

[9] Cleomar Valois B. JrMarcius Armada de Oliveira. “Recommender systems in social networks”. JISTEM J.Inf.Syst. Technol. Manag. 8 (3) • Dec 2011 • https://doi.org/10.4301/S1807-17752011000300009

[10] LIU Liling. “Summary of recommendation system development”. 2019 *J. Phys.: Conf. Ser.* 1187 052044