International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Recommendation Systems Using Event-Based Temporal Data Model

[1]Mr. Vinod B. Ingale, [2]Dr. E. Saikiran

Submitted: 14/09/2023 **Revised**: 27/10/2023 **Accepted**: 12/11/2023

Abstract: Despite challenges like concept drifts, or temporal dynamics in RS, RS has grown in popularity due to its usefulness in meeting customers' needs by helping them find things they might like based on past purchases and interests. Despite their great effectiveness in generating recommendations, conventional RS techniques fall short when it comes to providing accurate ideas due to problems with concept drift. The development of temporal models to account for concept drifts and guarantee more accurate recommendations has been the focus of a lot of research in the wake of these issues, giving rise to dynamic recommender systems (DRSs). However, the bulk of the effort needed to address the drift of interest is put in developing long-term and short-term models for users. It is not possible to dynamically track users' changing interests. We express doubts in this position paper about the practice of computing evaluation metrics for recommender systems as single numbers since these values only represent average effectiveness over a very long time period (for example, a year or longer). This approach only provides a vague, unchanging picture of the facts. To better understand the performance of recommender systems, we propose that researchers compute metrics across time series such as weeks or months and present the results visually, using a line chart, for instance. Insightful forecasts about an algorithm's future performance can be made with the use of results that show how an algorithm's effectiveness evolves over time. As a result, we'll be able to make more informed selections about which algorithms to utilize in a recommender system by collecting more data on an algorithm's performance over time, spotting trends, and developing more accurate forecasts about an algorithm's future performance.

Keywords**:** *Recommendation systems include dynamic recommender systems, time series analysis, and algorithms.*

1. Introduction

In order to manage the explosion of online data and provide users with relevant, personalized recommendations, recommendation systems (RSs) were created. Offering support to customers Current applications of RSs include increasing the possibility of cross-selling and customer or consumer pleasure and loyalty by suggesting products that users may find interesting. These applications may be found in a wide variety of industries, including e-commerce, advertising, e-learning, document management, and journalism. The CF technique, which uses user similarity values (or items), is often used by RSs. In other words, the CF approach is based on the idea that customers will continue to choose similar products if they have always accessed them in the same way. The user's tastes might be influenced by factors including their current location, the time of day, the weather, and the type of gadget they are using. Using these standards, we can gain insight that can help us improve RS performance. In this study, we offer a novel recommendation system that accounts for the impact of users' reported rating timestamps. To that end, we first set up a representation model based on sequential patterns to capture the users' feedback. An historical sequence of user similarities is constructed for use in predicting future user similarity.

[1]*Research Scholar, CSE Department, Chaitanya Deemed to be University, Warangal*
Assistant Professor PVPIT Budhgaon

[2]*CSE Department, Chaitanya Deemed to be University, Warangal*

1.1 Recommender system

Recommendation systems can be valuable tools when discovering objects in a huge collection of data. Information in the form of text, articles, videos, audio, etc. may be included in such collections. Sites like eBay, Netflix, and Spotify all make use of recommender systems [1]. Since the advent of collaborative filtering in the 1990s, RSs have been an integral part of every information-based business available on the internet, from bookselling to video streaming to ad recommendations [2].

Figure 1 [3] shows the development of RS over time, with the initial recommendations being provided by content-based filtering based on the attribute values of data. The late 1990s saw the rise of collaborative filtering (CF). The usage of hybrid algorithms for recommendations has developed from the adoption of ontology-based RS in the early 2000s [3].

There has been a noticeable shift towards storing more information online. It seems to reason that as data volumes increase, it will become ever more challenging to locate specific pieces of information. As a result, there is still a need for cutting-edge, lightning-fast recommendation algorithms. The criteria also vary depending on the specifics of the field being studied [2]. While traditional recommendation algorithms have their shortcomings, the demand of hybrid RSs has increased.

Any of the algorithms covered in the rest of this paper can be used to create these hybrid algorithms. In order to keep up with rising expectations for reliability, developers of hybrid recommendation systems are increasingly using neural networks and deep learning algorithms.

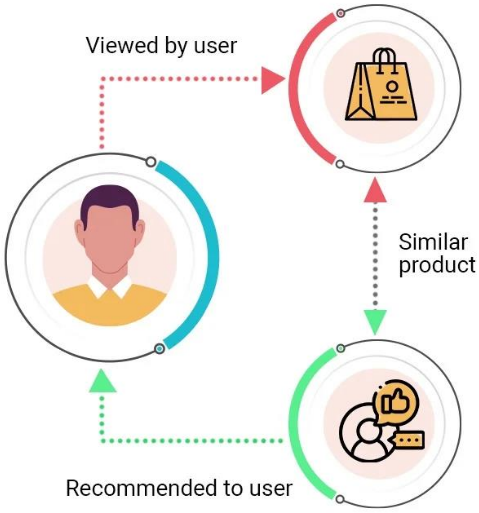

Fig 1.1 Recommendation Systems

Products that many customers are interested in, demographic data, and previous purchasing activity are analyzed by a recommender system so that recommendations can be made that best meet the needs of each individual customer [8]. Services that cater to the preferences of each customer have become increasingly important in online shopping. In the context of the Internet, personalization refers to the practice of rapidly catering to each individual customer's special requirements.

Web customization is described as action conducted on the Internet by an individual in response to his/her interests or tastes [14]. The importance of personalized service lies in the fact that it allows clients to spend less time looking for what they need. By recommending the right items, businesses not only strengthen ties with their online shoppers but also boost client loyalty [15].

1. 2 Personalization techniques

The personalization techniques for recommender systems include:

(1) Content-based recommender system: This system examines product data and makes suggestions based on that analysis. Texts, documents, news articles, and websites with copious and easily analyzed content are good candidates for this method of recommendation. [16].

(2) Rule-based filtering: Information profiles of consumers are compiled with this approach by questioning customers about their preferences. Profiles are built from users' answers to questions about their interests and preferences. This filtering system will advise or provide information on products that may interest the user based on their interests and personality attributes. [15].

(3) Demographic filtering: User information including age, sex, and education level are used to inform the system's recommendation making [17]. Users' preferences for specific products and categories can be easily analyzed with the use of demographic information.

(4) Collaborative filtering: This system makes recommendations using users' information such as age, sex, and education level [17]. Demographic attributes have an advantage of making an easy analysis of users' preferences regarding various kinds of items and item categories.

(5) Learning agent-based filtering system: Log files, which include information on when and from where a user accessed a website, can be parsed for clues about a user's attributes, habits, and preferences using this kind of personalization [16].

A recommender system is a piece of software that uses data about previous buyers or sellers to generate educated guesses about what an individual would be interested in. Many studies on recommender systems have aimed to determine how well they can predict whether or not a consumer will be happy with a certain purchase. Most often, comparable items are sorted using a technique called collaborative filtering. Collaborative filtering systems often employ the neighborhood-based algorithm depicted in Fig. 1. The active user determines how far away each other user is and then chooses as neighbors however many people are closest to them. Pearson's correlation coefficient, mean-square difference, and vector similarity can all be used to determine how far apart two users are.

In [18], the Pearson correlation coefficient yielded a better result than the vector similarity, and in [19], the Pearson correlation coefficient yielded a better result, though its predictive power could be diminished through the selection of either too few or too many neighbors.

1.3 Approaches to collaborative filtering

There are two kinds of collaborative filtering: user-based and item based collaborative filtering.

(1) User-based collaborative filtering: In order to quantify how well two users coincide with one another in terms of a shared feature, we can utilize distance calculations. The distance between two users is zero if, for instance, they both rate the same movie at the same level. On the other hand, if their ratings are different, the gap will be greater.

(2) Item-based collaborative filtering: In contrast to user-based methods, recommender systems often employ item-based collaborative filtering. For instance, two films are relatively close when users who enjoy movie No. 1 also like movie No. 2. User-based and item-based algorithms confront issues when an item is recommended. This is due to the stringent nature of the criteria for application, which precludes the use of anything even remotely similar as a starting point. For instance, if someone with a strong preference for action films is requested to rate content as part of a recommender system, the default algorithm will not be able to tell. This is where the idea of "dimensional reduction" is presented. An enormous matrix constructed to explore preference relations among many persons and numerous films can provide an explanation for dimensional reduction, for instance. The process of collecting data based on movie reviews should include an abstraction phase. Many people are categorized using the same criteria, and then additional items are added to the same categories based on the similarity criterion. A recommendation's efficacy increases as active dimensional reduction is applied.

Research and applications of collaborative filtering have been conducted in many different contexts. Collaborative filtering has been used by many websites to recommend content, including Amazon, CD Now, Drug store, and Movie Finder; Group Lens [5], a news article recommendation system; Video Recommender [4]; Ringo [6]; and PHOAKS [20], a user-related search engine on the World Wide Web.

1.4 Detection of attacks on collaborative filtering systems

In a recent article [21], Dellarocas outlined the many forms of assaults that have been undertaken against popular e-commerce platforms like eBay. We provide examples of these kinds of attacks on existing collaborative filtering algorithms and related systems and provide a prediction technique to mitigate their fallout.

Lam and Riedl [22] analyzed altered ratings by users and categorized suggestion assaults by kind. Attackers can influence a recommender system by directly rating things if they get access to the system. Finally altered by the attackers caused the system to make incorrect decisions. Lam and Riedl investigated factors including skewed evaluations that impacted the system's suggestions. Although research into recommender system attacks is ongoing, no reliable method for foreseeing unpredictable assaults has been developed. Forecasting possible random attacks on recommender systems through analysis of rating stream trends could be an effective approach for doing so. [23].

2. Exiting work

In [1], the author proposes KERS, a multi-armed bandit technique for patients to use while deciding on medical treatment. The first part, called "exploration," identifies areas in which customers have latent interest. A database of information compiled by specialists backs this up. When a user's focus moves, we enter a new exploitation phase during which we recommend products from these categories. The writers make an effort to lessen exploratory work while raising user satisfaction.

Ahmedi et al. [3] provide a personalized technique that may also be applied in more broad recommendation scenarios using user profiles. Using Collaborative Topic Regression, they extract correlation rules from past user interaction logs.

Together Focused Collaboration In [4], the Autoencoder is presented as a deep learning-based model for general recommendation that specifically addresses the issue of data sparsity. Modelling latent variables in users and publications, they employ probabilistic matrix factorization and textual data.

In [5], we create PR-HNE, a probabilistic paper recommendation model that is both personalised and built on a common representation of authors and papers. They make their paper suggestions using graph data including citations, co-authors, conference information, and themes. Author embeddings are represented using SBERT, and topic embeddings are represented using LDA.

Users and papers are represented in a bipartite graph in [19]. Word2Vec or BERT embeddings of a paper's content reflect the paper, while user vectors consist of representations of papers with which a given user has interacted. After that, we use elementary graph convolution to sum the vectors.

Current user interest is at the heart of [22]'s technique, which makes use of k-Means and KNN. Users' profiles are constructed using the works they have published. User recommendations are based on the most highly cited publications in the cluster that is most similar to them. In a subsequent attempt, they increased the size of their research team to once again focus on the same issue. Again, Bulut et al. [21] zero down on user details. Users are reduced to a collection of characteristics in their writings. Next, the vector representations of all articles are compared to these to find the most similar ones. Documents can be represented using TF-IDF, Word2Vec, or Doc2Vec vectors.

In [25], the authors propose a strategy to article recommendations that makes indirect use of direct elements collected from qualities of publications (such as keyword diversity, text complexity, and citation analysis).

Later, in [26], the authors propose employing indirect indicators, including the quality of the publication, for group recommendation. Users' profiles are compiled from the articles they've read. After that, a smaller group of them got to work on a plan with similar goals. The general Hybrid Topic Model is provided by Chaudhuri et al. [24] and is utilized to provide article suggestions. LDA and Word2Vec work together to deduce a user's preferences and goals. They extract user interest from probability distributions of clicked papers' phrases and dominating subjects in publications. In this study, we present CPM, a recommendation algorithm that uses thematic clustering to rank users based on their disclosed interests [27]. From these groups, they construct user requirement models using LDA and pattern equivalence class mining. In order to find the most suitable suggestions, candidate papers are compared to user need models.

As a recommender as a service, the authors of [28] propose deploying Mr. DLib, their paper recommendation system. Article representations in Doc2Vec, TF-IDF vectors, and a recommender based on keywords are compared. Du et al. [29] present HNPR, a heterogeneous network strategy that utilizes two distinct graphs. The technique merges citation information, co-author relationships, and published research fields. Using a technique called "random walk on networks," they build vector representations of articles.

3. Proposed work:

Our research suggests the following guidelines for selecting a recommendation algorithm:

1. The user's profile length may influence the quality of the recommendation.

2. the popularity of the products a user evaluates may affect the quality of the recommendation, in that a user may rate more popular items higher than less popular ones, so affecting the quality of the advice.

3. contrasting findings might be found by comparing two algorithms using different metrics (such mean absolute error and recall).

Due to the complexity of the algorithms, RS often integrate recommendation algorithms that provide users with a limited set of recommendations.

Methodology for evaluating and comparing recommendation algorithms is divided into three steps:

Step 1: statistical analysis and partitioning

Step 2: optimization of the model parameters

Step 3: computation of the recall for the top-N recommendation.

3.1 Statistical analysis and partitioning

M-fold cross-validation is used to divide the dataset into subsets.

In addition to dividing the dataset, we perform an analysis to determine which users belong to which groups and which products have the most ratings. This will allow us to determine how the model responds depending on the depth of the user profile and the level of interest in the recommended goods. The following procedure is used to organize users into groups:

In order to analyses the behavior of a learning algorithm with respect to the length of the profiles, we first sort users by the length of their profiles (i.e. the number of ratings), then split them into two groups so that each group contains the 50% of the ratings, and finally split each group into two subgroups (as described in b) to get a more granular view of the data.

In a similar vein, we use the following structure to identify the most well-liked products:

The items are: a) sorted by total ratings; b) split in half so that each half comprises 50% of the total ratings.

3.2 Optimization of the model parameters

By manipulating the threshold at which an item is classified as "to recommend" or "not to recommend," ROC curves can be utilized to see the trade-off between

TPR (i.e., recall) and FPR. A set of ROC curves can be obtained by varying the model's parameters (like the latent size of the singular value decomposition, which will be discussed in Section 4). In order to optimize the model parameters, according on the kind of dataset, we implement two strategies based on ROC curves: ROC1 and ROC2. Leave-k-out and the random withholding of 25% of the rated items for each user are the two methods.

4. Result Analysis

Movie Analytics

Movie Analytics is an online service that streamlines the process of renting and analyzing films. The homepage of the work contains two buttons, one for customers to book a movie and the other for the admin to examine stats. Including a landing page, client page, and admin, this work provides a comprehensive solution for movie analytics.

Booking and analyzing films is a breeze with Movie Analytics. It's a one-stop shop for the film business thanks to its interactive infographics and ability to estimate the best possible offer. This paper offers an end-to-end solution for movie analytics, covering everything from ticketing to analysis to forecasting. The results of an examination of our suggested system using several existing approaches and a wide range of parameters are shown below.

4.1 Performance Parameters

Metrics

ï»¿	Metric	Value
0	R-squared	0.6692550191622055
1	MAE	3.7519109184475155
2	MSE	20.019626195688545
3	RMSE	4.4743296923325335

Statistics

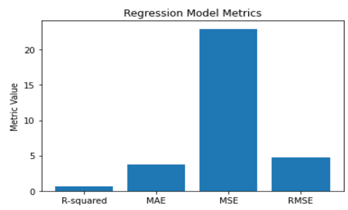

Fig 02: Performance analysis using different Parameters of the proposed system

4.1 Comparative Analysis

In order to analyses the practical application of a novel methodology, the results of the proposed system were analyzed using established methods for the various parameters, as shown in the figure below.

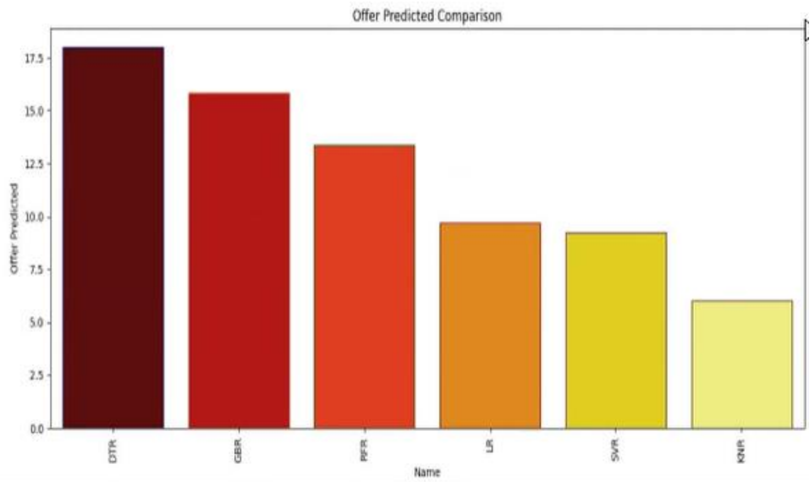

Fig 03: Comparative analysis using a proposed system with existing one

5. Conclusion:

The trials revealed that the system's accuracy is above average for more than 90% of the users. Also, those who have reviewed between 60 and 100 films had the highest accuracy. Because collaborative recommenders struggle on such user models, this is an advantage of the content-based approach recommender. Combined features might be viewed as a potential future optimization for the stated prediction technique to further enhance accuracy. Our proposed method enhances the recommender's prediction accuracy by identifying situations in which two attributes appear together, which often leads to a bad proposal.

References:

[1] Casillo, M.; Gupta, B.B.; Lombardi, M.; Lorusso, A.; Santaniello, D.; Valentino, C. Context Aware Recommender Systems: A Novel Approach Based on Matrix Factorization and Contextual Bias. Electronics **2022**, 11, 1003.

[2] Shanmugasundar, G.; Fegade, V.; Mahdal, M.; Kalita, K. Optimization of Variable Stiffness Joint in Robot Manipulator Using a Novel NSWOA-MARCOS Approach. Processes **2022**, 10, 1074.

[3] Kalita, K.; Ghadai, R.K. Optimization of Plasma Enhanced Chemical Vapor Deposition Process Parameters for Hardness improvement of Diamond Like Carbon Coatings. Sci. Iran. **2022**.

[4] Kalita, K.; Ghadai, R.K.; Chakraborty, S. Parametric optimization of CVD process for DLC Thin film coatings: A comparative analysis. S¯adhan¯a **2022**, 47, 57.

[5] Kalita, K.; Ghadai, R.K.; Bansod, A. Sensitivity Analysis of GFRP Composite Drilling Parameters and Genetic Algorithm-Based Optimisation. Int. J. Appl. Metaheuristic Comput. **2022**, 13, 1–17.

[6] 60. Shankar, R.; Ganesh, N.; Cˇ ep, R.; Narayanan, R.C.; Pal, S.; Kalita, K. Hybridized Particle Swarm—Gravitational Search Algorithm for Process Optimization. Processes **2022**, 10, 616.

[7] 61. Rajendran, S.; Ganesh, N.; Cˇ ep, R.; Narayanan, R.C.; Pal, S.; Kalita, K. A Conceptual Comparison of Six Nature-Inspired Metaheuristic Algorithms in Process Optimization. Processes **2022**, 10, 197.

[8] 62. Kalita, K.; Pal, S.; Haldar, S.; Chakraborty, S. A Hybrid TOPSIS-PR-GWO Approach for Multi-objective Process Parameter Optimization. Process Integr. Optim. Sustain. **2022**, 1–16.

[9] 63. Joshi, M.; Ghadai, R.K.; Madhu, S.; Kalita, K.; Gao, X.-Z. Comparison of NSGA-II, MOALO and MODA for Multi-Objective Optimization of Micro-Machining Processes. Materials **2021**, 14, 5109.

[10] 64. Pal, S.; Kalita, K.; Haldar, S. Genetic Algorithm-Based Fundamental Frequency Optimization of Laminated Composite Shells Carrying Distributed Mass. J. Inst. Eng. Ser. C **2022**, 103, 389–401.

[11] 33. Ge, Y.; Zhao, S.; Zhou, H.; Pei, C.; Sun, F.; Ou, W.; Zhang, Y. Understanding echo chambers in e-commerce recommender systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020.

[12] Himeur, Y.; Sayed, A.; Alsalemi, A.; Bensaali, F.; Amira, A.; Varlamis, I.; Eirinaki, M.; Sardianos, C.; Dimitrakopoulos, G. Blockchain-based recommender systems: Applications, challenges and future opportunities. Comput. Sci. Rev. **2022**, 43, 100439.

[13] Singh, M. Scalability and sparsity issues in recommender datasets: A survey. Knowl. Inf. Syst. **2018**, 62, 1–43.

[14] Vellaichamy, V.; Kalimuthu, V. Hybrid Collaborative Movie Recommender System Using Clustering and Bat Optimization. Int. J. Intell. Eng. Syst. **2017**, 10, 38–47.

[15] Wang, Q.; Ma, Y.; Zhao, K.; Tian, Y. A Comprehensive Survey of Loss Functions in Machine Learning. Ann. Data Sci. **2020**, 9, 187–212.

[16] Farashah, M.V.; Etebarian, A.; Azmi, R.; Dastjerdi, R.E. A hybrid recommender system based-on link prediction for movie baskets analysis. J. Big Data **2021**, 8, 32.

[17] Ortega, F.; Mayor, J.; López-Fernández, D.; Lara-Cabrera, R. CF4J 2.0: Adapting Collaborative Filtering for Java to new challenges of collaborative filtering based recommender systems. Knowl. Based Syst. **2020**, 215, 106629.

[18] Al-Bakri, N.F.; Hashim, S.H. Reducing Data Sparsity in Recommender Systems. Al-Nahrain J. Sci. **2018**, 21, 138–147.

[19] Shen, J.; Zhou, T.; Chen, L. Collaborative filtering-based recommendation system for big data. Int. J. Comput. Sci. Eng. **2020**, 21, 219–225.

[20] Kalita, K.; Dey, P.; Joshi, M.; Haldar, S. A response surface modelling approach for multi-objective optimization of composite plates. Steel Compos. Struct. **2019**, 32, 455–466.

[21] Katarya, R.; Verma, O.P. An effective collaborative

movie recommender system with cuckoo search. Egypt. Inform. J. **2017**, 18, 105–112.

[22] Kumar, B.; Sharma, N. Approaches, Issues and Challenges in Recommender Systems: A Systematic Review. Indian J. Sci. Technol. **2016**, 9, 1–12.

[23] Colomo-Palacios, R.; García-Peñalvo, F.J.; Stantchev, V.; Misra, S. Towards a social and context-aware mobile recommendation system for tourism. Pervasive Mob. Comput. **2017**, 38, 505–515.

[24] Alyari, F.; Navimipour, N.J. Recommender systems: A systematic review of the state of the art literature and suggestions for future research. Kybernetes **2018**, 47, 985.

[25] Caro-Martinez, M.; Jimenez-Diaz, G.; Recio-Garcia, J.A. A theoretical model of explanations in recommender systems. In Proceedings of the ICCBR, Stockholm, Sweden, 9–12 July 2018.

[26] Gupta, S. A Literature Review on Recommendation Systems. Int. Res. J. Eng. Technol. **2020**, 7, 3600–3605.

[27] Abdulla, G.M.; Borar, S. Size recommendation system for fashion e-commerce. In Proceedings of the KDD Workshop on Machine Learning Meets Fashion, Halifax, NS, Canada, 14 August 2017.

[28] Aggarwal, C.C. An Introduction to Recommender Systems. In Recommender Systems; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1–28.

[29] Jóhann, Þorvaldsson, Koskinen, P., Meer, P. van der, Steiner, M., & Keller, T. Improving Graduation Rates in Engineering Programs Using Machine Learning. Kuwait Journal of Machine Learning, 1(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/110

[30] Chaudhury, S., Dhabliya, D., Madan, S., Chakrabarti, S. Blockchain technology: A global provider of digital technology and services (2023) Building Secure Business Models Through Blockchain Technology: Tactics, Methods, Limitations, and Performance, pp. 168-193.