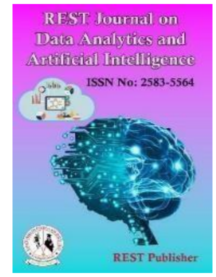
REST Journal on Data Analytics and Artificial Intelligence
Vol: 4(2), June 2025
REST Publisher; ISSN: 2583-5564
Website: http://restpublisher.com/journals/jdaai/

DOI: https://doi.org/10.46632/jdaai/4/2/5



Recommendation Systems: A Comprehensive Survey

***Sachin Soni, Sanjay Shah**
Kadi Sarva Vishwavidyalaya, Gandhinagar, Gujarat, India.
Corresponding author: *Sachinsoni.cs@gmail.com*

Abstract*: To assist users in discovering goods, services, and content, recommendation systems have become a critical part of modern digital landscapes. The fundamental concepts, methods, challenges, and advancements in recommendation systems are discussed within this survey. We review a number of types of recommendation systems, including collaborative, content-based, and hybrid schemes. We also discuss the techniques that enable these systems, such as deep learning, matrix factorization, and graph-based methods. Industry applications, open problems, emerging trends, and key evaluation measures are all thoroughly examined*.

Keywords: *Content Filtering, Collaborative Filtering, Hybrid, LLM*

1. INTRODUCTION

Recommendation systems (RS) are algorithms that suggest the most suitable items for users, like products, films, or music [1]. Their applicability extends to e-commerce, entertainment, healthcare, and learning. As the amount of available information increases, RS become crucial for personalized content provision. The present paper offers a full review of recommendation systems, including methods, metrics for assessment, difficulties, and prospective future research directions.

2. BACKGROUND AND DEVELOPMENT OF RECOMMENDATION SYSTEMS

Recommendation systems have their genesis in the early 1990s with the employment of simple rulebased models. As machine learning and data mining improved, more advanced methods came into place [2]. Mid-1990s saw the arrival of collaborative filtering (CF), which was then joined by contentbased filtering (CBF) and then hybrid approaches. Big data and deep learning have further transformed RS capabilities [3].

3. RECOMMENDATION SYSTEM TYPES

Recommendation systems can be generally categorized into three broad categories: Content-Based Filtering, Collaborative Filtering, and Hybrid Approaches. These approaches vary in the way they provide re commendations to users.

A. Content-Based Filtering

Content-based filtering (CBF) suggests items to users based on the content of the items they have already engaged with and recommends similar items. It targets the item features or item attributes instead of depending on user interactions. [4]

Key Concepts:

- User Profile: CBF stores a profile per user in terms of their interaction history. The profile is made up of the features (or attributes) of the items a user has interacted with, e.g., genre, tags, keywords, or other such attributes.

- Item Features: Item features are compared directly. For example, in a movie scenario, features may be genre, director, actors, etc.

Techniques:

- TF-IDF (Term Frequency-Inverse Document Frequency): It is a common method for text content (such as movie synopsis or product descriptions). It measures the significance of a word in a document against a set of documents.
- Cosine Similarity: A metric that computes the cosine of the angle between two vectors. It is commonly used to calculate the similarity between item feature vectors.
- Keyword Matching: This means matching items due to shared keywords in their description, like matching books with comparable subjects or titles of movies.

Example:

- Movie Recommendation: A content-based movie recommendation would consider a user's previous preference (e.g., if they enjoyed sci-fi films like Interstellar), then suggest other movies in the same. genre, maybe from the same director or having the same actors.
- E-commerce: Amazon tends to apply content-based filtering by reviewing the items a user has shopped for or viewed before and suggesting similar products on the basis of attributes such as category, price range, or brand.

Benefits:

- No Cold Start Problem for Users: New users do not require past interaction history data, only a minimal profile.
- Transparency: It is simple to describe why a specific item was suggested on the basis of its attributes. Drawbacks:
- Serendipity is restricted: Because the system only offers things that resemble items the user likes, fewer chances exist to interact with novel things.
- Overfitting: The system is inclined to focus on the past like of the user too heavily and not propose those things, although the user may like them eventually but haven't interacted with them as of yet.

B. Collaborative Filtering

One of the most widely used types of recommendation is Collaborative Filtering (CF). It recommends content by identifying people with similar preferences and recommending products that they liked [5].

Key Concepts:

- User-based Collaborative Filtering: It recommends items by finding users who have shared interaction history. The premise is that if user A and user B have rated nearly identical items in the past, user A will enjoy items that user B has enjoyed.
- Item-based Collaborative Filtering: It suggests items that are similar to the ones that the user has previously engaged with. It emphasizes identifying items that have been liked by similar users.

Techniques:

- Nearest Neighbor Search: The approach calculates the similarity of users (user-based CF) or objects (item-based CF). These similarities are calculated using cosine similarity, Pearson correlation, and Jaccard similarity.
- Matrix Factorization: An emerging technique used for collaborative filtering where a sparse user-item matrix is factored into two sparse matrices. Commonly used approaches include Singular Value Decomposition (SVD) and Alternating Least Squares (ALS).
- K-Nearest Neighbors (K-NN): In CF, this technique identifies the K nearest users or items and employs their interactions to make recommendations.

Example:

a. Netflix: Netflix's recommendation system mainly employs collaborative filtering to recommend TV shows or movies based on the viewing history of users who are similar to you. If lots of users who watched Stranger Things also liked Dark, it would suggest Dark to a user who watched Stranger Things.
b. Amazon: Collaborative filtering is employed by Amazon to suggest items on the basis of other users' behavior with similar buying patterns.

Advantages:

- Effective at Scaling: Collaborative filtering can handle big data without any manual feature extraction.
- Discovery of Novel Items: Users have a higher chance of discovering varied items since the system is not dependent on item features alone.

Limitations:

- Cold Start Problem: Collaborative filtering has a problem with new users or new products since there is not enough interaction history to provide recommendations.
- Sparsity: In big systems, most of the users could have interacted with only a small set of items, hence resulting in sparse data and making it challenging to identify the correct recommendations.

C. Hybrid Systems

Hybrid systems integrate more than one recommendation method to take advantage of each and reduce the drawbacks. The aim is to improve recommendation accuracy and solve problems such as the cold start problem, sparsity, and low diversity [6].

Key Concepts:

- Weighted Hybridization: Various recommendation algorithms are employed, and each algorithm is given a weight. The ultimate recommendation is the aggregation of weighted scores across varied algorithms.
- Switching Hybridization: Depending on some condition or situation, the system alternates among various forms of recommendations. For instance, it can employ content-based filtering if a user has sparse interaction history and collaborative filtering when there is sufficient user information.
- Feature Augmentation: In this method, the output of one method is taken as an extra feature in another method. For instance, content-based filtering's output can be taken as an extra input to collaborative filtering.

4. TECHNIQUES AND MODELS IN RECOMMENDATION SYSTEMS

The performance of a recommendation system relies greatly on the techniques and models utilized to create recommendations. The techniques can be of varying complexity, from basic methods such as nearest neighbor algorithms to complex techniques such as deep learning. Some of the most commonly used techniques are listed below: A. Matrix Factorization Matrix factorization is a method often employed in collaborative filtering; especially product/movie/other item recommendation based on user-item interaction information. It is very effective in dealing with sparse data, which is common in big scale recommendation systems.[7]

Key Concept:

- The overall principle behind matrix factorization is to factorize the user-item interaction matrix (a matrix in which each row corresponds to a user, and each column corresponds to an item) into two lower dimensional matrices that capture latent user and item factors. Latent factors are the hidden patterns within user preferences and item properties.

Techniques:

- Singular Value Decomposition (SVD): A widely used matrix factorization method that breaks down the user-item matrix into three matrices: user factors, item factors, and singular values. It reduces the dimensionality while keeping the vital data intact. Example: In Netflix's recommendation system, SVD can factor a user-movie matrix into latent factors such as user preferences and movie attributes (e.g., genre, director).

- Alternating Least Squares (ALS): A matrix factorization technique that alternates between solving for item factors and user factors to optimize the interaction matrix. It's commonly applied to implicit feedback systems (where ratings aren't provided but are instead inferred from actions such as clicks or views).

Advantages:
- Suits large-scale datasets.
- Dimensionality reduction facilitates the discovery of underlying patterns in user tastes and item attributes.
- Scalable to process billions of items and users.

Limitations:
- Cold Start Problem: Matrix factorization can't handle new users or products with too little data.
- Computationally intensive and may need to be optimized. B. Deep Learning Techniques Deep learning techniques are popular nowadays since they can easily represent complex, non-linear relations between data. Deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are applied to recommendation problems more and more. [8]
- Deep learning models are capable of automatically extracting features from raw data, and hence they are extremely effective for tasks such as image, text, or sequential recommendation. Deep models are especially strong in identifying complex interactions that might be lost by simpler models such as collaborative filtering.

Techniques:

Deep Neural Networks (DNNs): They are employed to make predictions about user-item interactions by learning a hierarchical feature representation from the data. For example, autoencoders, which are a form of neural network, can be employed for collaborative filtering, learning compressed user and item representations.

Example:
YouTube employs deep learning to suggest videos by examining both video content (e.g., descriptions, metadata) and user behavior.
- Convolutional Neural Networks (CNNs): CNNs, which are traditionally employed for image recognition, may also be employed in recommendation systems, particularly content-based filtering. Through learning of image, video, and product photo features, CNNs are able to make recommendations based on visual content. Example: For online stores such as Amazon, CNNs examine product photos to recommend visually similar products to customers.
- Recurrent Neural Networks (RNNs): RNNs are utilized in sequential recommendation tasks. RNNs are especially effective when it comes to forecasting future action from past activity, such as suggesting the next movie a person will watch from their viewing record.

Example:
- RNNs are applied by Netflix in forecasting what shows or movies an individual will most likely watch next from their previous viewing history.
- Flexible: Deep learning is capable of processing various data types, such as structured data (purchase history, ratings) and unstructured data (text, images).
- Suitable for intricate patterns: They are able to recognize non-linear data relationships that may be overlooked by conventional models.

Limitations:
Needs large datasets to train well, thus is not ideal for compact systems. •Computationally demanding and needs high hardware, particularly in applications at large scales.

D. Graph-Based Methods

Graph-based methods describe users, items, and their interaction as nodes and edges in a graph. The methods encode interactions between users and items as well as relationships between users or items. Graph-based models are most effective in representing complicated relationships in recommendation tasks.

Key Concept:

- Users and items are nodes in a graph, and the user item interactions (e.g., ratings, clicks, purchases) are edges. These methods can then do graph traversal or node embeddings to make predictions of user preferences.

Techniques:

- Graph Neural Networks (GNNs): GNNs have been very promising in recommendation systems. They enable the aggregation of information from neighboring nodes in the graph, which aids in capturing the local and global structure of the recommendation network. [9,10,11]

Example: GNNs can be used in social networks or e-commerce websites, where users engage with various products and other users. The model can then make predictions about which products to recommend based on user-item connections.

- Personalized PageRank (PPR): An extension of the PageRank algorithm, PPR orders items in terms of their proximity to the user in the graph. It's applied to identify the most relevant items by traversing the graph structure.

Example: For a social network site, PPR can be employed to suggest content by evaluating how close a piece of content is to a user, on the basis of social interactions or similar user activity.

Advantages:

- Efficiently captures complex relationships: GNNs are very effective in capturing complicated relationships among users, items, and other system entities.
- Versatile: They can be used for both user-user and item-item interaction graphs, making them versatile for many types of recommendation systems (e.g., product recommendations, social recommendations). Shortcomings:
- Computational scalability problems: GNNs become computationally intensive with extremely large datasets because of the nature of graph computations.
- Needs meticulous graph building and can be harder to apply than other techniques.

E. K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) is a basic yet effective instance-based learning algorithm applied to content-based as well as collaborative filtering recommendations. The central idea is to recommend items by identifying the K most relevant users or items to the target user.

Key Concept:

- For user-based K-NN, the system finds K users who are similar to the target user and recommends the items these similar users liked. For item-based KNN, the system recommends K similar items for the interactions of the user.

Techniques:

- User-based K-NN: For instance, if user A has liked movies X and Y, and user B has also liked movie X, then user A will be regarded as a neighbor of user B. Depending on the movies user A has liked, the system can suggest new movies to user B.

- Item-based K-NN: In this, the system suggests items that are similar to the ones the user has already engaged with. If a user has viewed Matrix, the system may suggest Inception based on genre, director, or theme similarities.

Advantages:

- Easy to implement and comprehend.
- Efficient for small data sets and has low training needs.

Limitations:

- Scalability: K-NN can become inefficient when the number of users and items increases because of the pairwise similarity calculations required.
- Cold Start Problem: K-NN can have issues with new users or items that lack enough interaction history.

5. RECOMMENDATION SYSTEM CHALLENGES

A. Cold Start Problem: Cold start problem is the issue of recommending products to new users or products, as there is not enough data to make good predictions [12].
B. Data Sparsity: Sparse interaction matrices present problems for classic collaborative filtering algorithms since there may be too few user-item interactions to produce solid recommendations [13].
C. Scalability: As datasets get larger, recommendation systems have to scale efficiently and, in the process, require better algorithms and computational resources [14].
D. Fairness and Bias: Data bias can result in biased recommendations. Maintaining fairness in recommendations, especially for diverse user groups, is an area of interest with growing relevance [15].
E. Ethical Concerns
   - Filter Bubbles: Over personalization restricts exposure to diverse content.
   - Transparency: Users call for justifications of recommendations (e.g., "Why was this suggested?").

Recent Trends

A. Explainable Recommendations: Explainable AI (XAI) methods enable transparency within recommendation systems by presenting comprehensible explanations for suggestions proposed to consumers [16].
B. Federated Learning: Federated learning allows for privacy-guaranteed recommendation models by making it possible for decentralized training without transferring user information across systems [17].
C. Reinforcement Learning: Reinforcement Learning (RL) formulates recommendation systems as sequential decisionmaking problems, and models learn how to enhance recommendations with time depending on user feedback [18].
D. Large Language Models (LLMs)
   - Intent Understanding: LLMs such as GPT4 process unstructured text (e.g., reviews, questions) to derive preferences.
   - Conversational RS: Facilitates natural-language interactions (e.g., shopping assistants powered by ChatGPT).

Applications Across Industries

A. E-commerce: Amazon employs recommendation systems to create sales by providing users with product recommendations that match their previous buy and browse records [19].
B. Entertainment: Recommendation systems utilized by Netflix and Spotify provide targeted content to the users based on their viewing and listening habits [20].
C. Education: Recommendation systems in educational institutions recommend products and study material to users by analyzing their school background and studies [21].
D. Healthcare: Personalized treatment prescriptions in medicine apply patient information to recommend best treatments and diagnoses by individual traits [22].

Future Directions

1. Cross-Domain Recommendations: Transfer learning to map models across domains (e.g., movie to book tastes).
2. Ethical AI Frameworks: Fairness metrics standardization and regulatory adherence.
3. Quantum Computing: Solving optimization problems on large scales in MF.
4. Integration with IoT: Realtime recommendation based on data from wearable devices.

6. CONCLUSION

Recommendation systems have come a long way with the incorporation of advances from graph theory, machine learning, and deep learning. Active research fields include cold start, bias, and data sparsity. Topics for future consideration are explainability, privacy-protecting models, and systems based on reinforcement learning

REFERENCES

[1]. G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, 2005.
[2]. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in Proc. ACM Conf. Computer Supported Cooperative Work, 1994.
[3]. X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, 2009.
[4]. M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in The Adaptive Web, Springer, 2007, pp. 325–341.
[5]. J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI), 1998.
[6]. R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.
[7]. Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.
[8]. W. Zhang, T. Du, and J. Wang, "Deep Learning over Multi-Field Categorical Data," in Proc. European Conf. Information Retrieval (ECIR), 2016, pp. 45–57.
[9]. R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," in Proc. 24th ACM SIGKDD Intl. Conf. Knowledge Discovery & Data Mining (KDD), 2018, pp. 974–983.
[10]. J. McNee, S. Riedl, and J. Konstan, "Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems," in Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI), 2006, pp. 1097–1101.
[11]. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," in Proc. 25th Conf. Uncertainty in Artificial Intelligence (UAI), 2009.
[12]. S. Schein, A. Popescul, L. Ungar, and D. Pennock, "Methods and Metrics for Cold-Start Recommendations," in Proc. 25th Annual Intl. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2002.
[13]. H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King, "Recommender Systems with Social Regularization," in Proc. 4th ACM Intl. Conf. Web Search and Data Mining (WSDM), 2011, pp. 287–296.
[14]. J. Leskovec, L. A. Adamic, and B. A. Huberman, "The Dynamics of Viral Marketing," ACM Trans. on the Web, vol. 1, no. 1, pp. 5:1–5:39, 2007.
[15]. M. D. Ekstrand, M. Ludwig, J. T. Konstan, and J. Riedl, "All the Cool Kids, How Do They Fit In? Popularity and Demographic Bias in Recommender Evaluation and Effectiveness," in Proc. 12th ACM Conf. Recommender Systems (RecSys), 2018, pp. 172–176.
[16]. Q. Zhang and B. Zhu, "Explainable Recommendation: A Survey and New Perspectives," Foundations and Trends in Information Retrieval, vol. 14, no. 1, pp. 1–101, 2020.
[17]. P. Hardy, A. Fawaz, M. Baracaldo, and R. Zhou, "Private Recommender Systems: An Overview and Research Opportunities," arXiv preprint arXiv:2003.06229, 2020.
[18]. X. Zhao, L. Zhang, Y. Liu, J. Xie, J. Zhao, and W. Lu, "Deep Reinforcement Learning for List-wise Recommendations," in Proc. Intl. Conf. Learning Representations (ICLR), 2018.
[19]. G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[20]. C. D. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," ACM Trans. Management Information Systems, vol. 6, no. 4, pp. 1–19, 2016.
[21]. M. Thai-Nghe, L. Drumond, T. Horvath, A. Krohn-Grimberghe, and L. Schmidt-Thieme, "Matrix and Tensor Factorization for Predicting Student Performance," in Proc. Intl. Conf. Computer Supported Education, 2011, pp. 144–151.
[22]. J. Sun, D. Wang, S. Hu, and H. Edabollahi, "Patient Outcome Prediction through Integrated Modeling of Patient Data and Knowledge," PLoS One, vol. 7, no. 11, pp. e48264, 2012.