भारतीय सूचना प्रौद्योगिकी संस्थान, नागपुर
Indian Institute of Information Technology, Nagpur
"An Institution of National Importance by an Act of Parliament"
Survey No. 140,141/1 Behind Br. Sheshrao Wankhade Shetkari Soot Girni, Village: Waranga,
Po: Dongargaon (Butibori), District : Nagpur (Maharashtra) -441108
Website: www.iiitn.ac.in Email: director@iiitn.ac.in, registrar@iiitn.ac.in Phone: 9405215010

Topic: Online Learning Algorithms

Group: G7

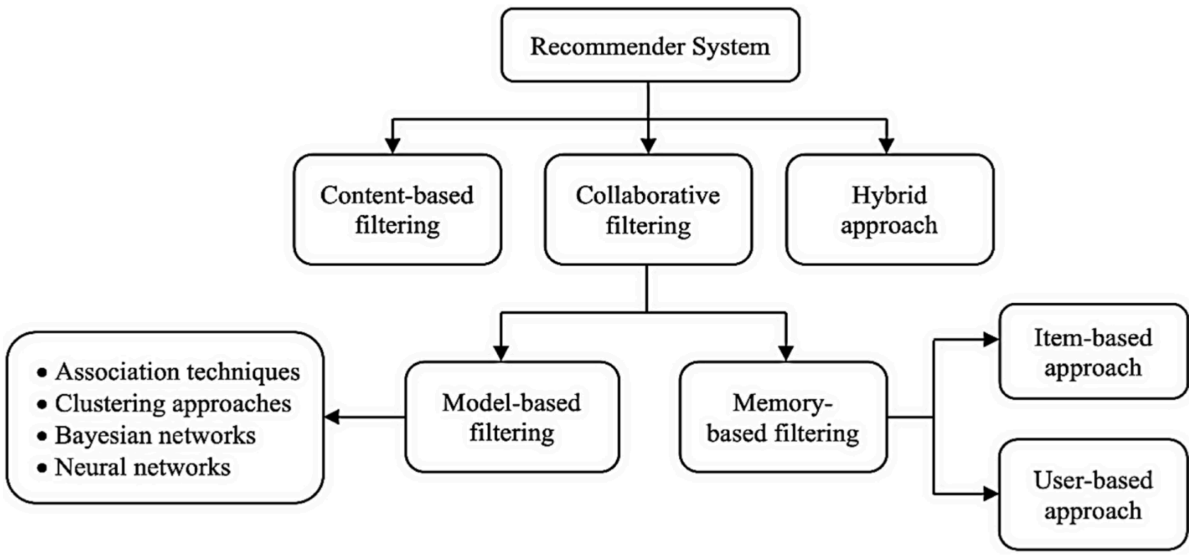
Members:
Om Varma [BT22CSE216]
Vinayak Gupta [BT22CSE222]
Kamal Chandra [BT22CSE223]
Harsh Agarwal [BT22CSE228]

Under the guidance of:
Dr. Jitendra V. Tembhurne

Abstract
Online learning represents an important family of machine learning algorithms, in which a learner attempts to resolve an online prediction (or any type of decision-making) task by learning a model/hypothesis from a sequence of data instances one at a time [1]. Recommender systems are an extensive class of Web Applications that aim to forecast user responses to options by considering preferences and objective behaviours [2]. Owing to their ability to enhance user experience by streamlining the decision-making process, these have become indispensable tools in various domains. However, despite their widespread adoption, recommender systems encounter certain impediments to their efficacy, accuracy and reliability. Some examples include the cold start, sparsity and over-specialisation problems. This paper investigates the underlying reasons behind the prevalence of these challenges and offers insights into potential avenues for mitigating their impact. Through a systematic review of existing literature and empirical analysis, this study illuminates the nuanced complexities inherent in recommender systems and underscores the imperative of addressing their limitations to realise their full potential.

Introduction
Recommender systems leverage algorithms to analyse user preferences and behaviours, thereby facilitating personalised recommendations tailored to individual users' interests. Despite their pervasive adoption and evident utility, recommender systems grapple with multifaceted challenges that undermine their effectiveness and reliability.

Types of recommender systems
Recommender systems are broadly categorised into three different types namely, content-based recommender systems, collaborative recommender systems and hybrid recommender systems.



Content-based recommender system
All the data items are collected into different item profiles based on their description or features. When a user gives a positive rating to an item, then the other items present in that item profile are aggregated together to build a user profile. Items present in the user profile are then recommended to the user.

Collaborative filtering-based recommender system
Collaborative filtering recommender systems analyse user behaviour to make predictions about items or products they might like. It works by collecting and comparing preferences or past interactions to find similar patterns among users. By identifying similarities, it recommends items that other users with similar preferences have liked or interacted with. This approach doesn't require explicit knowledge of items' features but relies on user feedback to generate recommendations, making it effective for diverse domains like movies, music, and e-commerce.

Hybrid filtering
Hybrid filtering combines multiple recommendation techniques, such as collaborative filtering, content-based filtering, and sometimes knowledge-based approaches, to enhance recommendation accuracy and coverage. By leveraging the strengths of different methods, hybrid systems can overcome individual limitations and provide more personalised and diverse recommendations. This approach is widely used in various applications, including e-commerce, streaming platforms, and news websites, to offer users tailored and relevant suggestions.

Challenges Encountered by Recommender Systems
This section briefly describes the various challenges present in current recommender systems.

1. Cold Start Problem
The cold start problem occurs when a recommender system struggles to provide accurate recommendations for new users or items with limited data. For new users, the system lacks historical interaction data to understand preferences, while for new items, there's insufficient feedback to gauge their relevance to users. This results in less personalised and potentially inaccurate recommendations, impacting user satisfaction and system performance. [3]

2. Sparsity Problem
The sparsity problem in recommender systems arises when the available data matrix, representing user-item interactions, contains mostly empty or missing entries. In such cases, the system struggles to accurately infer user preferences or item similarities due to the lack of sufficient information. This sparsity hampers recommendation quality and can lead to poor user experiences.

3. Overspecialization Problem
This issue arises when the suggested items exhibit too much similarity to one another. For instance, consider a scenario where a user frequently purchases groceries from an online shopping platform. Upon visiting the website, they consistently receive recommendations for sugar, albeit from different brands, due to its high frequency of purchase. Seeing the same item repeatedly without any variation may lead the user to lose interest in the recommendations, potentially prompting them to explore other platforms that offer more diverse and engaging suggestions. [4]

4. Shilling Attacks
Shilling attacks are manipulative actions aimed at corrupting the integrity of recommender systems by injecting fake or biassed ratings or reviews. Perpetrators create numerous fake user accounts or artificially inflate ratings to promote certain items or demote others, distorting the system's recommendations. Detecting and mitigating shilling attacks is crucial to maintaining the trustworthiness and effectiveness of recommender systems.

5. Scalability
Scalability is the property of the system which the weather system will be able to cope with when the system grows. In the case of recommender systems, scalability can be understood as a situation where a recommender system is performing very well in the case of a few users like 1000 users but as the user grows to 10000 or 100000 it starts performing in a way which is not desirable. When the system faces scalability issues it becomes slow it starts feeling it start giving problems which it has never given when a load of users recommendation were less. [4]

6. Latency Problem
Latency issues arise in collaborative filtering (CF) recommenders when new items are added frequently to the database, as they are only able to suggest items already rated, leaving new additions unexplored. While content-based (CB) filtering can mitigate waiting times, it may lead to overspecialization. To tackle this challenge, a category-based approach, coupled with user stereotypes, proves effective. Moreover, leveraging clustering techniques and offline calculations can boost performance, while model-based CF approaches offer scalability and enhanced efficiency.

7. Synonymy
The synonymy problem in natural language processing occurs when different words or phrases have similar meanings but are not recognized as such by the system. This leads to ambiguity and inconsistency in understanding text, affecting tasks like information retrieval and sentiment analysis. The variation in using descriptive terms is greater than

commonly thought and the excessive usage of synonym words decreases the performance of CF recommenders. Since item contents are thoroughly ignored, therefore, the recommender does not consider the latent association between items. This is the reason why new items are not recommended as long as these are rated by the users.

Problem Statement

Our work centres around tackling the **Sparsity Problem** within the context of recommender systems. Additionally, we aim to address the **Cold Start Problem**, which is a distinct subset of the broader Sparsity Problem. Our work will concentrate on developing methodologies, algorithms, or frameworks to mitigate the detrimental effects of data sparsity and the Cold Start Problem, thereby enhancing the effectiveness and efficiency of recommender systems.

Research Gaps

In the recent decade, recommender systems have arisen as tools for mitigating information overload, showcasing their utility across a spectrum of disciplines. Nonetheless, these systems confront a multitude of challenges elucidated below:
1. Deployment challenges such as cold start, scalability, sparsity.
2. Enhancing Real-Time User Feedback Handling: Developing techniques to improve the handling of real-time user feedback in recommender systems, focusing on capturing and processing feedback efficiently to enhance recommendation quality in dynamic environments.
3. Developing standardised evaluation criteria or methodologies to assess the performance of different recommender systems comprehensively, facilitating fair comparisons across systems and domains.
4. Integration of deep learning models into recommender systems, exploring new variants of hybrid meta-heuristic approaches to improve recommendation quality and efficiency.
5. Developing Evaluation criteria or metrics to assess user satisfaction in real-time, enabling the evaluation of recommender system performance based on user feedback and preferences as they interact with the system.
6. Application of recommender systems in diverse contexts beyond traditional domains, such as psychology, mathematics, and computer science, including methods to adapt recommender algorithms to different contexts effectively.

Literature Review

- Fu Jie Tey et al. presented a social network-based recommender system, which is a user-centred recommender system to exclude the products that users are disinterested in according to user preferences and their friends' shopping experiences so as to make recommendations effective.
- YiBo Chen et al. proposed direct and indirect similarities between users and computed similarity matrix through the relative distance between user's rating, realised a new collaborative filtering approach to alleviate the sparsity problem and improved the quality of recommendation. [11]
  Conceptually, the cold-start problem can be viewed as a special instance of the sparsity problem, where most elements in certain rows or columns of the consumer–product interaction matrix A are 0 [5].
- Another proposed approach, dimensionality reduction, aims to reduce the dimensionality of the consumer–product interaction matrix directly. A simple strategy to reduce the dimensionality is to form clusters of items or users and then use these clusters as basic units in the prediction. More advanced techniques can be applied to achieve dimensionality reduction. Examples are statistical techniques such as Principal Component Analysis (PCA) [9]. Essentially, dimensionality reduction approaches deal with the sparsity problem by generating a denser user-item interaction matrix that considers only the most relevant users and items. Predictions are then made using this reduced matrix. Empirical studies indicate that dimensionality reduction can improve recommendation quality significantly in some applications, but performs poorly in others, the potentially useful information might be lost during this reduction process [6].
- Researchers have also attempted to combine collaborative filtering with content-based recommendation approaches to alleviate the sparsity problem [7][8]. In addition to user-item interactions, such techniques also consider similarities between items derived from their content, which allow them to make more accurate predictions. However, the hybrid approach requires additional information regarding the products and a metric to compute meaningful similarities among them. In practice, such product information may be difficult or expensive to acquire and a related similarity metric may not be readily available. Another category of methods consider the data as a bipartite graph where nodes represent the users and items, and an edge (i, j)

exists between a user i and an item j if i has rated j. Moreover, edge (i, j) is given a weight corresponding to the rating given by i to j. These methods then derive global similarities between users or items using graph theoretic measures. For instance, one such method computes similarities between two users as the average commute time between their respective nodes in a random-walk of the graph. Other graph theoretic measures were also investigated, such as the minimal hop distance between nodes of the graph, and the spread activation of the nodes in the graph. The main drawback of these approaches is that there is often no good interpretation of the similarity measures in the context of the prediction problem [10].

Studies for the Recommendation Systems to Alleviate the Cold-Start Problems

- Prior to applying the original matrix (OM) to collaborative filtering (CF), filter the OM based on the item category information and regenerate the matrix into a more suitable form for the user. In the conventional method, the OM is applied to CF as input and the prediction results are derived. Compared to the conventional CF, here analyse the user selection propensity and extract a matrix that reflects the user preferences from the OM. First extract the category ratio of the selected items by user and regenerate the OM based on the category percentage. The regenerated matrix (RM) is assumed to be more user-appropriate than the OM and is applied to the CF. We conduct experiments using the MovieLens database, and consider the genres that exist in movie information as the category information. Accordingly, we take the movie database as an example to explain the proposed method.
- Hybrid systems addressing collaborative filtering (CF) and content-based filtering (CBF) in various manners. CBF has been applied not only to e-commerce and e-learning, but also to news recommendation and user preference analysis [12–16]. These approaches utilised item or user features, such as category or demographic information [17–20]. Several CBF methods have used item or user features, such as category or demographic information to deal with cold-start problems for new items or new users [21-23]. The studies utilise item or user features in recommendation processes. In other words, items are classified based on the item features such as category and used to derive the recommendation results, or features are used as input from deep neural networks to learn the features together with numerical values [24-25].
- In addition to this, there is also a method of reconfiguring the input matrix by reusing the results of the Matrix Factorisation to improve the performance of the recommendation [26].
- Gantner et al. [27] attempted to cluster new items with no user responses by addressing the feature data. The clustering method based on feature data mitigated the item-side cold-start problem; that is, the authors applied CBF concepts using side information, such as item features to mitigate cold-start in recommender systems.
- Sun et al. [28] clustered items using the attribute data and preferences, and created a decision tree that could be applied to new and existing items, and could predict preferences for new items.
- In [29], the authors proposed a Bayesian network model incorporating user, item, and feature nodes. The proposed model was based on a combination of CF and CBF, as it used various features to derive predictions through CF. Superior recommendation quality was provided based on the proposed model.
- In [30], the authors constructed user features based on the action history of the users, following which the similarities between users and the items (website content) were derived to recommend items.
- Duong et al. [31] generated the tag genome of movie data by applying a natural language processing (NLP) technique. The authors also proposed a three-layer autoencoder to create a more compact representation of the tags. Thereafter, they provided recommendation results by implementing MF.
- Chen et al. [32] proposed a hybrid recommendation algorithm. They used a latent Dirichlet allocation topic model to reduce the user data dimension and generated a user theme matrix that could reduce the data sparsity for CF. The VGG16 deep learning model was used to extract the feature vectors. The generated matrix and vectors were used as input for content based recommender systems, following which the recommendation results were derived.
- Mehrabani et al. [33] proposed a method to extract the item features as words based on the NLP method word2vec. The vectors were used to calculate the similarities between features. After calculating the similarities, the proposed system derived the recommendation results according to the content-based concept.

Studies for the Recommendation Systems to Improve the Sparsity Issues for Inputs

- Zhao et al. [34] proposed a new item-based CF algorithm based on Kullback–Leibler (KL) divergence to measure item similarity. They first try to improve the accuracy of similarity results. Then adjusted prediction results, more rating information is integrated with explicit user preferences in prediction processes. The results of the proposed algorithm show better recommendation quality in the sparsity dataset.
- Jiang et al. [35] propose a recommendation model for service API based on knowledge graph and collaborative filtering. They applied latent dimensions in collaborative filtering for analysing the potential relations between mashups and APIs to reduce the impact of data sparsity. Based on the proposed model, authors have significantly improved the accuracy of service recommendation.
- Ahmadian et al. [36] propose a novel recommendation method to address the issues that the existing recommendation methods focused on accuracy of recommendation without the time factor of users. The proposed method first incorporates the temporal issues Mathematics 2023, 11, 292 5 of 26 based on the effectiveness of the users' rating by utilising a probabilistic approach. They measure the quality of the prediction with respect to the changes of users' preferences over time since the proposed method addresses temporal reliability and data sparsity. Through their approaches, the method can remove ineffective users in the neighbourhood which means the set of similar users based on the changes of users' preferences over time. For this step, authors can show the temporal reliability of their recommendation approaches.
- Ajaegbu [37] focused on addressing the sparsity and cold-start situations in collaborative filtering by improving the conventional similarity measurements, such as Cosine similarity, Pearson correlation coefficient, and Adjusted cosine similarity. In existing collaborative filtering, by adjusting similarity measurement, the author improves the accuracy of recommendation results in sparsity and cold-start situations compared with the results of the conventional similarity measurements.
- Khaledian et al. [38] propose a trust-based matrix factorization technique (CFMT) that addresses trust networks in user data. They utilise the social network data in recommendation processes as trustees and trustees. By using the trust network and integrating ratings and trust statements authors alleviate the sparsity and cold-start problems in a recommendation model.
- Liu et al. [39] suggest a novel framework DAAN based on a deep adversarial and attention network. They tried to integrate model based collaborative filtering, which means matrix factorization with deep adversarial via an attention network. The proposed framework is leveraged to common features in two domains and adjusts the degree of effect between domain-shared and domain specific knowledge. Graph collaborative filtering methods that leverage the interaction graph based on users' preferences for items can positively affect the results of recommendation; however, the methods still have side effects, such as data sparsity in real situations. Although there are approaches to reduce the data sparsity using contrastive learning in graph collaborative filtering, the approaches conventionally construct the contrastive pairs ignored for the relationship between users or items.

Motivation

In order to alleviate the cold-start and sparsity problem many studies currently construct and propose a method of predicting users' preferences for items using information, such as the feature of items. In addition, various studies have been conducted to predict users' evaluation by extracting and analysing feature vectors of items using deep learning approaches. However, current studies are attempting research that transforms the shape of the existing system using metadata of users or items. Namely, it proposes methods that can produce users' evaluations in cold-start and sparse situations to produce more accurate results. The previous studies have the advantage of improving or alleviating the current recommendation performance or the problems of existing recommendation techniques. However, more analysis is required to use the proposed techniques in real situations. The approaches currently being studied can produce more accurate results in specific situations, but have the disadvantage of not having universality, such as conventional collaborative filtering. In addition, the sparsity problems persist since most studies still utilise the same input form.

Proposed Solution

Having multiple features in a model, suppose for a movie recommender system we have different features such as action, horror, thriller, etc. Some movies have been rated by users, most aren't, i.e., we have sparsity of data.
Some features are correlated to each other, for example a horror movie will have some component of thriller. This is useful, for instance, recommending horror and action movies to someone who likes thrillers.
We then create a rating matrix where each row represents a user and each column represents a movie. This matrix will store the ratings given by users to movies. The NaN (Not a Number) values indicate missing ratings, as not every user may have rated every movie.
Another matrix is initialised where each row represents a movie and each column represents a movie genre. This matrix helps in analysing the distribution of movie genres across the dataset.
Each cell in this matrix indicates whether a particular movie belongs to a specific genre. This is represented by binary values (0 or 1), where 1 indicates the presence of the genre and 0 indicates its absence.
Any NaN values in the rating matrix (which represent missing ratings) are filled with 0. This simplifies subsequent analysis by treating missing ratings as neutral (i.e., a rating of 0).
These matrices serve as the foundational data structures for various types of analyses, such as collaborative filtering for recommendation systems, genre-based analysis, or user-item matrix factorisation for dimensionality reduction.
For each user, an empty dictionary is initialised to store genre preferences for that user. The dictionary is initialised with genre names as keys and initial preference values set to 0 for each genre. We then filter the ratings dataframe to retrieve movies rated by the current user.
Given a movie, the genre preferences for the current user based on the movie genres and ratings are updated. Genre preferences are calculated by multiplying the genre value (1 if the movie belongs to the genre, 0 otherwise) by the rating given by the user for that movie and adding it to the corresponding genre preference.
After calculating genre preferences, we normalise them using Min-Max normalisation. This ensures that preferences are scaled between 0 and 1, making them comparable across users. This step prevents biases caused by users who might rate movies differently on average.
Generally, to find similarities between users we take dot products of their vectors. The closer two vectors are, the more similar their interests are, so we can suggest similar contents. However, a disadvantage is that we have to take dot products of all the pairs of users, and dot product itself is matrix multiplication. So, this entire process is time consuming.
After this, we identify users who have a strong preference (liking) for each movie genre, based on their normalised genre preferences derived earlier. This provides insights into user preferences, enabling targeted recommendations or genre-based analysis.
For each genre, users who have a preference greater than 0.65 (arbitrarily chosen threshold) are filtered for the current genre. This threshold indicates users who have a strong liking for that genre.
Then, the filtered users are sorted in decreasing order of their likeness for the current genre. Likeness is determined by the value of the genre preference for each user.
When a movie name is passed as input, the ID associated with it is looked up and the genres associated with the specified movie are retrieved.
Subsequently, we work on a variable called, 'recommendation score'. The recommendation score indicates how likely a user is to enjoy the specified movie based on their genre preferences.
For each user, it calculates a recommendation score based on the user's preferences for genres that match the genres of the specified movie. It iterates through the genres of the specified movie and checks if the user has a preference for each genre. If the user has a preference for a genre associated with the movie, it adds the preference value to the recommendation score.
Once recommendation scores are calculated for all users, it sorts the users based on their recommendation scores in descending order. Users with higher recommendation scores are prioritised as they are deemed more likely to enjoy the specified movie.
We also generate movie recommendations for a given user based on the preferences of similar users.
Users similar to the specified user are identified based on their genre preferences. For each genre, if the specified user is among the users who have a strong preference for that genre, it adds all other users who also like that genre to the set of similar users.

We then check the ratings of similar users for highly rated movies, here 3 (rated above this arbitrarily chosen threshold).
The 'recommended movies' set is updated with these movies.
The number of recommended movies are limited to the top n movies.

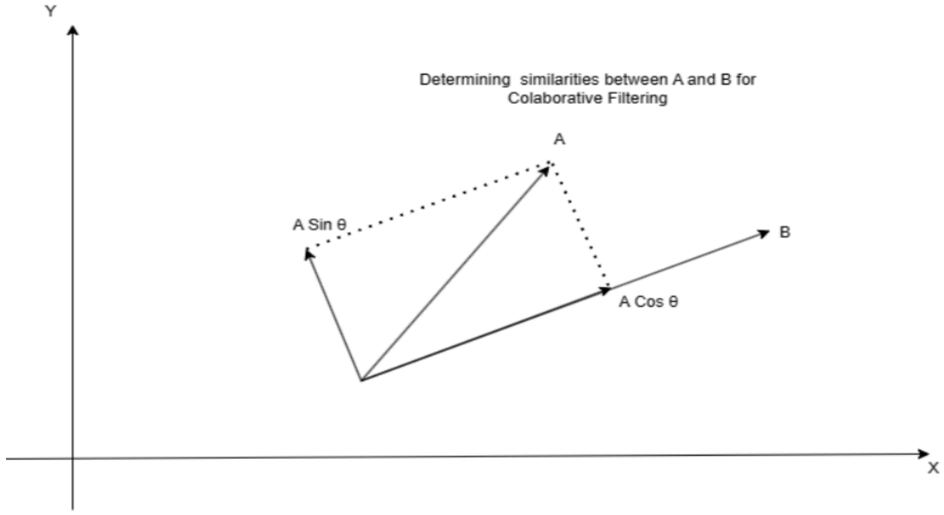
Reconsidering the dot product

Generally, to find similarities using cosine similarity between users we take dot products of their vectors. The closer two vectors are, the more similar their interests are, so we can suggest similar contents.
However, a disadvantage is that we have to take dot products of all the pairs of users, and dot product itself is matrix multiplication. So, this entire process is quite expensive.
On the other hand, we can just break the vector in its components and club the users with high interests on each axis to form a group of users interested in similar topics. Imagine each genre as a different axis, with the normalised score on each of them plotted for different users.
The advantage with this method is that we don't need to compare each pair here, also if two users have similar interests we know exactly in which category and how much they have similar interests.
For Example, if two users have similar interests but one user is more inclined to the other axis then recommending only those content which is extremely liked by the second user in the first category will be beneficial. This operation is not as expensive as the previous approach.

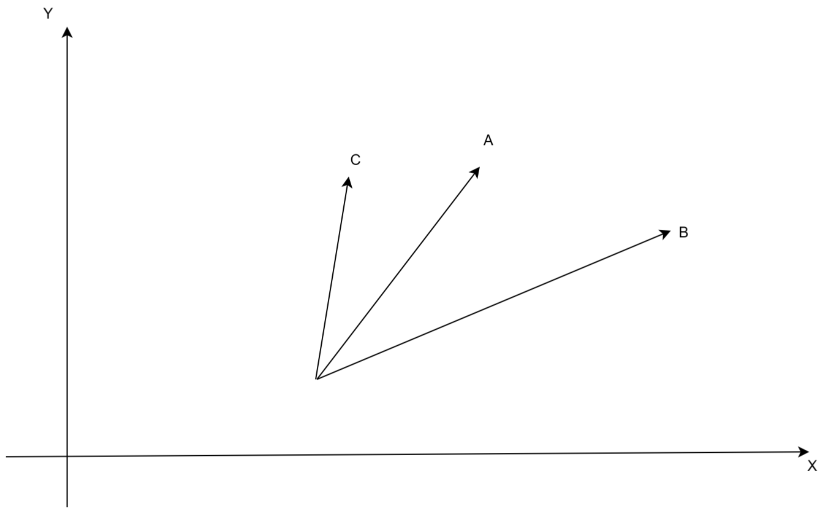
Time complexity Analysis

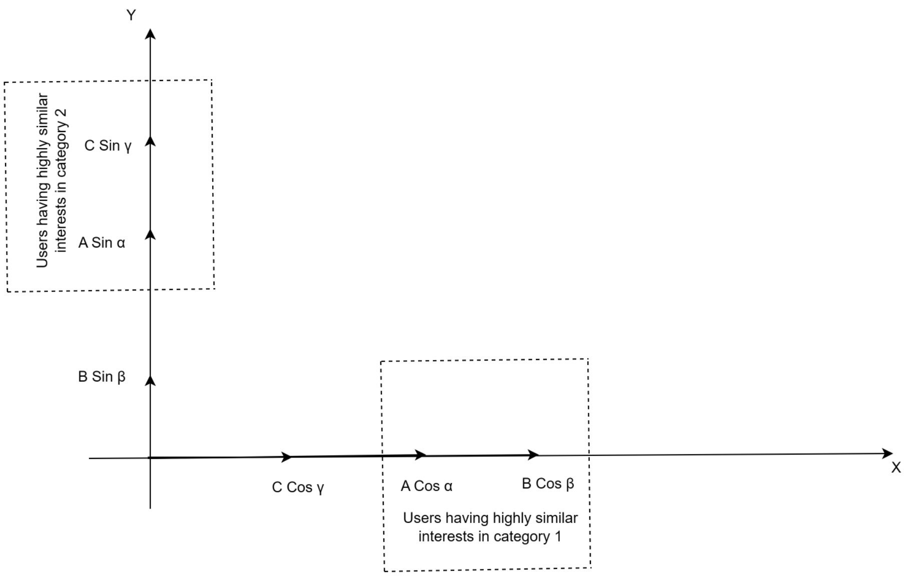
Previous research focused on conventional similarity measurements like Cosine similarity and Pearson correlation coefficient to tackle sparsity and cold-start situations in collaborative filtering.
In this approach, calculating similarities between users involves computing the Pearson correlation coefficient for each pair of users. This process has a time complexity of $O(n^2 * m)$, where n is the number of users and m is the number of movies. Additionally, generating recommendations for each user involves iterating over similar users and their ratings, which can also be computationally expensive.

By adjusting similarity measurements, the accuracy of recommendation results is enhanced compared to conventional methods in sparse and cold-start scenarios:
The new approach involves grouping on each dimension = m (No. of features)
Sorting time for each dimension = $n*\log n$
Total Time : $O(n\log n * m)$



Old approach by using Dot Product





New Approach without using dot product

Addressing the Cold Start Problem

The strategies used for solving the cold start problem are, recommending the popular items to users based on their geolocation, if we have information about users' connections then recommending based on it.
Some other approaches include collecting context about the user from the data available on his device and connected accounts to recommend items. Such as, recommending a user based on his browsing history and collaborating with other platforms for finding users' interests. However, these methods do give rise to privacy issues.
Another method to find out users' interests is asking users' interests upfront at time of login and providing him access to modify his interest any time by choosing topics.

References

[1] Steven C.H. Hoi et al, Online Learning: A Comprehensive Survey, Cornell University
[2] Mining of Massive Datasets: Jure Leskovec Stanford Univ., Anand Rajaraman Milliway Labs, Jeffrey D. Ullman Stanford Univ.
[3] Roy, D., Dutta, M., "A systematic review and research perspective on recommender systems." J Big Data 9, 59 (2022)
[4] Nitin Mishra et al 2021 J. Phys.: Conf. Ser. 1717 012002, "Research Problems in Recommender systems"
[5] Zan Huang, Hsinchun Chen, et al. Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. ACM Transactions on Information Systems, Vol. 22, No. 1, January 2004, 116–142.
[6] Sarwar, B., Karypis, G., et al. Application of dimensionality reduction in recommender systems: A case study. In Proceedings of the WebKDD Workshop at the ACM SIGKKD. ACM, New York.2000.
[7] Good, N., Schafer, J., et al. Combining collaborative filtering with personal agents for better recommendations. In Proceedings of the 16th National Conference on Artificial Intelligence, 1999, 439–446.
[8] Huang, Z., Chung, W., et al. A graph-based recommender system for digital libraries. In Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (Portland, Ore.). ACM, New York, 2002, 65–73.
[9] Goldberg K, Roeder T, et al. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval. 2001, 4(2):133-151.
[10] Chrsistian Desrosiers, George Karypis. Solving the Sparsity Problem: Collaborative Filtering via Indirect Similarities. Technical Report. Department of Computer Science and Engineering University of Minnesota 4-192 EECS Building 200 Union Street SE Minneapolis, MN 55455-0159 USA. 2008.
[11] Chen, Yibo et al. "Solving the Sparsity Problem in Recommender Systems Using Association Retrieval." J. Comput. 6 (2011): 1896-1902.
[12] Çano, E.; Morisio, M. Hybrid Recommender Systems: A Systematic Literature Review. CoRR 2019, abs/1901.03888. Available online: https://arxiv.org/abs/1901.03888 (accessed on 12 January 2019).
[13] Rojsattarat, E.; Soonthornphisaj, N. Hybrid Recommendation: Combining Content-Based Prediction and Collaborative Filtering. In Proceedings of the Intelligent Data Engineering and Automated Learning; Springer: Berlin/Heidelberg, Germany, 2003; pp. 337–344.
[14] Lang, K. NewsWeeder: Learning to Filter Netnews. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; pp. 331–339.
[15] Krulwich, B. Learning user interests across heterogeneous document databases. In Proceedings of the 1995 AAAI Spring Symposium Series, Palo Alto, CA, USA, 27–29 March 1995; pp. 106–110.
[16] Chughtai, M.W.; Selamat, A.; Ghani, I.; Jung, J. E-Learning Recommender Systems Based on Goal-Based Hybrid Filtering. Int. J. Distrib. Sens. Netw. 2014, 2014.
[17] Choi, S.M.; Ko, S.K.; Han, Y.S. A movie recommendation algorithm based on genre correlations. Expert Syst. Appl. 2012, 39, 8079–8085.
[18] Pirasteh, P.; Jung, J.J.; Hwang, D. Item-Based Collaborative Filtering with Attribute Correlation: A Case Study on Movie Recommendation. In Proceedings of the Intelligent Information and Database Systems–6th Asian Conference (ACIIDS '14), Bangkok, Thailand, 7–9 April 2014; pp. 245–252. 30. Zhang, J.; Peng, Q.; Sun, S.; Liu, C. Collaborative filtering recommendation algorithm based on user preference derived from item domain features. Phys. Stat. Mech. Its Appl. 2014, 396, 66–76.
[19] Christensen, I.; Schiaffino, S. A Hybrid Approach for Group Profiling in Recommender Systems. J. Univers. Comput. Sci. 2014, 20, 507–533.
[20] Lekakos, G.; Giaglis, G. A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. User Model. User-Adapt. Interact. 2007, 17, 5–40.
[21] Choi, S.M.; Ko, S.K.; Han, Y.S. A movie recommendation algorithm based on genre correlations. Expert Syst. Appl. 2012, 39, 8079–8085.
[22] Lika, B.; Kolomvatsos, K.; Hadjiefthymiades, S. Facing the cold start problem in recommender systems. Expert Syst. Appl. 2014, 41, 2065–2073.
[23] Carrer-Neto, W.; Hernández-Alcaraz, M.L.; Valencia-García, R.; García-Sánchez, F. Social knowledge-based recommender system. Application to the movies domain. Expert Syst. Appl. 2012, 39, 10990–11000.

[24] He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW 17), Perth, Australia, 3–7 April 2017; pp. 173–182.
[25] Liang, D.; Krishnan, R.G. Variational Autoencoders for Collaborative Filtering. In Proceedings of the 2018 World Wide Web Conference (WWW 18), Lyon, France, 23–27 April 2018; pp. 689–698.
[26] Ghazanfar, M.A.; Prügel-Bennett, A. The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations. Informatica (Slovenia) 2013, 37, 61–92.
[27] Gantner, Z.; Drumond, L.; Freudenthaler, C.; Rendle, S.; Schmidt-Thieme, L. Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In Proceedings of the 10th IEEE International Conference on Data Mining (ICDM '10), Sydney, Australia, 13–17 December 2010; pp. 176–185.
[28] Sun, D.; Luo, Z.; Zhang, F. A novel approach for collaborative filtering to alleviate the new item cold-start problem. In Proceedings of the 11th International Symposium on Communications and Information Technologies (ISCIT '11), Hangzhou, China, 12–14 October 2011; pp. 402–406.
[29] de Campos, L.; Fernández-Luna, J.; Huete, J.; Rueda-Morales, M. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. Int. J. Approx. Reason. 2010, 51, 785–799.
[30] Wen, H.; Fang, L.; Guan, L. A hybrid approach for personalised recommendation of news on the Web. Expert Syst. Appl. Int. J. 2012, 39, 5806–5814.
[31] Duong, T.N.; Vuong, T.A.; Nguyen, D.M.; Dang, Q.H. Utilising an Autoencoder-Generated Item Representation in Hybrid Recommendation System. IEEE Access 2020, 8, 75094–75104.
[32] Chen, S.; Huang, L.; Lei, Z.; Wang, S. Research on personalised recommendation hybrid algorithms for interactive experience equipment. Comput. Intell. 2020, 36, 1348–1373.
[33] Mehrabani, M.M.; Mohayeji, H.; Moeini, A. A Hybrid Approach to Enhance Pure Collaborative Filtering Based on Content Feature Relationship. Available online: https://arxiv.org/abs/2005.08148 (accessed on 17 May 2020).
[34] Zhao, W.; Tian, H.; Wu, Y.; Cui, Z.; Feng, T. A New Item-Based Collaborative Filtering Algorithm to Improve the Accuracy of Prediction in Sparse Data. Int. J. Comput. Intell. Syst. 2022, 15, 1–15.
[35] Jiang, B.; Yang, J.; Qin, Y.; Wang, T.; Wang, M.; Pan, W. A Service Recommendation Algorithm Based on Knowledge Graph and Collaborative Filtering. IEEE Access 2021, 9, 50880–50892.
[36] Ahmadian, S.; Joorabloo, N.; Jalili, M.; Ahmadian, M. Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. Expert Syst. Appl. 2022, 187, 115849.
[37] Ajaegbu, C. An optimised item-based collaborative filtering algorithm. J. Ambient. Intell. Humaniz. Comput. 2021, 12, 10629–10636.
[38] Khaledian, N.; Mardukhi, F. CFMT: A collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships. J. Ambient. Intell. Humaniz. Comput. 2022, 13, 2667–2683.
[39] Liu, H.; Guo, L.; Li, P.; Zhao, P.; Wu, X. Collaborative filtering with a deep adversarial and attention network for cross-domain recommendation. Inf. Sci. 2021, 565, 370–389.