Journal of King Saud University – Computer and Information Sciences 34 (2022) 6385–6399

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Clus-DR: Cluster-based pre-trained model for diverse recommendation generation

Naina Yadav, Sukomal Pal *, Anil Kumar Singh, Kartikey Singh

Department of Computer Science and Engineering, IIT(BHU), Varanasi, Uttar Pradesh 221005, India

ARTICLE INFO

Article history:
Received 15 September 2021
Revised 17 January 2022
Accepted 10 February 2022
Available online 7 March 2022

Keywords:
Aggregate diversity
Diversification
Collaborative filtering
Matrix factorization
Recommender system

ABSTRACT

Recommender Systems are a predictive model for personalized suggestions utilizing past interactions and experiences. Collaborative filtering is the most popular and successful approach. The core idea behind this approach is that the users expressing similar preferences in the past are considered similar and will continue to like similar recommendations in future. The similarity among the items between past and future references, however, affects diversity and coverage of the recommendation system. In this work, we focus on a less usual direction for recommendation systems by increasing the probability of retrieving unusual and novel items in the recommendation list, which are, or can be, also relevant to the users. Most of prevailing techniques for incorporating diversity are based on re-ranking methodology, which shrinks the domain of user's exposure to serendipitous items. To overcome this issue, we propose a methodology Clus-DR (Cluster-based Diversity Recommendation) that uses individual diversity of users and then pre-trained model for diverse recommendation generation. Instead of relying on re-ranking approach, we use different clustering techniques to have different groups of users with similar diversity. Experimental results using datasets of diverse domains indicate the effectiveness of the proposed **Clus-DR** methodology in diversity and coverage while maintaining acceptable level of accuracy.

1. Introduction

Nowadays, users have access to extensive collections of items from a wide variety of sources and devices that can be used from anywhere and at any time because of the easy availability and easy accessibility. With the emergence of many web and mobile applications like Amazon, Netflix, Pandora, Spotify, and Amazon Music, Goodbooks, the use of recommender systems have facilitated both the users and vendors alike and increased the business tremendously in the last decade (Beel et al., 2016; Tavares and Collares, 2020). Early approaches for building Recommender Systems (RS) were largely based on selecting a relevant and accurate item to the user based on their past interaction with the system. There are very well-known and popular techniques used for recommender systems based on Collaborative Filtering (CF), Content-Based Filtering (CBF), hybrid methods, and network-based methods that are mainly used to predict ratings of non-rated items for a user or a set of users (Ricci et al., 2011; Portugal et al., 2018; Abdollahi and Nasraoui, 2018). Performance evaluation of these approaches is typically done based on their ability to predict accurate user-item interactions. In other words, the evaluation set-up and, thus recommender systems are accuracy-centric. However, focus only on relevant and accurate items on a collaborative basis often leads to recommendations too evident and less interesting, and it may, over time, make user interaction with the system monotonous. Introducing diversity is a way out of this problem, which was introduced in 2001 by Bradley and Smyth (2001). Therefore, the introduction of diversity offers a lot of potential for enriching the experience from recommendation systems. A number of research works were motivated by the work of Bradley and Smyth to introduce different algorithms for diversification of results in a recommender system (Bradley and Smyth, 2001; Adomavicius and Kwon, 2011; Karakaya and Aytekin, 2018). Most methods that aim to improve the recommendation diversity consider diversity optimization as a post-processing step regardless of the recommendation generation model. At first, recommendations are generated for each user. The list is then shortened and modified with some re-ranking and intent-based methods that consider the user's intent and item diversity (Adomavicius and Kwon, 2011; Ziegler et al., 2005; Wasilewski and Hurley, 2017).

Motivation- In the above works, the primary goal of the recommender systems is considered to provide diverse recommendations to users with highly unique or personalized items. One

* Corresponding author.
E-mail address: spal.cse@itbhu.ac.in (S. Pal).

https://doi.org/10.1016/j.jksuci.2022.02.010
1319-1578/

major drawback of these approaches is the coverage of the dataset in hand. If an algorithm recommends items from a specific domain, then it will re-rank items within that list, so exploring a broad domain is impossible (Kaminskas and Bridge, 2016). In CF based algorithms, a recommendation list is generated from the user's past interaction history and like-minded users' history. For example, suppose a target user has restricted herself to only comedy and horror movies, and her similar users' movie genres are comedy and action. In that case, the recommendation to the target user is also generated from all the genres involving both the target user and her collaborators. This means that existing re-ranking algorithms can cover horror, action, and comedy domains and not the other domains present in the dataset. We believe that diversity is a personal requirement, and therefore, optimal diversity can vary for each user based on their past interaction data. It can be justified with an example. Suppose there are two users in a system. While the first user restrictively prefers only classical music, the other is interested in all kinds of music (including classical). The first one has less diversity, while the second one's taste is more diverse. Most of the existing recommendation techniques do not consider this personal preference of diversity. Even if they address diversity, they consider during post-processing and at the macro-level taking into account all the users together. But we argue that it should be considered much before during recommendation, and at the individual user level. We also strongly feel that it is important to balance accuracy and diversity during recommendations. We therefore, calculate Individual Diversity Score (IDS) and use it at the core of our recommendation generation algorithm.

Contribution- In this paper, we propose a model named Cluster-based Diversity Recommendation (**Clus-DR**) that uses Individual Diversity Score (IDS) and Item Feature Score (IFS) as two decisive criteria for recommendation generation based on collaborative filtering models. The novelty of our proposed approach lies in presenting a unified matrix factorization model, utilizing metadata and rating information, and considering users' personalized diversification in rating prediction. The primary goal of the proposed approach is to train the model for the automatic diverse recommendation generation for the target user. The key objective of this approach is to train the model to provide a diverse recommendation while keeping each user's diversification level in mind using their past interaction history and using feature information of items rated by the users. We propose IDS and IFS scores for each user. The IFS score considers item-feature information that each item is associated with. E.g., for movies or music data, it can be genre information, and for books data, it can be user-defined tags information like good, bad, average etc. The concept behind this IFS score is to generate a score for each user based on the user-feature matrix. The user-feature matrix is generated using the user-item interaction matrix and item-feature matrix. The user-feature matrix contains the entry of IFS score. The IFS score is the ratio of the number of total features used by the user in their past preferences and the total number of unique features present in the dataset. Both IDS and IFS are real numbers between 0 and 1 and constant for a given user.

This research contributes to using personalized diversity scores and item feature scores of each user for users clustering. To the best of the authors' knowledge, this is the first work that includes personalized user diversity in collaborative filtering models for diverse recommendation generation. IDS and IFS capture two aspects of individual user's diversity level and are used for clustering the users. Such clustering increase the model's ability to find a diverse set of items for users. However, in most approaches, accuracy and diversity have an inverse relation, i.e., diversity will decrease for the accurate recommendation. Any attempt to increase the diversity of the recommendations list for each user will undoubtedly reduce their model accuracy. This may be seen as a problem with using accuracy (or similar personalized preference-based metrics) as the main metric. Conceivably we need better metrics for evaluating diversity in the context of accuracy. In the absence of that, apart from accuracy, we also validate our results by reporting on various evaluation metrics commonly used for recommendation systems. The principal contributions of the paper are as follows.

- We propose individual diversity score (IDS) and individual feature score (IFS) for each user which are calculated from users' past preferences. These two scores work as two decisive criteria for each user's diversity inclusion in their recommendation list.
- Next, we include a cluster-based algorithm to group users. The basic strategy for user clustering is to quantify more diverse users using their past interaction history. We used two clustering strategies for users: (1) individual diversity of the user and (2) using a content feature of the item.
- Our proposed Clus-DR model involves a tunable hyper-parameter (ILD-Score) that users can use to adjust the diversity levels of their recommendations list. The goal of model training for target users with more aggregate diversity is to include interaction domains among diverse users so that the exploration domain of the target user is increased.
- We also present an extensive experiments on three real-world datasets to demonstrate our Clus-DR model effectiveness with various baseline algorithms for diversification in the recommendation system.

2. Related work

We find that the related research done so far can be seen from two viewpoints: i) recommendation generation and ii) personalized diversification for users. While very little systematic research are available on diversity, a number of studies in machine learning, deep learning, and other related applications are highly relevant and they inspire us to incorporate diversity in a recommendation system. We consider to study diversity in the CF-based recommender systems. We, therefore, consider the related work on various collaborative filtering approaches, and then on diversity in recommender system.

2.1. Collaborative filtering

In the early 1990s, collaborative filtering began to arise as a solution for a recommender system. In collaborative filtering, the prediction step is to generate a relevance score for the items for which the system does not have any precise information about user preferences. We consider three representative and most suitable algorithms for our evaluations. The idea behind this proposal is to reduce the user-item interaction matrix into low dimensional latent space, from which we find a d-dimensional matrix. Advantage of this approach is that it takes less computation time than the original user-item interaction matrix. Another variant of matrix factorization technique is SVD (Singular Value Decomposition). SVD is a prominent algorithm in information retrieval for the identification of latent semantics (Ranjan et al., 2019). Another modification of the matrix factorization approach is SVD++, where implicit feedback is included for better accuracy and predictions. Earlier in SVD, implicit feedback was absent, so data sparsity problem arose, but in SVD++, the inclusion of implicit feedback solves this problem (Jia et al., 2014).

However, it may be claimed that both matrix factorization and SVD suffer from the over-fitting problem (Koren et al., 2009), so another variant of matrix factorization is used, like regularized kernel matrix factorization and non-negative matrix factorization (NNMF), which can avoid the issue efficiently. NNMF is used to extract latent vectors from non-negative large sparse matrices using an intrinsically non-negative latent factor model (Luo et al., 2016; Bobadilla et al., 2017). The CF-based techniques cannot give recommendations for the real-time user, so a Reinforcement Learning (RL) based technique has been introduced in the recommender system to overcome these issues.

Chen et al. (2019) proposed a method based on RL to overcome the inconsistency between the continuous action representation using the tree-based policy gradient (TPGR) method. Choi et al. (2018) proposed a method for an RL-based recommender system using the bi-clustering technique. They use the Markov-decision process (MDP) and the RL technique to solve the large space discrete action problem in the recommender system. Chen et al. (2020a) proposed an approach for itinerary planning. They used the RL technique to model recommendations for an itinerary where users will enter their beginning and end destination with budget constraints.

They formulate route recommendation problems using MDP and variational agents optimized through a deep Q-learning algorithm. Ie et al. (2019) proposed an approach for a slate recommendation system using Q-learning. In a slate-based recommender system, recommended items are selected from an extensive collection of items and ranked basis user behavior and interests. The model is based on MDP and Q-learning, where Deep Q Network with experience replay is used to learn these Q-values. Liu et al. (2018) proposed an approach for a slate recommendation system using Q-learning. In a slate-based recommender system, recommended items are selected from an extensive collection of items and ranked basis user behavior and interests. The model is based on MDP and Q-learning, where Deep Q Network with experience replay is used to learn these Q-values.

2.2. Diversity in recommender system

As mentioned above, this study is a go-between of two research areas on the use of diversity in the collaborative filtering models in recommendation generation. In this section, studies related to diversity have been reviewed for recommender systems. Here, we first discuss the various definitions of diversity and multiple algorithms to increase diversification in the recommendation system. Subsequently, we also review numerous metrics for measuring the diversity of recommendations.

2.2.1. Definitions and diversity metrics

Diversity was first introduced in IR (Information Retrieval). In IR, it has been noted that the retrieved document is influenced by the document similarity with another document as well as with the query posted by the user. Suppose a user searches for ‘apple’, the term apple may refer to an fruit or an electronics brand. If there were not much information to disambiguate the user’s interest for a respective search, then at that point the diversity plays a significant role by suggesting both the items at the same time. Ziegler et al. (2005) define diversity as intra-list dis-similarity metric. Diversity in the recommendation system refers to individual diversity and aggregate diversity. Individual diversity aims to maximize the coverage and diversity for each user individually. Individual diversity measure is defined as topic coverage as well as α- nDCG (variant of Normalised Discounted Cumulative Gain). Vargas (2014) proposed an approach for diversity that uses the intent as users requirement. They proposed a measure for diversity, which includes ambiguity, diversity, redundancy, and novelty into a single action. Lathia et al. (2010) define individual diversity using temporal information of users because the data changes over time as users behavior towards their preferences. Castells et al. (2015) define diversity as a product of items relevance, similarity value, and rank in the recommendation list. Fleder and Hosanagar (2009) also use diversity to assess the effect of the recommender system in sales diversity. The authors only focused to provide diversity for a specific domain in terms of sales diversity. Some recent research in diversity includes the personalized diversity measure as an algorithm basically for long-tail item recommendation (Yadav et al., 2019).

2.3. Recent work on diversity in recommendation systems

Diversity in recommendation system has mostly been seen as a post-processing procedure by many researchers (Anelli et al., 2021; Wasilewski and Hurley, 2017; Hamedani and Kaedi, 2019; Hu et al., 2017; Kyriakidi et al., 2017; Abdollahpouri et al., 2021). Here we discuss some recent and promising approaches for diversity in the recommendation system. Apart from the greedy re-ranking algorithm, intent-Aware Diversification is used in recommendation systems to re-rank the recommendation list using various aspects of users. These aspects can be defined explicitly or implicitly. Implicit aspects can be derived from the user’s history, like latent factors, and explicit aspects are a set of features that are used as a description for items and users. These are some machine learning-based approaches. Apart from these, some deep learning approaches are also used for diversity awareness in the recommender system. Esmeli et al. (2019) proposed a session-based personalization for diverse recommendation. Diversity was included in the recommendation list by adding items that depend on the diversity level of the last interacted item of the session. Similarly Hu et al. (2017) also proposed a deep learning model for a session-based recommender system that uses the context information to personalized diversity. This approach is also a rearranging approach which re-rank the recommendation list using the relevance of users to the item based on some given context. Apart from single context consideration in RS, many authors incorporated multiple aspects in improving diversity in the recommendation. Oliveira et al. proposed a multiobjective method for diversity and accuracy consideration, which includes content information like contemporaneity, gender, genre, and locality. The multiobjective optimization is achieved using Pareto optimality (Oliveira et al., 2017). Nassife et al. proposed a diversification method in music recommendation using the Jaccard swap diversity and submodular diversity optimization method (Nassif et al., 2018). Volokhin et al. proposed a diversity infused recommendation model based on users intent. The user’s intent is obtained from a survey and used as a context value for music recommendations (Volokhin and Agichtein, 2018). He et al. proposed multi-source subtopics, a framework for subtopic modeling that uses a random walk-based approach to estimate the similarity between subtopics extracted from multiple Web sources and then regularize the similarity relations to construct document content (He et al., 2012). Recent works are using the traditional approach for a recommender system. Su et al. (2013) propose a set-oriented framework for diversity using a matrix-factorization method based on users context information explicitly for acquiring personalized diversity. Wang et al. (2016) included personalized diversity using the similarity network for better user influence on the recommender system. This approach connects the similarity function and bipartite graph to improve the resource-allocation process using a similarity network. Chen et al. (2020b) proposed a deep learning approach for diversification. They used a sequential recommendation model with intent mining for diversity enhancement. The method uses an implicit intent mining approach to mine user intent automatically. Another concept introduced in recommendation system for diversity using a machine learning algorithm, is called DPP (Determinantal Point Process). DPP is a complementary and encouraging approach used as a probabilistic model for the negative correlation, sampling, conditioning, marginalization, and many other inference tasks.

Wilhelm et al. (2018) included the DPP process in their work for YouTube video recommendation. DPP is a naive approach for learning diversity without using any side information, so the computational cost of this model grows exponentially because possible sets of items also increase exponentially.

2.4. Diversity trends in recommendation system

In recent years lots of research work based on temporal dynamics in recommender systems have been proposed. In these works, most of the research focuses on adding temporal dynamics of the users and their interaction with the item. Another trend in recommendation is to include multiple aspects of users and items in recommendation generation. In a recommendation system, the user's interests may change during the time. Recently, lots of studies have been performed that included the users temporal dynamics information. At the very early stage, Lathia et al. (2010) proposed an approach for temporal diversity measure and their impact on the recommendation. They offered a standard for temporal diversity based on users surveys for different time span. Wang et al. (2019) proposed an approach for repeat consumption of items that consider the users temporal dynamics. They offered a model based on Hawkes Process into Collaborative Filtering considering long and short-term effects. Ma et al. (2020) proposed an approach named HRNN-meta based on contextual features of users and items with varying time effects. The model is based on the session-based matrix factorization approach, which considers temporal orders of events, contextual features, and metadata about items. You et al. (2019) proposed an approach HierTCN for sequential multi-session interactions with items based on a temporal convolution network utilizing both the long-term and the short-term interactions within sessions to predict the next interaction. Najafipour et al. (2020) proposed an approach based on a neural network that generated the sub-graph from the short micro blog text. This approach calculated a relevance score between two users as an edge weight for community detection among users. We also summarize related work in the Table 1.

2.5. Diversification algorithms: baseline

In this section we discuss the diversification algorithm that we use as part of a comparative analysis with some accuracy centric approaches. **Greedy Re-ranking Approach** is used as a post-processing step to improve diversity in the recommender system. The underlying assumption for this approach is that it takes a conventional recommender system to generate a prediction score and generate a recommendation list. Using the Greedy Re-rank algorithm, we explicitly re-rank the items in the recommendation list based on the dissimilarity value. Ziegler and Cai-Nicolas proposed the greedy re-rank algorithm in 2005 (Ziegler et al., 2005). Next approach is **xQuAD** (Explicit Query Aspect Diversification) proposed by Abdollahpouri et al. (2019) and Abdollahpouri et al. (2021) uses the personalized diversification to improve the recommendation and reduce the popularity bias. They use the concept of re-ranking the recommendation list using a sub-profile aware diversification method with accuracy and diversity at the same time. In addition to this, for better diversity, they included the relevance for long tail and short head using marginal likelihood over both categories for each user. Another diversity-centric baseline for our model is **Maximal Marginal Relevance (MMR)** which is a concept used in the Information Retrieval process to re-rank documents based on the query relevance. MMR attempts to reduce the redundancy of the retrieved documents and increase the diversity of the documents (Carbonell and Goldstein, 1998). The goal of this proposed algorithm is to maximize the objective function which is used to balance the trade-off between accuracy and diversity using a single function.

3. Problem description

Let us have a set of users $U = \{u_1, u_2, u_3 \dots u_m\}$ and a set of items $I = \{i_1, i_2, i_3 \dots i_n\}$ and we have set of item features $F = \{f_1, f_2, f_3 \dots f_k\}$. Therefore we have a user-item interaction matrix R of size $m \times n$ and item-feature matrix of size $(n \times k)$. In interaction matrix each user has rated some items, the rating for item i by user u is r_{ui}, the rating value ranges between integer value 1 to 5 (5 being the best and 1 the worst). The item feature matrix S is a binary matrix, the entry (s_{ik}) is either 1 if item i has feature k else 0 (item I does not have feature k).

Given the interaction matrix $R_{m\times n}$, item-feature matrix $S_{n\times k}$, we generate user-feature matrix $(G_{m\times k})$. For a set of target users, the goal of our proposed model is to predict the ratings of all the non-rated items for the target users and recommend top-N items for each of the target users.

Table 1
Summary of literature review.

Ref	Category	Approach	Description
Ribeiro et al. (2012)	Machine learning	Reranking method	They proposed a relevance score, a combination of accuracy, diversity, and novelty. Then used this score within a joint model that includes three variants of the matrix factorization approach for diverse recommendation generation.
Hurley (2013)	Machine learning	Reranking Method	Proposed a pairwise reranking model that learns user and item factors by minimizing an objective function that includes item dissimilarity irrespective of recommendation list size.
Su et al. (2013)	Machine learning	Reranking method	Proposed a pairwise reranking model that works on item set level. Training data is prepared based on pairs of item sets and comparing each pair of item sets using both relevance and diversity criteria.
Xie et al. (2021)	Graph Neural Network	Reranking method	The model builds a heterogeneous preference network to record user preferences and uses a heterogeneous graph attention network for node aggregation. Two different stages matching stage concentrate on accuracy, and the ranking module more concentrate on diversity.
Anderson et al. (2020)	Deep Learning	Classification	They proposed a method to quantify users based on their music consumption diversity, i.e., generalist and specialist. They investigate how algorithmic recommendations relate to consumption diversity.
Karako and Manggala (2018)	Deep Learning	Reranking method	They proposed a method to fairness-aware variation of the Maximal Marginal Relevance (MMR) reranking method using the user's demographic information. They used a pre-trained CNN model with MMR to obtain fairness in the recommendation.
Abdollahpouri et al. (2019)	Machine learning	Reranking method	They proposed a model xQuAD based on the re-ranking method for long-tail items. The model is used for the control of popularity bias in the recommendation system.
Sonboli et al. (2020)	Deep Learning	Multi-aspect	This work is used to find out the subgraph of a similar author for the target author using multiple aspects like temporal context (time, day, etc.), short text (tweets).
Najafipour et al., 2020	Deep Learning	Multi-aspect re-ranking	They proposed a re-ranking approach to a fairness-aware recommendation that learns individual preferences across multiple fairness dimensions for users (gender, sector, rate, etc.) to provide a fair recommendation.

4. Proposed methodology

This section presents the overall architecture and operational methodology of our proposed Clus-DR model (Fig. 1). We use two inputs: a user-item interaction matrix ($R_{m\times n}$), and item-feature matrix ($S_{n\times k}$). While $R_{m\times n}$ can be readily used, $S_{n\times k}$ needs to be processed to transform to a user-feature matrix.

The workflow of our proposed Clus-DR model can be divided into three parts. First, individual diversity calculation and item feature score calculation for the user; second, clustering of users based on item features and individual user's diversity score; and third, training our model using Non-Negative Matrix Factorization (NNMF) technique to generate a diverse recommendation. Apart from NNMF, we also evaluate our model performance for SVD and SVDpp algorithms. Our goal is to exhibit the personalized diversity that should be included in the recommendation system before rating prediction. The flow chart of the proposed Clus-DR model is shown in the Fig. 2. The overall architecture of our proposed methodology is divided into two parts online and offline part. The online part contains input where we calculate IDS and IFS scores. These scores will update whenever a new user and item are introduced in the system, and in the offline part, we train these input information for recommendation generation.

4.1. Individual diversity score for user

We describe here an attempt to quantify the user's need for diversity. Every user has different preferences in their daily life. For example, in the movie domain users have different choices in terms of artists, genres etc. We factor in this user-specific preferences for diversity into our model and introduce an Individual Diversity Score (IDS) for users, representing how a user liked diversity in her past preferences. This diversity score of an user is used as a minimum threshold for diversification of our recommendation. We define the diversity in terms of cosine similarity. The similarity between items will be calculated using the Eq. (2). The average dissimilarity between items rated by user u is considered as the individual diversity score for the user u. In Eq. (1), diversity is calculated for the user u, where i_j and i_k are two different items rated by user u from their past interacted item list of size n.

$$IDS(u) = \frac{1}{n(n-1)}\sum_{j=1}^{n}\sum_{k=1(j\neq k)}^{n}(1 - sim(i_j, i_k)) \tag{1}$$

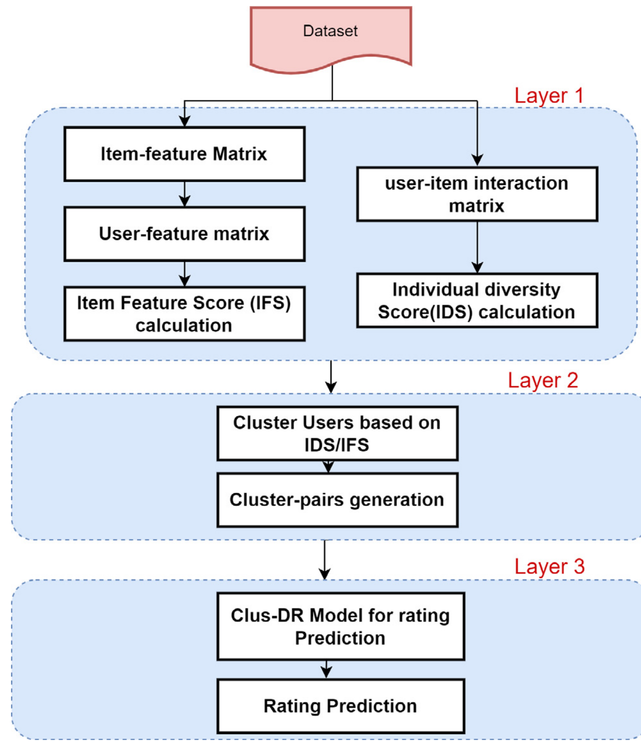


Fig. 2. Flow-chart of the proposed Clus-DR recommendation model.

$$similarity(i_j, i_k) = \frac{\sum_{p=1}^{m} r_{pj}.r_{pk}}{\sqrt{\sum_{p=1}^{m}(r_{pj})^2.\sum_{p=1}^{m}(r_{pk})^2}} \tag{2}$$

In Eq. (2) we calculate a cosine similarity between two items i_j and i_k rated by user u where r_{pj} and r_{pk} are the ratings for item i_j and i_k by user p and m is the number of users present in the dataset. IDS(u) is calculated for each user and is termed as individual diversity score for user u. We consider IDS(u) as a threshold for the user u, which decide the user u's minimum diversity level.

4.2. Item feature score for user

In users item selection, items features information also plays a significant role. Users often select items based on their content

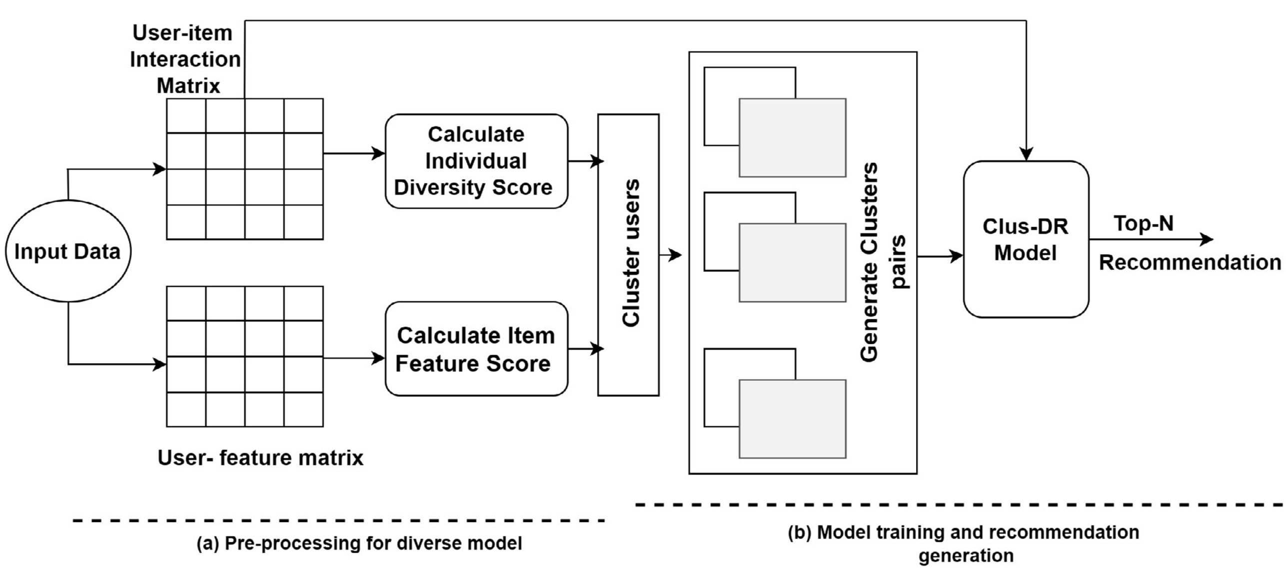


Fig. 1. Architecture of the proposed Clus-DR recommendation model.

information like genre, year, artist, etc. So we argue that item feature information will also be a decisive criterion in diversity enhancement in the recommendation system. We propose an item feature score for each user based on this assumption. Here in our proposed algorithm, we use genre information of movies and music and tags information of books to calculate a feature score for each user, which defines each user's preference for a particularized genre. For generalization apart from genre score, we can also use other content information of items like in music domain, artists, genre, language, etc. We use MovieLens, LastFM, and Goodbooks datasets, where each movie has at least two genres, and in LastFM and Goodbooks, we use the tag information of each artist to calculate the feature score. The tags are the user-defined tags/shelves/genres in each artist/book dataset. In other terms, we encode the item's metadata information for clustering of users.

$ItemFeatureSource(u)$ is calculated for each user based on a portion of particular user interaction information. Let user u has rated n_1 items from a complete set of n items ($n_1 <= n$). The calculation $ItemFeatureSource(u)$ is calculated as follows-

$$Item\ Feature\ Source\ (u) = \frac{\sum_{j=1}^{n_1}\sum_{k=1}^{Q} G(i_j, f_k)}{\sum_{j=1}^{n}\sum_{k=1}^{Q} G(i_j, f_k)} \tag{3}$$

Here in Eq. (3) we calculate $ItemFeatureSource(u)$ for user u. where function $G(i_j, f_k)$ is a boolean function defined as follows-

$$G(i_j, f_k) = \begin{cases} 1, & \text{if } i_j \text{ has feature } f_k \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

and $f_1, f_2, \ldots f_Q$ represents a set of Q distinct features (genre) and $j \in \{1, \ldots, n\}$ where n is total number of items present in the dataset.

4.3. Clustering algorithm

The second step of our proposed algorithm is the clustering of users based on IDS and IFS. In our proposed algorithm, clustering is performed using two different techniques:

1. Clustering is first performed using IFS
2. Clustering is also performed using IDS.

The flexibility of our approach is that we can use any feature information apart from genre value and use any unsupervised clustering algorithm. First clustering is performed using the feature information of an item, i.e., the genre information of the movie and second clustering is performed using individual diversity of a user. The agenda for both the algorithm is different. Inclusion of genre score will carry items that have the similar genre-ratio information. Similarly, the other clustering will cluster users with the same diversity level. We use k-means clustering algorithm. However, any unsupervised algorithm can also used instead. The k-means algorithm in our approach works as follows.

- Define the number of clusters as total number of distinct genres present in the data-set.
- Selecting a random set of k data points and then calculate the centroid without data shuffling.
- Keep iterating until the centroid value stops changing. It also means that the data value assigned to the cluster is not changed.
- Compute Euclidean distance from one data point to other data points and assign data points to each cluster with which they have the minimum distance and then compute the centroid.

4.4. Aggregate cluster diversity

Next, in our proposed algorithm, we need to calculate aggregate cluster diversity. The aggregate cluster diversity is the average of each user's individual diversity present in the cluster. The urge behind the calculation of aggregate cluster diversity is that it includes users with similar diversity levels. The further use of aggregate cluster diversity is to make cluster pairs for the training of our model. The intention behind the training of our model with these clusters having more diverse users will help other less diverse users so that the interaction domain of these users will expand, and the model will generate more diverse recommendations. The pseudo code for aggregate cluster diversity is described in Algorithm 1.

Algorithm 1: Pseudo-code for Aggregate Diversity for a Cluster

```
Input User set U ∈ (u_1, u_2, ... u_m) Users in Cluster
    (u_{c_k}), Total Cluster(C_k), Individual diversity of
    user u (Div_u)
Output ADivClus ← [ ] ;          ▷ aggregate cluster
    diversity
for C_k = 1, 2, ... k do
    for each u_c ∈ U do
        if u_c ∈ C_k then
            ADivClus ← Div(u_c)
        end
    end
    return ADivClus(C_k)
end
```

The aggregate diversity for the ML-1M dataset is calculated after applying the k-means clustering algorithm for clusters size (k) = 18. The optimal value for cluster size(k) is chosen form a set of experiment performed from various k values, the detailed analysis is present in the Section 6. The aggregate cluster diversity for clusters are shown in Tables 2 and 3 and their graphical representation is also presented in Figs. 4 and 5.

4.5. Clus-DR model

In this section, we explain the different prediction algorithms used for our Clus-DR model training. After finding the clusters of users and individual diversity scores for the users, three types of the prediction models are used for recommendation generation. The proposed Clus-DR model generates the top-k prediction for each user which are more diverse than the other state-of-the-art recommendation algorithm. The architecture of the proposed approach is depicted in Fig. 1. The pre-processing for the Clus-DR model is the calculation of feature score for each user, where we first calculate the individual diversity score and item feature score for each user. This score is further used to cluster users into groups based on the assumption that more diverse users will be in the same group. The user clusters $C_i = (C_1, C_2, \ldots C_k)$ are obtained by using unsupervised clustering algorithm, in our proposed model, we use the k-means algorithm. Once we obtained the clusters, the model training comes in the role where we adopt the SVD, SVDpp and non-negative matrix factorization approach. The training of our proposed model is based on constraint non-negative matrix factorization approach. Our approach considers NMF with

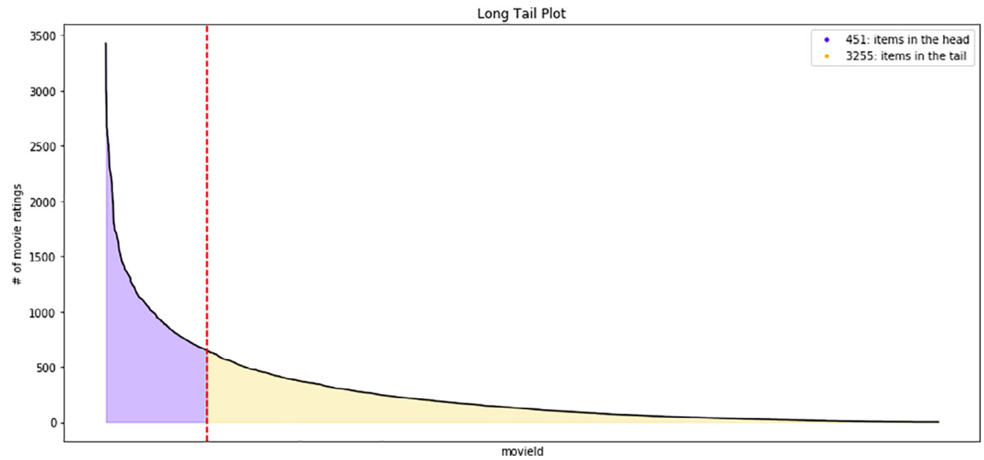


Fig. 3. The Long-tail of popularity of items in the MovieLens dataset.

a constraint that defines additional data apart from the user-item input matrix, cluster information for each user. In our proposed method, clusters obtained from pre-processing of data in the Section 4.3 worked as training pairs for Clus-DR model. We used typical matrix factorization algorithms (SVD, SVDpp and NNMF) to generate diverse recommendation. We achieved the best predictions using the NNMF model. A typical NNMF is used as a dimensionality reduction algorithm which can be written as $R \approx U.V$ where U and V are two matrices having non-negative values (Aghdam et al., 2015). In recommendation system U and V are the column matrices used for user and item representation. In general matrix R contains some unknown value which are the values of user-item rating information which is not available in dataset. The data is represented into a user-item interaction matrix $R \in \mathbb{R}^{m \times n}$. Our goal is to represent it by the product of two low dimensional matrices where $U \in \mathbb{R}^{m \times d}$and $V \in \mathbb{R}^{d \times n}$. The NNMF problem is $\min_{U \geqslant 0 V \geqslant 0} \|R - (U.V)\|$. The objective of the cost function is to minimize the error between the original rating matrix R and the product of the latent vectors of user U and item V. The cost function for NNMF is often depends on the probability distribution of data so the simple way is to use the Frobenius-norm measure. The NNMF cost function for error minimization is defined as follows -

$$D_F(R||UV^T) = \left\|R - UV^T\right\|_F^2 = \sum_{u,v}(R_{uv} - \sum_{d=1}^{k} U_{uf}.V_{fv}) \quad (5)$$

In Eq. (5) we calculated the error for NNMF algorithm where the error is nothing but the l_2 norm of the matrix of size $(M \times N)$ which is a difference matrix obtained from original matrix R and the

Table 2
Aggregate Diversity of Clusters based on Genre-score Clustering for ML-1M dataset.

Cluster_ID	Agg-Diversity	Cluster_ID	Agg-Diversity
1	0.9934	7	0.9535
17	0.9905	4	0.9277
12	0.9838	0	0.9231
5	0.9810	2	0.9221
16	0.9741	6	0.9101
9	0.9717	10	0.9087
11	0.9580	13	0.8899
3	0.9569	8	0.8758
15	0.8750	14	0.8581

Table 3
Aggregate Diversity of Clusters based on user individual diversity score clustering for ML-1M dataset.

Cluster_ID	Agg-Diversity	Cluster_ID	Agg-Diversity
17	0.9956	14	0.9702
15	0.9928	4	0.9603
5	0.9905	12	0.9381
6	0.9898	9	0.9046
0	0.9808	1	0.9001
2	0.9799	8	0.8948
16	0.9758	7	0.8814
3	0.9748	11	0.8604
10	0.9708	13	0.8582

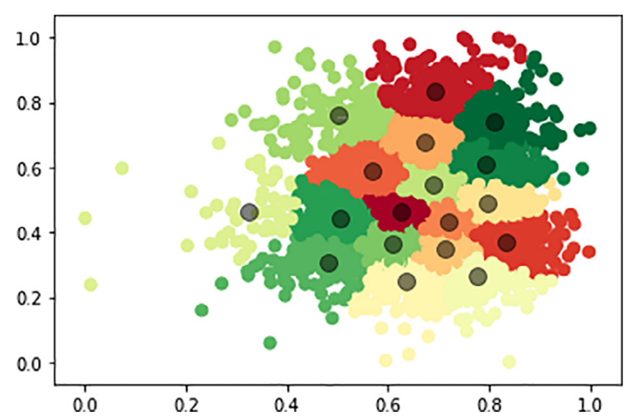


Fig. 4. Clustering of a user based on item feature score using k-means.

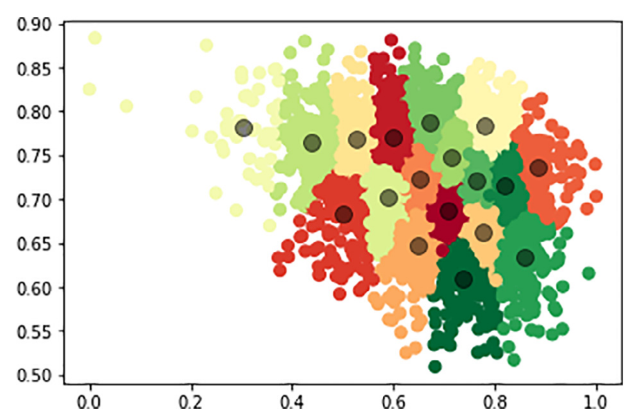


Fig. 5. Clustering of a user based on individual diversity score using k-means.

resultant matrix of $U.V^T$ where U and V are the latent vectors of the users and items respectively. Here M is the total number of users and N is the total items in the dataset and R_{uv} is the original rating of item v given by the user u. The above cost functions are convex concerning either the entries of the matrix U or the matrix V, but not both. Therefore, it is impossible to solve this problem in the sense of finding a global minimum. However, many numerical optimization methods can be applied to discover local minimums, so we adopted gradient descent for faster convergence.

The update rule for the user (U) and item (V) is derived using gradient descent. Gradient descent for multi variable function $F(\alpha)$ is defined and differentiable in a neighborhood of a point A. So it follows rule $\beta \leftarrow \alpha - \eta \triangledown f(\alpha)$ for very small learning rate (η) then $f(\alpha) \geqslant f(\beta)$. Now the update rules for U and V that reduce the divergence are as follows:

$$U_{uf} = U_{uf} + \eta_{uf}\left[\sum_{m=1}^{M} V_{fm}^{T} \frac{R_{um}}{(U.V^{T})_{um}} - \sum_{m=1}^{M} V_{fm}^{T}\right] \tag{6}$$

$$V_{fv} = V_{fv} + \eta_{fv}\left[\sum_{n=1}^{N} U_{nf}^{T} \frac{R_{nu}}{(U.V^{T})_{nu}} - \sum_{n=1}^{N} U_{nf}^{T}\right] \tag{7}$$

These rules are used to calculate the user and item latent factor. The update rule for user U_{uf} is defined in Eq. (6) and update for item V_{fv} is defined in Eq. (7), where η_{uf} and η_{fv} are the learning rate. We have used divergence based cost function to minimize the difference between the original rating matrix R and the matrix obtained from user and item latent factors. In Eq. (6), (7) we have set the value of learning rate as follows-

$$\eta_{uf} = U_{uf} / \sum_{\eta=1}^{M} V_{f\eta}^{T} \tag{8}$$

$$\eta_{fi} = V_{fv} / \sum_{\eta=1}^{N} U_{\eta f}^{T} \tag{9}$$

Still, the main aim is to increase the diversity of the algorithm, so training of the algorithm is the core part that induces diversity, which we discussed below in the Section 4.6.

4.6. Clus-DR model training and recommendation generation

In this section we define our training procedure for the Clus-DR model. We use movie, book and music domain data to train our model. We first calculate the user-item interaction matrix, then we calculate individual diversity and item feature scores for each user discussed in Section 4.1 and 4.2 respectively. After that, we apply the k-means algorithm for user clustering based on individual diversity and item feature scores. The assumption over the clustering algorithm using individual diversity score is to group users following the same diversity levels. The speculation over the clustering of the using item feature score is users following the same item features will be in one group. At a time, we use only one score for clustering to check which score performs better in terms of diversity. Another essential motive is to check whether content information is important or not for diversification in recommendation. Once we get the clusters of users, after that we calculate aggregate cluster diversity for each cluster which is defined in Algorithm 1. The overall goal of the aggregate cluster diversity is used as a threshold value for training pair generation for users based on their diversity level. The aggregate value for each cluster is nothing but an average of the individual diversity score of each user present in the cluster. Suppose there are three clusters C_1, C_2 and C_3 and their aggregate diversity are $0.988, 0.899$ and 0.901, then training pairs are $(C_1, C_2), (C_2, C_3)$ and (C_1, C_3). So the input for our Clus-DR model training are the clusters pairs, and their corresponding user-item interaction pairs which will process according to the model description discussed in Section 4.5. For model training we need to define some hyper-parameter for model training first is k value defines number of clusters. We set $k = 18$ for our model and also check our results for $k = 10$. Another hyper-parameter is learning rate $\eta = 0.01$. After cluster pair generation and hyper-parameter setting we train our model for three prediction algorithms. First we use *(Clus-NMF)* algorithm to train our model, apart from Clus-NMF we also use SVD *(Clus-SVD)*, and SVDpp *(Clus-SVDpp)* algorithm for our proposed model, but the result obtained from *Clus-NMF* are good enough. In the Clus-DR model, training is based on the cluster of more diverse users, which leads to sparsity. We strictly limit ourselves to only those users who belong to the same cluster or more diverse cluster for recommendation generation. This may lead to some information loss in the model and leads to sparsity. The cluster-pair formation restricts our model in search for similar users for target user recommendation generation. For explanation, unlike the traditional CF-based algorithm, while searching for similar users from the complete list of user sets, we will search similar users in clusters. So this might be the reason for information loss and leads to the accuracy drop of our proposed model.

5. Experiments

In this section, the experiment setup and the dataset used for the proposed algorithm are described. Later on, performance comparison of the proposed model with the baseline recommendation model is reported. The baseline for the recommendation model is described in Section 2.5.

5.1. Dataset

The recommender system is based on the purchase history of users items and their interaction history. Based on that, it predicts the best suitable item for users. So, an essential requirement for these models is a set of user items and their feedback that describe the products. To prove the efficiency of our proposed methodology, We use ML-1M[1] and ML-100k,[2] an established dataset for movie recommender systems, LastFM(2k)[3] dataset for the music domain and Goodbooks(10k)[4] dataset form book domain. The statistics of the datasets are present in Table 4. The 2(k) and 10(k) are abbreviations used in the dataset, indicating the number of distinct users and interaction. The LastFM(2k) is a music dataset for 2000 different users' music listening datasets. Similarly, the Movielens(100k) is a movie dataset for 943 distinct users having 100000 interactions of user and movie in the dataset. For ML-1M and ML-100k dataset, every movie is marked with genre information. In these datasets, a total of 18 different genres are present, and every movie is described with more than one genre, e.g. Genre Fantasy, Sci-fi describe the movie Star Wars. Before using our approach for recommendation generation the first step is to preprocess the raw dataset. Similarly, we also use a different domain dataset for our model's flexibility, and we use a music dataset LastFM. In this dataset, there is a play count value that describes each user's total number of plays for a particular artist instead of rating. As a pre-processing step, we converted the play count value into a rating scale of 1 to 5. Similarly in Goodbook dataset we have user, book and their corresponding rating with author and tag information of each book. For result generation, we need to transform this every preprocessed dataset into a

[1] https://files.grouplens.org/datasets/movielens/ml-1m.zip.
[2] https://files.grouplens.org/datasets/movielens/ml-100k.zip.
[3] http://millionsongdataset.com/lastfm
[4] https://www.kaggle.com/zygmunt/goodbooks-10k.

Table 4
Statistics of the dataset used for evaluation of our approach.

Dataset	Users	Movies	Interaction
LastFM (2k)	1892	17632	92834
ML-1M	71567	10681	10000054
Goodbooks (10k)	53424	10000	981756
ML-100k	943	1682	100000

user-item interaction matrix. This matrix is then used for recommendation generation.

5.2. Evaluation metrics

In this section, we discuss evaluation metrics used for the evaluation of our proposed model. There are numerous metrics available for evaluation of a recommendation model. We use Root Mean Square Error (RMSE) and MAE (Mean Absolute Error) to evaluate our model's accuracy. The accuracy metric for this model is not sufficient because the fundamental goal for this proposed model is to generate some set of diverse as well as accurate recommendations. Aggregate diversity is an essential feature of the diversity of recommendation systems. Aggregate Diversity of a recommended list is calculated using the average dissimilarity between items present in the list. Diversity for top-n recommendation is calculated using the intralist similarity measure defined in Eq. (10). We also include user coverage result which shows the coverage of item set for each user. For further evaluation we use accuracy measure in terms of RMSE and MAE (mean absolute error) with top-k Precision, Recall and nDCG (Normalized Discounted Cumulative Gain) which are important measure for recommender system evaluation. Aggregate Diversity of a recommended list is calculated using Intralist Diversity (IL-D) measure defined in Eq. (10). IL-D is the average dissimilarity between items present in the recommendation list size of R for each user.

$$IL-D=\frac{1}{|R||R-1|}\sum_{i\in R}\sum_{j\in R,i\neq j}1-d(i,j) \tag{10}$$

In recommendation, system coverage is how many items appear in the recommended results or top-n recommended result. Coverage is define as below-

$$Coverage(M)=\bigcup_{u}R_{M,k}(u) \tag{11}$$

Coverage for model M is $R_{M,k}$ is test set R for user u retrieved from model M. The higher value for coverage is better and shows that the model recommends a wide range of items.

5.3. Experimental results and baseline

In this Section, the results of the experiments are presented. Here, we shows our model's performance for different evaluation metrics used for the recommendation model, followed by comparative analysis with baseline algorithms used for diversification of the recommendation system.

5.3.1. Performance of the Clus-DR model

In the experiment of our Clus-DR model, we selected SVD, SVDpp, and NNMF approach for training the model. The number of clusters (k) chosen to be 18 with a learning rate of 0.01. The value of cluster size has been experimented with several hyperparameters, and the optimal value is 18 for our proposed model, so, for further evaluation, we use cluster size(k) =18. The accuracy of the Clus-DR model is calculated using RMSE and MAE in Table 5. The best accuracy of our model is for ML-1M dataset where RMSE is 1.1107 and MAE is 0.9250 is for Clus-NMF algorithm. We used three different datasets of the movie, books and music domain to prove the generalization of our approach. Statistics of these datasets are shown in Table 4. The MovieLens dataset suffers from the long tail problem. The long-tail graph for ML-1M dataset is shown in Fig. 3. The long-tail graph shows the distribution of items popularity in the dataset, where the x-axis shows the items, and y-axis shows the popularity of items in terms of rating provided by the users. This long-tail problem shows the uneven distribution of the rating behavior of the user, which makes the dataset sparse. Next, following our Clus-DR model algorithm, we need to group users based on the individual diversity and item-feature score. For this, we applied k-means clustering algorithm, and users clusters for the ML-1M dataset are depicted in Figs. 4 and 5. After cluster generation, we need to calculate the aggregate diversity of clusters, which is nothing but the average of the individual diversity of users present in the cluster. For each datasets, we created 18 clusters, so the aggregate diversity for each cluster for ML-1M dataset are represented in Tables 2, 3. We create 18 clusters because, in this MovieLens dataset, there are 18 different genres available. The flexibility of our approach is we can use any unsupervised clustering algorithm in place of the k-means. The reason to use the k-means algorithm is because we can maximize users similarity within clusters by utilizing this algorithm and minimize the similarity of users in different clusters. Moving forward, we need to check our model's efficiency in terms of accuracy and diversity. As per the accuracy-diversity trade-off, our model loses some accuracy compared with the baseline algorithms for the recommender system. For our comparative model study, we use RMSE and MAE for accuracy analysis. All the results for our Clus-DR model over various dataset are presented in Table 5. Table 5 shows our models accuracy measures in context of different datasets. This article's main aim is to propose an approach that gives diverse recommendations for each user. Apart from using any post-processing step, we tried to train our model to generate diverse recommendations. The table shows our model's accuracy in terms of RMSE and MAE for ML-1M, Ml-100k, LastFM(2k), Goodbooks(10k) dataset.

Table 5
Comparison of error evaluation metrics for the proposed Clus-DR model in terms of RMSE (Lower the better value) and MAE (Lower the better value).

Dataset	Proposed Model	Algorithm	RMSE	MAE
ML-1M	Clus-DR (IDS)	Clus-SVD	1.1107	0.9707
		Clus-SVDpp	1.1181	0.8987
		Clus-NMF	**1.1107**	**0.9250**
	Clus-DR (IFS)	Clus-SVD	1.1120	0.8906
		Clus-SVDpp	1.1181	0.8987
		Clus-NMF	1.1141	0.9765
ML-100k	Clus-DR (IDS)	Clus-SVD	1.0783	0.866
		Clus-SVDpp	1.0776	0.8649
		Clus-NMF	1.1228	0.9296
	Clus-DR (IFS)	Clus-SVD	1.0991	0.8772
		Clus-SVDpp	1.0891	0.9055
		Clus-NMF	1.1196	0.8866
LastFM (2k)	Clus-DR (IDS)	Clus-SVD	1.1196	0.8866
		Clus-SVDpp	1.1228	0.8833
		Clus-NMF	1.1008	0.9368
	Clus-DR (IFS)	Clus-SVD	1.1233	0.8987
		Clus-SVDpp	1.1229	0.9033
		Clus-NMF	1.0988	0.9299
Goodbooks (10k)	Clus-DR (IDS)	Clus-SVD	1.2026	0.9885
		Clus-SVDpp	1.2118	0.9442
		Clus-NMF	1.1803	0.9003
	Clus-DR (IFS)	Clus-SVD	1.2103	0.9987
		Clus-SVDpp	1.2269	0.9893
		Clus-NMF	1.1991	0.9221

Clus-DR(IDS) and Clus-DR(IFS) are the model based on IDS score and item feature score respectively.

From Table 5, we can conclude that the ML-1M is the best performing dataset for our Clus-DR model. For performance evaluation of the proposed Clus-DR model, we use precision, nDCG, IL-D (intralist diversity), and coverage. The complete result is shown in Table 6. The results discussed in Table 6 are for top-15 recommendations for each user.

5.3.2. Comparative analysis

The comparison of our proposed model Clus-DR model for diverse recommendation system is conducted over a large and real-time dataset MovieLens-1M, ML-100k, LastFM (2k) and Goodbooks (10k) datasets. We compare our model with two different techniques:

1. Accuracy Centric – We use basic collaborative algorithms like SVD (Singular value decomposition) (Ranjan et al., 2019), SVDpp (Singular value decomposition++) (Jia et al., 2014), NNMF (Non-negative Matrix Factorization) (Koren et al., 2009). All are discussed in Section 2. We also evaluated our proposed approach with reinforcement learning-based recommendation model's TPGR (Chen et al., 2019) and SlateQ (Ie et al., 2019) for three datasets ML-100k, Ml-1M, and LastFM (2k).
2. Diversity Centric – Next we evaluation our model for diversity so we use state of the art diversification algorithm first is Greedy algorithm (Ziegler et al., 2005) second is xQuAD (Abdollahpouri et al., 2019) and the third one is MMR (Maximal Marginal Relevance) (Carbonell and Goldstein, 1998). All approach's are discussed in Section 2.5. These algorithms are used as re-ranking approaches used as a post-processing step for diversity in the recommendation system.

For comparative analysis of our Clus-DR model, we use two different decisive algorithms. First, we use state-of-the-art algorithms for accurate recommendation generation, and second, algorithms are used for diversity enhancement of the recommendation system. Results for our proposed approach are shown in Tables 7 and 8. In terms of accuracy, our model performance is limited. The accuracy of the reinforcement-based learning methods is

Table 6
Comparison of various accuracy metrics for the proposed Clus-DR model for precision (higher value is better), nDCG (higher value is better), diversity, and coverage.

Dataset	Proposed Model	Algorithm	Precision	nDCG	IL-D	Coverage
		Clus-SVD	**0.2831**	**0.2781**	0.7336	53.98
		Clus-SVDpp	0.2646	0.2358	0.7703	50.33
	Clus-DR (IDS)	Clus-NMF	0.2697	0.2361	**0.8028**	**55.62**
		Clus-SVD	0.2557	0.2667	0.7674	51.41
		Clus-SVDpp	0.2203	0.2273	0.7872	52.41
ML-1M	Clus-DR (IFS)	Clus-NMF	0.2332	0.2443	0.8058	55.67
		Clus-SVD	0.3435	0.3552	0.7138	52.16
		Clus-SVDpp	0.2422	0.2398	0.7071	49.32
	Clus-DR (IDS)	Clus-NMF	0.1900	0.2039	0.7875	82.67
		Clus-SVD	0.3214	0.3364	0.6621	41.03
		Clus-SVDpp	0.2215	0.1933	0.6229	42.93
ML-100k	Clus-DR (IFS)	Clus-NMF	0.1792	0.1822	0.7003	48.55
		Clus-SVD	0.2245	0.2247	0.5336	39.22
		Clus-SVDpp	0.2117	0.2338	0.5662	40.33
	Clus-DR (IDS)	Clus-NMF	0.2998	0.2445	0.6028	41.22
		Clus-SVD	0.2007	0.1998	0.5028	37.44
		Clus-SVDpp	0.2109	0.2188	0.5331	35.22
LastFM (2k)	Clus-DR (IFS)	Clus-NMF	0.2398	0.2239	0.5773	39.88
		Clus-SVD	0.2008	0.1947	0.4036	29.83
		Clus-SVDpp	0.1988	0.2038	0.4632	25.09
	Clus-DR (IDS)	Clus-NMF	0.2198	0.2105	0.5088	38.98
		Clus-SVD	0.1992	0.2083	0.4108	27.38
		Clus-SVDpp	0.1889	0.1798	0.3931	25.07
Goodbooks (10k)	Clus-DR (IFS)	Clus-NMF	0.2038	0.2089	0.4273	35.88

The diversity value ranges from 01 and higher values are better and values for coverage are given in percentage (%). All results are for top-5 value.

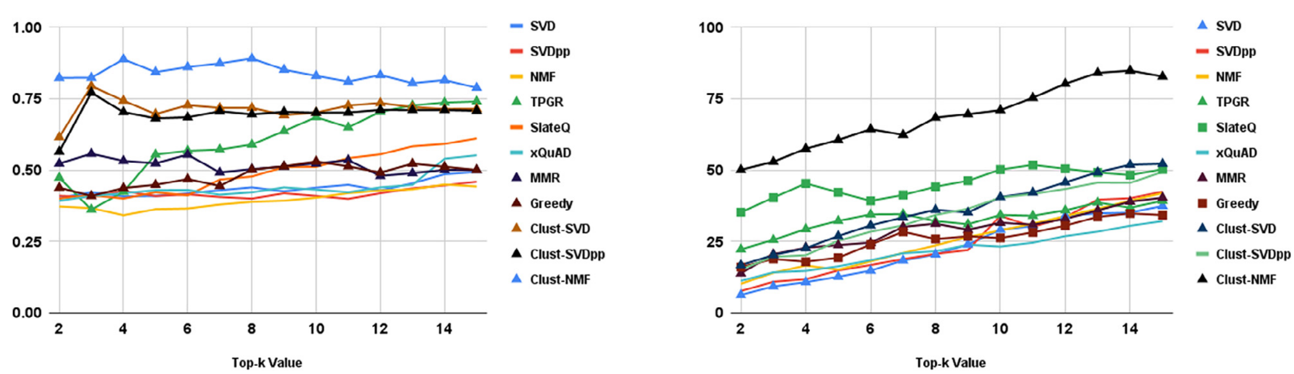
Table 7
Comparison of accuracy measures and top-15 results for the proposed model with the baseline model for precision and nDCG measure.

Proposed Model	Algorithm	RMSE	MAE	Precision	nDCG
	SVD (Ranjan et al., 2019)	0.974	0.8319	0.1572	0.1654
	SVDpp (Jia et al., 2014)	**0.8648**	**0.6022**	0.1361	0.1444
	NMF (Koren et al., 2009)	1.1058	0.9928	0.1453	0.1579
	TPGR (Chen et al., 2019)	1.0998	0.9823	0.2544	0.2779
Baseline (Accuracy based algorithm)	SlateQ (Ie et al., 2019)	1.0811	0.9466	0.2311	0.2841
	xQuAD (Abdollahpouri et al., 2019)	–	–	0.2422	0.2422
	MMR (Carbonell and Goldstein, 1998)	–	–	0.2196	0.2347
Baseline (Diversity based algorithm)	Greedy (Ziegler et al., 2005)	–	–	0.2016	0.2051
	Clus-SVD	1.1107	0.9707	**0.2831**	**0.2781**
	Clus-SVDpp	1.1181	0.8987	0.2646	0.2358
Clus-DR (individual diversity score)	Clus-NMF	1.1107	0.925	0.2697	0.2361
	Clus-SVD	1.112	0.8906	0.2557	0.2667
	Clus-SVDpp	1.1181	0.8987	0.2203	0.2273
Clus-DR (item feature)	Clus-NMF	1.1141	0.9765	0.2332	0.2443

perfect. The reason behind this accuracy drop in our proposed model is the diversity-accuracy trade-off and the sparsity of the dataset which is insignificantly enhanced due to the clustering and pair-wise training. However, the main aim of our approach is to increase the model performance for diversity and coverage, and our model performs best in terms of diversity and coverage which is shown in Table 6. The comparison of the diversity in top-*n* recommendation generation is calculated using intralist-diversity (IL-D) measure defined in the Eq. (10). The experiment of the proposed diverse recommendation generation method with other baseline methods is implemented with the ML-1M dataset. The results obtained from the Clus-DR model are analyzed using various accuracy metrics. In Table 6 shows that precision and nDCG for the Clus-DR model are more reliable than the baseline recommendation model. From Table 6, we concluded that using an individual diversity score for user clustering is good enough for diversity and accuracy instead of item feature scores. Accuracy drop in Clus-NMF and Clus-SVD are almost similar, but in terms of

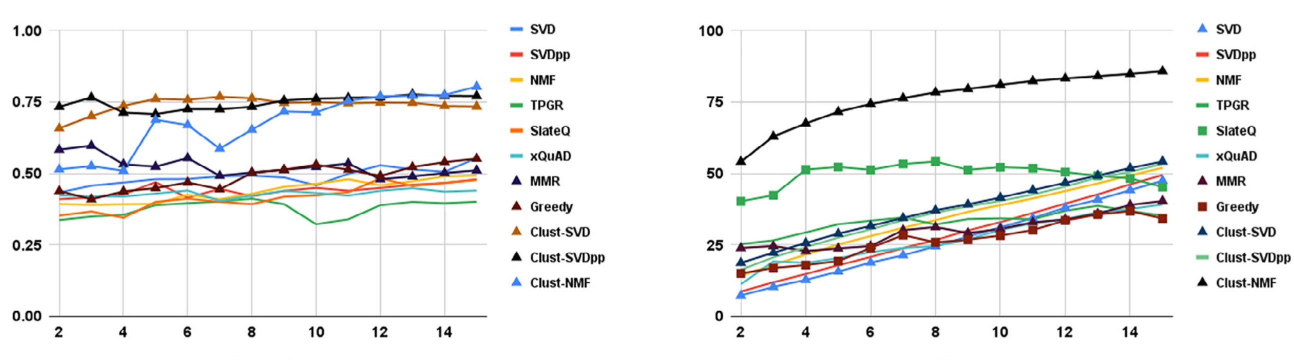
Table 8
Comparison of various results for the proposed model with the baseline model for diversity and coverage for top 5,10,15 recommendations.

		Intralist Diversity			Coverage		
Proposed Model	Algorithm	Top-5	Top-10	Top-15	Top-5	Top-10	Top-15
	SVD (Ranjan et al., 2019)	0.4794	0.4582	0.5535	15.65	31.11	47.41
	SVDpp (Jia et al., 2014)	0.4688	0.4492	0.4788	17.86	32.95	49.41
	NMF (Koren et al., 2009)	0.3928	0.4632	0.4923	25.12	38.88	51.92
	TPGR (Chen et al., 2019)	0.3885	0.3221	0.4008	32.19	34.25	35.22
Baseline (Accuracy based algorithm)	SlateQ (Ie et al., 2019)	0.4002	0.4229	0.4771	45.22	52.14	51.29
	xQuAD (Abdollahpouri et al., 2019)	0.4293	0.4311	0.4398	20.33	30.22	39.11
	MMR (Carbonell and Goldstein, 1998)	0.5233	0.5227	0.5103	23.74	30.55	40.33
Baseline (Diversity based algorithm)	Greedy (Ziegler et al., 2005)	0.4482	0.5299	0.5512	19.33	28.22	34.22
	Clust-SVD	**0.7601**	0.7489	0.7336	28.9	41.53	54.16
	Clust-SVDpp	0.7069	**0.7607**	0.7703	27.5	40.34	53.32
Clus-DR (Individual Diversity)	Clust-NMF	0.687	0.713	**0.8028**	**71.48**	**80.92**	**85.67**
	Clust-SVD	0.7571	0.745	0.7036	25.65	41.11	51.41
	Clust-SVDpp	0.7244	0.7641	0.7223	27.86	42.95	52.41
Clus-DR (item feature)	Clust-NMF	0.4251	0.7531	0.7728	41.48	50.92	55.67



(a) Diversity Graph (b) Coverage Graph

Fig. 6. A demonstration for top-k diversity and coverage analysis for ML-100k dataset for baseline and our proposed Clus-DR model.



(a) Diversity Graph (b) Coverage Graph

Fig. 7. A demonstration for top-k diversity and coverage analysis for ML-1M dataset for baseline and our proposed Clus-DR model.

diversity and coverage, Clus-NMF performance is better than Clus-SVD.

We also compared our model's diversity with other baseline algorithms (Accuracy Centric, Diversity Centric) using intra-list diversity measure defined in Eq. (10). Table 8 shows the comparison result for diversity for various recommendation list size, and it presents a better result for the Clus-DR model. From the comparative results, it is notable that we got on average **51.2%** more diverse items in the top-15 recommendation list than items recommended from the basic collaborative algorithm. Next, the comparative results are for Coverage analysis, which are for item space coverage, and shows the range of items that a recommender system can predict. Here, in this comparison graph, item space coverage is evaluated based on the genre of movies. We evaluated Clus-DR model for coverage analysis, and Table 8 shows the comparison result for Clus-DR model and another recommendation model for various recommendation list sizes. In coverage also we got average improvement in **29.8**% in item space. The comparative analysis for diversity and coverage for datasets ML-1M, ML-100k, LastFM and Goodbooks (10k) are shown in Figs. 6–9. Similarly, Figs. 10–12 represent the obtained results and their comparison graph of precision, recall, nDCG measures using a bar graph for ML-1M dataset.

5.4. Accuracy-diversity trade-off

We first study the trade-off when the models are trained for rating prediction using ranking technique. The descending order of predicted rating gives a ranking of recommendation list generated for the user. Now we evaluate the trade-off between accuracy and

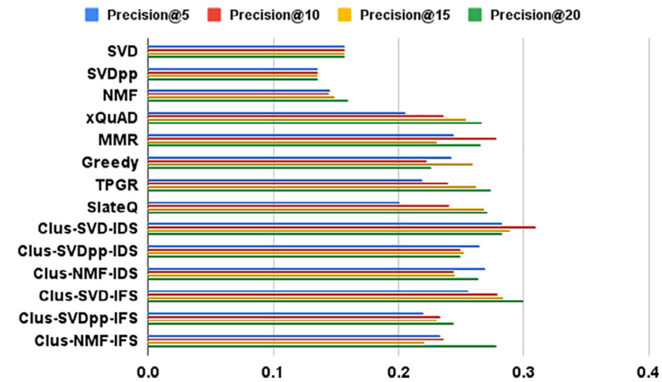
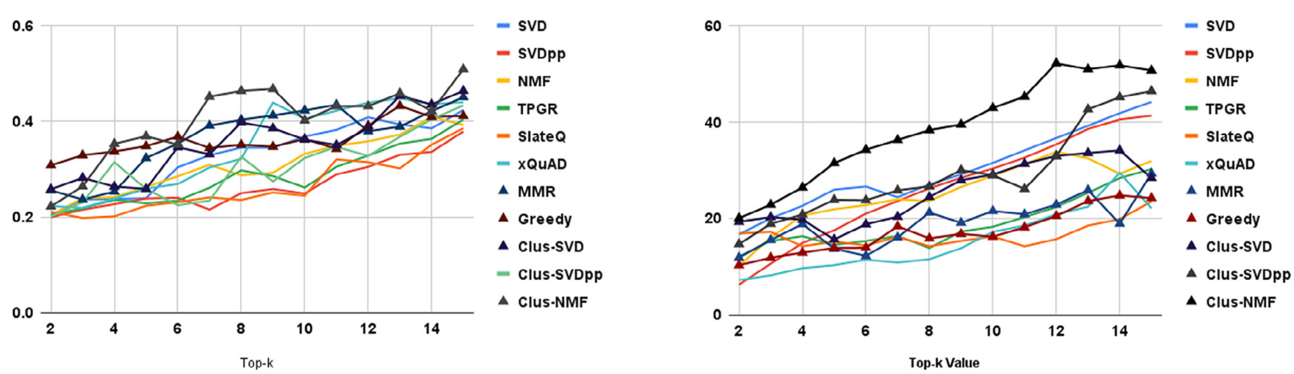


Fig. 10. A demonstration for top-*k* precision analysis for ML-1M dataset for baselines and proposed Clus-DR model.

diversity. The trade-off is plotted for the top-k recommendation list evaluation using nDCG and intralist diversity in Fig. 13a. The higher value of nDCG leads to a decrease in IL-D and vice versa. Similarly, we plotted a trade-off between nDCG and coverage in Fig. 13b, showing similar behavior like IL-D, where coverage increases while nDCG decreases.

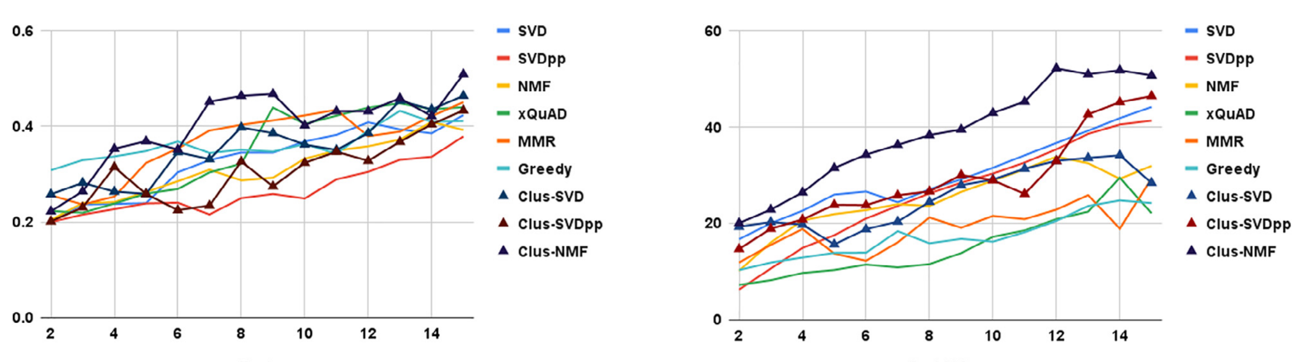
6. Ablation study

Effect of cluster size and algorithm- In our approach, we prove the flexibility of the approach by changing the cluster number and the clustering algorithm. We evaluated our model's performance



(a) Diversity Graph (b) Coverage Graph

Fig. 8. A demonstration for top-k diversity and coverage analysis for LastFM dataset for baseline and our proposed Clus-DR model.



(a) Diversity Graph (b) Coverage Graph

Fig. 9. A demonstration for top-k diversity and coverage analysis for Goodbooks (10k) dataset for baseline and our proposed Clus-DR model.

for various k (cluster size) values, and the results are shown in the Fig. 15. In Table 10, we show the result obtained from k = 10 using k-means algorithm. Similarly, we also shift the clustering algorithm from k-means to MiniBatchk-means, all the comparative results are shown in Table 11. The coverage result for cluster size (k = 10) and for cluster algorithm (Minibatchk-means) are shown in the Table 12. The results were obtained by changing the number of clusters value ($k = 10$) and replacing k-means with MiniBatchk-means shows some improvement in RMSE and MAE value. Still, IL-D and coverage results are almost similar to proposed results discussed in the subSection 5.3.1.

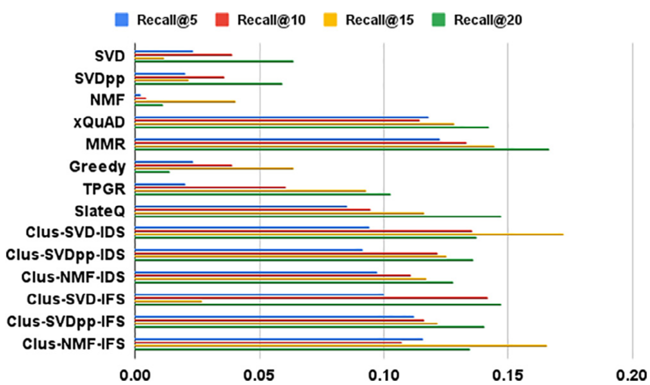


Fig. 11. A demonstration for top-k recall analysis for ML-1M dataset for baselines and proposed Clus-DR model.

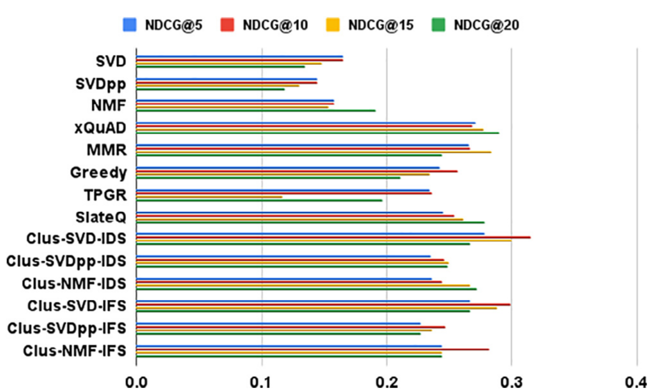


Fig. 12. A demonstration for top-k nDCG analysis for ML-1M dataset for baselines and proposed Clus-DR model.

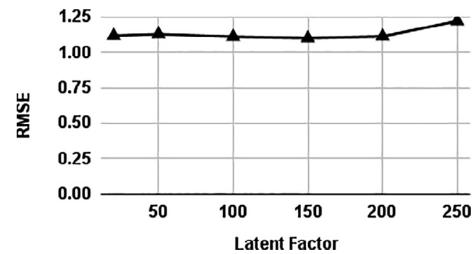


Fig. 14. Performance analysis for various latent factor for our proposed approach.

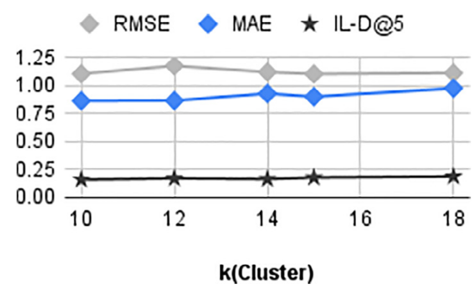


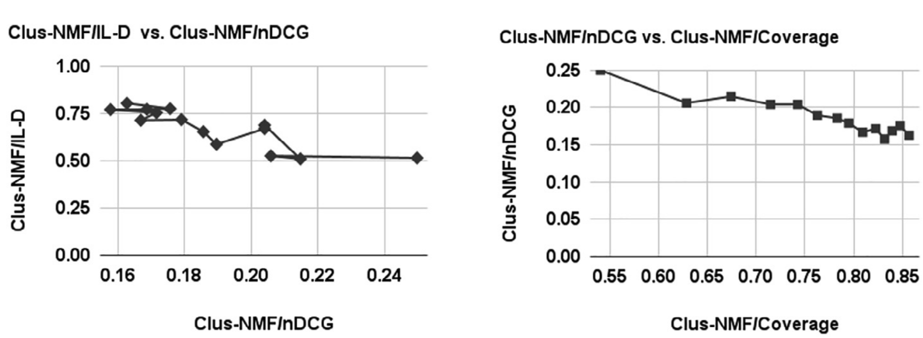
Fig. 15. Effect of cluster value (k) on RMSE, MAE and on diversity for top 5 recommendation.

Table 9
Cold-start prediction performance for the Clus-DR model.

Group	Approach	RMSE	MAE
Group 1 (0–5)	SVD	1.9662	0.9985
Group 2 (5–15)	SVD	1.8112	0.9887
Group 3 (15–30)	SVD	1.7665	0.9668
Group 1 (0–5)	Clus-DR	1.1985	0.9089
Group 2 (5–15)	Clus-DR	1.1507	0.9205
Group 3 (15–30)	Clus-DR	1.0797	0.9021

Effect of user and item latent vector- We obtained the best result for the individual diversity score for the Clus-NMF approach, where the rating prediction is based on the user and item latent factor size. We performed an analysis for accuracy using various latent factor sizes (20,50,100,150,200,250). The accuracy results obtained from Clus-NMF are shown using a line graph in Fig. 14 where we conclude that for latent factor size 100, we got the best results.

Online Experiment- The online evaluation of the recommendation system is a more realistic experiment. Our proposed model



(a) nDCG vs. intralist diversity	(b) nDCG vs. coverage

Fig. 13. Accuracy-diversity trade-off for top-k recommendation for Clus-NMF algorithm.

Table 10
Comparison of various results for the proposed Clus-DR model with the different cluster $k = 10$ value.

	Clus-SVD	Clus-SVDpp	Clus-NMF
RMSE	1.2141	1.1281	1.0529
MAE	0.9765	0.8987	0.8280
MAP@20	0.0131	0.0145	0.0197
nDCG@20	0.2300	0.2245	0.2522
Precision@20	0.2198	0.2200	0.2489
Recall@20	0.0330	0.0379	0.0473
IL-D Top-20	0.7985	0.7861	0.7735

Table 11
Comparison of various results for the proposed model with the Minibatchk-means algorithm.

	Clus-SVD	Clus-SVDpp	Clus-NMF
RMSE	1.2141	1.1281	1.0506
MAE	0.9765	0.8987	0.8298
MAP-20	0.0134	0.0166	0.0173
nDCG-20	0.2363	0.2384	0.2241
Precision-20	0.2255	0.2326	0.2164
Recall-20	0.0337	0.0407	0.0416
IL-D Top-20	0.7985	0.7861	0.7735

Table 12
Comparison of coverage results (%) for the proposed Clus-DR model for cluster size (k) = 10 and for Minibatachk-means algorithm.

	Clus-SVD	Clus-SVDpp	Clus-NMF
k = 10	46.47	48.57	49.20
Minibatchk-means	15.59	51.57	52.35

focuses on the top-n recommendation generation for each user, so we need to find out models efficiency for the new user and unseen items. For new user and unseen item, we experiment with leaving one out cross-validation where we remove one data from each user's top-n training data list and use that data to evaluate our model's performance. For cold-start prediction we followed the experimental analysis followed by Shi et al. (2018) where they divided users into three groups according to their rating records $(0,5), (5,15)$ and $(15,30)$. We compare our results with traditional SVD, SVDpp and NMF approach. The comparative performance of our proposed Clus-DR model is shown in the Table 9.

7. Statistical significance test

We perform a statistical significance test for performance comparison of our proposed approach Clus-DR and the second-best performing approach. A statistical significance test is based on a few simple ideas: hypothesis testing, the normal distribution, and p values. We performed a significance test for analysis of results obtained from Clus-DR to check whether they are significant or not. We did a paired t-test as a shred of statistical evidence, which shows that the mean difference between different paired results on a distinct outcome is significantly different from zero. The parametric t-test is defined as-

$$t = \frac{m}{s/\sqrt{n}} \tag{12}$$

where m is mean s is standard deviation of the difference between all pairs and n is the total number of samples we taken for testing. We test our hypothesis using a one-sided t-test where our hypothesis says that the difference between the mean of two different samples is greater than zero. For this test, we considered precision@k and nDCG@k values for top 5, 10, 15, 20 respectively. We also use the actual and predicted rating mean for the Clus-DR for the same hypothesis to check the stability and prove the significance of the proposed Clus-DR against other state-of-the-art techniques. For null hypothesis prove the statistic is supposed to follow the t-distribution with n-1 degree of freedom for a total number of samples. The p-value should be less than the x (x = 0.01, 0.05). We reject the null hypothesis for nDCG@k value paired samples and observed effect size d is medium 0.4 and $SD(standard\ deviation) = 0.017221$ and for second group $mean = 0.34080$ and $SD = 0.00509$. We reject the null hypothesis with a 95% confidence interval of this difference from -0.01121 to 0.01121. Similarity we rejected the hypothesis for precision with 95% confidence interval with t value = 0.007638.

8. Conclusion and future work

The primary objective of the recommender system is to provide a better recommendation to users through users' interest and relevance, which may change with time. This paper proposes a model based on the clustering of users to include a diverse and accurate recommendation to address the accuracy-diversity trade-off. Also, we achieved better results for the Clus-DR model for coverage and diversity and good comparative accuracy from other diversification algorithms. The proposed approach gives recommendations, which balances accuracy and diversity simultaneously. The experiments performed by our proposed algorithm suggest that a clustering-based collaborative algorithm provides comparable diverse recommendations and improves online performance significantly. In the future, we plan to include diversity based on session information with temporal dynamics in the recommendation system as future work. In our experimental analysis, we use reinforcement learning approaches for comparative analysis; from this, we can conclude consideration of reinforcement in the future for diversity enhancement will lead to a better recommendation. We also intend to extend our approach using a graph neural network to include diverse users' self-identification in the training phase for recommendation generation. We use four datasets from the movie and music domain to compare and evaluate our approach in this proposed work. However, there are many domains for user interaction in real-time applications like book and e-commerce, explored through different datasets like Amazon, Book-crossing, etc.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdollahi, B., Nasraoui, O., 2018. Transparency in fair machine learning: the case of explainable recommender systems. Human and machine learning. Springer, 21–35.
Abdollahpouri, H., Burke, R., Mobasher, B., 2019. Managing popularity bias in recommender systems with personalized re-ranking. In: The thirty-second international flairs conference.
Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B., Malthouse, E., 2021. User-centered evaluation of popularity bias in recommender systems. In: Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, pp. 119–129.
Adomavicius, G., Kwon, Y., 2011. Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans. Knowl. Data Eng. 24, 896–911.
Aghdam, M.H., Analoui, M., Kabiri, P., 2015. A novel non-negative matrix factorization method for recommender systems. Appl. Math. Inf. Sci. 9, 2721.
Anderson, A., Maystre, L., Anderson, I., Mehrotra, R., Lalmas, M., 2020. Algorithmic effects on the diversity of consumption on spotify. Proceedings of The Web Conference 2020, 2155–2165.

Anelli, V.W., Deldjoo, Y., Di Noia, T., Ferrara, A., Narducci, F., 2021. How to put users in control of their data in federated top-n recommendation with learning to rank. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing, pp. 1359–1362.

Beel, J., Gipp, B., Langer, S., Breitinger, C., 2016. paper recommender systems: A literature survey. Int. J. Digit. Libr. 17, 305–338.

Bobadilla, J., Bojorque, R., Esteban, A.H., Hurtado, R., 2017. Recommender systems clustering using bayesian non negative matrix factorization. IEEE Access 6, 3549–3564.

Bradley, K., Smyth, B., 2001. Improving recommendation diversity. In: Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland, Citeseer. pp. 141–152.

Carbonell, J., Goldstein, J., 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 335–336.

Castells, P., Hurley, N.J., Vargas, S., 2015. Novelty and diversity in recommender systems. Recommender systems handbook. Springer, 881–918.

Chen, H., Dai, X., Cai, H., Zhang, W., Wang, X., Tang, R., Zhang, Y., Yu, Y., 2019. Large-scale interactive recommendation with tree-structured policy gradient. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3312–3320.

Chen, S., Chen, B.H., Chen, Z., Wu, Y., 2020a. Itinerary planning via deep reinforcement learning. In: Proceedings of the 2020 International Conference on Multimedia Retrieval, pp. 286–290.

Chen, W., Ren, P., Cai, F., Sun, F., de Rijke, M., 2020b. Improving end-to-end sequential recommendations with intent-aware diversification. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 175–184.

Choi, S., Ha, H., Hwang, U., Kim, C., Ha, J.W., Yoon, S., 2018. Reinforcement learning based recommender system using biclustering technique. arXiv preprint arXiv:1801.05532.

Esmeli, R., Bader-El-Den, M., Abdullahi, H., 2019. Improving session based recommendation by diversity awareness. UK Workshop on computational intelligence. Springer, 319–330.

Fleder, D., Hosanagar, K., 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. Manage. Sci. 55, 697–712.

Hamedani, E.M., Kaedi, M., 2019. Recommending the long tail items through personalized diversification. Knowl.-Based Syst. 164, 348–357.

He, J., Hollink, V., de Vries, A., 2012. Combining implicit and explicit topic representations for result diversification, in. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp. 851–860.

Hu, L., Cao, L., Wang, S., Xu, G., Cao, J., Gu, Z., 2017. Diversifying personalized recommendation with user-session context. In: IJCAI. pp. 1858–1864.

Hurley, N.J., 2013. Personalised ranking with diversity. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 379–382.

Ie, E., Jain, V., Wang, J., Narvekar, S., Agarwal, R., Wu, R., Cheng, H.T., Lustman, M., Gatto, V., Covington, P., et al., 2019. Reinforcement learning for slate-based recommender systems: A tractable decomposition and practical methodology. arXiv preprint arXiv:1905.12767.

Jia, Y., Zhang, C., Lu, Q., Wang, P., 2014. Users' brands preference based on svd++ in recommender systems. In: 2014 ieee workshop on advanced research and technology in industry applications (wartia), IEEE, pp. 1175–1178.

Kaminskas, M., Bridge, D., 2016. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Trans. Interactive Intell. Syst. 7, 1–42.

Karakaya, M.Ö., Aytekin, T., 2018. Effective methods for increasing aggregate diversity in recommender systems. Knowl. Inf. Syst. 56, 355–372.

Karako, C., Manggala, P., 2018. Using image fairness representations in diversity-based re-ranking for recommendations. In: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, pp. 23–28.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer 42, 30–37.

Kyriakidi, M., Stefanidis, K., Ioannidis, Y., 2017. On achieving diversity in recommender systems. In: Proceedings of the ExploreDB'17. pp. 1–6.

Lathia, N., Hailes, S., Capra, L., Amatriain, X., 2010. Temporal diversity in recommender systems. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 210–217.

Lathia, N., Hailes, S., Capra, L., Amatriain, X., 2010. Temporal diversity in recommender systems. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 210–217.

Liu, F., Tang, R., Li, X., Zhang, W., Ye, Y., Chen, H., Guo, H., Zhang, Y., 2018. Deep reinforcement learning based recommendation with explicit user-item interactions modeling. arXiv preprint arXiv:1810.12027.

Luo, X., Zhou, M., Shang, M., Li, S., Xia, Y., 2016. A novel approach to extracting non-negative latent factors from non-negative big sparse matrices. IEEE access 4, 2649–2655.

Ma, Y., Narayanaswamy, B., Lin, H., Ding, H., 2020. Temporal-contextual recommendation in real-time. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2291–2299.

Najafipour, S., Hosseini, S., Hua, W., Kangavari, M.R., Zhou, X., 2020. Soulmate: Short-text author linking through multi-aspect temporal-textual embedding. IEEE Trans. Knowl. Data Eng.

Nassif, H., Cansizlar, K.O., Goodman, M., Vishwanathan, S., 2018. Diversifying music recommendations. arXiv preprint arXiv:1810.01482.

Oliveira, R.S., Nóbrega, C., Marinho, L.B., Andrade, N., 2017. A multiobjective music recommendation approach for aspect-based diversification., in: ISMIR, pp. 414–420.

Portugal, I., Alencar, P., Cowan, D., 2018. The use of machine learning algorithms in recommender systems: A systematic review. Expert Syst. Appl. 97, 205–227.

Ranjan, A.A., Rai, A., Haque, S., Lohani, B.P., Kushwaha, P.K., 2019. An approach for netflix recommendation system using singular value decomposition. J. Comput. Math. Sci. 10, 774–779.

Ribeiro, M.T., Lacerda, A., Veloso, A., Ziviani, N., 2012. Pareto-efficient hybridization for multi-objective recommender systems. In: Proceedings of the sixth ACM conference on Recommender systems, pp. 19–26.

Ricci, F., Rokach, L., Shapira, B., 2011. Introduction to recommender systems handbook. Recommender systems handbook. Springer, 1–35.

Shi, C., Hu, B., Zhao, W.X., Philip, S.Y., 2018. Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. 31, 357–370.

Sonboli, N., Eskandanian, F., Burke, R., Liu, W., Mobasher, B., 2020. Opportunistic multi-aspect fairness through personalized re-ranking. In: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, pp. 239–247.

Su, R., Yin, L., Chen, K., Yu, Y., 2013. Set-oriented personalized ranking for diversified top-n recommendation. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 415–418.

Su, R., Yin, L., Chen, K., Yu, Y., 2013. Set-oriented personalized ranking for diversified top-n recommendation. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 415–418.

Tavares, T.F., Collares, L., 2020. Ethnic music exploration guided by personalized recommendations: system design and evaluation. SN Appl. Sci. 2, 1–9.

Vargas, S., 2014. Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, p. 1281.

Volokhin, S., Agichtein, E., 2018. Understanding music listening intents during daily activities with implications for contextual music recommendation. In: Proceedings of the 2018 Conference on Human Information Interaction & Retrieval, pp. 313–316.

Wang, C., Zhang, M., Ma, W., Liu, Y., Ma, S., 2019. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. The World Wide Web Conference, 1977–1987.

Wang, X., Liu, Y., Xiong, F., 2016. Improved personalized recommendation based on a similarity network. Physica A 456, 271–280.

Wasilewski, J., Hurley, N., 2017. Personalised diversification using intent-aware portfolio. In: Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization. pp. 71–76.

Wilhelm, M., Ramanathan, A., Bonomo, A., Jain, S., Chi, E.H., Gillenwater, J., 2018. Practical diversified recommendations on youtube with determinantal point processes. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 2165–2173.

Xie, R., Liu, Q., Liu, S., Zhang, Z., Cui, P., Zhang, B., Lin, L., 2021. Improving accuracy and diversity in matching of recommendation with diversified preference network. arXiv preprint arXiv:2102.03787.

Yadav, N., Mundotiya, R.K., Singh, A.K., Pal, S., 2019. Diversity in recommendation system: A cluster based approach. International Conference on Hybrid Intelligent Systems, Springer, 113–122.

You, J., Wang, Y., Pal, A., Eksombatchai, P., Rosenburg, C., Leskovec, J., 2019. Hierarchical temporal convolutional networks for dynamic recommender systems. The world wide web conference, 2236–2246.

Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G., 2005. Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web, pp. 22–32.