Received: 28 April 2024 | Revised: 12 June 2024 | Accepted: 8 July 2024 | Published online: 18 July 2024

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2025, Vol. 4(4) 397–412
DOI: 10.47852/bonviewJCCE42023272

BON VIEW PUBLISHING

Diversity and Serendipity Preference-Aware Recommender System

Kexin Yin[1] and Junqi Zhao[2,*]

[1]*Institute for Financial Services Analytics, University of Delaware, USA*

[2]*College of Engineering, The Pennsylvania State University, USA*

Abstract: Diversity and novelty are essential objectives in recommender systems to improve stakeholders' benefits by reducing user's discovery efforts and improving business operators' sales and revenue. Existing diversity and novelty-based methods indifferently increase diversity or novelty for every user, which inevitably induces the trade-off dilemma between relevance and accuracy. Moreover, different users have different preferences for recommendation diversity and novelty. Such preference should be considered by a recommendation algorithm, thereby avoiding the trade-off dilemma and increasing the prediction accuracy. To address this research gap, we propose a new Diversity and Serendipity-Aware Recommender System (DSPA-RS) problem and its solution method. The MovieLens-2k data are used to evaluate our proposed DSPA-RS method against seven widely used recommendation methods in recommender systems as benchmarks. The test results demonstrate our method shows a superior performance than the benchmarks by a range of 34.30% to 108.27%, indicating that the movies recommended by our method best satisfy users' diversity and serendipity preference. For recommendation accuracy, our DSPA-RS method outperforms the most accurate method by 34.62% in Precision, 7.71% in Recall, and 24.37% in F1 score. The improvement in recommendation accuracy indicates that DSPA-RS's consideration and utilization of diversity preference and novelty momentum greatly improves recommendation quality.

Keywords: diversity and serendipity preference, recommender system, deep learning, optimization

1. Introduction

With the wide penetration of the Internet, how to manage the explosion of information becomes a crucial problem that urges effective solutions. Recommender systems, a research area that attracts extensive and consistent efforts from both academia and industry, are an efficient solution to mitigate the information overload that benefits both the users and the business operators. For users, a recommender system can proactively filter the massive available information and show the personalized recommendations, e.g., movies, products, and news, to a user. For business operators, a well-developed recommender system can promote product exposure, increase sales, and eventually boost revenue. For example, 80% of movies watched on Netflix came from recommendations [1]. Recommendation system has become one of the most ubiquitous user-centered artificial intelligence applications in modern information systems [2]. With the rapid development of accuracy-oriented recommender systems, researchers realize the importance of customer satisfaction and the discovery functionality of recommender systems, thus making diversity and novelty the two trending research topics of the area. Diversifying the recommendation can enrich user's system experience and increase user satisfaction by reducing duplicated recommendations [3], and adding novelty can improve the discovery functionality of the recommender system thus further improving user experience [4].

Recommendation diversity is a set-level metric which measures the difference among items in a recommendation list. Diversification methods aim to increase the diversity of a recommendation list to reduce duplicated and tedious recommendations. Existing diversification methods mainly aim at keeping a balanced performance between recommendation diversity and accuracy, as improving one performance usually comes with the cost for the other [3]. In this regard, the objective function of a typical diversification method is usually a sum of recommendation accuracy and diversity with different weights. This line of diversification methods, however, inevitably encounters the diversity accuracy dilemma [5]. In this vein, some research proposes personalized diversification methods to customize the diversity level of different individual user and alleviate the trade-off impact [6]. The existing adaptive diversification methods make attempts for personalized diversification but still rely on adjusting the trade-off parameter to control the diversification level. The personalized diversification, if well defined, should directly reflect the diversity level preferred by every single user, and a good personalized diversification method should greatly alleviate the accuracy and diversity trade-off. Meanwhile, traditional diversification methods often re-rank a recommendation list that recommends highly relevant items to a user using recommender like collaborative filtering (CF). The CF-based methods, however, push users to the same set of products and lead to a concentration of recommending popular products at user aggregate

***Corresponding author:** Junqi Zhao, College of Engineering, The Pennsylvania State University, USA. Email: junqi.zhao@alumni.psu.edu

level [7]. A promising solution for alleviating the concentration of recommending popular items is to improve recommendation novelty and optimize novelty synchronously with diversity. The improvement of novelty promotes the exposure of non-popular or long-tail products for business operators and leads to the system-level sale diversity for business operators and serendipitous product discovery for users [8].

Recommendation novelty can be improved by adding serendipity. Recommendation serendipity refers to the recommendation that is novel, i.e., different from the items a user historically viewed or purchased, but also useful to the user. Appropriately adding serendipity to recommender system not only improves users' experience and their long-term perceived recommendation diversity but also increases the exposure of cold items and broadens sales categories for the business operators [8]. Research about serendipity originates from the research of novelty. Recommendation novelty measures how a recommended item is different from the items a user previously seen or experienced [3]. However, focusing too much on the freshness of recommended items may lead to a dilemma that items are unexpected but also useless to the target user [4]. To emphasize more on the utility of the novel recommendations, serendipity is proposed to overcome the limitation of novelty-driven methods [9]. To increase recommendation serendipity, many algorithms have been proposed and the serendipity-oriented methods can be mainly classified into three categories: re-ranking algorithms, serendipity-oriented modification, and new algorithms [10]. Existing serendipity-oriented methods increase serendipity indifferently for all the user and overlook user's preference for novel items. According to Ziarani and Ravanmehr [4], different user has different level of novelty-seeking, and such difference can result from different personal attributes, e.g., personality, age, and gender. For example, because adolescent has lower self-restraint ability and emotional regulation, when compared with adults, adolescent typically has higher level of novelty-seeking. According to the novelty-seeking theory, user's personal novelty preference should be estimated according to her historical behavior. The novelty level of the recommendations should be adjusted to satisfy user's preference for novelty. In addition, studies are seldom found to simultaneously optimize diversity and novelty towards user's preference. Thus, a recommender system, which simultaneously considers user's preference to optimize diversity and novelty, is needed.

This study contributes to the extant body of knowledge by addressing the research gaps discussed above. Specifically, we propose a novel recommender system research problem and its corresponding method, both of which simultaneously take user's diversity preference and needs for novelty into consideration. Psychology studies discovered that familiarity and novelty are the two basic drivers that shape people's preference [5]. Anchoring in the theory, we divide user's preference into two components: diversity preference reflecting user's preferred distribution of familiar items and serendipity preference reflecting user's needs for novel items among relevant candidates. The metric of user diversity preference is an extension of diversity preference of link recommendation in our previous study [11]. Specifically, we define a user novelty-seeking momentum to measure a user's preference towards novel items. The novelty-seeking momentum is integrated into the re-ranking framework to pick items that meet both the preferred unexpectedness and usefulness, which satisfies a user's serendipity preference. The diversity preference and needs for novelty are optimized simultaneously in a sum-of-ratios optimization problem to select recommendations that best satisfy a user's preference. We demonstrate that our proposed method outperforms 8 different benchmarks on the MovieLens-2k data set.

The paper is organized as follows: Section 2 provides a review of existing research in recommender system considering diversity and serendipity preference and identifies the research gaps. The theoretical foundation supporting this study is discussed in Section 3. In Section 4, we describe the problem formulation and propose the DSPA-RS method as a solution. The empirical evaluation of the proposed DSPA-RS method is conducted in Section 6, with key findings discussed in Section 7.

2. Literature Review

In this section, we first review the representative studies in the field of accuracy-oriented methods in recommender system (see Section 2.1), ranging from the conventional methods [12, 13] to more recent methods leveraging deep neural networks (DNN) [14, 15]. Next, we conduct the review of current methods balancing diversity [5, 16, 17], novelty, and serendipity [3, 4, 10, 18, 19] with recommendation accuracy in recommender systems. The reviewed studies in this section help identify the research gap for existing recommender system, specifically the need for better estimation of user's novelty-seeking based on the historical behavior.

2.1. Accuracy-oriented methods in recommender system

2.1.1. Conventional recommendation system

Recommendation systems rely on the recommendation algorithms to estimate a user's preference towards items thus recommending items according to the user's preference [20]. The recommendation models can generally be classified into three categories, namely CF, content-based, and hybrid. CF leverages historical user-item interactions for recommendation, where the interactions could be either explicit ratings or implicit feedback (e.g., browsing history) from the user. Content-based recommendation mainly considers the auxiliary information (e.g., images or text) of both items and users. Hybrid models combine more than two types of recommendation strategies [12]. Popularized by the Netflix challenge, CF has become the mainstream recommender model for almost a decade (from 2008 to 2016) [21]. The CF aims to make the most of all users' collaborative behaviors when predicting a specific user's behavior. The early adoption of CF is based on the direct estimation of behavior similarity of either users or items. The Matrix Factorization (MF) based models later became popular, which collectively finds the latent spaces to encode the interaction matrix between users and items user-item [13]. More recent studies also try to improve the learning for latent factors, such as integrating $L_1$and L_2 norms in the loss function to balance the robustness and stability in recommender systems [22]. The MF becomes the de facto method for latent factor model-based recommendation.

Despite research efforts that have been devoted to enhancing the MF, it is worth noting two outstanding issues hindering the performance improvement of MF models. On the one hand, one user only has a limited number of behaviors when comparing with a large 80 volume of items; therefore, the critical challenge lies in CF is learning user-item representation accurately from the sparse interactions between users and items [2]. On the other hand, MF relies on an interaction function to operate on learned representations for users and items, thus generating the model output for recommendation decisions. However, it is well-known that MF performance

can be limited by the simple inner production as the interaction function, as the linear nature of interaction function in MF models is ineffective for complex user-item interactions from large-scale dataset [14].

2.1.2. DNNs-based recommender systems

The boom of DNN since the mid-2010s has revolutionized speech recognition, natural language processing, and computer vision. The power of deep learning comes from its capability in learning deep representations and abstractions from data [14]. The great success of DNN relies on its advantages for learning complicated patterns from extensive data. The advancement of DNN naturally sheds light upon overcoming the limitations within the conventional recommendation systems, particularly CF-based models. For representation learning, DNN is effective in learning the latent factors and useful representations from input data. In a real-world application, there exists a great volume of information regarding both items and users. Leveraging such information deepens our understanding of items and users, thus improving the recommender. Applying DNN in representation learning for recommendation models becomes a natural choice. In terms of modeling user-item interactions, DNN can model the nonlinearity with various nonlinear activation functions (such as relu, sigmoid, and their variants). This property allows capturing complicated interactions between users and items. As such, applying DNN in recommendation models is becoming one of the most thriving research topics in recommendation, which has achieved concrete progresses and demonstrates the potential of becoming technical foundations for next-generation recommender systems [2].

The representation learning models vary with the modeling techniques and input information. Relevant studies can be grouped into three categories: user-item historical interaction embedding, autoencoder-based model, and graph-based representation learning. The first group of studies focuses on embedding user-item historical interactions as part of the learned user representation. Representative models include Factored Item Similarity Model (FISM), which constructs user representation vector by pooling the interacted item embeddings [23], and SVD++ [24] model, which constructs final user representation via adding historical embedding learned by the FISM user representation. However, historical interactions were given equal or heuristic weights in these models (e.g., FISM and SVD++), which is not reasonable given different historical items should contribute differently for modeling users' representations or preferences. Therefore, some researchers integrate the attention mechanism to learn historical interactions more effectively. For instance, Deep Item-based CF model (DeepICF) [25] is a representative model adopting attention mechanisms for representation learning. As such, incorporating the attention mechanisms helps to model historical interactions in a more intelligent approach for learning representation.

The second group of studies utilize the idea of leveraging autoencoder for reconstructing input for representation learning. The autoencoder models use the interaction between user and item as input and then learn hidden representations for either user or item with the encoder [26]. One natural extension of autoencoder-based recommendation model is leveraging the variants of autoencoders into CF, such as applying variational autoencoders [27, 28].

Another trending group of studies is representation learning based on graph. The graph provides a new perspective on learning interactions of user-item and even user. Within the graph of user-item interaction, the interaction history between user and item represents the first-order connectivity among users. Thus, a further extension is exploring the higher-order connectivity from the graph of user-item interaction. Specifically, a user's second-order connectivity includes similar users co-interacting with the similar items. With the breakthrough of Graph Neural Networks (GNNs) for data in graph structure [29], recent research has put great efforts on modeling the bipartite user-item graph structure as neural graph-based representation learning. Simplified neural graph CF models eliminating unnecessary operations, such as Linear Residual Graph Convolutional Network (LR-GCCF) [30] and Light Graph Convolution Network, can also demonstrate superior performance in practices.

In addition to DNN-based representation learning in recommender systems, recent studies have also started to improve the DNN model design to address the challenges in the noisy training data, such as OR-AutoRec [31] developed for data with outliers and debiasing autoencoder [32] designed to overcome the bias. Additionally, given the impact of hyperparameters on the DNN model performance, more recent research efforts have also been put into designing a learning framework to optimize the hyperparameters in DNN-based recommender systems [33].

2.1.3. User-item interaction modeling

Conventional CF-based models rely on the inner production when scoring the user-item pairs, which fails to capture the complex user-item relationship [15]. To capture the complex interaction, researchers have replaced the inner production with MLPs, which is a general function that can be used to approximate any complex continuous function. An exemplary model is the Neural Collaborative Filtering (NCF) [15], which improves the recommendation quality by integrating MF's linearity and MLP's nonlinearity. Despite the improvement of DNN-based CF for capturing higher-order correlation between user-item dimensions, such improvement comes with the cost of increased computational cost, particularly for a large volume of data.

In summary, the studies in DNN-based recommendation models have been thriving recently, which suggests the boom of deep learning in recommender system research. Representation learning and modeling interaction function are two major advantages when leveraging DNN to enhance conventional CF models. It is worth noting that, among the diverse representation learning models for CF, GNN models have shown superiority for learning latent representation of users and items. According to a recent review [2], the success of GNNs can be attributed to two reasons. The bipartite graph between the user and item nodes can be used to represent the user-item interactions. By information propagation, GNNs encode the CF signal of user-item interactions. For interaction modeling, MLPs models have shown improved performances when replacing the simple inner production. Moreover, MLPs can naturally integrate with DNN-based representation learning models as an end-to-end recommendation model.

2.2. Diversity in recommender system

Considering the importance of the diversity within the recommended items [17], researchers have proposed various diversification methods for recommender systems. To meet the diversity requirement, most existing methods will re-rank recommended items by the accuracy-maximization recommender system [16, 17, 34]. Therefore, such methods are generic for being integrated with any existing recommender system [16]. In the subsequent paragraphs, we focus on reviewing these re-ranking methods as they

can be integrated with existing recommender system algorithm for diversifying recommendations.

Diversification methods, in general, strive to achieve the balance between recommendation accuracy and diversity, as the improvement of one usually comes with the cost of sacrificing the other. Accordingly, the objective function of such diversification methods is formulated as a weighted sum of recommendation accuracy and diversity, where the weights adjust a recommender system's focus on accuracy and diversity. Recommendation diversity can be measured using the average pair-wise dissimilarity of recommended items [17, 34]. Inspired by the maximal marginal relevance (MMR) method from information retrieval, Ziegler et al. [17] propose a greedy diversification method to select one item at a time for arriving at the final diversified recommendation list. An item will be selected if it maximizes the weighted sum of its own relevance score and its average dissimilarity to each item already selected in the final recommendation list. However, the greedy method bears the risk of achieving a sub-optimal solution given error accumulation in each iteration [35]. Instead of using a greedy heuristic, Zhang and Hurley [34] formulate the recommendation diversification as an optimization problem, where the weighted sum of recommendation accuracy and diversity will be maximized, given the constraint of recommending a predetermined number of items. Next, the solution to the optimization problem will be the diversified recommendation list. The diversification method developed in Chen et al. [16] also aims to balance the trade-off between recommendation accuracy and diversity but the novel diversity measure models the relationship among all recommended items rather than their pair-wise relationships. Based on an elegant probabilistic model,the determinantal point process (DPP), they propose a greedy algorithm to search for the final recommendation list.

This line of diversification methods, however, inevitably encounters the diversity accuracy dilemma [5]. In this vein, some research proposes personalized diversification methods to customize the level of diversity for individual user and ameliorate the trade-off impact. For example, items can be adapted to a user according to the user's taste range [36]. Existing adaptive diversification methods have attempted to achieve personalized diversification but still rely on adjusting the trade-off parameter to control the diversification level. Some users may prefer one certain genre of movie, e.g., thrillers or comedies, others may enjoy high-quality movies from different genres. Such preference can also be multi-dimensional. For example, a fanatical fan of thrillers may enjoy thrillers from all over the world, as long as the movie can make her hair stand on end. The other user may enjoy all kinds of movies, but all of those are Chinese movies. Thus, a movie recommender can apply a user's movie-watching history to construct her multi-dimensional diversity preference and utilize this diversity preference as the target to adjust the level of diversity within the recommendations to best satisfy a user's taste.

2.3. Novelty and serendipity in recommender system

Recommendation serendipity refers to the recommendation that is different from a user's historically viewed or purchased items but also useful to the user. Adding serendipity appropriately to recommender system not only increases users' experience and their long-term perceived recommendation diversity but also increases the exposure of cold items and broaden sales product categories for the business operators. In consideration of the benefits of serendipity, there is a bulk of research about serendipity emerging in the past decade. This section provides a systematic review about serendipity in recommender system.

The research of serendipity starts from the research about novelty. Novelty of recommendation measures how an item is different from the items a user previously seen or experienced [3]. The metric of recommendation novelty is usually formulated as the compliment of familiarity and measured using a set function, e.g., the maximum number of unseen items in a recommendation list. A recommender can increase novelty using a re-ranking method as the one utilized in traditional diversification methods and control the trade-off between recommendation accuracy and novelty using a trade-off parameter. Or even simpler, novelty can be achieved by filtering out items that has been encountered before. However, over-emphasizing the freshness of recommended items may lead to a dilemma that items are unexpected but useless to the target user [4].

To emphasize more on the utility of the novel recommendations, serendipity is proposed to overcome the limitation of novelty-driven methods. For example, Zuva and Zuva [19] measure unserendipity for a user using the cosine similarity between the item and the items that a user has experienced

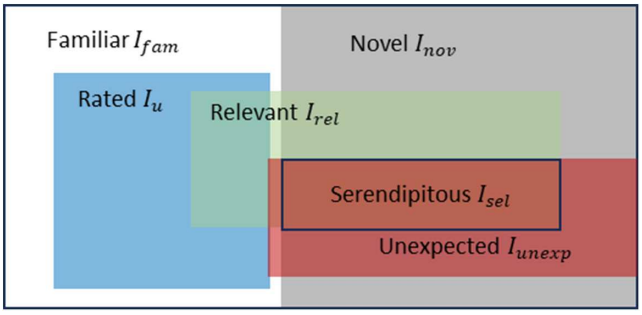
$$\text{Unserendipity}_u = \frac{1}{|H_u|} \sum_{h \in \mathbb{H}_u} \sum_{i \in R_u} \frac{\cos(i,h)}{|R_u|} \tag{1}$$

where H_u is the set of items that the user has experienced, Ru is the recommendation list to the user u. Serendipity measures that consider both unexpectedness and usefulness are also proposed in literature. A general way to measure usefulness of recommended item is using the relevance score produced by a baseline recommender [37]. Ge et al. [38] measure serendipity using the equation where R_{unexp} is the set of items that are not recommended by the baseline recommender. Adamopoulos and Tuzhilin [18] define R_{unexp} as $R \backslash R_{exp}$, which is the recommendations excluding those that are expected by the user. R_{exp} is composed of rated items and items that are duplicated with the rated items, where the item duplication is measured by feature similarity, i.e., cosine similarity or Jaccard coefficient. De Gemmis et al. [39] define highly related but not yet rated items as serendipity items and use the equation below to evaluate the serendipity at top N of a recommendation

$$\text{Serendipity}_u = \frac{|R_{unexp} \bigcap R_{useful}|}{|R|} \tag{2}$$

where $S(i) = 1$ represents the recommendation is serendipitous, otherwise zero, and recommendation list R contains the top N recommendations for user u. An illustration about item serendipity's relation with other item metrics in recommender system is given below.

Figure 1
An Euler illustration about the serendipity definition



Serendipity-based recommender system can be generally classified into three classes: Re-ranking algorithms, serendipity-oriented modification, and new algorithms [10]. Re-ranking algorithms improve serendipity by re-ranking a recommendation candidate list generated by a well-known relevancy-oriented algorithm. Adamopoulos and Tuzhilin [18] propose a re-ranking-based algorithm that can increase serendipity by re-ranking any candidate recommendation generated by a relevancy-oriented algorithm. This re-ranking algorithm balances the unexpectedness and relevance of recommendation items by first filtering out items that are irrelevant or too obvious based on the relevancy score and then re-rank the remaining items based on the weighted summation of unexpectedness utility and relevant utility [18]. Zhang et al. [40] propose a Full Auralist algorithm that can increase diversity, novelty, and serendipity at the same time without hurting accuracy too much. The Full Auralist algorithm suite consists of a relevancy predictor, a diversification component, and a delustering component which selects unexpected items. The Full Auralist outputs items with the highest weighted summation of the three scores, i.e., relevancy, diversity, and novelty, as the final recommendation. Because the overlap of relevancy and unexpectedness leads to serendipity (as shown in Figure 1), Full Auralist is the first method that considers diversity and serendipity at the same time for recommendation.

Serendipity-oriented modification algorithms modify the relevancy-oriented algorithm, e.g., k-nearest neighbor (KNN), to increase serendipity. The new algorithms are specifically designed for increasing serendipity based on proposed serendipity measurement. Nakatsuji et al. [41] propose a modified KNN algorithm to improve recommendation serendipity by replacing the distance measure in KNN into a relatedness measure, where the relatedness is predicted using random walk with restarts on a user distance graph. The modified KNN formulates a user's neighborhood by picking users that are related to the target user, i.e., high visiting probability in random walk, but are dissimilar, i.e., having low similarity score in terms of rated items [41].

The last group of serendipity methods directly designs new algorithms for the purpose of improving serendipity. De Gemmis et al. [39] improve recommendation's serendipity by the algorithm Random Walk with Restarts enhanced with Knowledge Infusion (RWR-KI). RWR-KI first creates an item similarity graph using information collected from Wikipedia and WordNet. Then, RWR-KI performs random walk on the similarity graph for each user where user's previously rated items are utilized as starting nodes and returns item relatedness score for each user. RWR-KI can recommend items having high relevancy to user's rated items, but the algorithm cannot guarantee the user's perceived level of novelty from the recommended items [10].

The current serendipity-based recommender systems increase serendipity indifferently for all the users and overlook user's preference for novelty level. According to Zeigler-Hill and Shackelford [42], different user has different level of novelty-seeking. Therefore, user's personal needs for novelty should be estimated based on their historical behavior. The level of novelty regarding the selected recommendation items among highly relevant candidates should be adjusted to achieve serendipity preference-aware recommendation.

3. Theoretical Foundation

Two factors shape people's preference [43]: familiarity and novelty. Familiarity is often said to be a major driver of preference, as revealed by many studies, a stimulus object's attractiveness monotonically increases with repeated exposure to the object [43]. Besides, novelty is the other main driver of people's preference, because people tend to prefer a novel stimulus over the older ones [44]. Echoing the two factors of people's preference, a preference-aware recommender system should model user's preference in terms of the interaction between the familiarity driver and the novelty driver.

Familiarity-driven preference should be modeled using a diversification method built upon an accuracy-oriented recommender, because diversification methods re-rank the familiar items fetched by accuracy-oriented methods to create a less monotonous recommendation list. Existing accuracy-oriented recommenders generate recommendations by learning user's preference towards familiarity items. According to our literature review, user's preference is learned by analyzing their past rating or purchasing behaviors and the items that are most similar to user's historical records will be recommended [2]. The accuracy-oriented systems aim only at maximizing the recommendation accuracy. As a result, items recommended by the accuracy-oriented systems tend to be similar to each other [34]. Duplicated recommendations make users feel tedious and harm user's satisfaction toward the system [45], while diversification technique is an effective solution to decrease duplication. A diversification method typically re-ranks the recommendation list generated by the accuracy-oriented system 91 to reduce the duplication [16]. Thus, applying diversification methods together with an accuracy-oriented recommender can generate refined ranking order of familiarity-based recommendations.

However, existing diversification methods indifferently increase diversity for every user and induce the accuracy-diversity dilemma, where the increase of diversity decreases recommendation accuracy. Diversity should be considered according to the users' different preference about diversity to avoid the accuracy-diversity dilemma. Psychological research suggests that people with different personalities have different preference for object's (e.g., movie) diversity. For example, people having low conscientiousness prefer higher level of overall diversity. Furthermore, people's diversity preference can be different across different feature dimensions [46], e.g., movies' genre, director, actor/actress. For example, people who are more nervous and reactive (i.e., people who are high in neuroticism) typically prefer a diverse set of directors; people who are creative and imaginative (i.e., people who are high in openness to experience) more likely to choose movies with diverse actors and actresses. Anchoring in the psychology theory, recommendation diversity should be added according to user's diversity preference towards the object's different feature dimensions, thus reducing accuracy-diversity dilemma and increasing the quality of familiarity-based recommendations.

Novelty-driven preference should be added to a preference-aware recommendation system. According to our literature review, novelty is an important element of recommendation system to increase users' experience and business operators' sales diversity. Existing novelty or serendipity-based recommenders indifferently increase novelty or serendipity for users regardless of how much and what kinds of novelty is wanted by the target user. Thus, the increase of novelty or serendipity always induces the decrease of recommendation accuracy. Novelty should be added according to the users' different preference about novelty. People prefer different level of novelty because novelty-seeking is a kind of personal trait [42], and different people have different levels of novelty-seeking. The difference results from 92 different personalities, genders, age, etc. For example, adolescent has higher level of novelty-seeking than adults because adolescent lack self-restraint and emotional regulation [42]. To design a generic method that considers people's novelty-seeking preference, we re-rank the recommendation items, i.e., high-relevance items, retrieved by an accuracy-oriented recommender. Because the novelty-seeking behavior is evaluated

based on a set of items with high relevance, according to our literature review, we name the novelty-seeking preference among relevant items as serendipity preference. Considering the interactive effect between the two drivers on preference, we optimize diversity preference and serendipity preference simultaneously to best capture the subtle formation of user's preference inside a system.

4. Problem Formulation

We first define the concept of diversity preference and novelty momentum. Then, we formulate the DSPA-RS problem. Let $\mathbb{U} = \{u_i\}, i = 1, 2, \cdots, n$ be the set of users and $\mathbb{V} = \{v_j\}, j = 1, 2, \cdots, m$ be the set of items. Matrix $\boldsymbol{R} \in \mathbb{R}^{n \times m}$ stores ratings users giving on items, where r_{ij} in $\boldsymbol{R}$ denotes the rating given by u_i to item v_j. Let $\mathbb{V}^i$ be the set of items that have been rated by user u_i, then we use the subset $\mathbb{V}^{i+}$, $\mathbb{V}^{i+} \subseteq \mathbb{V}^i$ to denote the set of items preferred by user u_i, where

$$\mathbb{V}^{i,+} := \{v_j | v_j \in \mathbb{V}^i, r_{ij} \geq r^+\}$$

and r^+ is a preferred rating threshold. Each item is described by a H-dimensional item profile, e.g., profile dimension< *genre*, *director*, *actor*, *studio* > for movies.

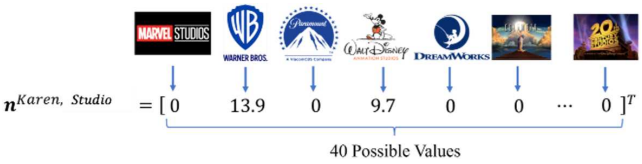
4.1. Diversity preference's definition and objective

The profile distribution of a user's high-rated items reflects the user's diversity preference in selecting items. For instance, if 6 of 10 high-rated movies from Karen is thriller, this indicates a high preference of thrillers for Karen. On contrary, if John's movie ratings are distributed evenly over genres, this suggests John has a diverse preference for movies. A recommendation algorithm can recommend items by leveraging user's diversity preference to best match this user's preferred item distribution, thereby the recommended items are more likely to be selected and get high rating from one user. Therefore, a user's diversity preference can be defined as below.

Definition 1. Diversity preference. Given a recommender system with user set $\mathbb{U}$, item set $\mathbb{V}$, and H-dimensional profiles of items in $\mathbb{V}$, user u_i's diversity preference for dimension h, $h = 1, 2, \ldots,$, is a Z-dimensional vector $d^{i,h} \in \mathbb{R}^{Z_h}$, where Z_h is the number of possible values of dimension h. The z_{th} element of $d^{i,h}$, $d_z^{i,h}$, represents u_i's preference on the z_{th} value on dimension h and is measured as the summation of ratings of items $v_j \in \mathbb{V}^{i+}$ that have the value at dimension h.

Example 1. If Karen has 18 high-rated movies in an online streaming platform, e.g., Netflix. The movie studio dimension is 40 unique values (i.e., $Z_{\{studio\}} = 40$), e.g., Universal, Marvel Studios, Warner Bros, Paramount, and Columbia. Among Karen's 16 high-rated movies, 10 are produced by Marvel Studios, 3 are produced by Warner Bros, 2 are produced by Walt Disney, and 1 is produced by Paramount. Table 1 below gives the information (i.e., movie's name, studio, and rating from Karen) of Karen's 18 high-rated movies.

According to Definition 1 and the movie ratings given in Table 1 below, Karen's diversity preference for movie studio $d^{\{Karen, studio\}}$ is



where T is the transpose for a vector.

An item v_j is potentially selected by user u_i if the item hasn't been rated by u_i, i.e., $v_j \notin \mathbb{V}^i$. The predicted rating $r_{\{ij\}}$ is the rating that u_i will give to v_j after selecting the item, i.e., watching the movie or purchasing the product, which can be estimated using an existing accuracy-oriented recommender [2]. A candidate's preferred item

Table 1
High rated movies from Karen (rating scale: 0–5)

Movie	Studio	Karen's Rating
Iron Man	Marvel Studios & Paramount	4.5
Iron Man 2	Marvel Studios & Paramount	4.3
Iron Man 3	Marvel Studios	4
Marvel's The Avengers	Marvel Studios	4.9
Avengers: Age of Ultron	Marvel Studios	3.8
Avengers: Infinity War	Marvel Studios	4.2
Avengers: Endgame	Marvel Studios	4.7
Thor: The Dark World	Marvel Studios	4.0
Thor: Ragnarok	Marvel Studios	4.2
Captain America: Civil War	Marvel Studios	4.1
Inception	Warner Bros	5
I am Legend	Warner Bros	5
Batman v Superman: Dawn of Justice	Warner Bros	3.9
Frozen	Walt Disney	4.7
Frozen II	Walt Disney	5
Forrest Gump	Paramount	4.9

of u_i is a potential selecting item of u_i with a relatively high predicted rating $r_{\{ij\}}$. Specifically, the u_i's set of preferred candidates is defined as $\mathbb{C}^i = \{v_j | rank(r_{ij}) < l, v_j \notin \mathbb{V}^i, j = 1, 2, \ldots, m\}$, which contains u_i's top-l potential selecting items as ranked by predicted rating.

The function $rank(\cdot)$ returns the rank of u_i's predicted rating on v_j among the predicted ratings given by u_i to each of u_i's potential selecting items. Given $\mathbb{C}^i$ as candidate set and $\mathbb{C}^i$ represents the profile of each candidate item, the candidate profile matrix $\boldsymbol{C}^{i,h} \in \mathbb{R}^{Z_h \times |\mathbb{C}^i|}$ can be constructed for each profile dimension h, $h = 1, 2, \cdots, H, where Z_h$ is the possible values at h dimension and the number of u_i's preferred candidates is $|\mathbb{C}^i|$. An element $C^{i,h}_{z,q}$ of $C^{i,h}$ is 1 if candidate q takes the z_{th} value in dimension h, otherwise 0.

Example 2. Given Karen and her preferred candidate set $\mathbb{C}^{Karen} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, there are 40 unique values at the movie studio dimension, i.e., $Z_{studio} = 40$. Among Karen's preferred candidates, movies v_1, v_2 are produced by Marvel Studios, v_3 is produced by Marvel Studios and Paramount together, v_4 is produced by Disney, v_5 is produced by Disney and Warner Bros together, and v_6 is produced by Universal and Warner Bros together.

Karen's candidate profile matrix for the movie studio dimension, $C^{Karen,studio} \in \mathbb{R}^{40 \times 6}$, is as follows:

$$C^{Karen,studio} = \begin{array}{c} \\ MarvelStudio \\ WarnerBros \\ Paramount \\ WaltDisney \\ DreamWorks \\ Columbia \\ \vdots \\ \vdots \\ 20th\ Century\ Studio \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & & \cdots & \cdots & & 0 \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ 0 & & \cdots & \cdots & & 0 \end{bmatrix} \end{array}$$

Given u_i's preferred candidate $\mathbb{C}^i$ and k items recommended to u_i, the diversity preference-aware recommender will select k candidates from $\mathbb{C}^i$ that meet $u_i's$ diversity preference, where $k < |\mathbb{C}^i|$. Specifically, the aim is maximizing the similarity between u_i's diversity preference and the diversity of the k preferred candidate items recommended to u_i. Let $\boldsymbol{y}^i = \begin{bmatrix} y^i_1 \cdots y^i_{|\mathbb{C}^i|} \end{bmatrix}^T$ be the recommendation decision vector, where $y^i_j = 1$ if the j^{th} candidate in $\mathbb{C}^i$ is recommended to u_i and $y^i_j = 0$ otherwise, $j = 1, 2, \cdots, |\mathbb{C}^i|$. The diversity of candidates recommended to u_i for dimension h, $\boldsymbol{r}^{i,h} \in \mathbb{R}^{Z_h}$, can be calculated as

$$\boldsymbol{r}^{i,h} = \boldsymbol{C}^{i,h} \boldsymbol{y}^i \tag{3}$$

Example 3. Given Karen's candidate set $\mathbb{C}^{Karen}$ and her candidate profile matrix for the movie studio dimension $\boldsymbol{C}^{Karen,studio}$ as described in Example 2. Let $\boldsymbol{y}^{Keran} = [111010]^{\mathrm{T}}$, i.e., candidates v_1, v_2, v_3, and v_5 will be Karen's recommendation. The distribution for the recommended candidates, $\boldsymbol{r}^{Karen,studio}$, is calculated as:

$$\boldsymbol{r}^{Karen,studio} = \boldsymbol{C}^{Karen,studio} \boldsymbol{y}^{Karen} \tag{4}$$

$$= \begin{array}{c} \\ MarvelStudio \\ WarnerBros \\ Paramount \\ WaltDisney \\ DreamWorks \\ Columbia \\ \vdots \\ \vdots \\ 20th\ Century\ Studio \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & & \cdots & \cdots & & 0 \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ 0 & & \cdots & \cdots & & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{array}{c} \\ MarvelStudio \\ WarnerBros \\ Paramount \\ WaltDisney \\ DreamWorks \\ Columbia \\ \vdots \\ \vdots \\ 20th\ Century\ Studio \end{array} \begin{bmatrix} 3 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

Among the 4 candidates recommended to Karen, 2 are produced by Marvel Studio, 1 is produced by Warner Bros and Disney together, and 1 is produced by Marvel Studio and Paramount together.

Definition 2. Diversity preference-aware objective. Given $d^{\{i,h\}}$, user u_i's diversity preference on item profile dimension h, and $\boldsymbol{r}^{i,h}$, the distribution of preferred candidates recommended to u_i, the cosine similarity is utilized to measure how much the recommended items match u_i's diversity preference:

$$cos\left(\boldsymbol{d}^{i,h}, \boldsymbol{r}^{i,h}\right) = \frac{\boldsymbol{d}^{i,h^T} \boldsymbol{r}^{i,h}}{||\boldsymbol{d}^{i,h}|| ||\boldsymbol{r}^{i,h}||}, \tag{5}$$

where $||\bullet||$denotes the L2 norm of a vector. Taking u_i's H-dimensional profiles into account, the diversity preference-aware objective is to maximize the total similarity between u_i's diversity preference and the diversity of the k recommended friends across all H dimensions

$$\sum_{h=1}^{H} \frac{\boldsymbol{d}^{i,h^T} \boldsymbol{r}^{i,h}}{||\boldsymbol{d}^{i,h}|| ||\boldsymbol{r}^{i,h}||}. \tag{6}$$

4.2. Novelty momentum and serendipity preference objective

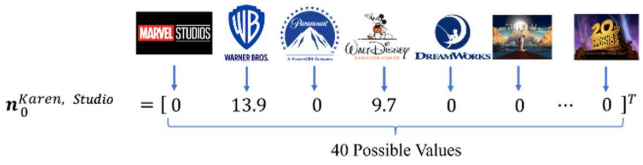
In recommender system, novelty refers to item features that never appear before. The task of a novelty-aware recommender is to find out the next most possible novel item features for u_i which has never been experienced before. To do so, we target the next most possible novel item features for u_i according to the following two drives:

1) Among all the unexperienced item features, those features that are most similar to user u_i's current novelty-seeking tendency more likely become the next novel features explored by u_i;
2) The higher the rating of a feature in u_i's currently novelty-seeking tendency, the more likely u_i would like to explore novel features that are similar to high-rated ones in her novelty-seeking tendency.

First, we define the calculation of a user u_i's current novelty-seeking tendency. The novel preference values appear in user u_i's preference within recent time t measures a u_i's novelty-seeking tendency towards certain item types. For example, in the past 2 months, Karen preferred 3 movies produced by Warner Bros and 2 movies produced by Disney, and both Warner Bros and Disney never appeared in Karen's preference for movie studios before. The newly appearing movie studios in Karen's profile reflect her novelty-seeking tendency for movies produced by Warner Bros and Disney recently. Thus, we measure a user u_i's novelty-seeking tendency using vector $\boldsymbol{n}^{i,h} \in \mathbb{R}^{|Z_h|}$, where the z_{th} element is the rating

summation of novel items preferred by u_i that take the value in dimension h.

Example 4. Karen's novelty-seeking tendency $\boldsymbol{n}^{Karen,studio} \in \mathbb{R}^{40}$ in the past 3 months is as following:



The trace-back period[1] of a user's novelty-seeking tendency can be extended to longer historical period. We create a user's novelty-seeking tendencies for past T trace-back periods, $\boldsymbol{n}_t^{i,h}$, $t = 1, 2, \ldots, T$, and conclude a user's novelty-seeking tendency as the weighted summation of the T historical novelty-seeking tendencies

$$\boldsymbol{n}^{i,h} = \sum_{t=1}^{T} \delta^t \boldsymbol{n}_t^{i,h} \tag{7}$$

where δ is the time decay parameter, $0 < \delta < 1$.

Second, we derive the calculation of feature similarity matrix $\boldsymbol{S}^h \in \mathbb{R}^{|Z_h|\times|Z_h|}$ for each profile dimension h. Let matrix $\boldsymbol{F}^h \in \mathbb{R}^{m\times|Z_h|}$ represent the item feature matrix on profile dimension h, where the element F^h_{jz} equals 1 if the item v_j takes the z_{th} value. Let matrix $\boldsymbol{R}^h \in \mathbb{R}^{n\times|Z_h|}$ represent the user-feature rating matrix for item profile dimension h, where $\boldsymbol{R}^h = \boldsymbol{R}\boldsymbol{F}^h$, and element R^h_{iz} denotes the rating user u_i giving to the z_{th} feature on profile dimension h. The value of element R^h_{iz} in matrix $\boldsymbol{R}^h$ is calculated as the summation of u_i's ratings to items that take the z_{th} feature on profile dimension h.

Example 5. Suppose in an online streaming platform, e.g., Netflix, matrix $\boldsymbol{R}$ stores movie ratings given by users u_i, $i = 1, \cdots, 6$, to movies $movie_j$, $j = i, \cdots, 6$. The matrix $\boldsymbol{F}^{studio}$ contains movies' profile for each studio values, where MS denotes Marvel Studio, PA denotes Paramount, WD denotes Walt Disney, DW denotes DreamWorks, and WB denotes Warner Bros. The following equations give examples of calculating user-feature rating matrix $\boldsymbol{R}^{studio}$.

$$\boldsymbol{R}^{Studio} = \boldsymbol{R}\boldsymbol{F}^{Studio} =$$

$$\begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} movie_1 & movie_2 & movie_3 & movie_4 & movie_5 & movie_6 \end{array} \\ \begin{bmatrix} 4.2 & 0.0 & 4.3 & 5.0 & 0.0 & 4.9 \\ 0.0 & 3.8 & 4.2 & 4.8 & 0.0 & 5.0 \\ 2.1 & 3.5 & 0.0 & 5.0 & 0.0 & 0.0 \\ 4.4 & 0.0 & 4.6 & 4.8 & 0.0 & 0.0 \\ 0.0 & 2.2 & 3.5 & 0.0 & 4.8 & 4.7 \\ 0.0 & 0.0 & 0.0 & 0.0 & 4.3 & 4.5 \end{bmatrix} \end{array} \begin{array}{c} \\ movie_1 \\ movie_2 \\ movie_3 \\ movie_4 \\ movie_5 \\ movie_6 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} MS & PA & WD & DW & WB \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{8}$$

$$= \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} MS & PA & WD & DW & WB \end{array} \\ \begin{bmatrix} 13.4 & 0.0 & 5.0 & 4.3 & 9.1 \\ 9.2 & 3.8 & 4.8 & 4.2 & 5.0 \\ 2.1 & 3.5 & 5.0 & 0.0 & 2.1 \\ 9.0 & 0.0 & 4.8 & 4.6 & 4.4 \\ 8.2 & 2.2 & 0.0 & 3.5 & 9.5 \\ 4.5 & 0.0 & 0.0 & 0.0 & 8.8 \end{bmatrix} \end{array}$$

The similarity between two profile features can be calculated based on the user-feature rating matrix $\boldsymbol{R}^h$. In matrix $\boldsymbol{R}^h$, each column $R^h_{:,z}$ is a user-feature rating vector that represents the ratings a feature gets from the users in the recommender system. The similarity between the two user-feature rating vectors indicates the similarity of ratings the two features get from all the users. To measure the similarity between two vectors, we use dot product, a popular vector similarity metric utilized in natural language process [47]. Dot product is chosen here for two reasons: (1) dot product not only calculates the similarity of two user-feature rating vectors but also reflects the feature popularity, i.e., number of rated times, of an item feature, and item popularity is an important feature in attracting user's novelty-seeking behavior; and (2) computational efficient. Accordingly, a high dot product similarity between a pair of item features indicates: (1) the two features are simultaneously rated by many users, and (2) the two features are similarly preferred (i.e., highly rated) by many users in the system. Thus, the item features similarity matrix $\hat{\boldsymbol{S}}^h$ on profile dimension h is calculated as $\hat{\boldsymbol{S}}^h = \boldsymbol{R}^{h^T}\boldsymbol{R}^T$, where each element $\hat{\boldsymbol{S}}^h_{Z_1,Z_2}$ of the similarity matrix is the inner product of the user-feature rating vector $\boldsymbol{R}^h_{:,z_1}$ and $\boldsymbol{R}^h_{:,z_2}$. Because an item feature's similarity with itself is useless for finding the next possible novel features, the diagonal elements of the final feature similarity matrix S are set to zeros and therefore

$$\boldsymbol{S}^h = \hat{\boldsymbol{S}}^h - diag\left(\hat{S}^h_{zz}\right), where \hat{\boldsymbol{S}}^h = \boldsymbol{R}^{h^T}\boldsymbol{R}^h \tag{9}$$

Example 6. Given the user-feature rating matrix for movie studio, $\boldsymbol{R}^{Studio}$, in Example 5, the movie studio similarity matrix $\boldsymbol{R}^{Studio}$ is calculated as below.

$$\begin{aligned} \boldsymbol{S}^{Studio} &= \hat{\boldsymbol{S}}^{Studio} - diag(\hat{\boldsymbol{S}}^{Studio}_{zz}) \\ &= \boldsymbol{R}^{Studio^T}\boldsymbol{R}^{Studio} - diag\left(\left(\boldsymbol{R}^{Studio^T}\boldsymbol{R}^{Studio}\right)_{zz}\right) \end{aligned}$$

$$= \begin{array}{c} \\ MS \\ PA \\ WD \\ DW \\ WB \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} MS & PA & WD & DW & WB \end{array} \\ \begin{bmatrix} 0.0 & 60.35 & 164.86 & 166.36 & 329.45 \\ 60.35 & 0.0 & 35.74 & 23.66 & 47.25 \\ 164.86 & 35.74 & 0.0 & 63.74 & 101.12 \\ 166.36 & 23.66 & 63.74 & 0.0 & 113.62 \\ 329.45 & 47.25 & 101.12 & 113.62 & 0.0 \end{bmatrix} \end{array} \tag{10}$$

With user's current novelty-seeking tendency and items' feature similarity defined, according to the two drives we mentioned at the beginning of Section 4.2, u_i's novelty momentum towards the z_{th} feature in profile dimension h is

$$p^{i,h}_z \propto \boldsymbol{S}^h_{z,\cdot}\boldsymbol{n}^{i,h} \tag{11}$$

which means the probability that u_i will explore the z_{th} novel feature is proportional to: (1) how the z_{th} novel feature is similar to u_i's current novelty-seeking tendency, and (2) how u_i prefers her current novelty-seeking tendency. We formally give the definition of u_i's novelty-seeking tendency for next time period, namely u_i's novelty momentum, as below.

Definition 3. Novelty momentum. Given the item feature similarity matrix $\boldsymbol{S}^h$, $h = 1, 2, \ldots, H$, and user u_i's current novelty tendency $\boldsymbol{n}^{i,h}$, $h = 1, 2, \ldots, H$, user u_i's novelty momentum at h dimension, is a Z_h-dimensional vector $\boldsymbol{p}^{i,h} \in \mathbb{R}^{Z_h}$, where Z_h is the values that is possible at the dimension h. The z_{th} element of $\boldsymbol{p}^{i,h}$, $p^{i,h}_z$, represents u_i's novelty momentum regarding the Z_h value of

[1] For different recommender systems, the length of the trace-back period should be different. For example, the trace-back period for movies should be longer than the trace-back

dimension h and will be zero if this value has appeared in u_i's current preference, else is measured as $\boldsymbol{S}^h_{z,:}\boldsymbol{n}^{i,h}$.

Example 7. Let's go back to Karen's example, where a larger movie feature similarity matrix is created as more movie studio are involved. Suppose the similarity matrix of movie studio $\boldsymbol{S}^{Studio}$ is

$$\boldsymbol{S}^{Studio} = \begin{array}{c} \\ MS \\ WB \\ PA \\ WD \\ DW \\ CO \\ \vdots \\ 20C \end{array} \begin{array}{c} \begin{array}{cccccccc} MS & WB & PA & WD & DW & CO & \dots & 20C \end{array} \\ \begin{bmatrix} 0.0 & 329.45 & 264.86 & 166.36 & 69.80 & 100.31 & \dots & 135.43 \\ 329.45 & 0.0 & 233.21 & 189.37 & 55.20 & 34.60 & \dots & 52.53 \\ 264.86 & 233.21 & 0.0 & 33.74 & 21.12 & 133.78 & \dots & 289.97 \\ 166.36 & 189.37 & 33.74 & 0.0 & 113.62 & 44.86 & \dots & 29.45 \\ 69.80 & 55.20 & 21.12 & 113.62 & 0.0 & 64.62 & \dots & 29.32 \\ 100.31 & 34.60 & 133.78 & 44.86 & 64.62 & 0.0 & \dots & 103.86 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 135.43 & 52.53 & 289.97 & 29.45 & 29.32 & 103.86 & \dots & 0.0 \end{bmatrix} \end{array}$$

Notes: MS: Marvel Studio, WB: Warner Bros, PA: Paramount, WD: Walt Disney, DW: Dream Works, CO: Columbia, 20C: 20th Century Studios.

Karen's novelty momentum for studio Marvel Studio, $p^{Karen,Studio}_{z=MS}$, is zero, because Marvel Studio has appeared in Karen's current diversity preference as shown in Example 1. And Karen's novelty momentum for studio Dream Works is measured as

$$p^{Karen,Studio}_{z=DW} = \boldsymbol{S}^{Studio}_{z=DW,:}\boldsymbol{n}^{Karen,Studio} = 1869.39.$$

Taking together, Karen's novelty momentum for all movie studios is

With novelty moment defined, the serendipity preference-aware objective is formally defined below.

Definition 4. Serendipity preference-aware objective. Given $\boldsymbol{p}^{i,h}$, user u_i's novelty momentum on item profile dimension h, and $\boldsymbol{r}^{i,h}$, the distribution of preferred candidates recommended to u_i, the serendipity preference-aware objective is to achieve the maximized similarity between u_i's novelty momentum and the distribution of the recommended k friends across H dimensions

$$\sum_{h=1}^{H} \frac{\boldsymbol{p}^{i,h^T}\boldsymbol{r}^{i,h}}{\|\boldsymbol{p}^{i,h}\|\|\boldsymbol{r}^{i,h}\|} \tag{12}$$

Notice that the objective here called "Serendipity Preference" not "Novelty Preference" because only items that are highly relevant to u_i are in the candidate set and participate in the re-ranking. According to our literature review in Section 2.3, the overlap between novelty and relevancy is serendipity. Thus, the set of relevant items that can best match a user's novelty momentum is a set of items that having serendipity.

4.3. Diversity and serendipity preference-aware recommendation problem

Before formally proposing the diversity and serendipity-aware recommendation problem, we introduce the preference balance parameter $\alpha^i \in \mathbb{R}^+$. According to the theoretical foundation in Section 3, the two factors, familiarity and novelty, shape people's preference and different individual relies differently on the two factors. Thus, the preference balance parameter α^i is a personalized parameter for every user to balance the relative importance of familiarity, which is measured by the diversity preference using familiar items, and the importance of novelty, which is measured by novelty momentum using novel items. We calculate the personalized preference balance parameter α^i using the ratio of u_i's preferred novel features by

$$\alpha^i = \sum_{h=1}^{H} \frac{\|\boldsymbol{n}^{i,h}\|_0}{\|\boldsymbol{d}^{i,h}\|_0} \tag{13}$$

where $\|\cdot\|_0$ denotes the cardinality of a vector that calculates the number of non-zero elements in a vector. A high ratio of $\frac{\|\boldsymbol{n}^{i,h}\|_0}{\|\boldsymbol{d}^{i,h}\|_0}$ indicates that among u_i's preferred features, a great portion of the features are novel and different from the features experienced by the users before the T time periods[2]. Thus, the higher the value of α^i, the more user u_i prefers novel items over familiar items. In addition to the personalized preference balance parameter α^i, a system-level parameter $\mu \in \mathbb{R}^+$ is also added to control the system-level propensity towards novelty discovery.

We formally propose the diversity and serendipity-aware recommendation problem below.

Definition 5. Diversity Serendipity Preference-Aware Recommendation (DSPA-RS) problem. Given a recommender system with the H-dimensional profiles of items, user u_i's diversity preference $\boldsymbol{d}^{i,h}$and novelty momentum $\boldsymbol{p}^{i,h}$ for profile dimension h, $h = 1, 2, \dots, H$, as well as the candidates $\mathbb{C}^i$ of user u_i, recommend k items in $\mathbb{C}^i$ to u_i, $k < |\mathbb{C}^i|$, such that the weighted summation of diversity preference-aware objective below is maximized.

$$\sum_{h=1}^{H} \frac{\boldsymbol{d}^{i,h^T}\boldsymbol{r}^{i,h}}{\|\boldsymbol{d}^{i,h}\|\|\boldsymbol{r}^{i,h}\|} + \mu\alpha^i \sum_{h=1}^{H} \frac{\boldsymbol{p}^{i,h^T}\boldsymbol{r}^{i,h}}{\|\boldsymbol{p}^{i,h}\|\|\boldsymbol{r}^{i,h}\|} \tag{14}$$

5. Method

By definition, the DSPA-RS problem for a given user can be formulated as the following optimization Problem (1):

$$\underset{y}{\text{maximize}} \sum_{h=1}^{H} \frac{\boldsymbol{d}^{h^T}\boldsymbol{r}^h}{\|\boldsymbol{d}^h\|\|\boldsymbol{r}^h\|} + \mu\alpha \sum_{h=1}^{H} \frac{\boldsymbol{p}^{h^T}\boldsymbol{r}^h}{\|\boldsymbol{p}^h\|\|\boldsymbol{r}^h\|}$$

$$\text{subject to} \mathbf{1}^{\mathrm{T}}\boldsymbol{y} = k$$

$$y_j \in \{0, 1\}, j = 1, 2, \cdots, c \qquad \text{Problem (1)}$$

where $\boldsymbol{r}^h = \boldsymbol{C}^h\boldsymbol{y}, \boldsymbol{C}^h$ represents item's candidate profile matrix for dimension h, c denotes the user's candidate friends, $\boldsymbol{y} = [y_1, y_2, \cdots, y_c]^{\mathrm{T}}$ is the decision vector, and $y_j = 1$ if the j^{th} candidate is recommended to the user and $y_j = 0$ otherwise. Since the

[2]A user's novelty-seeking tendency $\boldsymbol{n}^{i,h}$ concludes a user's novelty-seeking behavior for dimension h in the past T trace-back periods.

optimization problem is the same for every user, to simplify notations, we remove the user index i in Problem (1).

Theorem 1. Problem (1) is NP-hard.

Proof. It suffices that the Problem (1) is NP-hard if the NP-hardness when $H = 1$ is proven. Given $H = 1$, the objective function of Problem (1) can be rewritten as

$$\left(\frac{\boldsymbol{d}^T}{\|\boldsymbol{d}\|} + \mu\alpha\frac{\boldsymbol{p}^T}{\|\boldsymbol{p}\|}\right)\frac{\boldsymbol{C}\boldsymbol{y}}{\|\boldsymbol{C}\boldsymbol{y}\|} \tag{15}$$

Let $\boldsymbol{Q} = \boldsymbol{C}^T\boldsymbol{C}$ and $\bar{\mathrm{q}}^T = \left(\frac{\boldsymbol{d}^T}{\|\boldsymbol{d}\|} + \mu\alpha\frac{\boldsymbol{p}^T}{\|\boldsymbol{p}\|}\right)\boldsymbol{C}$ the Problem (1) for $H = 1$ can be rewritten as

$$\underset{y}{\text{maximize}}\frac{\bar{\mathrm{q}}^T\boldsymbol{y}}{\sqrt{\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y}}}$$

$$\text{subject to}\,\mathbf{1}^{\mathrm{T}}\boldsymbol{y} = k$$

,

$$y_j \in \{0,\ 1\},\ j = 1, 2, \cdots, m$$

This is identical to the diversity preference-aware link recommendation (DPA-LR) problem in our previous study [11] which has been proved by the NP-hardness. Therefore, the NP-hardness for Problem (1) is proven.

Following the common approach for solving NP-hard problems, we relax Problem (1) and derive its approximate solution. By letting $\bar{\boldsymbol{d}}^h = \frac{\boldsymbol{d}^h}{\|\boldsymbol{d}^h\|}$ and $\bar{\boldsymbol{p}}^h = \frac{\boldsymbol{p}^h}{\|\boldsymbol{p}^h\|}$ and substituting $\boldsymbol{r}^h$ with $\boldsymbol{C}^h\boldsymbol{y}$, the objective function of Problem (1) can be simplified as

$$\begin{aligned}&\sum_{h=1}^{H}\bar{\boldsymbol{d}}^h\frac{\boldsymbol{C}^h\boldsymbol{y}}{\|\boldsymbol{C}^h\boldsymbol{y}\|} + \mu\alpha\sum_{h=1}^{H}\bar{\boldsymbol{p}}^h\frac{\boldsymbol{C}^h\boldsymbol{y}}{\|\boldsymbol{C}^h\boldsymbol{y}\|}\\ &= \sum_{h=1}^{H}\frac{(\bar{\boldsymbol{d}}^h + \mu\alpha\bar{\boldsymbol{p}}^h)\boldsymbol{C}^h\boldsymbol{y}}{\|\boldsymbol{C}^h\boldsymbol{y}\|}\end{aligned} \tag{16}$$

By letting $\bar{\boldsymbol{a}}^h = \bar{\boldsymbol{d}}^h + \mu\alpha\bar{\boldsymbol{p}}^h$ and relaxing the binary integer constraint $y_j \in \{0, 1\}$ to $0 \le y_j \le 1$, the continuous relaxation problem of Problem (1) is

$$\underset{y}{\text{maximize}}\sum_{h=1}^{H}\frac{\bar{\boldsymbol{a}}^{h^T}\boldsymbol{C}^h\boldsymbol{y}}{\|\boldsymbol{C}^h\boldsymbol{y}\|}$$

$$\text{subject to}\,\mathbf{1}^{\mathrm{T}}\boldsymbol{y} - k = 0$$

$$\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b} \preccurlyeq \mathbf{0} \qquad \text{Problem (2)}$$

Notice that the vector $\boldsymbol{a}^h$ in Problem (2) is the weighted summation of normalized diversity preference $\bar{\boldsymbol{d}}^h$ and novelty momentum $\bar{\boldsymbol{p}}^h$, and the parameters multiplication $\mu\alpha$ together adjusts the relative importance of the two preference components.

Here we propose the DSPA-RS method. The Problem (2) is identical to problem DPA-LR in our previous study [11] and can be solved using the same iterative algorithm. The iterative algorithm presented in our previous study [11] finds a stationary point solution y^s to Problem (2). In order to solve Problem (1) based on the solution for Problem (2), the DSPA-RS method sorts the elements in vector y^s in descending and assigns 1 to the top-k elements and 0 to the rest.

6. Empirical Evaluation

The DSPA-RS method is evaluated using the MovieLens-2k data set, which is an extension of MovieLens10M data set [48]. In this section, we present the data summary and benchmark methods, evaluation procedure, and evaluation result.

6.1. Data set and benchmark methods

The MovieLens-2k data set was published by the GroupLens research group in 2011. In the MovieLens-2k data set, movies are linked with their introduction pages at Internet Movie Database and Rotten Tomatoes. The MovieLens-2k data set has been widely utilized for evaluating recommender system methods [49, 50]. Moreover, the empirical evaluation of our method requires the datasets has a fine-grained movie profile library and enough evolving time of the movie rating for defining user's serendipity and novelty preference. To the best of our knowledge, the Movielens-2k was then the only suitable dataset for our experiment requirement.

The Movielens-2k data set used in this study contains data about user's rating of movies from September 1997 to January 2009, and movies' profiles, e.g., genres, directors, production countries, film locations, and actors. The summary statistics of the MovieLens-2k data set is given in Table 2 below. As reported, there are 2,113 users, 10,197 movies, and 855,598 ratings in the data set, where the ratings have a scale of 0.5–5.

In the MovieLens-2k, each movie has a profile with six dimensions: movie genres, directors, countries, locations, actors, and tags. Table 3 below shows the summary statistics of the movie profile. In Table 3, $\boldsymbol{n}_h$ is the number of unique values at the movie profile dimension h. For example, there are 20 unique movie genres in the data set. The notation avg_h and max_h indicate the average and maximum number of profile values a movie has on profile dimension h, respectively. For example, a movie has up to 8 genres and 2.04 genres on average in the data set.

Table 3
Summary statistics of the cleaned profile data

Dimension (h)	n_h	max_h	avg_h
Movie genres	20	8.0	2.04
Directors	4.060	1.0	1.0
Countries	72	1.0	1.0
Locations	47.899	87.0	4.7
Actors	95.321	220.0	22.78

Table 2
Summary statistics of the MovieLens-2k data

#of Users	#of Movies	#of Ratings	Ratings per user	Ratings per movie
2,113	10,197	855,598	404.92	84.64

Table 4
Summary of methods compared in the evaluation

Method group	Method	Note
Our proposed method	DSPA-RS	Diversity Serendipity Preference-Aware RS
Accuracy-oriented methods	NCF	Neural Collaborative Filtering
	LR-GCCF	Linear Residual Graph Constitutional Network
Diversification methods	MMR	Maximum Marginal Relevance
	MSD	Max-Sum Diversification
	DPP	Determinantal Point Process-based method
	DiRec	Clustering-based method
Serendipity methods	M-PSVD	Modified PureSVD
	RWR-KI	Knowledge Infusion + Random Walk with Restarts

Given the DSPA-RS problem is a novel research problem, there is no existing recommender system algorithm that can directly solve the DSPA-RS problem. Therefore, we use three kinds of method as benchmarks: (1) pure accuracy-oriented recommender system algorithm, (2) diversification methods in recommender system, and (3) novelty or serendipity-oriented methods in recommender systems. For the accuracy-oriented group of methods, we choose two state-of-the-art recommendation algorithms (as reviewed in Section 2.1) NCF [15] and Linear Residual Graph Constitutional Network [30] as benchmarks. Notice that we use the prediction results from NCF to construct movie recommendation candidate sets for users. For the diversification methods, the four most representative generic diversification methods [2, 3] are adopted as benchmarks: the MMR-based method (MMR) [17], the max-sum diversification method (MSD) [34], the DPP method (DPP) [16], and the DiRec method [51]. Among the selected diversification benchmarks, the first three methods, i.e., MMR, MSD, and DPP, re-rank the recommendation candidates by balancing the trade-off between diversity and accuracy using a parameter, while Direc diversifies recommendations by clustering items based on their pairwise dissimilarities and then selecting one item with the highest relevance score from each cluster to form the final recommendation list [51]. For the last group of benchmark methods, i.e., the novelty and serendipity-oriented methods, we select the modified PureSVD (M-PSVD) [52] and the Random Walk with Restarts enhanced by Knowledge Infusion method (RWR-KI) [39]. In RWR-KI, we calculate the item similarity graph using the movie profile information. Table 4 above summarizes all the recommendation methods compared in this study. For the DSPA-RS method, we set the convergence threshold ε to 10^{-3}.

6.2. Evaluation procedure

We construct movie recommendation candidate set for each user, which will be utilized as the input to every re-rank-based methods, i.e., DSPA-RS, MMR, MSD, DPP, and DiRec. The movies in a candidate set $\mathbb{C}^i$ are the movies that haven't been rated by user u_i but have relatively high predicted rating by an accuracy-oriented recommendation algorithm. We adopt the NCF[3] [23] to predict the baseline rating from a user to all the unrated movies. Each user's top-100 predicted preferred movies are selected into the candidate sets. Specifically, because every user's historical rating behaviors are needed for constructing diversity preference and novelty momentum, we split training, validate, and test data according to each single user's rating timeline in the MovieLens-2k data set. In terms of each user rating timeline, we put the first 64% ratings into the training data, the next 16% ratings into the validation data, and the last 20% data into the test data. Figure 2 below gives an example of a single user's data split. Among the 13 rated movies of a user, the first 8 (i.e., $13 \times 0.64 = 8.32$) movies go to training set, the next 2 (i.e., $13 \times 0.16 = 2.08$) movies go to validation set, and the last 3 (i.e., $13 \times 0.20 = 2.60$) movies go to test set. During the evaluation, the optimal hyper-parameter configuration for NCF is determined by search. After experimenting with different model architectures, we finalize the model integrating 3 linear layers that gives higher precision and train the models using Adam optimizer with a learning rate of 1×10^{-3}. The hidden layer dimension of NCF is set to 150, and the dropout technique [53] is applied to all the linear layers. The model is trained for 2500 epochs with the batch size of 2000. Figure 3 presents the changes of training and validation loss during the 2500-epoch training of NCF.

All the re-ranking methods, i.e., DPA-LR and its benchmarks MMR, MSD, DPP, and DiRec, take candidate sets generated by NCF as input. We evaluate the recommendation performance using two types of metrics in recommender system: preference matching score and accuracy metrics. To evaluate if user's preference and recommendation are matched, we define the scores: diversity and serendipity preference matching score (DSPMS). The DSPMS is the averaged objective function of the DSPA-RS problem across H profile dimensions.

$$\text{DSPMS}_i = \frac{1}{H}\left(\sum_{h=1}^{H} \frac{\boldsymbol{d}^{i,h^T}\boldsymbol{r}^{i,h}}{\|\boldsymbol{d}^{i,h}\|\,\|\boldsymbol{r}^{i,h}\|} + \mu\alpha^i \sum_{h=1}^{H} \frac{\boldsymbol{p}^{i,h^T}\boldsymbol{r}^{i,h}}{\|\boldsymbol{p}^{i,h}\|\|\boldsymbol{r}^{i,h}\|}\right) \quad (17)$$

The DSPMS_i value of a recommendation list falls between 0 to 1. A higher DSPMS_i denotes better recommendations meet diversity and serendipity preference of the user.

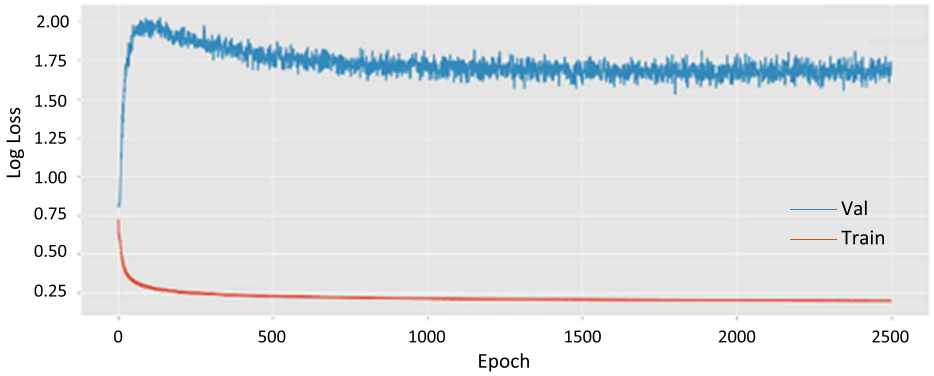
To evaluate the recommendation accuracy, we adopt precision, recall, and F1 score. The three metrics together measure the recommendation accuracy. Let TP_i denote the number of movies recommended that actually get high rating from the user u_i (i.e., rating > 3.0), and let P_i be the number of movies that truly get high rating from u_i (i.e., rating > 3.0). For each user u_i, the movie recommendation accuracy is defined as the percentage of the k recommended movies that are actually rated by u_i:

$$\text{precision}_i = \frac{\text{TP}_i}{k}. \quad (18)$$

[3]In our evaluation, NCF was implemented using code provided at https://www.kaggle.com/code/gpreda/neural-collaborative-filtering.

Figure 2
Training, validation, and test set split for MovieLens-2k

Figure 3
NCF training process



The recall denotes the percentage of the movies highly rated by u_i that also appear in the k recommendations generated by the method:

$$\text{recall}_i = \frac{\text{TP}_i}{\text{P}_i}. \tag{19}$$

The F1 score is the harmonic mean of precision and recall:

$$\text{F1Score}_i = \frac{2 \times \text{precision}_1 \times \text{recall}_i}{\text{precision}_i + \text{recall}_i}. \tag{20}$$

For evaluating a method, the DSPMS, precision, recall, and F1 Score, are based on the average of all the users, e.g.,

$$\text{DSPMS} = \frac{\sum_{i=1}^{n} \text{DSPMS}_i}{n}. \tag{21}$$

6.3. Results

We evaluate the recommendation performance of all methods in Table 5 with the top-10 recommendations, i.e., $k = 10$. For our proposed method DSPA-RS, we set the system-level novelty propensity μ to 1, which indicates a neural system preference for novelty discovery. For diversification methods having trade-off parameter, i.e., methods MMR, MSD, and DPP, we set the parameter θ to 0.5[4].

Table 5 below reports the performance of DSPA-RS and all the benchmarks on our evaluation metrics. As shown, our proposed DSPA-RS method shows the highest performance among all the metrics. For DSPMS, our method outperforms the benchmarks by 34.30% to 108.27%, indicating that the movies recommended by our method best satisfy users' diversity and serendipity preference. For recommendation accuracy, our DSPA-RS method outperforms the most accurate method by 34.62% in Precision, 7.71% in Recall, and 24.37% in F1 score. The improvement in recommendation accuracy indicates that DSPA-RS's consideration and utilization of diversity preference and novelty momentum greatly improves recommendation quality. Pair t-test is conducted between DSPA-RS with all the benchmarks for all the metrics, and DSPA-RS significantly outperforms benchmark methods ($p < 0.001$) with respect to all the evaluation metrics.

6.4. Performance analysis

After demonstrating DSPA-RS's superior performance over benchmarks, we perform performance analysis to unveil the secret of DSPA-RS's methodological novelty. DSPA-RS performs re-ranking on the prediction generated by NCF through optimizing the diversity preference objective and serendipity preference objective simultaneously, which constitutes the key methodology novelty of this study. If we optimize one objective at a time, the diversity serendipity preference-aware method is reduced to two methods: diversity preference-aware recommender system (DPA-RS) and serendipity preference-aware recommender system (SPA-RS). If we further drop the entire methodology novelty of the DSPA-RS method, it is reduced to a pure NCF method. Table 6 reports the performance comparison between DSPA-RS with three reduced methods: DPA-RS, SPA-RS, and NCF. As reported in Table 5, the DSPA-RS achieves the best DSPMS and prediction accuracy. Compared with DPA-RS and SPA-RS, DSPA-RS significantly ($p < 0.001$) improves DSPMS by a range of 5.81% to 106.21%, improve precision by 4.29-92.09%, recall by 5.26%-42.14%, and F1 score by 4.82%-67.49%. The improvements of prediction accuracy for DSPA-RS imply that user's preference is coordinately driven by the two factors, i.e., familiarity reflected by diversity preference and novelty reflected by novelty momentum. Thus, solely optimizing one of the drivers only produce sub-optimal predictions.

[4] These diversification methods optimize the weighted neural sum of diversity and relevancy. The parameter θ is the weight for diversity, and $1 - \theta$ is the weight for relevancy [7, 8, 10].

Table 5
Comparison between DSPA-RS and benchmark methods (k = 10)

Method	DSPMS	Precision	Recall	F1Score
DSPA-RS	0.8304	0.2126	0.0503	0.0700
NCF	0.4333 (91.65%)	0.1434 (48.32%)	0.0467 (7.71%)	0.0563 (24.37%)
LR-GCCF	0.4178 (98.75%)	0.1297 (63.97%)	0.0422 (19.30%)	0.0529 (32.32%)
MMR (θ = 0.5)	0.4108 (102.15%)	0.0811 (162.17%)	0.0251 (100.33%)	0.0315 (122.06%)
MSD (θ = 0.5)	0.4414 (88.11%)	0.1371 (55.13%)	0.0431 (16.70%)	0.0527 (32.83%)
DPP (θ = 0.5)	0.6043 (37.41%)	0.1580 (34.62%)	0.0434 (15.93%)	0.0560 (24.98%)
DiRec	0.6183 (34.30%)	0.1115 (90.70%)	0.0332 (51.64%)	0.0424 (65.25%)
M-PSVD	0.3987 (108.27%)	0.0933 (128.03%)	0.0282 (78.34%)	0.0340 (106.00%)
RWR-KI	0.4080 (103.54%)	0.1024 (107.64%)	0.0317 (58.87%)	0.0399 (75.57%)

Note: The number in the parentheses denotes the percentage improvement of our method over a benchmark.

Table 6
Comparison between DSPA-RS and benchmark methods (k = 10)

Method	DSPMS	Precision	Recall	F1Score
DSPA-RS	0.8304	0.2126	0.0503	0.0700
DPA-RS	0.7848 (5.81%)	0.1959 (8.55%)	0.0478 (5.26%)	0.0658 (6.41%)
SPA-RS	0.4027 (106.21%)	0.1107 (92.09%)	0.0354 (42.14%)	0.0418 (67.49%)
NCF	0.4333 (91.65%)	0.1434 (48.32%)	0.0467 (7.71%)	0.0563 (24.37%)

Note: The number in the parentheses denotes the percentage improvement of our method over a benchmark.

Table 7
Comparison between DSPA-RS and benchmark methods (k = 10)

Method	DSPMS	Precision	Recall	F1Score
DSPA-RS	0.8304	0.2126	0.0503	0.0700
DSPA-MMR (σ = 0.1)	0.4931 (68.42%)	0.1241 (71.28%)	0.0409 (22.97%)	0.0493 (42.12%)
DSPA-MMR (σ = 0.5)	0.5642 (47.18%)	0.0969 (119.50%)	0.0311 (61.59%)	0.0378 (85.22%)
DSPA-MMR (σ =0.9)	0.7503 (10.67%)	0.1897 (12.09%)	0.0475 (6.04%)	0.0643 (8.88%)

Note: The number in the parentheses denotes the percentage improvement of our method over a benchmark.

To demonstrate the superior of DSPA-RS method, we modify the benchmark MMR by changing the diversity component in MMR's objective function to diversity and serendipity preference-aware objective, with the diversity serendipity preference parameter $\sigma = 0.1, 0.5, 0.9$. The modified MMR, namely DSPA-MMR, greedily optimizes the weighted summation of relevance and user preference. Table 7 above summarizes the performance comparison between DSPA-RS and DSPA-MMR, where our proposed DSPA-RS method significantly ($p < 0.001$) and consistently outperforms the DSPA-MMR across different values of θ. The results in Table 7 above further illustrate the methodology superiority of DSPA-RS.

7. Discussion and Conclusion

In this study, we extend the DPA-LR method to recommender systems. In recommender systems, diversity and novelty are essential objectives to improve stakeholders' benefits by reducing users' discovery efforts and improving business operators' sales and revenue. Existing diversity and novelty-based methods indifferently increase diversity or novelty for every user, which inevitably induces the trade-off dilemma with accuracy. Moreover, different users have different preferences for recommendation diversity and novelty [42]. Such preference should be considered by a recommendation algorithm, thereby avoiding the trade-off dilemma and increasing prediction accuracy. To address the research gap, we propose the new Diversity and Serendipity-Aware Recommender System (DSPA-RS) problem and its solution method. Using the MovieLens-2k data set, we demonstrate the superior predictive power of the DSPA-RS method over eight different benchmarks.

This study has several theoretical implications. First, this study informs the new direction of combined optimization of diversity and novelty–diversity and serendipity preference-aware recommendation. Psychology theories unveil that different users have different preferences towards the distribution of familiar and novel items, and such preference can be different on different item feature dimensions [42]. However, existing diversification and novelty-based methods indifferently increase diversity and novelty regardless of users' different preferences. As a result, methods aiming at maximizing diversity or novelty inevitably encounter the trade-off dilemma with accuracy. Our method finds the effective solution to avoid the trade-off by introducing recommendation diversity and novelty according to the user's preference, thus increasing recommendation accuracy. Second, we provide the extension of the DPA-LR method proposed in the first project, and DPA-LR's method generality is evident by DSPA-RS's good performance.

The current study can be extended regarding the following limitations. First, the current solution method of the DSPA-RS problem relaxes the integer assignment problem to a continuous problem and approximates the solution of the relaxed problem using an iterative algorithm. The integrality gap between the optimal solution of the integer problem and the rounding algorithm is hard to prove because of the inclusion of multiple relaxing operations. More research is needed to find an approximation solution with a tighter integrality gap in the future. Second, the current empirical evaluation only includes a MovieLens data set. Experiments on data sets from other types of recommender systems are needed. Additionally, although two representative methods, namely NCF and LR-GCCF, were used as benchmarking, empirical tests with more recent methods as benchmarking is necessary for future works for assessing the proposed method.

Acknowledgement

The authors are grateful to the Institute for Financial Services Analytics at University of Delaware for providing the GPU servers needed for experiments in this study.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The MovieLens10M data set that support the findings of this study are openly available at https://doi.org/10.1145/2827872, reference number [48]. The MovieLens-2k data set that support the findings of this study are openly available in Grouplens at https://grouplens.org/datasets/hetrec-2011/.

Author Contribution Statement

Kexin Yin: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Junqi Zhao:** Writing – original draft, Writing – review & editing, Visualization, Project administration.

References

[1] Wang, Y. (2022). Netflix: How to keep a continued success. In *Proceedings of the 2022 2nd International Conference on Enterprise Management and Economic Development*, 1215–1219. https://doi.org/10.2991/aebmr.k.220603.197

[2] Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2023). A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering, 35*(5), 4425–4445. https://doi.org/10.1109/tkde.2022.3145690

[3] Castells, P., Hurley, N., & Vargas, S. (2022). Novelty and diversity in recommender systems. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender systems handbook* (pp. 603–646). Springer. https://doi.org/10.1007/978-1-0716-2197-4_16

[4] Ziarani, R. J., & Ravanmehr, R. (2021). Serendipity in recommender systems: A systematic literature review. *Journal of Computer Science and Technology, 36*(2), 375–396. https://doi.org/10.1007/s11390-020-0135-9

[5] Stitini, O., Kaloun, S., & Bencharef, O. (2022). An improved recommender system solution to mitigate the over-specialization problem using genetic algorithms. *Electronics, 11*(2), 242. https://doi.org/10.3390/electronics11020242

[6] Liu, Q., Reiner, A. H., Frigessi, A., & Scheel, I. (2019). Diverse personalized recommendations with uncertainty from implicit preference data with the Bayesian Mallows model. *Knowledge-Based Systems, 186,* 104960. https://doi.org/10.1016/j.knosys.2019.104960

[7] Lee, D., & Hosanagar, K. (2019). How do recommender systems affect sales diversity? A cross-category investigation via randomized field experiment. *Information Systems Research, 30*(1), 239–259. https://doi.org/10.1287/isre.2018.0800

[8] Ziarani, R. J., & Ravanmehr, R. (2021). Deep neural network approach for a serendipity-oriented recommendation system. *Expert Systems with Applications, 185,* 115660. https://doi.org/10.1016/j.eswa.2021.115660

[9] Xu, Y., Yang, Y., Wang, E., Han, J., Zhuang, F., Yu, Z., & Xiong, H. (2020). Neural serendipity recommendation: Exploring the balance between accuracy and novelty with sparse explicit feedback. *ACM Transactions on Knowledge Discovery from Data, 14*(4), 50. https://doi.org/10.1145/3396607

[10] Kotkov, D., Wang, S., & Veijalainen, J. (2016). A survey of serendipity in recommender systems. *Knowledge-Based Systems, 111,* 180–192. https://doi.org/10.1016/j.knosys.2016.08.014

[11] Yin, K., Fang, X., Chen, B., & Sheng, O. R. L. (2023). Diversity preference-aware link recommendation for online social networks. *Information Systems Research, 34*(4), 1398–1414. https://doi.org/10.1287/isre.2022.1174

[12] Walek, B., & Fojtik, V. (2020). A hybrid recommender system for recommending relevant movies using an expert system. *Expert Systems with Applications, 158,* 113452. https://doi.org/10.1016/j.eswa.2020.113452

[13] Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems,* 240–248. https://doi.org/10.1145/3383313.3412488

[14] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys, 52*(1), 5. https://doi.org/10.1145/3285029

[15] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182. https://doi.org/10.1145/3038912.3052569

[16] Chen, L., Zhang, G., & Zhou, E. (2018). Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 5627–5638. https://dl.acm.org/doi/10.5555/3327345.3327465

[17] 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,* 22–32. https://doi.org/10.1145/1060745.1060754

[18] Adamopoulos, P., & Tuzhilin, A. (2015). On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Transactions on Intelligent Systems and Technology, 5*(4), 54. https://doi.org/10.1145/2559952

[19] Zuva, K., & Zuva, T. (2017). Diversity and serendipity in recommender systems. In *Proceedings of the International Conference on Big Data and Internet of Thing,*120–124. https://doi.org/10.1145/3175684.3175694

[20] Patel, K., & Patel, H. B. (2020). A state-of-the-art survey on recommendation system and prospective extensions. *Computers and Electronics in Agriculture, 178,* 105779. https://doi.org/10.1016/j.compag.2020.105779

[21] Pipergias Analytis, P., & Hager, P. (2023). Collaborative filtering algorithms are prone to mainstream-taste bias. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 750–756. https://doi.org/10.1145/3604915.3608825

[22] Wu, D., Shang, M., Luo, X., & Wang, Z. (2022). An L_1-and-L_2-norm-oriented latent factor model for recommender systems. *IEEE Transactions on Neural Networks and Learning Systems, 33*(10), 5775–5788. https://doi.org/10.1109/tnnls.2021.3071392

[23] Zhao, X., Zeng, W., & He, Y. (2021). Collaborative filtering via factorized neural networks. *Applied Soft Computing, 109,* 107484. https://doi.org/10.1016/j.asoc.2021.107484

[24] Wang, S., Sun, G., & Li, Y. (2020). SVD++ recommendation algorithm based on backtracking. *Information, 11*(7), 369. https://doi.org/10.3390/info11070369

[25] Xue, F., He, X., Wang, X., Xu, J., Liu, K., & Hong, R. (2019). Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems, 37*(3), 33. https://doi.org/10.1145/3314578

[26] Pan, Y., He, F., & Yu, H. (2020). Learning social representations with deep autoencoder for recommender system. *World Wide Web, 23*(4), 2259–2279. https://doi.org/10.1007/s11280-020-00793-z

[27] Truong, Q.-T., Salah, A., & Lauw, H. W. (2021). Bilateral variational autoencoder for collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 292–300. https://doi.org/10.1145/3437963.3441759

[28] Polato, M. (2021). Federated variational autoencoder for collaborative filtering. In *2021 International Joint Conference on Neural Networks*, 1–8. https://doi.org/10.1109/ijcnn52387.2021.9533358

[29] Waikhom, L., & Patgiri, R. (2023). A survey of graph neural networks in various learning paradigms: Methods, applications, and challenges. *Artificial Intelligence Review, 56*(7), 6295–6364. https://doi.org/10.1007/s10462-022-10321-2

[30] Chen, L., Wu, L., Hong, R., Zhang, K., & Wang, M. (2020). Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(1), 27–34. https://doi.org/10.1609/aaai.v34i01.5330

[31] Hu, Y., Wang, X., Liang, C., Li, J., Wu, D., & He, Y. (2022). OR-AutoRec: An outlier-resilient autoencoder-based recommendation model. In *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles*, 1918–1923. https://doi.org/10.1109/smartworld-uic-atc-scalcom-digitaltwin-pricomp-metaverse56740.2022.00277

[32] Huang, T., Liang, C., Wu, D., & He, Y. (2024). A debiasing autoencoder for recommender system. *IEEE Transactions on Consumer Electronics, 70*(1), 3603–3613. https://doi.org/10.1109/tce.2023.3281521/mm1

[33] Wu, D., Sun, B., & Shang, M. (2023). Hyperparameter learning for deep learning-based recommender systems. *IEEE Transactions on Services Computing, 16*(4), 2699–2712. https://doi.org/10.1109/tsc.2023.3234623

[34] Zhang, M., & Hurley, N. (2008). Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, 123–130. https://doi.org/10.1145/1454008.1454030

[35] Wu, Q., Liu, Y., Miao, C., Zhao, Y., Guan, L., & Tang, H. (2019). Recent advances in diversified recommendation. *arXiv Preprint:1905.06589*. https://doi.org/10.48550/arXiv.1905.06589

[36] Deldjoo, Y., Anelli, V. W., Zamani, H., Bellogin, A., & di Noia, T. (2021). A flexible framework for evaluating user and item fairness in recommender systems. *User Modeling and User-Adapted Interaction, 31*(3), 457–511. https://doi.org/10.1007/s11257-020-09285-1

[37] Alhijawi, B., Awajan, A., & Fraihat, S. (2023). Survey on the objectives of recommender systems: Measures, solutions, evaluation methodology, and new perspectives. *ACM Computing Surveys, 55*(5), 93. https://doi.org/10.1145/3527449

[38] Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010). Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems,* 257–260. https://doi.org/10.1145/1864708.1864761

[39] de Gemmis, M., Lops, P., Semeraro, G., & Musto, C. (2015). An investigation on the serendipity problem in recommender systems. *Information Processing & Management, 51*(5), 695–717. https://doi.org/10.1016/j.ipm.2015.06.008

[40] Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., & Jambor, T. (2012). Auralist: Introducing serendipity into music recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, 13–22. https://doi.org/10.1145/2124295.2124300

[41] Nakatsuji, M., Fujiwara, Y., Tanaka, A., Uchiyama, T., Fujimura, K., & Ishida, T. (2010). Classical music for rock fans?: Novel recommendations for expanding user interests. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 949–958. https://doi.org/10.1145/1871437.1871558

[42] Zeigler-Hill, V., & Shackelford, T. K. (Eds.). (2020). *Encyclopedia of personality and individual differences*. Switzerland: Springer. https://doi.org/10.1007/978-3-319-24612-3_301867

[43] Fang, X., Singh, S., & Ahluwalia, R. (2007). An examination of different explanations for the mere exposure effect. *Journal of Consumer Research, 34*(1), 97–103. https://doi.org/10.1086/513050

[44] Nussenbaum, K., Martin, R. E., Maulhardt, S., Yang, Y. J., Bizzell-Hatcher, G., Bhatt, N. S., ..., & Hartley, C. A. (2023). Novelty and uncertainty differentially drive exploration across development. *eLife, 12,* e84260. https://doi.org/10.7554/eLife.84260

[45] Cui, Z., Zhao, P., Hu, Z., Cai, X., Zhang, W., & Chen, J. (2021). An improved matrix factorization based model for many-objective optimization recommendation. *Information Sciences, 579,* 1–14. https://doi.org/10.1016/j.ins.2021.07.077

[46] Dhelim, S., Aung, N., Bouras, M. A., Ning, H., & Cambria, E. (2022). A survey on personality-aware recommendation systems. *Artificial Intelligence Review, 55*(3), 2409–2454. https://doi.org/10.1007/s10462-021-10063-7

[47] Chandrasekaran, D., & Mago, V. (2022). Evolution of semantic similarity—A survey. *ACM Computing Surveys, 54*(2), 41. https://doi.org/10.1145/3440755

[48] Harper, F. M., & Konstan, J. A. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems, 5*(4), 19. https://doi.org/10.1145/2827872

[49] Gupta, M., & Kumar, P. (2020). Recommendation generation using personalized weight of meta-paths in heterogeneous information networks. *European Journal of Operational Research, 284*(2), 660–674. https://doi.org/10.1016/j.ejor.2020.01.010

[50] Thakkar, P., Varma, K., Ukani, V., Mankad, S., & Tanwar, S. (2019). Combining user-based and item-based collaborative filtering using machine learning. In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018*, 173–180. https://doi.org/10.1007/978-981-13-1747-7_17

[51] Boim, R., Milo, T., & Novgorodov, S. (2011). Diversification and refinement in collaborative filtering recommender. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 739–744. https://doi.org/10.1145/2063576.2063684

[52] Zheng, Q., Chan, C.-K., & Ip, H. H. S. (2015). An unexpectedness-augmented utility model for making serendipitous recommendation. In *Advances in Data Mining: Applications and Theoretical Aspects: 15th Industrial Conference,* 216–230. https://doi.org/10.1007/978-3-319-20910-4_16

[53] Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: An overview. In *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020,* 599–608. https://doi.org/10.1007/978-981-15-3383-9_54

How to Cite: Yin, K., & Zhao, J. (2025). Diversity and Serendipity Preference-Aware Recommender System. *Journal of Computational and Cognitive Engineering*, *4*(4), 397–412. https://doi.org/10.47852/bonviewJCCE42023272