International Journal of Applied and Advanced Scientific Research (IJAASR)
International Peer Reviewed - Refereed Research Journal, Website: www.dvpublication.com
Impact Factor: 5.655, ISSN (Online): 2456 - 3080, Volume 11, Issue 2, July - December, 2026

HYBRID RECOMMENDER MODEL COMBINING RANDOM FOREST METHOD AND K-NEAREST NEIGHBOURS USING USER-RECOMMENDER INTERACTION

K. Dhiyaneshwaran* & P. Sharmila**

* Assistant Professor, Department of Computer Technology and Information Technology, Kongu Arts and Science College (Autonomous), Erode, Tamil Nadu, India

** Principal, Navarasam Arts and Science College for Women, Arachalur, Erode, Tamil Nadu, India

Cite This Article: K. Dhiyaneshwaran & P. Sharmila, "Hybrid Recommender Model Combining Random Forest Method and K-Nearest Neighbours Using User-Recommender Interaction", International Journal of Applied and Advanced Scientific Research, Volume 11, Issue 2, July - December, Page Number 1-11, 2026.

Type of Review: Peer Reviewed as per |C|O|P|E| Guidance.

DOI: https://doi.org/10.5281/zenodo.21139275

Abstract:

In today's digital and e-commerce world, RSs (recommender systems) play an important part in decisions taken for many sectors, such as purchasing a product, using a service, rating a movie, and so on. Existing RSs take into account only one form of user's behaviour, which is implicit. In most cases, user-recommender interactions are overlooked in real-time scenarios. A hybrid model based on interactions between users and recommenders is proposed in this work. The scheme can be used for examining performances of systems using evaluation metrics. User recommended interactions are defined in phase one. RSs accept user's questions for their N recommendations while recording user's decisions. If any of the goods are in the users' favour, browsing is continued and recommenders continue to function until the all recommendations are done. In the second phase, a hybrid model combining RFs (random forests) and KNNs (K-nearest neighbours) is developed. The performances are evaluated using a variety of measures based on the modified scenario. This work's experiments were carried out on Movie Lens data, and the results of the suggested method showed that it performed better than non-hybrid methods.

Key Words: Recommender Systems, K-Nearest Neighbour and Random Forest

1. Introduction:

Recommendations from RSs have been extensively studied for their displays on many things including movies, music, and books [1-3]. Techniques used for recommendations by RSs can based on Memories or models or hybrid systems [4] where memory based RSs use similarity measures [5] to compute distances between people or items. RSs based on models [6] establish suggestions using demographic, content, or aggregated data. Hybrid techniques for RSs [7, 8] integrate different types of recommenders for improved performances.

Most existing RSs assume user behaviours like surfing, rating, and acting non-sequentially. Users are in control of visited objects in browsing [9]. Scores are assigned in rating behaviours [10] where preferred item sequences have no connections with sequence-independent approaches [11]. RSs very rarely consider user's interactive scenarios in recommendations.

Hybrid RSs which recommend based on user's interactions are proposed in this paper where both user-recommended behaviours and recommender algorithms are built on these interactions. The study uses randomized algorithm to overcome cold start issues caused by sparse data. Recommendations are made incrementally till certain data thresholds are reached. In user recommended systems, the order of choices influence subsequent recommendations. interactions and results in active learning settings.

This work is divided into four sections where user-recommender interactions are defined first. The recommender behaviours are based on three parameters namely acceptance of user requests, recommendation of products to users based on algorithms, and maintaining user's preferences for future recommendations. User's actions are also divided into three stages: (1) receiving item recommendations, (2) comparing them to user's interests, and (3) choosing items for further investigations. In the proposed hybrid interactive RSs based on incremental learning, recommendations do not exist during initialization phases and products are recommended using random algorithms resulting in scarce data. Subsequently, items are suggested using a hybrid method that combines RFs and kNNs. Data scales reach specific points during steady stages. Primarily Items are proposed by kNNs followed by measures for recall and diversity [12] which assess the quality of RSs. The count of recommendations of rounds serve as covers for evaluating randomized recommendations, NNs (Neural networks) based recommendations and hybrid algorithm's recommendations.

The paper makes three contributions. (1) Interactions between users and recommenders are defined as they are more realistic with practical implications. (2) The commonly used accuracy metrics based on error like RMSEs, MAEs [13] are not a natural fit for evaluating interactive RSs and are evaluated by redefining two criterions namely recall and variety based on new scenarios. (3) The proposed hybrid algorithm is tested with interactive recommender processes.

Experiments on Movie Lens dataset (http://www.movielens.org/) show that (1) when ratios of random recommendations are very high, they do not significantly affect performances; (2) RSs can use NNs for recommendations; and (3) hybrid algorithms outperform non-hybrid algorithms. The remainder of the paper is structured as follows. The assumptions for the interactive recommendation scenario are made in Section 2. In our interactive scenario, we establish the user and recommender behaviours. Section 3 creates a new hybrid recommender system based on the interaction of the user and the recommender. Section 4 offers experimental results on the Movie Lens dataset using two recall and diversity metrics. The proposed hybrid algorithm's suitable parameters are examined in detail. Academic literatures on user-recommender interactions, active learning, NNs, hybrid filters and granular computing are summarised in Section 5 while Section 6 caries closing remarks.

2. Literature Survey:

Duch et al. (2002) suggested hybrid models by incorporating disparate measures [14]. Their proposed model when tested on ML-20M-GroupLens dataset showed that it could improve precisions by 8.02% in over collaborative filters.

Geetha et. al. (2021) proposed PDD scheme that was a combination of DMTs (data Mining Techniques) and predicted type 2 Diabetes in patients [15]. The study used K-Means Clustering combined with RFs for their proposed predictive model, PDD, which outperformed hierarchical clustering techniques and clustering using BNs (Bayesian Networks) with RFs in terms of prediction accuracies.

C. Y. Lin et al. [19] proposed hybrid RI-SGDs (real-time incremental stochastic gradient descents) strategy for updating feedbacks implicit on RSs using MFs (matrix factorizations) [16] as gathering implicit feedback evaluation scores are easier than explicit feedback evaluation scores, but have issues in MFs based RSs while transforming raw data into user's preferences. The study implemented their proposed updates on music RSs where their numerical results showed that their proposed RI-SGDs achieved equivalent recommendation accuracy when compared to completely retrained models but consuming only 0.02% of retraining times.

Yang and his associates (2020) presented BRS cS (BPR-Review-Score-Social), a hybrid recommendation model for top-N recommendations and by integrating social data, scores, and reviews. The study optimized ranks using upgraded BPR models which were trained by joint representative learning [17]. User trust model rated and reviewed social relationships while PV-DBOW model processed reviewed data and fully connected NNs rated reviewed data. The study's proposed BRSCS algorithm outperformed previous RSs like BRS c, User CF, and HRS c in experiments on Yelp dataset.

Zhang and colleagues (2019) suggested items to group of users called aggregated suggestions which was more generic than user-oriented or item-oriented techniques and thus more ideal for cold-start suggestions. These aggregated RSs were created using RFs [18]. The scheme was used to forecast how a set of users would rate specific types of items. Users, items, and rating data were combined in pre-processing for creating aggregated decision tables where rating were the decision attributes. The study also modelled data conversion procedures for new users/items. Forests were built for aggregated training sets in training stages where leaves were assigned discrete rating distributions. Four forecasting algorithms evaluated values based on distributions of trees in testing. The study's experimental results on Movie Lens dataset showed acceptable accuracy levels for the aggregation technique.

K. A. Fararni et al. (2019) proposed hybrid RSs for sequential suggestions called U2CMS. The proposed scheme combined collaborative and content-based similarity models with Markov chains and established associations between items by considering sequential patterns and content information [19]. The model used higher order Markov chains along with textual content information for describing sequential patterns in multiple time steps. The study's several experiments on Amazon datasets showed that their proposed U2CMS was capable of handling sparse data issues, wider ranges of N-ordered Markov Chains, and could accurately recognize similarities between items. The study's U2CMS outperformed many other techniques based RSs including RSs based on DLTs (deep learning techniques) and with better handling of sparse data. Moreover, the scheme's performances were consistent when compared with N-ordered systems.

This paper also examined interactions based on granular computing, user recommendations, KNNs, active learning and usage of hybrid filters. Interactions based on user recommendations are frameworks that generate appropriate suggestion lists after analysing user tasks and recommend based on algorithms [20]. User-recommender interactive interfaces receive user requests and a recommendation list from recommenders, (2) recommender algorithms use user information to generate recommendation lists (3) user's perceptions which become recommendations over a period of time can be called recommender's personality.

Active learning can be achieved using interactive suggestions which results in new knowledge from contexts of viewed information [21] and similar methods have been used in such types of learning [22].KNNs [23] are the most closely related approach as its classifications use nearest training instances to classify objects. Similarity techniques for users often compute similarities based on user item ratings where item-item NN versions determine similarities between them. Cosines, Pearson correlations, means squared differences, and Euclidean distances are conventional metrics used for computing similarities [24]. Random and NN algorithms were combined in the hybrid algorithm to improve performances and alleviate cold-start problems by hybrid filtering [25] which combines suggestions. Basic hybridization strategies include weights, mixers, switches, and feature combinations [26]. Bio-inspired or probabilistic approaches using hybrid filtering [4] include random algorithms [27], KNNs [28], and BNs [29].

The granules K and N are two different types of granules. The number of neighbours who participate in KNN suggestions are based the value of K while N implies the count of products recommended to users at any given time. The impact of these granules were also examined. Relational data mining, granular computing, and RSs are all combined in the Granular Association Rules [30]. The essences of granular computing are embodied in the description of information granules with various attribute-value pairings and sizes [31]. Granular computing [32] are novel approaches to data processing and provide unified frameworks for forming and processing information granules, and have recently been making significant progresses [33].

3. Defining Interactions:

3.1. Binary Relations for RSs:

Information system's binary relationships are defined in this section

Definition 1:

Let $M = \{ M_1, M_2, M_3, \ldots, M_n\}$ and $K = \{ K_1, K_2, K_3, \ldots, K_n\}$be two sets of objects. Any$R \subseteq M * K$ is a binary relation from R to M whose example is given in Table 1 where $M = \{ M_1, M_2, M_3, \ldots, M_n\}$ are user sets and $K = \{ K_1, K_2, K_3, \ldots, K_n\}$are movie sets. An information system can be viewed as a binary relation. However, it is more commonly stored in the database as a table with two foreign keys to save space.

Table 1: Binary Relation

MID/UID	K_1	K_2	K_3	K_4
M_1	1	1	0	1
M_2	1	0	0	1
M_3	0	1	0	0
M_4	0	1	0	1

3.2 Assumptions:

This work made assumptions listed below to simplify complexities of processing.

- User interests are pre-determined where small fixed subsets of objects interest them which may not be true due to changing user preferences.
- The item set remains unchanged. This may be true in slow-evolving applications. This is not the case with rapidly changing applications such as e-commerce, where things are often added and removed.
- The user set remains unchanged. This is analogous to the situation with the items set.
- RSs start without browsing histories i.e. all values of the browsing matrix are made 0.
- In each round, the user is recommended a set products and in web page searches, only a definite number of links to web pages or sites are displayed.

User interest matrices are constructed for assumption one and defined below:

Definition 2:

Let $M = \{ M_1, M_2, M_3, \dots, M_n\}$be the set of all users and $T = \{ T_1, T_2, T_3, \dots, T_n\}$the set of all items. The user interest matrix is defined as

$$MT_{nxm} \tag{1}$$

$$\text{Where } Um_{ij} = \begin{cases} 1, & if\ M_i\ interested\ in\ T_j\ ; \\ 0, & otherwise \end{cases} \tag{2}$$

3.3 User and Recommender Behaviours:

It is a dual way communication between recommenders and users when users log on to the system. RSs suggest one or more products which the user selects or ends the interaction. Based on inputs, users of RSs can be classified into two groups. When users specify browsing objects, they are referred to as browse users. When users assign scores to items, they are referred to as rate users. Throughout this study only browse users are considered.

Figure 1 depicts the user's behaviour. Browse users connect to systems and receive recommendations where users browse based on their interests and recommendations. The browse candidate sets can be defining as follows.

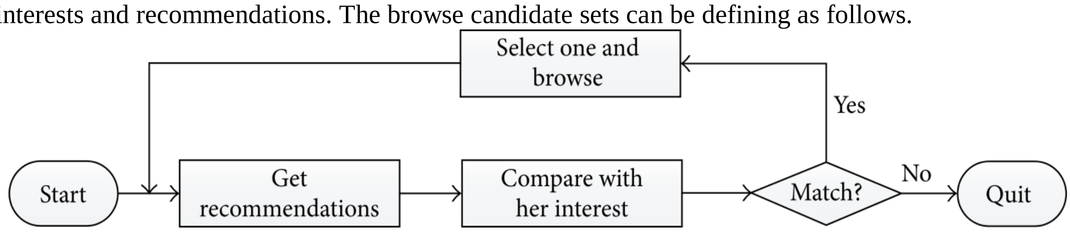

Figure 1: User Behaviour

Definition 3:

Let $T_i \subseteq T$ be the interests set of user u_j and $T_r = \{ t_{x1}, t_{x2}, \dots, t_{xn}\}$ the recommended set. $T_c = T_i \cap T_r$ is the candidate browse set. If $T_i \neq \emptyset$,u_j implies interest in selected items. If $L_r = \{ t_{x1}, t_{x2}, \dots, t_{xn}\}$ represents list of recommendations, users continue to browse for matches of$L_r$where the first match can be defined as

Definition 4:

If $t_x \in T_c$ and $\forall k \in \{1, 2, \dots, j-1\}$, $t_{xk} \notin T_c$, txj is the first match. If,$T_c = \emptyset$, u_j implies users have quit as recommendations do not match their interests and limited items are shown. Grey lists are defined for improving effective recommendations. Based on Definition 4, assume the list $L_o = \{ l_{x1}, l_{x2}, \dots, l_{xn}\}$ does not interest user U_j, then L_o items are added to grey lists to ensure they are not recommended again.

A matrix containing items browsed by users is known as the browse matrix and computed as:

$$bm_{mxn} \tag{3}$$

$$\text{Whereb} = m_{ij} = \begin{cases} \text{if } U_i, & interested\ in\ T_j \\ 0, & otherwise \end{cases} \tag{4}$$

Browse matrices are influenced by user behaviours and RSs like time intervals between browsing assuming most users do not return to RSs after quitting. A few patient users may visit RSs often called revisits. The behaviour of RSs is depicted in Figure 2.

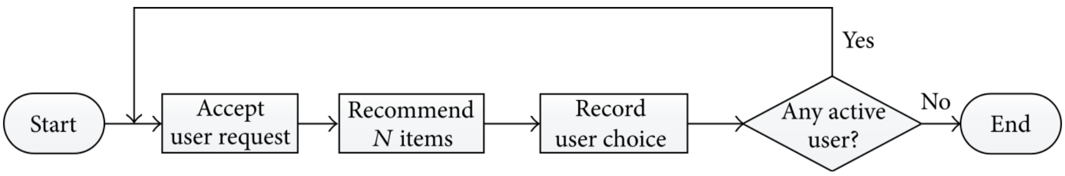

Figure 2: Recommender behaviour

When RSs receive login requests from users, they recommend N items to them. RSs then add user selections to browse matrices if they visit items which keep increasing as a result of more interactions (gradual learning). When users are unable to identify their interests from RS's N recommendations, they quit browsing.

4. Hybrid Recommendations for the Interactive Scenarios:

The creation of novel hybrid RSs based on user-recommender interactions are detailed in this section where 3 elements are considered: (1) new hybrid technique that compromises between interactive recalls and diversities; (2) modelling user-recommender interactions using new hybrid algorithm; and (3) proposing parameters for adjustments.

4.1 The Hybrid Recommendation Algorithm:

The proposed hybrid algorithm is a combination of random algorithms and kNNs. To overcome cold-start issues and maintain variety in RSs, random algorithms are used in this work. kNNs use cosine values to find comparable users where neighbours propose specific items. kNNs recommendations have a threshold value TR which determine on the usage of kNNs. The count of items browsed by users (KNN Switch) according to (4) is u_i is $|br(u_i)|$and given by:

$$\text{kNN switch} = \begin{cases} \text{true}, & if\ |br(u_i)| \geq TR; \\ \text{false}, & otherwise. \end{cases} \qquad (5)$$

RSs are centred around the proposed hybrid algorithm. Based on browse matrices data features, they can be divided into three steps in the initial stage namely (1) browse matrix is null; (2) browse items of logged user is null; and (3) items browsed by any user < threshold TR. Browse matrices exhibit two properties in their transitions:(1) they are not null, and (2) items browsed over TR > 0, but recommended items are not sufficient for kNNs to work. kNNs recommend few items, while the rest of the recommendations stem from the random algorithm.

When user viewed items cross TR values, kNN's can recommend sufficient items. The range of recommendations can be widened by the usage of random algorithm on specific of items.

Algorithm 1 with 4 steps is a hybrid incremental algorithm with user id (Uid) and three user-specified parameters as inputs namely total recommendations count (N), randomly recommended ratio (RT), and total recommended count (TRN).

- Step 1: There is no browse matrix history data in the initialization step and to overcome cold-start issues random algorithm is used. Lines 3-6 of the algorithm correspond to this stage.
- Step 2: TR values determine on the usage of kNNs. Initially $N_b$count of of browsed items by U_i is considered If$N_b \geq$ TR , kNNs are employed and correspondingly Lines 8-14 are executed.
- Step 3: If RSs are up and running random algorithm suggests fresh items for and correspondingly Lines 15-18 of the algorithm are executed.
- Step 4: To recommend some goods, the kNN algorithm is employed. In the transition and stable stages, the additional things are recommended based on a random algorithm. Lines 17-23 of the algorithm relate to this stage.

The algorithm's output is saved in Algorithm2's memory.

Algorithm 1:

Input $M_{id,}$ TR, RT, N; Output: Recommender sets; Method: Hybrid Algorithm:

Tb = browsed item set by Uid
bm=browser matrix of all users
if(bm = Θ or tb ≠ Θ) then
Tris the recommendation set by random algorithm
RETURN
end if
M={M1, M2, M3,…Mn}
Count=0
For each (Uiin U) do
Nb = Number of browsed items by Ui
If Nb>= TR THEN
Count = Count +1
End IF ; END FOR
If Count<=0 THEN
Tris the recommendation set by random algorithm
RETURN
END IF
Nk = N * (1-RT)
Tk = Recommended Nk items by kNN
Nrd = N - [Tk]
Trdis the recommendation set by random algorithm
Tr= Tk U Trd
Return Trd

Algorithm 2:

Input N, RT, TR:

Output: Recall, Diversity
Methods: Interactive Recommender

U = {u1, u2,.. un}
U1 = Array of U
T = {t1, t2, … tn}
Set Rf = TRUE
While (Rf) do Rf - FALSE
For each Uid U1 do

Tr= hybrid (N, RT, TR, Ui)
If (Ti∩Tr) ≠ 0 then
User quits browsing ELSE
Item recorded in Bm(Uid)
Rf = TRUE
END IF, FOR & WHILE
Calculate Recall and Diversity
Return {Recall, Diversity}

4.2 The User-Recommender Interaction:

This work's proposed RSs establish user-recommender interactions and can viewed as two parts namely users and recommenders. Users log in for browsing and recommenders do most of the work. User recommender interactions for users follow: (1) RSs recommend to users; (2) user's personal interest lists determine the browsing of items. When users click on items, process is continued else users quit as items suggested by RSs are not interesting. The proposed hybrid RSs interactions with users can be: (1) RSs accept user's requests and recommend items based on the hybrid algorithm. (2) RSs add user's selections to browse matrices when they brows recommended items else user's end their interactions with RSs (3) Users accesses to RSs are randomized in their orders.

Algorithm 2 describes the interactive recommender's procedure, which includes three user-specified parameters and three phases for users.

- Step 1: To get suggestions, call Algorithm 1. Line 8 of the algorithm represents this phase.
- Step 2: If the user is uninterested in the items suggested, the RS declines to make any more recommendations. Line 11 of the algorithm represents this phase.
- Step 3: If the user is interested in the suggested products, the RS saves the user's selection and changes the viewed matrix. Line 13 of the algorithm represents this phase.
- Due to the limited number of users, the user-recommender interaction will come to an end. After that, the interactive recall and diversity are evaluated. This step corresponds to Lines 18-19 of the algorithm.

4.3 A Running Example:

Table 2 shows an example of an interest matrix. There are eight movies and four users. In the beginning, the browse matrix is null. In the running example, N, RT, TR and K are assigned 3, 0.25, 1, and 1 respectively. Figure 3 shows a running example of hybrid recommendations of interactive scenarios. On the RSs, users log in random orders and assuming login sequences to be $\{u_1, u_2, u_3, \ldots, u_n\}$.

The interaction between u3 and the recommenders is depicted in Figure 3 where movies are recommended based on random algorithm as the user is the first login user with no neighbours. There are three rounds to the interactive process and assuming first round's recommended list is$\{m_5, m_6, m_2\}$. $\{m_6$ and $m_2\}$ are set the user's interests, but m5 is not. Users examinem6 as it is the first match. The browse matrix's corresponding element is set from 0 to 1.

The user's grey list now includes m5. For the second round{m7,m8,m4} list is recommended. User goes over m4 thus adding m7 and m8 to the grey list. For the third round, {m3, m1, m2} list is recommended. The user looks through m2, m3, and m1 gets added to the grey list. Users exit RSs as they are not given movie recommendations. Finally, users complete m2, m4, m6, and her grey list now includes {m1, m3, m5, m7, m8}.

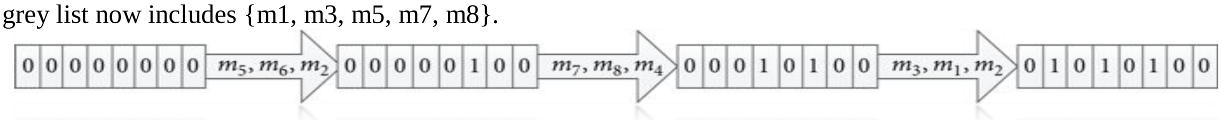

Figure 3: Running Example

The interaction between u2 and the recommender is depicted in Figure 4. There are three rounds to the interactive process. The random algorithm is used in the first round of recommendations. The first round's recommended list is {m8, m4, m1} . user looks through e m4 and adds m8 to grey list. The u3 becomes a neighbour because TR=1 and k=1. In the second round of recommendation, {m2, m6} are recommended using kNNs, while m7 is recommended using random algorithm {m2, m6, m7} = {m2, m6} U {m7} is the recommended set for the second round {m7, m2, m6} is the recommended list. Users look through m6 and adds {m7, m2} to grey list. The third round of recommendation is based on random algorithm because no movie can be recommended using kNNs {m1, m3, m5} is the recommended list. User drops out of RSs in the third round when recommended films are not appealing.

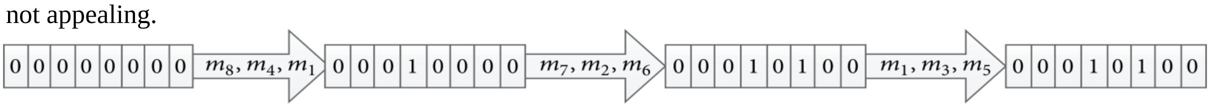

Figure 4

Figure 5 depicts the interaction between u1 and the recommender. The interactive process consists of six rounds. The first round of recommendation is based on random algorithm. The recommended list of the first round is {m3, m1, m7}. m2 is browsed by user and m3 is added to the grey list. Recommendations of 2ndround are based on kNNs and random algorithm where kNNs recommend N_k = [N *(1- RT)] = 2. Random recommended number isN_{rd} = N-N_k = 1 .U3 becomes a neighbor and recommends {m4.m6}. The two movies and the random recommended m7 constitute the list {m7, m4, m6}. m4 is browsed by user and m7 gets added to grey list. Recommendations of subsequent rounds are based on the random algorithm as neighbors u2 and u3 cannot recommend new movies.

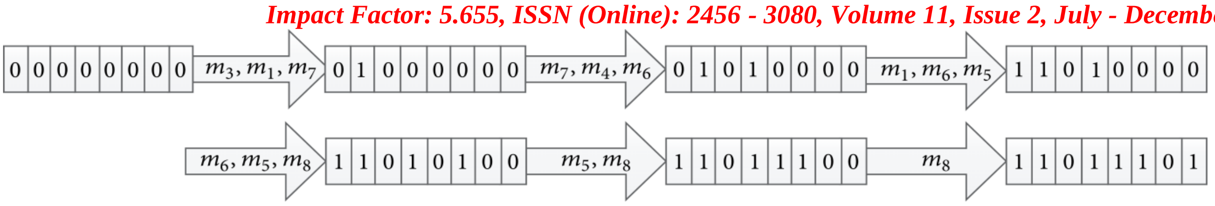

Figure 5

Figure 6 depicts the interaction between u4 and the recommender. The interactive process only includes one round because the recommended list {m4, m7, m6} is not in her interest list.

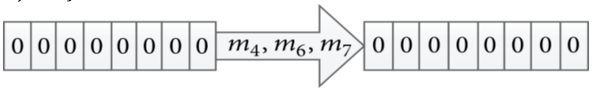

Figure 6

Figure 7 displays the browse matrix when there are no active users for RSs. If the user has a broad range of interests, it is more beneficial to recommend based on the end result. The interactive recollection is finally finished (U, T). Because there are 6 successfully recommended items, the interactive diversity is id (U, T). The user-recommender interaction creates an active learning scenario because the sequences of the users' login and choice affect subsequent recommendations.

MID/UID	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
M_1	1	1	0	1	1	1	0	1
M_2	0	0	0	1	0	1	0	0
M_3	0	1	0	1	0	1	0	0
M_4	0	0	0	0	0	0	0	0

Figure 7: The browse matrix

5. Experiments:

This study used two types of experimental schemes where first type consisted of experimental approaches for obtaining acceptable parameters for the proposed hybrid whereas the second type of experiments compared random and proposed hybrid algorithms. Because of the random process, each approach is repeated ten times with various recommended items. Through experimentation, we attempt to address the following questions.

- What proportion of random recommendations is appropriate?
- What is a reasonable NN recommendation threshold?
- What are the best kNN recommendations?
- How does it affect the algorithm's performance?
- Is the hybrid algorithm superior than the random algorithm?

5.1 Dataset:

The hypothesis was tested on Movie Lens dataset, extensively used in RSs. The following is the database schema:

- User (user ID, age, gender, and occupation),
- Movie (movie ID, year of release, and genre),
- Rates (user ID, movie ID).

The version used in this study has 943 users and 1,682 movies. We only look at whether a user has rated a movie, whereas the original rate relation includes movie ratings on five scales. Every user has seen at least one film, and the dataset contains approximately 100,000 film ratings but are incomplete as only 45% of the films were seen and only 20% of users have viewed more than 10% of films.

5.2 Experimental Design:

Movies watched by users are known in advance due to the initial dataset and are All users in this RSs have an interest matrix. All users' browsed matrix is initially empty. The browsed matrix is updated as new items are viewed during the incremental suggestion process. Finally, we use the browsed and interest matrix to compute interactive recall and diversity.

The performance of interactive recommendations is evaluated using recall and diversity criteria. The interactive recall will be the ratio of explored items to items that might be of interest to her.

The interactive recall of user Ui on a set of items T via RSs is defined as

$$\mathrm{ir}(u_i, T) = \mathrm{ir}(bm, um, u_i) = \frac{|\{t_j \in T \mid bm_{i,j} = 1\}|}{|\{t_j \in T \mid um_{i,j} = 1\}|} \tag{6}$$

$$= \frac{\sum_{k=1}^{m} bm_{i,j}}{\sum_{k=1}^{m} um_{i,k}} \tag{7}$$

Now we study the interactive recall of the recommender.

Definition 6:

The interactive recall of an RS is

$$\mathrm{ir}(U, T) = \mathrm{ir}(bm, um) = \frac{\sum_{i=1}^{n} \sum_{k=1}^{m} bm_{i,k}}{\sum_{i=1}^{n} um_{i,k}} \tag{8}$$

We can see that the interactive recall is linked to the user interest matrix in this example.

The recommender's interactive diversity is then investigated. In the user-recommender interaction, let BR(u,t) be the set of items browsed by user u. BR (U,T) = UuEU returns the set of all items successfully recommended to at least one user. BR(u,T) The interactive diversity is

$$Id(U,T) = \frac{BR\,|U,T|}{|T|} \quad (9)$$

A higher id (U,T) indicates more diverse recommendations because more items are recommended and browsed. The main goal is to increase interactive recalls and diversities, which means that users should look at as many items as they can, and recommenders should find as many user interests as they can. Two sets of experiments were designed where the first set of tests aims to find the best settings for the RT, TR, and K to hybrid algorithms. The second set of experiments pits random and hybrid algorithms against each other in terms of interactive recall and diversity.

In the first kind of experiments, we assign N=20. Let RT €{0,0.05,… 1} , TR €{1,2,3,4,… 20,40,60,80} and k €{1,2,3,4,… 80} . Six experiments are undertaken to answer the questions raised at the beginning of the section one by one. A parameter is allowed to make a change, and the other parameters are set to a fixed value in each experiment. Each experiment is repeated 10 times, and the average recall and diversity are computed.

5.3 Results:

Figure 7, the assigned values are TR=3, k-45, and modify the RT. When RT is changed from 0 to 0.4, the recall remains essentially constant. When RT exceeds 0.4, recall begins to decline significantly. Figure 8's parameter settings are the same as Figure 9's. According to the overall trend, as the ratio of random recommendations increases, so does the diversity. In Figure 10, we assign RT=0.25, k=45, and modify TR. With an increase in TR, the overall recall trend is decreasing. When the TR is increased from 1 to 20, the recall rapidly declines. When TR>=20, the recall gradually decreases..

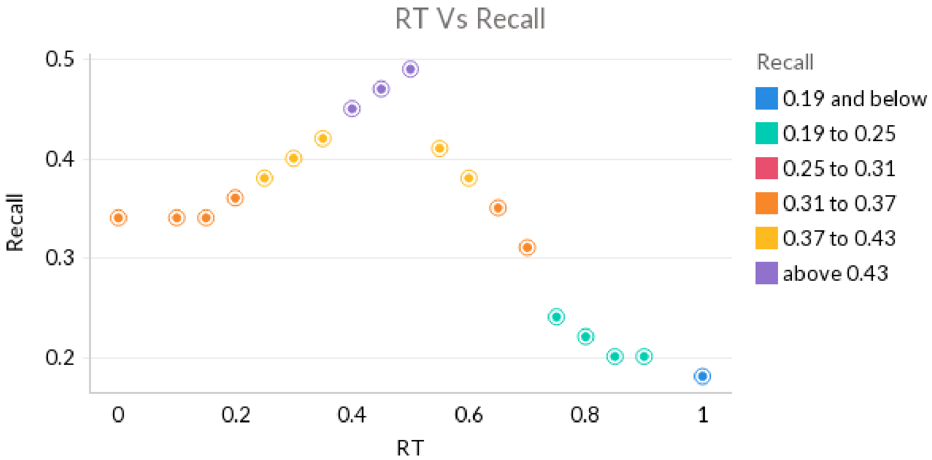

Figure 7: RT vs Recall

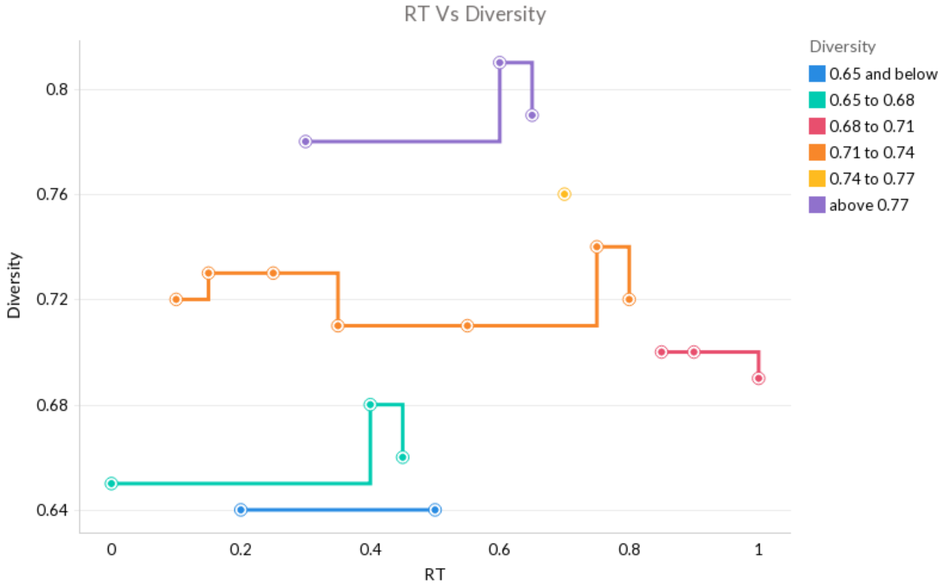

Figure 8: RT Vs Diversity

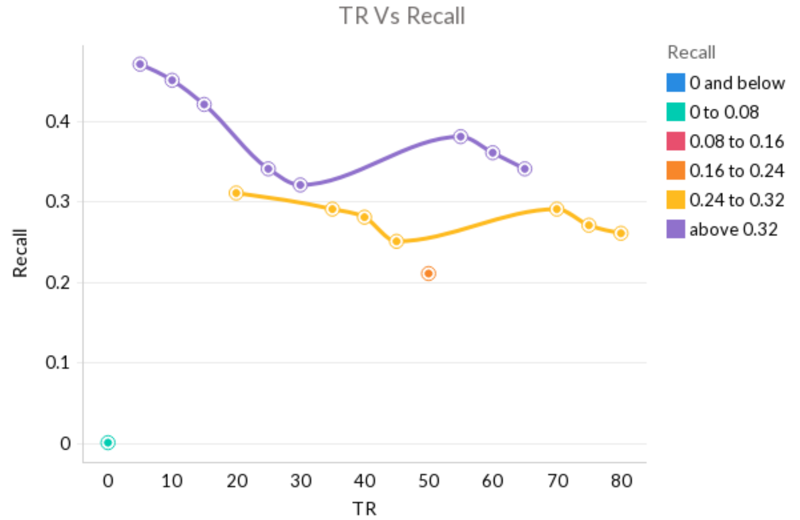

Figure 9: TR Vs Recall

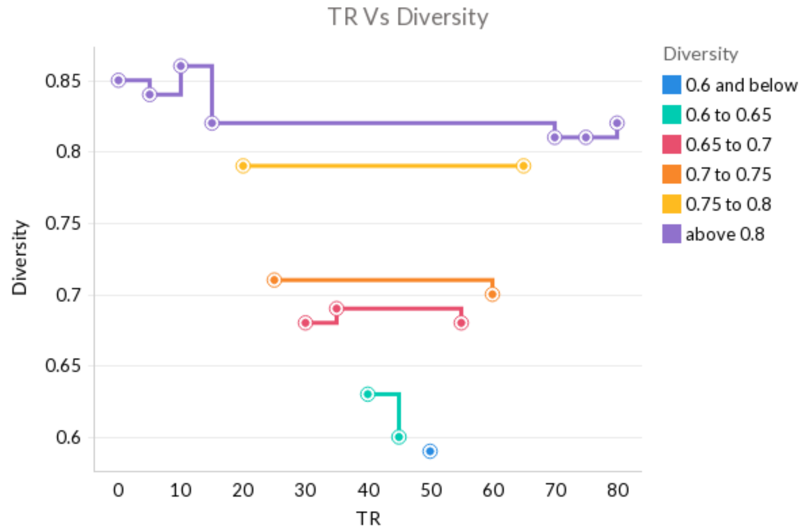

Figure 10: TR Vs Diversity

In the second kind of experiments, we assign RT=0.25 TR=3 and k=45 and make N and revisit change. We assign N €1,3,5...50 in Figure 11. With an increase in total number, the overall trend of recall is basically linear. The hybrid algorithm has a higher recall than the random algorithm. Figure 12 has the same parameter settings as Figure 11. When N is increased from 1 to 25, the overall trend of diversity increases in a linear fashion as the total number increases. When N is increased from 25 to 50, the overall trend of diversity remains constant. The hybrid algorithm's diversity is nearly identical to that of the random algorithm. We assign revisit €0,1,2,3...20 in Figure 13. The overall trend of recall is essentially increasing as the number of revisits increases. Figure 14 has the same parameter settings as Figure 13. The overall trend of diversity is essentially increasing with the increase in the number of revisits. The recall of the hybrid algorithm is higher than that of the random algorithm, but the diversity is lower.

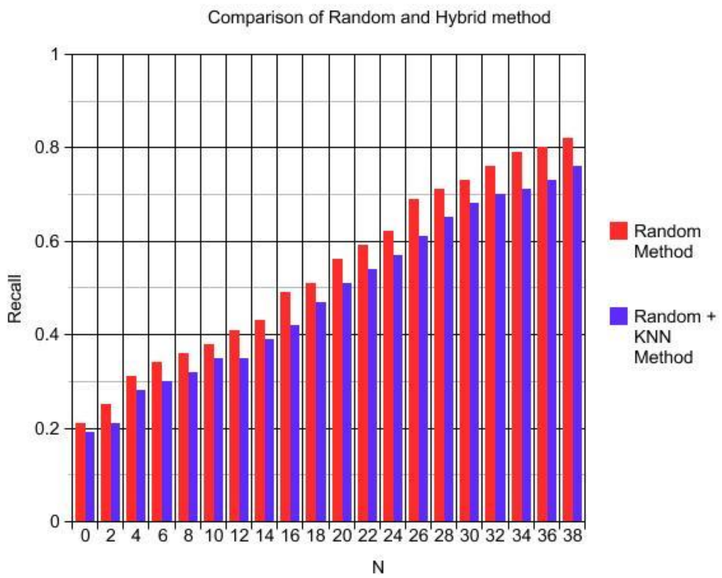

Figure 11: Random vs Hybrid (recall)

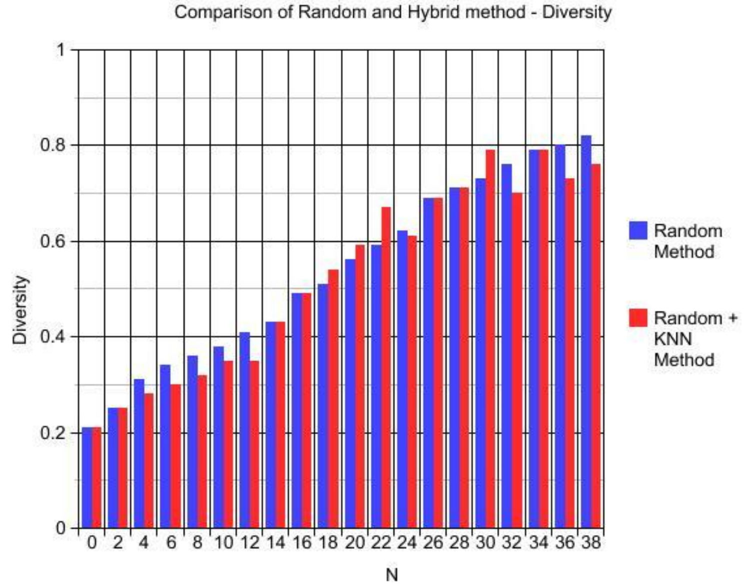

Figure 12: Random Vs Hybrid (Diversity)

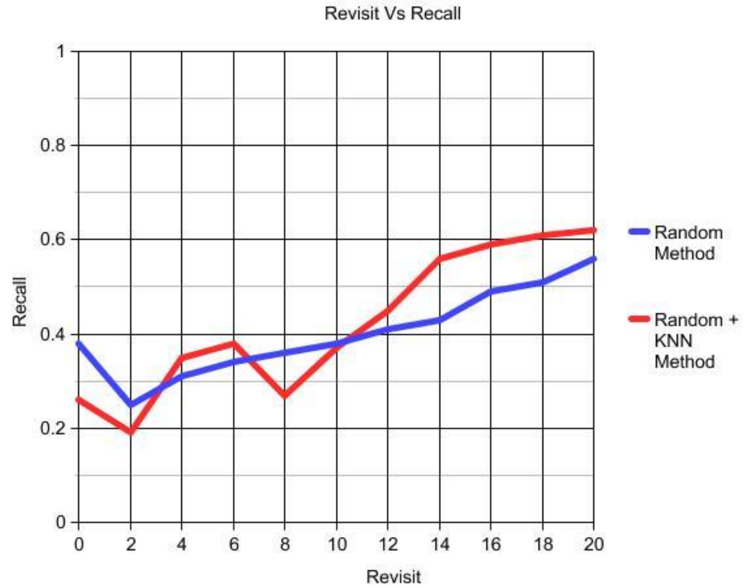

Figure 13: Revisit vs Recall

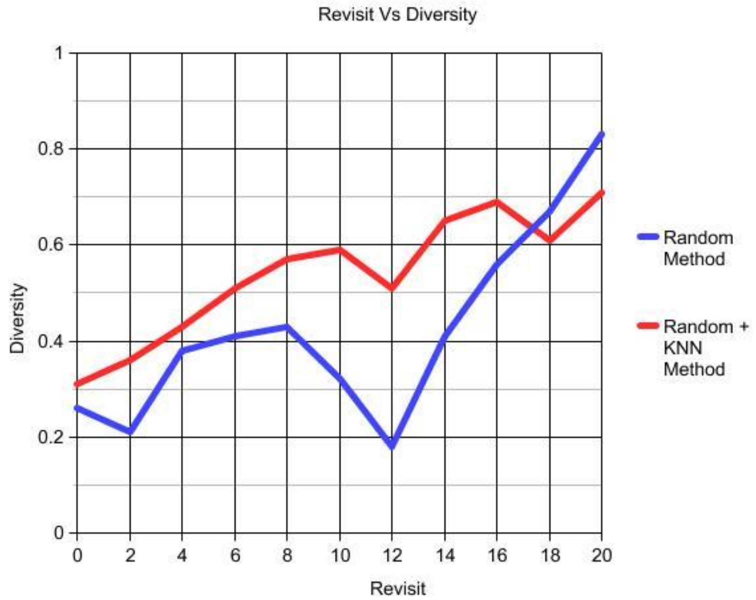

Figure 14: Revisit vs Diversity

6. Conclusion:

For the interactive scenario in this study, this paper proposes a hybrid recommender system. By Modifying, the following results were obtained in comparisons of random and hybrid algorithms and using the recommended parameters: (1) As long as the ratio of random recommendations isn't too high (e.g., less than 0.25), (2) kNN should be used as soon as possible, (3) the number of neighbours should be large enough (e.g., 45), (4) the recall increases nearly linearly with the number of recommendations in each round, and (5) the hybrid algorithm outperforms the random algorithm.

7. References:

1. L. Duan, W. N. Street, and E. Xu, “Healthcare information systems: data mining methods in the creation of a clinical recommender system,” Enterprise Information Systems, vol. 5, no. 2, pp. 169-181, 2011.
2. F. Min and W. Zhu, “Mining top-k granular association rules for recommendation,” in Proceedings of the IEEE IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS '13), pp. 1372-1376, 2013.
3. X. He, F. Min, and W. Zhu, “A comparative study of discretization approaches for granular association rule mining,” in Proceedings of the 26th IEEE Canadian Conference on Electrical and Computer Engineering, pp. 725-729, May 2013.
4. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” Knowledge-Based Systems, vol. 46, pp. 109-132, 2013.
5. J. Delgado and N. Ishii, “Memory-based weighted majority prediction,” in Proceedings of the SIGIR Workshop Recommender Systems, Citeseer, 1999.
6. H.-R. Zhang, F. Min, and X. He, “Aggregated recommendation through random forests,” The Scientific World Journal, vol. 2014, Article ID 649596, 11 pages, 2014.
7. H. R. Zhang, F. Min, and S. S. Wang, “A random forest approach to model-based recommendation,” Journal of Information and Computational Science, vol. 11, no. 15, pp. 5341-5348, 2014.
8. D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, “Collaborative filtering by personality diagnosis: a hybrid memory-and model-based approach,” in Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, pp. 473-480, 2000.
9. R. Burke, “Hybrid recommender systems: survey and experiments,” User Modelling and User-Adapted Interaction, vol. 12, no. 4, pp. 331-370, 2002.
10. P. Cremonesi, R. Turrin, and F. Airoldi, “Hybrid algorithms for recommending new items,” in Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 33-40, 2011.
11. T. Tran and R. Cohen, “Hybrid recommender systems for electronic commerce,” in Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04, pp. 78-83, AAAI Press, 2000.
12. V. Moscato, A. Picariello, and A. M. Rinaldi, “Towards a user based recommendation strategy for digital ecosystems,” Knowledge-Based Systems, vol. 37, pp. 165-175, 2013.
13. S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, “Ontological user profiling in recommender systems,” ACM Transactions on Information Systems, vol. 22, no. 1, pp. 54-88, 2004.
14. G. Adomavicius and Y. Kwon, “New recommendation techniques for multicriteria rating systems,” IEEE Intelligent Systems, vol. 22, no. 3, pp. 48-55, 2007.
15. T. T. Nguyen, D. Kluver, T.-Y. Wang et al., “Rating support interfaces to improve user experience and recommender accuracy,” in Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13), pp. 149-156, ACM, October 2013.
16. X. Luo, Y. Xia, and Q. Zhu, “Incremental collaborative filtering recommender based on regularized matrix factorization,” Knowledge-Based Systems, vol. 27, pp. 271-280, 2012.
17. Vu Duch , Martin (2021) A Hybrid Model for Recommendation Systems. In: Pan JS., Li J., Ryu K.H., Meng Z., Klasnja-Milicevic A. (eds) Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, vol 212. Springer, Singapore. https://doi.org/10.1007/978-981-33-6757-9_40
18. M. S. Geetha Devasena, R. Kingsy Grace and G. Gopu, "PDD: Predictive Diabetes Diagnosis using Datamining Algorithms," 2019 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1-4, doi: 10.1109/ICCCI48352.2020.9104108.
19. K. A. Fararni, F. Nafis, B. Aghoutane, A. Yahyaouy, J. Riffi and A. Sabri, "Hybrid recommender system for tourism based on big data and AI: A conceptual framework," in Big Data Mining and Analytics, vol. 4, no. 1, pp. 47-55, March 2021, doi: 10.26599/BDMA.2020.9020015.
20. C. Y. Lin, L. -C. Wang and K. -H. Tsai, "Hybrid Real-Time Matrix Factorization for Implicit Feedback Recommendation Systems," in IEEE Access, vol. 6, pp. 21369-21380, 2020, doi: 10.1109/ACCESS.2018.2819428.
21. Ji, Yang, Wang, et al. BRS cS: a hybrid recommendation model fusing multi-source heterogeneous data. J Wireless Com Network 2020, 124 (2020). https://doi.org/10.1186/s13638-020-01716-2
22. Zhang, Heng-Ru, 2019 " Aggregated Recommendation through Random Forests", The Scientific World Journal, Hindawi Publishing Corporation, https://doi.org/10.1155/2014/649596
23. S. M. McNee, J. Riedl, and J. A. Konstan, “Making recommendations better: an analytic model for human-recommender interaction,” in Proceedings of the Extended Abstracts on Human Factors in Computing (CHI '06), pp. 1103-1108, ACM, April 2006.
24. B. Settles, Active Learning Literature Survey, vol. 52, University of Wisconsin, Madison, Wis, USA, 2010.
25. K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, “Probabilistic Memory-Based Collaborative Filtering,” IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 1, pp. 56-69, 2004.
26. N. Rubens, D. Kaplan, and M. Sugiyama, “Active learning in recommender systems,” in Recommender Systems Handbook, pp. 735-767, Springer, 2011.

27. X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009.
28. J. Bobadilla, A. Hernando, F. Ortega, and J. Bernal, "A framework for collaborative filtering recommender systems," Expert Systems with Applications, vol. 38, no. 12, pp. 14609-14623, 2011.
29. B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th International Conference on World Wide Web, pp. 285-295, May 2001.
30. C. H. Lee, Y. H. Kim, and P. K. Rhee, "Web personalization expert with combining collaborative filtering and association rule mining technique," Expert Systems with Applications, vol. 21, no. 3, pp. 131-137, 2001.
31. J. Gemmell, T. Schimoler, M. Ramezani, and B. Mobasher, "Adapting k-nearest neighbor for tag recommendation in folksonomies," in Proceedings of the 7th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP '09), p. 528, 2009.
32. P. B. Kantor, L. Rokach, F. Ricci, and B. Shapira, Recommender Systems Handbook, Springer, 2011.