F O U N D A T I O N S O F C O M P U T I N G A N D D E C I S I O N S C I E N C E S
Vol. 46 (2021) No. 4
ISSN 0867-6356
DOI: 10.2478/fcds-2021-0023 e-ISSN 2300-3405

Assessment Methods for Evaluation of Recommender Systems: A Survey

Madhusree Kuanr , Puspanjali Mohapatra *

Abstract. The recommender system (RS) filters out important information from a large pool of dynamically generated information to set some important decisions in terms of some recommendations according to the user's past behavior, preferences, and interests. A recommender system is the subclass of information filtering systems that can anticipate the needs of the user before the needs are recognized by the user in the near future. But an evaluation of the recommender system is an important factor as it involves the trust of the user in the system. Various incompatible assessment methods are used for the evaluation of recommender systems, but the proper evaluation of a recommender system needs a particular objective set by the recommender system. This paper surveys and organizes the concepts and definitions of various metrics to assess recommender systems. Also, this survey tries to find out the relationship between the assessment methods and their categorization by type.

Keywords: Recommender System, Information Filtering System, Assessment Methods

1. Introduction

With the advent of the Internet and the age of e-commerce, businesses are opting for Recommender Systems (RSs) to raise sales. RSs have proved valuable for both service providers and users. RSs address the problem of the paradox of choice. The paradox of choice means the inability of making an effective choice from many available options. RSs also help companies to increase their sales by advertisement and promoting new products through the recommendation process. RS can be formally defined as: Let U denote the set of all users and let I denote the set of all items that can be used for recommendation. Let f be a utility function that is used to measure

*Department of Computer Science and Engineering, IIIT Bhubaneswar, India, {madhu.kuanr@gmail.com, puspanjali@iiit-bh.ac.in}

the utility of item i to user u i.e $f : U \times I \to R$, where R is an ordered set. Then for each user $u \in U$, the items $i' \in I$ will be recommended which maximize the user's utility. Hence more formally, it can be defined as $\forall u \in U, {i_u}' = argmax_{i \in I} f(u, i)$.

RSs can also be categorized into personalized RSs and non-personalized RSs. A personalized RS provides recommendations according to the behavior and interests of the user in the past. But a non-personalized RS provides blind recommendations to the users without considering their preferences and interests. RS can be used either for prediction or production of a top-n recommendation list where n is a positive integer. The schematic diagram of an RS is shown in Figure 1. RS collects the data

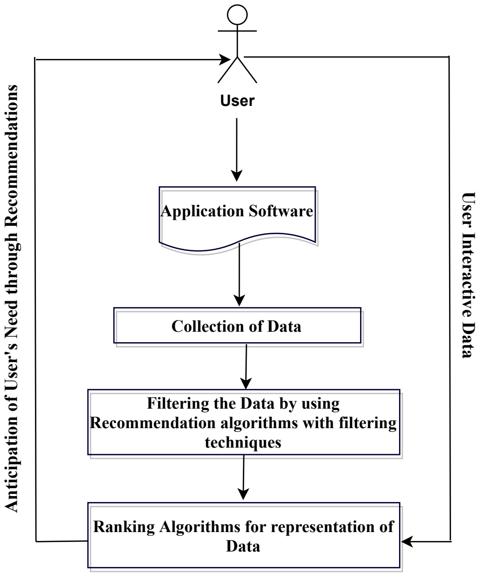

Figure 1. Framework of Recommender System

from users through the application software. It then filters the data using a recommendation algorithm and reverts some recommendations to the users by ranking the filtered data.

The rest of the paper is arranged according to the following ways. The next section contains a survey of various types of RS. Section 3 describes the different types of assessment methods for the proper evaluation of an RS. Section 4 outlines the relationships present between the assessment methods and the paper is finally concluded in section 5.

2. Recommender Types

Depending upon several factors, RS can be categorized into various types. RS has been categorized into many forms by various researchers in the past. Adomavicius

& Tuzhilin [2] have categorized RSs according to the approaches used, areas of application, and data mining techniques used in them. In this current survey, the RS has been categorized into 16 types according to its popularity of use in the different domains and applications which are discussed in the following subsections.

2.1. Collaborative RS

Collaborative recommender systems predict the rating for the querying user automatically by collecting information from collaborative users with similar preferences or interests (Adomavicius & Tuzhilin [2]). The major types of Collaborative RS are Memory-based and Model-based. Memory-based collaborative RS can also be categorized as user-based and item-based collaborative RS.

The memory-based approach uses user rating data to measure a person or item similarity. Li & Li [52] have proposed a user-based approach for collaborative filtering using the aggregation function to calculate the rating that the querying user will give for an item i by aggregating the rating values of similar users as shown in equation 1.

$$r_{u,i} = aggr_{u' \in U} r_{u',i} \tag{1}$$

Where U denotes the set of users similar to the active query user u who rated item i. Pearson correlation and vector cosine-based approaches are the different measures for similarity computation. The similarity between two users x and y according to the Pearson Correlation is defined as shown in equation 2.

$$sim(x,y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)^2} \sqrt{\sum_{i \in I_{xy}} (r_{y,i} - \bar{r}_y)^2}} \tag{2}$$

Where I_{xy} is the collection of items rated by both of the users x and y. The approach based on the cosine measure determines the cosine-similarity between the user x and y as seen in equation 3.

$$sim(x,y) = cos(\vec{x}, \vec{y}) = \frac{\vec{x}.\vec{y}}{\|\vec{x}\| \times \|\vec{y}\|} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}} \tag{3}$$

Models are constructed in model-based methodology using specific data mining or machine learning algorithms to predict the users' ratings for unrated items. Different model-based Collaborative Filtering algorithms are Bayesian networks, clustering models, latent semantic models like singular value decomposition, probabilistic latent semantic analysis, etc which provide recommendations by developing models of users' ratings. The Bayesian network approach develops a probabilistic model for the recommendation in collaborative filtering. The clustering approach considers it as a classification problem whereas latent semantic models try to capture the preferences or interests of a user by a variety of hidden factors.

2.2. Content-based RS

A content-based RS aims to provide recommendations according to the profile of the user. These approaches are best suited to situations where knowledge about an item is known but not about the querying user. According to Pal *et al.* [62], RS based on content treats recommendations as a user-specific classification issue and learns a classifier based on features of an item for the user's likes and dislikes. It has used similarity between items in terms of contents to provide practical recommendations.

2.3. Hybrid RS

Hybrid RS combines both Collaborative and Content-based approaches in different ways to take their complementary advantages. Hybridization can be achieved either by performing collaborative and content-based predictions individually and then combining them or by providing the solution of a collaborative approach into a content-based solution and vice-versa or by converging the approaches into one model (Kuanr & Mohapatra [50]). Several studies have proven that hybrid RS provides better recommendations compared to pure approaches. Thuan & Puntheeranurak [80] have proposed a novel hybrid RS using review helpfulness features and it is found to outperform the traditional collaborative and content-based RS in terms of accuracy.

2.4. Knowledge-based RS

A knowledge-driven RS makes suggestions based on a particular user request but not the user's rating background. Such systems are applied in cases where traditional approaches such as collaborative filtering and content-based filtering are impossible to apply. Samin & Azim [70] have proposed a Knowledge-based RS to recommend course teaching, research supervision, and industry-academia collaboration to users in academia using a probabilistic topic model. The results have been shown on real-world data and found to outperform the traditional RSs.

2.5. Demographic Filtering based RS

Demographic filtering-based RS uses a similarity measure as a metric. It tries to find similarities between user demographic information such as age, gender, occupation, etc. It stores the demographic information of users and produces a set of recommendations for a new active user according to his/her demographic information. Gupta & Gadge [36] have proposed an RS by combining item-based collaborative filtering with demographic filtering to address the cold start problem of the RS. Results present in that article have also revealed that it has outperformed the existing traditional collaborative filtering-based RSs.

2.6. Trust-based RS

Trust-based recommendations (TRS) are an improvement of traditional Collaborative recommendation approaches to increase the accuracy of the recommendation results. The rationale behind TRS is the use of graphs to represent the interaction between users and items based on their connection to the particular attributes. TRS is widely used in social networks to link a large number of users to the network. The trust between the users u and v is generally represented through trust-based weighted mean which represents a weighted value as presented in equation 4.

$$r(u, i) = \frac{\sum_{v}^{v \in R^T} t_{u,v} r_{v,i}}{\sum_{v}^{v \in R^T} t_{u,v}} \quad (4)$$

Tyagi & Bharadwaj [81] have proposed a trust-based RS using case-based reasoning with collaborative filtering and helped users in decision making that which viewpoints of other people they should trust more.

2.7. Context-Aware RS

Context-Aware RS (CARS) applies user context sensing and analysis to provide personalized services. Context Information in RS includes the use of data that characterizes a person to use it, where possible, as contextual knowledge for computing recommendations. Thiprak & Kurutach [79] have proposed a context-aware RS to recommend different kinds and types of Thai herbs to the users. It has used association rule mining to find out the relationships between the attributes present in the dataset.

2.8. Social network-based RS

Social network-based RS uses social network information, including preferences of the user, general acceptance of items, and social friends' influences. It is developed to make personalized recommendations from such information. Wongkhamchan *et al.* [85] have proposed an RS using collaborative filtering based on social networks to recommend top-n points of interest to the user. It has used collaborative filtering using rating values to find out the similar users and used user's check-in information from social networks to provide recommendations.

2.9. Soft computing Techniques-based RS

Soft computing technique-based RSs can handle the uncertainties present in the various models like business marketing activities. These RSs are usually designed based on the concepts like particle swarming, neural networks, genetics, evolution, etc.

2.10. Reclusive methods-based RS

Collaborative filtering is focused on finding user similarities. Unlike collaborative RS, the reclusive approach exploits the characteristics of objects and requires their representation. Hence this method is assumed as the complementary method of collaborative RS. Reclusive method-based RS finds the similarities between objects rather than users.

2.11. Rating-based RS

The RS based on rating uses empirical data (i.e. ratings) to forecast the taste of the user. Rating-based RS is generally incorporated with a model-based collaborative filtering RS. Ajesh *et al.* [5] have proposed a rating-based RS to provide recommendations considering ratings of the users, current trends, and the mindset of the active querying user for whom it will generate a recommendation.

2.12. Feature-based RS

The rating-based RSs use historical data (i.e. ratings) to predict user preferences. Nonetheless, one distinctive feature of the feature-based systems is that users are required to express their expectations about specific features of items directly. These RSs are particularly useful for recommending items that are rarely consumed during a user's lifetime or those that require a substantial monetary engagement like notebooks, vehicles, or digital cameras. Hayashi *et al.* [39], Spiekermann & Paraschiv [76] have proposed RSs using feature-based filtering to select meaningful items for providing the recommendations.

2.13. Case-based RS

The case-based recommender systems are derived from case-based reasoning (CBR) approaches (Brusilovsky *et al.* [14]). CBR systems solve new issues by using the case-base of prior problem-solving interactions rather than codified rules and basic domain constructs. It also finds a similar case and modifies the approach to the target situation (Bridge *et al.* [13]).

2.14. Utility-based RS

Utility-based RS is based on a calculation of the usefulness of every item for a user. Users may explicitly or indirectly indicate their preferences for a set of attributes that characterize multiple product-type attributes. The benefit of utility-based RS is that non-product property like product availability can also be added to utility computation. Zielinski [95] has proposed a case-based RS using a knowledge model. It has considered the current state and pedagogical strategies of the learner to recommend learning objectives according to the preferences of the learner.

2.15. Critiquing-based RS

Critiquing-based RS is an extended version of case-based and utility-based RS with critiquing as an additional component. It produces recommendations based on the needs of the user and the user feedbacks as critiques in each recommendation process. It tries to increase the accuracy of an RS by predicting the user's needs in the next recommendation cycle. Giustozzi *et al.* [30] have proposed a critiquing-based RS that collects the preferences of the user through interaction and critiques and recommends appropriate learning resources to the user.

2.16. Personality-based RS

Personality-based RS may offer more personalized services as it better understands the user from a psychological point of view. Unlike rating-based systems, personality-based RS provides both implicit and explicit means of evaluating the attributes of a user. Implicit methods infer user identities mainly by examining user behavior while executing a given task whereas explicit methods focus on personality questionnaires. Hariadi & Nurjanah [38] have proposed a book RS considering the personality of the user and attributes of the books. It has considered personality factors like preferences and interests to find out the most similar users and has used attribute factors to provide a recommendation.

3. Assessment Methods for Evaluation of a RS

The capability of the RS can be quantified through various assessment methods. RS should be properly evaluated before application, as they bring users' satisfaction to the system. The job of the RS is not only to suggest specific items to the user but also to be so accurate that the user should consume some of the items from the recommendation list. But users' interactions and consumption experiences are a big deal for the RS. Researchers have also sought various assessment methods for assessing RS instead of just using statistical performance and machine learning techniques (Herlocker *et al.* [41]). Suggestions from an RS should be measured by the value that the user can produce. The various concepts for the assessment of an RS are discussed in the following subsections.

3.1. Prediction Evaluation

The majority of RSs are based on the technique of prediction that predicts a user's opinion about items or the probability of usage. Prediction evaluation is one of the important evaluation categories which judges the prediction capability of an RS to gain the trust of users. An RS is assumed more efficient if it has a more accurate prediction capability. Prediction evaluation is independent of the user interface and therefore suitable for offline experiments. The various assessment methods for the prediction evaluation of an RS are discussed in the following subsections.

3.1.1. Accuracy

The accuracy assessment method determines the consistency in similarity to the facts or the real value the system produces. It is one of the useful tools for the assessment of RSs and it can be defined as shown in equation 5.

$$Accuracy = \frac{NumberofsuccessfulRecommendations}{TotalNumberofRecommendations} \tag{5}$$

Let *R(u,i)* represents the user u's predicted rating for the item i and *r(u,i)* represents the user u's actual rating for the item i. Then accuracy can be reformulated as shown in the following equation 6.

$$Accuracy = \frac{\sum_{(\forall u,i/rs(u,i)=1)} 1 - |r(u,i) - R(u,i)|}{M} \tag{6}$$

Where r and R are two binary functions and M is the number of recommended items shown to the user. Here, *rs(u,i)* represents the recommendation of the item i to the user u. *rs(u,i)* is 1 if the item i is recommended to the user u and 0 if the item is not recommended. Armstrong *et al.* [7] have proposed an RS to provide recommendations for appropriate hyperlinks to a particular search in the world wide web where it has used accuracy to evaluate the system. Rogers *et al.* [67] have used accuracy as an evaluation metric for evaluating the route prediction in a route advisor system. Billsus & Pazzani [10], Burke [15], Pazzani *et al.* [61] have used the accuracy as an assessment method for the evaluation of their proposed system.

3.1.2. Mean Absolute Error (MAE)

MAE is one of the assessment methods popularly used to evaluate RSs. It estimates the difference between the rating predicted by the RS and the rating given by the user. The MAE can be calculated by using the following equation 7.

$$MAE = \frac{\sum_{(\forall u,i)} |r(u,i) - R(u,i)|}{N} \tag{7}$$

Where N represents the number of observations that depends on the number of items that the user correctly rated. Several recommender systems like Breese *et al.* [12], Herlocker *et al.* [40], Shardanand & Maes [74] have used MAE for their assessment purposes.

3.1.3. Root of Mean Square Error (RMSE)

RMSE calculates a larger difference in rating prediction for larger errors (Ricci *et al.* [66]). It is one of the commonly used tools for prediction evaluation. It can be calculated by using the formulae as shown in equation 8.

$$RMSE = \sqrt{\frac{\sum_{i \in rs_u} (r(i) - R(i))^2}{N}} \tag{8}$$

Where N represents the number of recommendations to the user. The various RSs (Goldberg *et al.* [31], Ma *et al.* [55]) in the past have used this measure for their assessments.

3.1.4. Coverage

Coverage is usually used for evaluating the whole RS but not for the evaluation of the recommendation list. It can be categorized into three types as item space coverage, user space coverage, and genre space coverage (Ricci *et al.* [66], Vergas *et al.* [83]).

Item Space Coverage applies to the size of the things an RS forecasts. An RS with less coverage of items limits user recommendations (Silveira *et al.* [75]). It prevents the user from discovering valuable device items which in effect affects the user's overall satisfaction with the system. The item space coverage for prediction evaluation is defined in equation 9.

$$Coverage = \frac{|I_p|}{I} \tag{9}$$

Where Ip represents the predicted list of items and I represents the size of the set of items. Ricci *et al.* [66] have used the item space coverage for assessment purpose.

User space coverage deals with the proportion of users that an RS can predict items to those users. User space coverage would measure the rate of users who receive effective recommendations. No particular metric is available for user space coverage (Silveira *et al.* [75]).

Genre space coverage is characterized as the count of different varieties of items suggested to the user. Hence genres space coverage is more related to diversity recommendations rather than prediction evaluation (Hurley [45]). No particular metric is also available for genre space coverage in the literature.

Table 1. Confusion Matrix for retrieval of documents with two classes

	Relevant	Non-relevant
Retrieved	RR	RN
Non-retrieved	NR	NN

3.2. Evaluation for recommendation as sets

Sometimes, RS produces recommendations in terms of a set of items according to the preferences of the user. The items present in the list need to be evaluated for relevancy to check whether they meet the preferences of the user or not. The various assessment methods that are used for the evaluation of this set of items present in the recommendation list are discussed in the following subsections.

3.2.1. Precision, Recall and F-measure

Precision and recall are proved as the two important metrics for RS from the domain of Information Retrieval (IR) (Baeza-Yates & Ribeiro-Neto [9], Fisher *et al.* [28]). These two assessment methods are computed with the help of confusion metrics generated by RS as shown in Table 1. The non-binary rating values $r(u, i)$ can be converted into binary values by taking into account some unique threshold value to get the confusion matrix (Baeza-Yates & Ribeiro-Neto [9], Fisher *et al.* [28]).

Precision can be defined as the capacity of RS for showing only useful items to the user and tries to restrict the useless items not to be present in the recommendation list. Hence precision can be defined by the following equation 10.

$$Precision = \frac{RR}{RR + RN} \tag{10}$$

The recall is defined as the ability to get all the useful items in the recommendation list. Equation 11 represents the formula to obtain recall.

$$Recall = \frac{RR}{RR + NR} \tag{11}$$

With greater recall, the chances of removing useful items get increased and also increases the chances of removing all useless items. However, with greater precision, the chances of removing useful items get decreased and also decreases the chances of removing all useless items. Due to this property, both precision and recall are having opposing behaviors as shown by the point dotted line in Figure 2. One parameter can be raised at the expense of decreasing the other parameter, and so they are not addressed in isolation. Either the value of one metric is compared for a fixed level of the other one or both are combined into one metric. But a good RS always tries to optimize both the parameters as shown in the dashed-dotted line in Figure 2. The

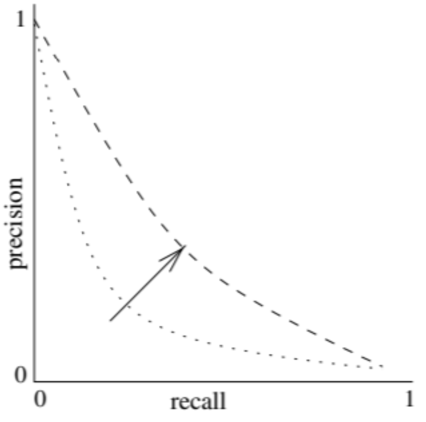

Figure 2. Behaviour between Precision and Recall

metric which is derived from the combination of these two metrics is F-measure which defines the system's overall utility as seen in equation 12.

$$F - measure(\beta) = \frac{Precision \times Recall}{(1-\beta) \times Precision + \beta \times Recall} \tag{12}$$

The most usual version of F-measure is *F1* by considering *beta=1/2* in equation 12. It is therefore also known as the harmonic mean of precision and recall as seen in equation 13.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

3.2.2. ROC Curve

ROC Curve is used as a measuring parameter of accuracy for the RSs. It is used in the recommendation process to measure the rate at which the user likes the items. It highlights items that were suggested but disliked by the user. It denotes recall against fallout where fallout is defined by equation 14. Optimizing the ROC curve is similar to the precision and recall optimization as shown in Figure 3 (Davis & Goadrich [25], Fawcett [27]).

$$Fallout = \frac{RN}{RN + NN} \tag{14}$$

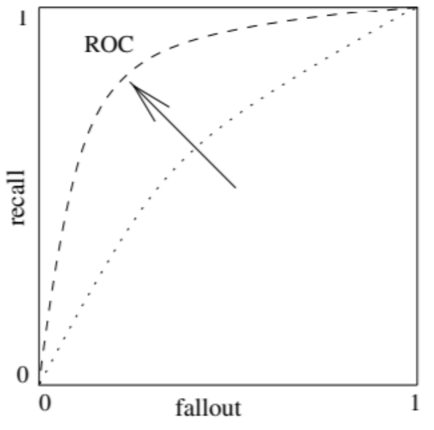

Figure 3. Behaviour between Fallout and Recall

3.3. Evaluation for recommendation as a ranked list

In many applications like Netflix, a user gets a list of recommendations that contains some items that the system predicts the user to like. The system tries to represent a ranked list with the correct ordering of items according to the choices of the user. The various assessment methods for the evaluation of these ranked lists are discussed in the following subsections.

3.3.1. Half-life

The half-life value is defined as the difference between the rating of the user for an item and that item's default rating. It is useful when the user is having a long recommendation list and the user consumes only a few items from the top of the list. The likelihood that the user finds a specific item on the ordered list is estimated using a function of exponential decay, parametrized by a parameter for half-life decay. It displays the rank of the item in the list, and there is a 50 percent chance that the user can see the item. It also influences the exponential decrease in the ranked list value of the positions. Considering $r(c, l_i)$ be the rating rated by the user u to the ith item of the recommendation list l and α as the half-life parameter, the expected utility of the item l_i is defined in equation 15. A half-life decaying model with matrix factorization for recommender system is proposed in Ardagelou & Arampatzis [6].

$$R_c = \sum_i \frac{max(u(c, l_i) - d, 0)}{2^{\frac{i-1}{\alpha-1}}} \tag{15}$$

3.3.2. Discounted cumulative gain (DCG)

DCG deals with the rank of highly relevant items so that they will not be ignored by the user while consuming (Jarvelin & Kekalainen [46]). DCG can be calculated by using the formulae as shown in equation 16 for a recommendation list with length l.

$$DCG(b) = \sum_{n=1}^{b} r_n + \sum_{n=b+1}^{l} \frac{r_n}{\log_b n} \tag{16}$$

Where r_n indicates the relevance of the nth ranked object and b is the persistence parameter which is usually taken as 2. The value of r_n is usually 1 for a relevant item and 0 for a non-relevant item.

3.4. Diversity Recommendation

Diversity is considered as the inverse meaning of similarity. Some situations in RS demand items to be diverse from each other to give the user the flexibility to choose a different kinds of items according to his/her preferences or interests. The various assessment methods used to check the diversity of the items present in a recommendation list are discussed in the following subsections.

3.4.1. Diversity

Diversity is a complementary approach to the similarity measure. The recommendation list generated by the RS should include the user's whole set of interests and there should be a wide variety in the recommendation list (Kunaver & Pozrl [51]). All the researchers in the past have argued that diversity presents a broad variety of items in a recommendation list. Kunaver & Pozrl [51] have proposed a intra-list similarity concept for the calculation of diversity in which the function $d(i,j)$ is used to calculate the distance between the two elements i and j in the recommendation list l_u as shown in the equation 17.

$$Diversity(l_u) = \sum_{i \in l_u} \sum_{j \in l_u i \neq j} d(i,j) \tag{17}$$

As stated by Zhang et al. [90], cosine similarity can also be used as a distance function to calculate diversity. The distance function therefore in equation 17 can be replaced by a cosine-similarity function as shown in equation 18.

$$Diversity(l_u) = \sum_{i \in l_u} \sum_{j \in l_u i \neq j} cosinesim(i,j) \tag{18}$$

The concept of Rank Information Retrieval can also be used for the calculation of diversity as shown in equation 19. This method has used a rank-discount differential function to determine the location of every pair of items being analyzed ($disc(k)$ and $disc(l|k)$) to access the intra-list similarity. It also uses a distance function ($d(i_k, i_l)$) between the items i_k and i_l. Cosine similarity measure can also be used as a distance function and hence equation 17 can be redefined as shown in equation 19 (Vargas & Castells [84]).

$$Diversity(l_u) = \sum_{k=1}^{|l_u|}\sum_{l=1}^{|l_u|} disc(k)disc(l|k)d(i_k, i_l)\forall i_k \neq i_l \tag{19}$$

3.4.2. Novelty

Novelty is described as the property for getting novel items in the list of recommendations. It can be broadly categorized into three types: Life-level novelty, System-level novelty, and Recommendation list novelty.

The user should be unaware of novel items (Ricci *et al.* [66]). Life-level novelty is the hypothetical way to measure the novelty as the RS has to inquire whether the user identifies the items. Zhang *et al.* [90] have concluded that items utilized by the users should be considered by RS when formulating predictions. Items that the users have not ever encountered before in their history are novel items. Developing methods to measure life-level novelty is not easy, as the RS has to find certain details to determine what the user knows and doesn't know. The researchers have not suggested any metrics in the past to measure the life level novelty.

System-level novelty indicates that a novel item for a user is something that the user has little or no expertise in it. All the researchers in the past have developed different system-level novelty metrics but they all have approached the same concept. They all have considered the system-level novelty as the dimension of the items unknown in the recommendation list. System-level novelty can be measured as the closeness between the items in the recommendation list l_u and the user's history h_u (Nakatsuji *et al.* [58]) as shown in equation 20.

$$Novelty(l_u) = \sum_{i \in l_u} min_{j \in h_u} d(class(i), class(j)) \tag{20}$$

Where *class(i)* represents the class of item i and d is the distance function. System-level novelty can also be determined by adding up the demand of items in the list generated by the user's RS (Ricci *et al.* [66], Gravino *et al.* [34]) as shown in equation 21 and 22.

$$Novelty(l_u) = \sum_{i \in l_u} \frac{\log_2 pop(i)}{|l_u|} \tag{21}$$

$$Novelty(l_u) = 1 - \frac{|popularity(i)|}{N} \tag{22}$$

Where *pop(i)* reflects the popularity of element i by taking into account the number of users who have already consumed it and N is the number of users.

Recommendation list novelty is associated with the RS at the recommendation list level. It is represented as the unrepeated items in the recommendation list which are not in the user's knowledge. User information is not required to find out recommendation list novelty. A recommendation list novelty by considering the intra-list similarity has been proposed by Bobadill *et al.* [11] as shown in equation 23.

$$Novelty(l_u) = \frac{1}{|l_u| - 1} \sum_{j \in l_u} 1 - d(i, j) \tag{23}$$

Where *d(i, j)* represents the distance between the item *i* and *j*. Another more efficient metric for the recommendation list novelty using the rank-discount differential function as used in section 3.4.1 for calculation of diversity has also been suggested which takes into an examination of the location of items in the recommendation list as shown in equation 24.

$$Novelty(l_u) = \sum_{k=1}^{|l_u|} disc(k)(1 - p(seen|i_k)) \tag{24}$$

3.4.3. Serendipity

The word serendipity means a lucky finding or a rewarding revelation (Silveira *et al.* [75]). Many of the researchers like Ge *et al.* [29] and Kotkov *et al.* [49] in the past have argued that serendipity is the users' perception of the recommendations that they receive. It can also be defined as surprising recommendations or surprising and interesting items for the user (Ricci *et al.* [66], Ge *et al.* [29]). Due to the involvement of unexpectedness and utility, serendipity can be divided into two types: primitive serendipity and non-primitive serendipity.

Primitive serendipity has been proposed using the relevance function and position of the item in the recommendation rank. The relevance function returns 1 if the item is relevant to the user and 0 otherwise. The position of the item in the recommendation rank can be found out as

$$\frac{count_k(k)}{k}$$

which is shown in equation 25.

$$Serendipity(l_u) = \sum_{k=1}^{|l_u|} max(R_u[k] - PM_u[k], 0) relevance(i_k) \frac{count_k(k)}{k} \tag{25}$$

Where PM_u is the Primitive Recommender. Another metric for serendipity using the unexpectedness of the item is proposed as shown in equation number 26.

$$Serendipity(l_u) = \frac{\sum_{i \epsilon Unexp_u} Utility(i)}{|RS_u|} \tag{26}$$

Where $UNEXP_u$ represents the surprising item for the user u and is calculated as $UNEXP_u$=RS_u- PMu. RS_u represents the recommendations for the user u. The utility function is maintained by the system.

Primitive serendipity can also be defined as the rate of not expected (RS_u-EP_u) items as shown in equation number 27.

$$Serendipity(l_u) = \frac{(RS_u - EP_u) \cap Useful_u}{|RS_u|} \tag{27}$$

Where EP_u represents the set of expected items and $Useful_u$ represents the set of useful items in the recommendation list.

The non-primitive serendipity using cosine similarity has been proposed in (Zhang *et al.* [90]). This uses the cosine similarity metric to determine the resemblance between the suggested items (RS_u) and the items consumed by the customer in the past (HC_u) as shown in equation 28.

$$Serendipity(l_u) = \frac{1}{HC_u} \sum_{i \epsilon HC_u} \sum_{j \epsilon RS_u} \frac{Cosinesim(i,j)}{RS_u} \tag{28}$$

The low similarity value represents high serendipity as the recommended items should not be very identical to the items consumed by the user in the past.

3.4.4. Unexpectedness

Unexpectedness can be considered as a part of serendipity and can be defined as deviating items expected to be consumed by the user (Murakami *et al.* [57]). The metrics for unexpectedness can also be divided into two types: primitive unexpectedness and non-primitive unexpectedness.

Ge *et al.* [29] have proposed a metric for primitive unexpectedness by considering unexpectedness as a deviation from the expected recommendation as shown in equation 29.

$$Unexp(RS_u) = RS_u - PM_u \tag{29}$$

Where PM_u represents the set of predicted items by the primitive recommender for the user u and RS_u represents the set of items in the recommendation list. Another metric for primitive unexpectedness has been proposed as the rate of unexpectedness. The set of unexpected items can be calculated as RS_u-EP_u. As EP_u=PM_u, this metric can also be defined in another way as shown in equation 30.

$$Unexp(RS_u) = \frac{RS_u - PM_u}{RS_u} \tag{30}$$

According to Kaminskas & Bridge [47], the metrics for non-primitive unexpectedness do not involve the predictions for the primitive recommender and only associated with the unexpectedness. This formula has compared the recommended items to the items that the user consumed in the past and tests if the user already understands the predictions as shown in equation 31.

$$Unexp(RS_u) = \sum_{i \epsilon RS_u} \sum_{j \epsilon HC_u} PMI(i,j) \tag{31}$$

$$PMI(i,j) = \frac{\log_2 \frac{p(i,j)}{p(i)p(j)}}{-\log_2 p(i,j)^2} \tag{32}$$

Where $p(i)$ is the probability that the users will rate item i. Another metric using the probability of co-occurrence and features of items has been proposed in Adamopoulos & Tuzhilin [4] as shown in equation 33.

$$Unexp(RS_u) = \frac{1}{\frac{1}{|F_i|}\sum_{v,w\epsilon F_i} \frac{I_v}{I_v+I_w-I_{v,w}}} \tag{33}$$

Where I_v determines the number of items that have feature v and I_v, w determines the number of items that have both features v and w.

3.5. Utility Recommendation for online evaluation

Utility-based assessment methods are used in online tests for assessing the effectiveness of the RSs for users. Generally, user experiments are made to evaluate the RS when it is implemented in the industry. The metrics used to assess RSs in the industry are click-through-rate (CTR) and retention. The accuracy-based assessment methods such as error metrics, precision, and recall can also be utilized for online evaluation as utility-based methods.

3.5.1. Click-through-rate (CTR)

The consistency-based metrics such as error thresholds, precision, and recall can also be utilized for online measurement as value-dependent thresholds. Click-through rate (CTR) is defined as the ratio of items consumed out of the total number of recommended items. Generally, the clicked or interacted items are considered as the consumed items. CTR is a very useful tool for the RSs that are implemented in the industry as it helps to determine how many items the user consumes successfully from the overall products in the recommendation list. This demonstrates the recommender system's efficacy in predicting important items for the user. The metric for CTR is shown in equation 34.

$$Utility(RS_u) = CTR = \frac{|NC_u|}{|RS_u|} \tag{34}$$

Where $|NC_u|$ is the number of items consumed by the user u and $|RS_u|$ is the number of items found in the user's recommendation list u. The various researchers in the past (Zhou *et al.* [94], Chu & Park [24]) have effectively used this measure for the assessment of their RSs.

3.5.2. Retention

Retention is another assessment tool used to test an RS online and it tests the capacity of an RS to encourage users to access items or use the system (Shani & Gunawardana

[73]). It is one of the measures needed to track how long users spend on their system and has been used as an evaluation measure of Netflix. According to Zhou *et al.* [94], the online retention checks are executed as A / B tests, and the retention delta is determined as shown in equation 35.

$$Utility(RS_u) = \triangle_{Retention} = P_t - P_c \tag{35}$$

Where P_t is the number of users in control and P_c is the number of users in test groups of the A/B test.

4. Relationship between Assessment Methods in Recommender System

The selection of suitable assessment methods for the evaluation of an RS plays an important role in the designing process of an RS. Various RSs in the literature have used different assessment methods for the evaluation process as shown in Table 2. From the various articles listed in Table 2, we can find that most of the RS have used MAE and RMSE as the assessment methods for prediction accuracy. These methods are mostly used to measure the closeness of the predicted value with the real value (Li & Li [52], Li & Murata [53]). Precision, recall, and F-measure are the important metrics that are used by many RSs as the recommendation of sets (Chen *et al.* [19], Asim & Khusro [8]). They are regarded as the metrics for decision support as they can be used to differentiate between right and wrong predictions in an RS. DCG is another metric found as the popular metric for the recommendation as a ranked list. RSs use this metric to rank the recommendations according to the preferences of the users (Zielinsk [95], Zheng & Pu [93]). Some accurate RSs may provide recommendations of the same type leading to a monotonous experience. The nontraditional methods like diversity and novelty are popularly used by the RSs to check the presence of this monotonicity in the recommendations (Castells *et al.* [17], Hurley & Zhang [44]). From the survey, it is also found that the click-through-rate metric is used by many of the RSs to evaluate users' engagement with recommendations. It can also be concluded that the assessment methods to evaluate an RS should be selected as per its utility so that it can properly convey the design implementation of the RS. However, there exists some implicit relations between the assessment methods as shown in Figure 4 which should be taken care of while evaluating an RS.

The methods of assessment can be categorized as user-dependent and user-independent according to their relation to user information. User-dependent methods are sensitive to user's information. Half-life, system-level novelty, unexpectedness, serendipity, RMSE, MAE, precision, and recall come under the category of user-dependent methods for evaluation of an RS. The user-independent methods simply evaluate an RS or a recommendation list without depending on the user's information. Diversity, recommendation list novelty, and coverage fall under the user-independent category.

The arrows present in Figure 4 indicate the relationships between the assessment

Table 2. Assessment Methods used for Different Recommender Systems

Sl.No.	Type of RS	Research Papers	Assessment Methods Used
1	Collaborative RS	[52], [20], [64], [71], [88]	MAE, RMSE, F-Measure, Precision, Accuracy, Coverage Ratio
2	Content-based RS	[60], [19], [8], [16]	MAE, Precision, Recall, Accuracy
3	Hybrid RS	[80], [59], [91]	MAE, Precision, Recall, F1-measure, Accuracy
4	Knowledge-based Rs	[70], [53]	Precision, Recall, NDCG
5	Demographic Filtering Based RS	[36], [48], [37], [62]	RMSE, MAE, Sensitivity, Specificity, Accuracy
6	Trust based RS	[81], [89], [92], [56]	MAE, RMSE, Coverage, Hit Ratio, Prediction shift, Precision, Recall
7	Context-aware RS	[1], [69], [3]	MAE, Coverage, Accuracy, RMSE
8	Social network-based RS	[85], [22], [43]	MAE, RMSE, Similarity Rank, Recall, Precision, F-score
9	Rating-based RS	[5], [86], [72], [26]	RMSE, Accuracy, MSE
10	Feature-based RS	[39], [23], [78]	Precision, Recall, MAE, RMSE
11	Case-based RS	[13], [32], [77], [35]	MAE, RMSE, Accuracy, Precision, Recall
12	Utility-based RS	[95], [93], [63], [54]	Ranking, DCG, NDCG, Rank, Utility
13	Critiquing-based RS	[30], [33], [21], [65]	Relevance, Utility, Accuracy, Confidence, Ranking
14	Personality-based RS	[82], [87], [42], [68], [17], [44]	F1-measure, Precision, Recall, Success ratio, Diversity, Novelty

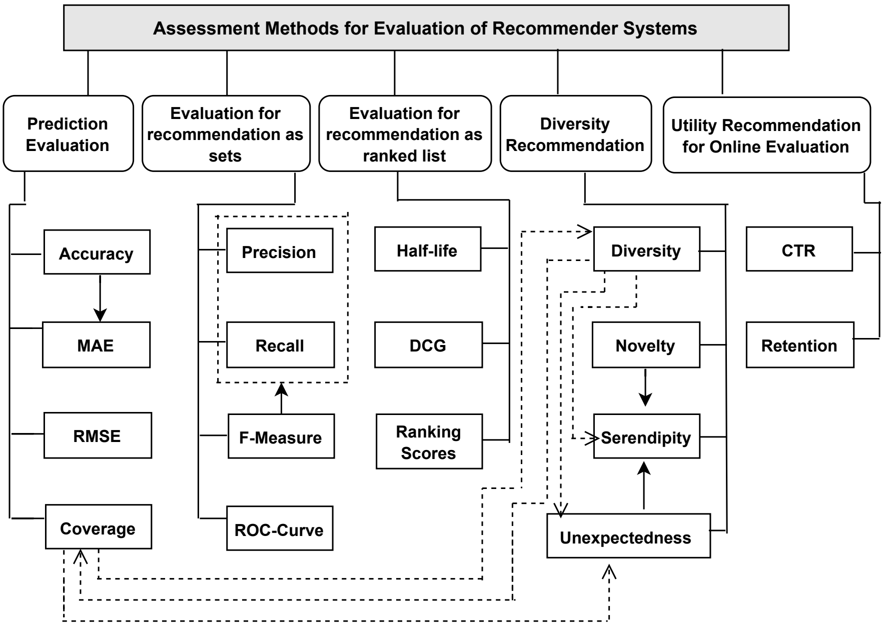

Figure 4. Assessment Methods for Evaluation of Recommender Systems

methods. Accuracy is related to MAE as $Accuracy = 1 - MAE$. The assessment method F-measure depends on the values of precision and recall as shown in equation 12. Similarly, unexpectedness and novelty are related to serendipity because serendipitous recommendations by nature are novel. After all, the user must not be aware of them (Ge *et al.* [29], Herlocker *et al.* [41]). Meanwhile, Serendipitous items need to be unexpected as they are the recommended items that are considered as a rewarding surprise to the user according to the definition (Ge *et al.* [29]).

The dashed arrows in Figure 4 present the possible correlation among the methods. Coverage would have a relationship with unexpectedness as uncovered items may increase unexpectedness to the user. Coverage could be related to diversity as shown by a dashed line in Figure 4. Similarly, a relation between diversity and serendipity and between diversity and coverage can also be possible. Unexpectedness may also have a relation to diversity.

5. Conclusion

This article reviews various assessment methods for the evaluation of an RS. The assessment methods are the primary factors in identifying and establishing an RS. The users always expect accurate recommendations and the performance of any RS is judged on this basis.

So, the assessment methods play the important role in designing an RS where various algorithms are compared and the best one is always selected for application design. The direct and indirect relationships between the assessment methods are also explored that affect the overall performance of the system.

The assessment methods evaluating RSs are mostly developed by the researchers or with the cooperation of the industry. The current real-life RSs include the assessment methods like retention, CTR, coverage, accuracy, and utility. In comparison to academic research, real-life RSs mostly adopt coverage and utility. However, in real-life RS, the engineering efforts also play a significant role to improve the performance of the RS. Hence sometimes, the system has to compromise with the efficiency to ignore the engineering effort to improve its performance.

This article can help the researchers to get an overall idea about the various assessment methods, their relationships, and their use in the designing of an RS.

References

[1] Abu-Issa, A., Nawawreh, H., Shreteh, L., Salman, Y., Hassouneh, Y., Tumar, I., Hussein, M. (2017, August). A smart city mobile application for multitype, proactive, and context-aware recommender system. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-5). IEEE.

[2] Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6), 734-749.

[3] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems (TOIS), 23(1), 103-145.

[4] Adamopoulos, P., Tuzhilin, A. (2014). On unexpectedness in recommender systems: Or how to better expect the unexpected. ACM Transactions on Intelligent Systems and Technology (TIST), 5(4), 1-32.

[5] Ajesh, A., Nair, J., Jijin, P. S. (2016, September). A random forest approach for rating-based recommender system. In 2016 International conference on advances in computing, communications and informatics (ICACCI) (pp. 1293-1297). IEEE.

[6] Ardagelou, P., Arampatzis, A. (2017). A Half-Life Decaying Model for Recommender Systems with Matrix Factorization. In TDDL/MDQual/Futurity@ TPDL.

[7] Armstrong, R., Freitag, D., Joachims, T., Mitchell, T. (1995, March). Webwatcher: A learning apprentice for the world wide web. In AAAI Spring symposium on Information gathering from Heterogeneous, distributed environments (Vol. 93, p. 107).

[8] Asim, M., Khusro, S. (2018). Content Based Call for Papers Recommendation to Researchers. 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), 42-47.

[9] Baeza-Yates, R., Ribeiro-Neto, B. (1999). Modern information retrieval. Essex: Addison Wesley.

[10] Billsus, D., Pazzani, M. J. (1998, July). Learning collaborative information filters. In Icml (Vol. 98, pp. 46-54).

[11] Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A. (2013). Recommender systems survey. Knowledge-based systems, 46, 109-132.

[12] Breese, J., Heckerman, D., Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. Uncertainty in artificial intelligence, proceedings of the fourteenth conference (pp. 43–52). Morgan Kaufman.

[13] Bridge, D., Göker, M. H., McGinty, L., Smyth, B. (2005). Case-based recommender systems. The Knowledge Engineering Review, 20(3), 315-320.

[14] Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) (2007). The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, vol. 4321. Springer-Verlag, Berlin.

[15] Burke, R. (2002). Hybrid recommender systems. User Modeling and UserAdapted Interaction, 12(4), 331–370.

[16] Cami, B. R., Hassanpour, H., Mashayekhi, H. (2017, December). A content-based movie recommender system based on temporal user preferences. In 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS) (pp. 121-125). IEEE.

[17] Castells, P., Vargas, S., Wang, J. (2011). Novelty and diversity metrics for recommender systems: choice, discovery and relevance.

[18] Chelliah, M., Sarkar, S. (2017, August). Product recommendations enhanced with reviews. In Proceedings of the Eleventh ACM Conference on Recommender Systems (pp. 398-399).

[19] Chen, H. W., Wu, Y. L., Hor, M. K., Tang, C. Y. (2017, July). Fully content-based movie recommender system with feature extraction using neural network. In 2017 International conference on machine learning and cybernetics (ICMLC) (Vol. 2, pp. 504-509). IEEE.

[20] Chen, J., Zhao, C., Chen, L. (2019). Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering. Complex Intelligent Systems, 1-10.

[21] Chen, L., Pu, P. (2006, July). Evaluating critiquing-based recommender agents. In AAAI (Vol. 6, pp. 157-162).

[22] Chen, L., Nayak, R., Xu, Y. (2011, July). A recommendation method for online dating networks based on social relations and demographic information. In 2011 International Conference on Advances in Social Networks Analysis and Mining (pp. 407-411). IEEE.

[23] Chen, S., Peng, Y., Mi, H., Wang, C., Huang, Z. (2018, March). A cluster feature based approach for QoS prediction in Web service recommendation. In 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE) (pp. 246-251). IEEE.

[24] Chu, W., Park, S. T. (2009, April). Personalized recommendation on dynamic content using predictive bilinear models. In Proceedings of the 18th international conference on World wide web (pp. 691-700).

[25] Davis, J., Goadrich, M. (2006). The relationship between precision, recall and roc curves. In Proceedings of the 23rd international conference on machine learning (ICML).

[26] Ding, J., Wang, Y., Wang, Q., Cao, Y. (2017, October). Football video recommendation system with automatic rating based on user behavior. In 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) (pp. 1-5). IEEE.

[27] Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8), 861-874.

[28] Fisher, M. J., Fieldsend, J. E., Everson, R. M. (2004). Precision and recall optimisation for information access tasks. In First workshop on roc analysis in AI. European conference on artificial intelligence (ECAI'2004), Valencia, Spain, August.

[29] Ge, M., Delgado-Battenfeld, C., Jannach, D. (2010, September). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In Proceedings of the fourth ACM conference on Recommender systems (pp. 257-260).

[30] Giustozzi, F., Casali, A., Deco, C., dos Santos, H.L. and Cechinel, C. (2016). Recommender system of educational resources: A critiquing-based proposal. 2016 XI Latin American Conference on Learning Objects and Technology (LACLO), San Carlos, 2016, pp. 1-8, doi: 10.1109/LACLO.2016.7751779.

[31] Goldberg, K., Roeder, T., Gupta, D., Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 4(2), 133–151.

[32] Gong, S. (2009, May). Joining case-based reasoning and item-based collaborative filtering in recommender systems. In 2009 Second International Symposium on Electronic Commerce and Security (Vol. 1, pp. 40-42). IEEE.

[33] Grasch, P., Felfernig, A., Reinfrank, F. (2013, October). Recomment: Towards critiquing-based recommendation with speech interaction. In Proceedings of the 7th ACM Conference on Recommender Systems (pp. 157-164).

[34] Gravino, P., Monechi, B., Loreto, V. (2019). Towards novelty-driven recommender systems. Comptes Rendus Physique, 20(4), 371-379.

[35] Guo, Y., Deng, G., Zhang, G., Luo, C. (2007, September). Using case-based reasoning and social trust to improve the performance of recommender system in e-commerce. In Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007) (pp. 484-484). IEEE.

[36] Gupta, J., Gadge, J. (2014, April). A framework for a recommendation system based on collaborative filtering and demographics. In 2014 international conference on circuits, systems, communication and information technology applications (CSCITA) (pp. 300-304). IEEE.

[37] Gupta, J. Gadge, J. (2015). Performance analysis of recommendation system based on collaborative filtering and demographics. Proceedings - 2015 International Conference on Communication, Information and Computing Technology, ICCICT 2015. 10.1109/ICCICT.2015.7045675.

[38] Hariadi, A. I., Nurjanah, D. (2017, November). Hybrid attribute and personality based recommender system for book recommendation. In 2017 International Conference on Data and Software Engineering (ICoDSE) (pp. 1-5). IEEE.

[39] Hayashi, A., Itoh, T., Nakamura, S. (2013, July). A visual analytics tool for system logs adopting variable recommendation and feature-based filtering. In 2013 17th International Conference on Information Visualisation (pp. 1-10). IEEE.

[40] Herlocker, J. L., Konstan, J. A., Borchers, A., Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In SIGIR'99: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval (pp. 230–237). New York, NY, USA: ACM Press.

[41] Herlocker, J. L., Konstan, J. A., Terveen, L. G., Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.

[42] Hu, R., Pu, P. (2010, June). A study on user perception of personality-based recommender systems. In International conference on user modeling, adaptation, and personalization (pp. 291-302). Springer, Berlin, Heidelberg.

[43] Huang, X., Tang, Y., Qu, R., Li, C., Yuan, C., Sun, S., Xu, B. (2018, May). Course Recommendation Model in Academic Social Networks Based on Association Rules and Multi-similarity. In 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)) (pp. 277-282). IEEE.

[44] Hurley, N., Zhang, M. (2011). Novelty and diversity in top-n recommendation–analysis and evaluation. ACM Transactions on Internet Technology (TOIT), 10(4), 1-30.

[45] Hurley, N. J. (2013, October). Personalised ranking with diversity. In Proceedings of the 7th ACM conference on Recommender systems (pp. 379-382).

[46] Jarvelin, K., Kekalainen, J. (2002). Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS), 20(4), 422-446.

[47] Kaminskas, M., Bridge, D. (2014, October). Measuring surprise in recommender systems. In Proceedings of the workshop on recommender systems evaluation: dimensions and design (Workshop programme of the 8th ACM conference on recommender systems).

[48] Katarya, R., Verma, O. P. (2015, October). Restaurant recommender system based on psychographic and demographic factors in mobile environment. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 907-912). IEEE.

[49] Kotkov, D., Veijalainen, J., Wang, S. (2016). Challenges of serendipity in recommender systems. In WEBIST 2016: Proceedings of the 12th International conference on web information systems and technologies. Volume 2, ISBN 978-989-758-186-1. SCITEPRESS.

[50] Kuanr, M., Mohapatra, P. (2021). Recent Challenges in Recommender Systems: A Survey. In Progress in Advanced Computing and Intelligent Engineering (pp. 353-365). Springer, Singapore.

[51] Kunaver, M., Požrl, T. (2017). Diversity in recommender systems–A survey. Knowledge-Based Systems, 123, 154-162.

[52] Li, X. Li, D. (2019). An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy. Mobile Information Systems. 2019. 1-11. 10.1155/2019/3560968.

[53] Li, X., Murata, T. (2010, October). Customizing knowledge-based recommender system by tracking analysis of user behavior. In 2010 IEEE 17Th International Conference on Industrial Engineering and Engineering Management (pp. 65-69). IEEE.

[54] Liang, S., Liu, Y., Jian, L., Gao, Y., Lin, Z. (2011, August). A utility-based recommendation approach for academic literatures. In 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (Vol. 3, pp. 229-232). IEEE.

[55] Ma, W., Shi, J., Zhao, R. (2017). Normalizing item-based collaborative filter using context-aware Scaled baseline predictor. Mathematical Problems in Engineering, 2017.

[56] Moghaddam, M. G., Elahian, A. (2014, May). A novel temporal trust-based recommender system. In 2014 22nd Iranian Conference on Electrical Engineering (ICEE) (pp. 1142-1146). IEEE.

[57] Murakami, T., Mori, K., Orihara, R. (2007, June). Metrics for evaluating the serendipity of recommendation lists. In Annual conference of the Japanese society for artificial intelligence (pp. 40-46). Springer, Berlin, Heidelberg.

[58] Nakatsuji, M., Fujiwara, Y., Tanaka, A., Uchiyama, T., Fujimura, K., Ishida, T. (2010, October). Classical music for rock fans? Novel recommendations for expanding user interests. In Proceedings of the 19th ACM international conference on Information and knowledge management (pp. 949-958).

[59] Ng, P. C., She, J., Cheung, M., Cebulla, A. (2016, April). An images-textual hybrid recommender system for vacation rental. In 2016 IEEE Second International Conference on Multimedia Big Data (BigMM) (pp. 60-63). IEEE.

[60] Pal, A., Parhi, P., Aggarwal, M. (2017, August). An improved content based collaborative filtering algorithm for movie recommendations. In 2017 tenth international conference on contemporary computing (IC3) (pp. 1-3). IEEE.

[61] Pazzani, M. J., Muramatsu, J., Billsus, D. (1996). Syskil Webert: Identifying interesting web sites. In Proceedings of the national conference on artificial intelligence (Vol. 1, pp. 54–61).

[62] Pandey, A. K., Rajpoot, D. S. (2016, December). Resolving cold start problem in recommendation system using demographic approach. In 2016 International Conference on Signal Processing and Communication (ICSC) (pp. 213-218). IEEE.

[63] Prangl, M., Bachlechner, R., Hellwagner, H. (2007, July). A hybrid recommender strategy for personalized utility-based cross-modal multimedia adaptation. In 2007 IEEE International Conference on Multimedia and Expo (pp. 1707-1710). IEEE.

[64] Pujahari, A., Padmanabhan, V. (2015, December). Group Recommender Systems: Combining user-user and item-item Collaborative filtering techniques. In 2015 International Conference on Information Technology (ICIT) (pp. 148-152). IEEE.

[65] Ricci, F., Nguyen, Q. N. (2005). Critique-based mobile recommender systems. OEGAI Journal, 24(4), 1-7.

[66] Ricci, F., Rokach, L., Shapira, B. Kantor, P. (2011) Recommender systems handbook. Springer, Berlin.

[67] Rogers, S., Flechter, C., Langley, P. (1999). An adaptive interactive agent for route advice. In Proceedings of the third international conference on autonomous agents (Agents'99) (pp. 198–205). Seattle, WA, USA: ACM Press.

[68] Roshchina, A. (2012). TWIN: Personality-based Recommender System. Institute of Technology Tallaght, Dublin.

[69] Salman, Y., Abu-Issa, A., Tumar, I., Hassouneh, Y. (2015, October). A proactive multi-type context-aware recommender system in the environment of internet of things. In 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (pp. 351-355). IEEE.

[70] Samin, H., Azim, T. (2019). Knowledge based recommender system for academia using machine learning: A case study on higher education landscape of Pakistan. IEEE Access, 7, 67081-67093.

[71] Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web (pp. 291-324). Springer, Berlin, Heidelberg.

[72] Shahjalal, M. A., Ahmad, Z., Arefin, M. S., Hossain, M. R. T. (2017, December). A user rating based collaborative filtering approach to predict movie preferences. In 2017 3rd International Conference on Electrical Information and Communication Technology (EICT) (pp. 1-5). IEEE.

[73] Shani, G., Gunawardana, A. (2009) Evaluating recommender systems. Microsoft Research. https://www.microsoft.com/en-us/ research/publication/evaluating-recommender-systems/

[74] Shardanand, U., Maes, P. (1995). Social information filtering: Algorithms for automating 'word of mouth'. In CHI'95: Proceedings of the conference of human factors in computing systems. ACM Press.

[75] Silveira, T., Zhang, M., Lin, X., Liu, Y., Ma, S. (2019). How good your recommender system is? A survey on evaluations in recommendation. International Journal of Machine Learning and Cybernetics, 10(5), 813-831.

[76] Spiekermann, S. Paraschiv, C. (2002). Motivating Human–Agent Interaction: Transferring Insights from Behavioral Marketing to Interface Design. Electronic Commerce Research. 2. 255-285. 10.2139/ssrn.1346865.

[77] Supic, H. (2012, October). An approach to integration of contextual information in case-based recommender systems. In 2012 IX International Symposium on Telecommunications (BIHTEL) (pp. 1-5). IEEE.

[78] Tian, F., Chen, Y., Wang, X., Lan, T., Zheng, Q., Chao, K. M. (2015, October). Common Features Based Volunteer and Voluntary Activity Recommendation Algorithm. In 2015 IEEE 12th International Conference on e-Business Engineering (pp. 43-47). IEEE.

[79] Thiprak, S., Kurutach, W. (2015, June). Ubiquitous computing technologies and context aware recommender systems for ubiquitous learning. In 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 1-6). IEEE.

[80] Thuan, T. T., Puntheeranurak, S. (2014, October). Hybrid recommender system with review helpfulness features. In TENCON 2014-2014 IEEE Region 10 Conference (pp. 1-5). IEEE.

[81] Tyagi, S., Bharadwaj, K. K. (2012, December). Trust-enhanced recommender system based on case-based reasoning and collaborative filtering. In 2012 2nd International Conference on Power, Control and Embedded Systems (pp. 1-4). IEEE.

[82] Uddin, M., F., Banerjee, S., and Lee, J.(2016).Recommender System Framework for Academic Choices: Personality Based Recommendation Engine (PBRE). 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), Pittsburgh, PA, 2016, pp. 476-483, doi: 10.1109/IRI.2016.70.

[83] Vargas, S., Baltrunas, L., Karatzoglou, A., Castells, P. (2014, October). Coverage, redundancy and size-awareness in genre diversity for recommender systems. In Proceedings of the 8th ACM Conference on Recommender systems (pp. 209-216).

[84] Vargas, S., Castells, P. (2011, October). Rank and relevance in novelty and diversity metrics for recommender systems. In Proceedings of the fifth ACM conference on Recommender systems (pp. 109-116).

[85] Wongkhamchan, T., Namvong, A., Surawanitkun, C. (2019, July). Personalized Recommender System Using A Social Network Based Collaborative Filtering Technique. In 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 846-849). IEEE.

[86] Xie, F., Xu, M., Chen, Z. (2012, March). RBRA: A simple and efficient rating-based recommender algorithm to cope with sparsity in recommender systems. In 2012 26th International Conference on Advanced Information Networking and Applications Workshops (pp. 306-311). IEEE.

[87] Yakhchi, S., Beheshti, A., Ghafari, S. M., Orgun, M. (2020). Enabling the Analysis of Personality Aspects in Recommender Systems. arXiv preprint arXiv:2001.04825.

[88] Yu, C., Tang, Q., Liu, Z., Dong, B., Wei, Z. (2018, July). A recommender system for ordering platform based on an improved collaborative filtering algorithm. In 2018 International Conference on Audio, Language and Image Processing (ICALIP) (pp. 298-302). IEEE.

[89] Zhang, F. (2011, November). Analysis of bandwagon and average hybrid attack model against trust-based recommender systems. In 2011 Fifth International Conference on Management of e-Commerce and e-Government (pp. 269-273). IEEE.

[90] Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., Jambor, T. (2012, February). Auralist: introducing serendipity into music recommendation. In Proceedings of the fifth ACM international conference on Web search and data mining (pp. 13-22).

[91] Zhang, Y., Liu, X., Liu, W., Zhu, C. (2016, September). Hybrid recommender system using semi-supervised clustering based on Gaussian mixture model. In 2016 International Conference on Cyberworlds (CW) (pp. 155-158). IEEE.

[92] Zheng, X. L., Chen, C. C., Hung, J. L., He, W., Hong, F. X., Lin, Z. (2015). A hybrid trust-based recommender system for online communities of practice. IEEE Transactions on Learning Technologies, 8(4), 345-356.

[93] Zheng, Y., Pu, A. (2018, December). Utility-based multi-stakeholder recommendations by multi-objective optimization. In 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI) (pp. 128-135). IEEE.

[94] Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., ... Gai, K. (2018, July). Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (pp. 1059-1068).

[95] Zielinski, A. (2015, July). A utility-based semantic recommender for technology-enhanced learning. In 2015 IEEE 15th International Conference on Advanced Learning Technologies (pp. 394-396). IEEE.

Received 18.08.2020, Accepted 20.07.2021