CrossMark

# Adaptive KNN based Recommender System through Mining of User Preferences

V. Subramaniyaswamy[1] · R. Logesh[1]

**Abstract** Research for the generation of reliable recommendations has been the main goal focused by many researchers in recent years. Though many recommendation approaches have been developed to assist users in the selection of their interesting items in the online world, still the personalization problem exists. In this paper, we present a new recommendation approach to address the problems such as scalability, sparsity, and cold-start in a collective way. We have developed a knowledge-based domain specific ontology for the generation of personalized recommendations. We have also introduced two different ontology-based predictive models as minion representation model and prominent representation model for the effective generation of recommendations to all types of users. The prediction models are induced by data mining algorithms by correlating the user preferences and features of items for user modeling. We have proposed a new variant of KNN algorithm as Adaptive KNN for the collaborative filtering based recommender system. The proposed recommendation approach is validated with standard MovieLens dataset and obtained results are evaluated with Precision, Recall, F-Measure, and Accuracy. The experimental results had proved the better performance of our proposed AKNN algorithm over other algorithms with the highly sparse data taken for the recommendation generation.

✉ V. Subramaniyaswamy
vsubramaniyaswamy@gmail.com

R. Logesh
logeshr@outlook.com

[1]    School of Computing, SASTRA University, Thanjavur, India

 Springer

## 1 Introduction

Recommender systems (RSs) as the web applications has been used in many domain areas to make predictions for user preferences and help active target user as a digital support tool in the search and selection of services or products. Currently, RSs have been implemented in various domains to make recommendations of the products based on the user preferences and evaluations of the similarities between the user interests and features of the items or products. Recommendations based on the collaborative filtering have been proved to be successful by addressing the traditional prediction and scalability problems [30, 33, 37, 40]. Generation of precise recommendations to overcome the sparsity problem by just evaluating the user preferences is not considered to be a sufficient. The incremental research carried out in the development of recommender systems has improved the efficiency of the generated recommendations. To deal with the data sparsity problems, data mining algorithms are applied to recommendation approaches to improve the performance as the recommendation approaches are not only based on the evaluation of items but also the prediction process considers other attributes too [43]. The induction of data mining algorithm in offline mode nowhere helps in the reduction of the user response time and reduces the precision based on the algorithm used. Hence, based on the requirement to address the sparsity problem, we need to discover appropriate data mining algorithms for the efficient personalized recommendations generations. Irrespective of the benefits of data mining algorithms in the utilization of recommendation generation process, there are few situations, the RS fails to provide recommendations. For an example, when a new item is introduced to the system without any former evaluations or ratings, the cold-start problem arises during the prediction and recommendation processes [13, 21].

Generally, RSs are widely used to generate suggestions for users on various things based on their preferences [10, 35, 46]. Still, the RSs are found to be incapable of predicting users' interests with absolute certainty due to various reasons like sparsity, cold-start and information overload issues. The recommender system is directed to present users with relevant information without them asking for it explicitly. In the recent, the development of RSs is found to have various tiers of complexity in the recommendation generation process of products that rely on users' associations in their previous transactions. Collaborative Filtering based recommender system (CFRS) make recommendations based on the similarity between users' previous ratings and user preferences in the prediction process. Few drawbacks in the conventional recommendation approaches were observed, particularly in the nearest neighbor algorithms, which cause grave performance and scalability problems during recommendation generation process. An efficient recommendation approach requires a very large number of evaluations for providing accurate recommendations and when the evaluations are not enough, it leads to the sparsity problem. The current web-based RSs utilized the improvements of the research carried out in the past years and generate effective recommendations.

In this paper, we present a novel recommendation framework to overcome the drawbacks of present recommendation models through employing semantic based context mining approach for effective recommendations. We have developed a knowledge-based domain specific ontology through applying an associative classification method with the help of annotated data of semantic metadata for personalized recommendations. In the user clustering process, the presented Adaptive K-Nearest Neighbor (AKNN) algorithm is used to increase the accuracy of the predictions by selecting appropriate attributes. Later, based on the user clusters, the personalized list of items are predicted and recommended to the active target user.

The remainder of the paper is organized as follows: Sect. 2 describes the related works carried out in the research and development of recommender systems and in Sect. 3, the proposed ontology-based recommendation framework is presented and explained in detail. Section 4 depicts the experimental setup and Sect. 5 portrays the obtained experimental results with in-depth discussions. Finally, Sect. 6 provides conclusions and future work directions.

## 2 Related Works

Generally, RSs are classified into three main categories based on the approaches used as Collaborative Filtering, Content-Based, and Hybrid [25]. CF based RSs are generally based on the nearest neighbour algorithms to make predictions on the user preferences for the items to be recommended based on the other similar users' opinions. The opinions of the other similar users are collected or organized from the ratings provided by the users or utilized from the previous purchase records with the time logs [39]. Initially, content-based approaches are introduced to recommend text documents through making comparisons on the contents of the documents and user profiles in the information recovery domain. The weights are computed for the words in the text documents and the computed weights are added to the corresponding words in the user profile when the user is interested in the page [25]. This approach has a major drawback as it fails to manage or compare the objects such as music, images, motion pictures, videos etc. In addition, the approach is incapable of processing a large number of words from the product reviews and attributes contents. The recommendation approach is extended to all kinds of items through replacing the document words to the attributes of items. Hybrid recommendation approach is crossbreed model for the recommendation generation by taking the advantageous features of collaborative filtering and content-based techniques.

At present, there are two main classifications in collaborative filtering approaches and they are model-based and memory-based algorithms. Memory-based algorithms are also known as the user-based algorithms or nearest-neighbor based algorithms. These memory-based algorithms use statistical techniques to treat the users and items to determine similar users with the similar preferences as neighbors. The recommendations are predicted based on the features of the neighborhood of the active target user. To determine the similarity between the items/users, there are many similarity functions exists and Pearson Correlation Coefficient (PCC) is most commonly used similarity metric. After the discovery of the similar items, the weighted average of the ratings provided by the neighbor users of the active target user is computed to make predictions. The advantage of memory-based algorithms is the utilization of most recent information and it has disadvantages of processing the high number of neighbors is very slow with larger user databases [40]. To overcome the drawbacks of memory-based CF algorithms, model-based algorithms are introduced through utilizing data mining techniques to model the users' ratings and to predict the user preferences based on modeled users' ratings. In the existing works, there are many data mining techniques used with the model-based CF approaches. Nearest neighbor based CF approach is modified for the classification problem by using the neural networks methodologies [6]. Similarly, Bayesian networks are used in the recommendation process as a standalone model or used in parallel with other techniques [7, 8]. The main disadvantages of these model-based CF algorithms need high computational resources to process the larger data. Support Vector Machines (SVM) are also used as a complementary approach in the RSs and some existing work employs SVM with other methods for the recommendation process [16, 47]. In the recent times, to experience and utilize the

advantageous strengths of hybrid methodologies, the collaborative and content-based filtering is used as the combinational model for the recommendation generation [4]. Some recent research works have added semantic information to the data available to formalize and categorize the user's attributes and products.

The memory-based collaborative filtering approach has major limitations that affect the quality of the generated recommendations. In this specific model, the rating prediction process is made with the homogeneous information available. The utilized information from the homogeneous domain leads to the scalability and sparsity issues, which is the serious factors that make only poor recommendations to the user [12]. Generally, sparsity problem occurs during the ratings prediction process while the ratings predicted by the collaborative filtering are lesser than the required ratings for the estimation of user's preferences for the items. In addition, the memory-based CF models also lag in the performance of the neighborhood estimation for the target user. The above-said problems specifically occur due to the requirement to process the huge user information. As the number of user and items increases, the computational time taken for the recommendation generation also increases linearly. The main aspect in the evaluation of the RS is the time taken for the recommendation and the model-based CF methods have advantageous lesser time in the recommendation generation. In the model-based CF methods, the model has been built in the off-line mode before the user goes online and this leads to the time reduction in the data processing for the recommendation generation. The major limitation of the model-based CF approaches is that the user's recent activity and feedback information does not make any impact with immediate effect for recommendation processing for the user.

Eirinaki et al. [18] suggested a clustering method that yields quick and better recommendations to the user. Semantically coherent clusters and Domain Ontology have been used for clustering and recommendation respectively. Mobasher et al. [32] have suggested a method called Profile Aggregation for web usage mining, in which the presented algorithm clusters the database based on similar transactions and pages that have been predicted through page view clustering. Nasraoui and Petenes [34] have suggested a smart web recommendation system using Fuzzy approximation reasoning method. They have utilized web usage mining to extract user profiles and grouped user information using Hierarchical Unsupervised clustering mechanism, and Fuzzy approximation reasoning has been used for recommendation generation. In Cho et al. [12] have proposed a solution for sparsity and scalability problems in collaborative filtering technique by the using decision tree induction, data warehousing technologies, and association rule mining algorithms. To find frequent patterns, they have used Apriori algorithm and utilized Decision tree induction method to classify the customers. Zhou et al. [50] have developed sequential pattern mining to predict the next web access patterns. Model Based Filtering technique generates the consecutive web access patterns, and it is also used for the generation of recommendation rules and user pattern matching.

Wang and Shao [45] have used association rule mining and web usage mining based clustering for a better recommendation. The "similar" users' interests are determined by making use of web usage mining techniques in two modes as Online and Offline. In Offline mode, they use Pattern Discovery, Data Pre-processing and Pattern Analysis methods and in online mode, they match the current user's profile with the aggregate usage profiles. Garcin et al. [20] have presented context trees for the accurate recommendation of stories and news to the users. The presented system, based on user interest provides the better recommendation, and the obtained results have been improved a lot in accuracy and recommendation quality. Khribi et al. [24] have utilized content mining and collaborative filtering techniques for the recommendation generation. In the offline user modeling process, they have used colony

clustering and Fuzzy C-means technique for user clustering. Recommendation of websites based on the interest of the user is made using data structures like User-Interest matrix, Frequent-Path matrix, Web-Interest Matrix, Class-Interest Matrix for better personalization and generated recommendations are based on user's interest [29].

There are many variants of KNN approaches developed recently and they have their own advantages and disadvantages too. Vincent and Bengio [44] suggested utilization of K-local hyperplane for the nearest neighbor algorithm. Chien and Wu [11] made use of the nearest feature space and nearest feature plane classifiers for facial recognition. Zheng et al. [49] presented pattern classification using two nearest neighbors, which was termed as nearest neighbor line and nearest neighbor plane. Li et al. [26] used local probabilistic centers of every class to propose a new nearest neighbor classifier. Gao and Wang [19] has developed a new center based nearest neighbor classifier. Cover and Hart [14] have used Bayes probability error to analyze the nearest neighbor classifier theoretically. Cevikalp et al. [9] have studied the leaping Hyper disk of every training class for the nearest neighbor approach. Parthasarathy and Chatterji [36] proposed a new way of using the KNN method for smaller sample size. Majumdar and Ward [31] done research on the possible ways to combine the random projection technique with the nearest neighbor classifier. Hu et al. [23] studied about the sample weight learning to increase the margin of the nearest neighbor classifier. Sáez et al. [38] observed the effect of noise filters on the nearest neighbor classifier. Argentini and Blanzieri [3] suggested the neighborhood counting measure by considering the similarity measure of KNN algorithm. Derrac et al. [15] improved the performance of KNN by adopting the Cooperative co-evolution method. Triguero et al. [42] addressed the shortcomings of traditional KNN classifier and optimized the positioning of prototypes by adopting differential evolution. Hernández-Rodríguez et al. [22] has determined the efficiency of KNN classifier and suggested 'k' most similar neighbor based on a tree structure.

Zhou and Yu [51] have developed a KNN classifier-based ensemble framework. Domeniconi and Yan [17] studied about KNN ensemble approach and also about their relationship according to error correlation and accuracy. Altinçay [2] aimed to design a multimodal perturbation-based nearest neighbor classifier ensemble and achieved it with successful experimentations. Acha et al. [1] classified the psychophysical experimental image by using nearest neighbor classifier. Yang et al. [48] have classified hyperspectral image data by using KNN classifier. Liu and Nakagawa [28] made a survey on KNN classifiers and studied about its applications to handwritten character recognition. Bhatia and Vandana [5] made an analysis of the key ideas, benefits, drawbacks of the nearest neighbor classifier.

In this paper, we present a new ontology based recommendation model to enrich the user data for better prediction of user's preferences. Our proposed idea is to classify the relevant and irrelevant items for the active target user through built user model. The utilization of the semantic information helps in the generation of improved recommendations in the real-time scenarios. The case-based reasoning with the ontological structures is the new initiative to obtain meaningful results. The later sections describe in detail about out proposed recommendation model for the better understanding of the readers.

## 3 Proposed Ontology Based Recommendation Framework

The main goal of the proposed recommendation framework is to address the major limitation of the RSs such as the sparsity, scalability and cold-start problems. Though some earlier works had tried to solve these problems, but they have not addressed all the problems jointly. In our proposed recommendation model, we try to generate better

recommendations based on the user's context data. Our predictive model adopts the data mining algorithms for the enhanced analysis of user's preferences. We split our prediction process into the online and offline process and it is depicted in Fig. 1. In the online process, the recommendations are generated based on the representations framed in the off-line process. Generally, the online process is invoked when the user requests for the recommendations. The on-line process differs as the user type differs. For the better recommendation generation and performance enhancement, the old users and new users are treated differently. As the new users might not have rated any products, their preferences are generally unknown. In the off-line process, the prediction models are induced before the users entering into the system for the recommendation generation. The off-line process is responsible to update the models in the periodical order to maintain consistent recommendations. The main role of the off-line process is to incorporate the new information of the updated new users list and items along with their corresponding preferences and characteristics such as ratings to help the on-line process to make better predictions.

In the off-line process, the two different user models are built based on the available historical user data. The minion representation model correlates the user's ratings with the
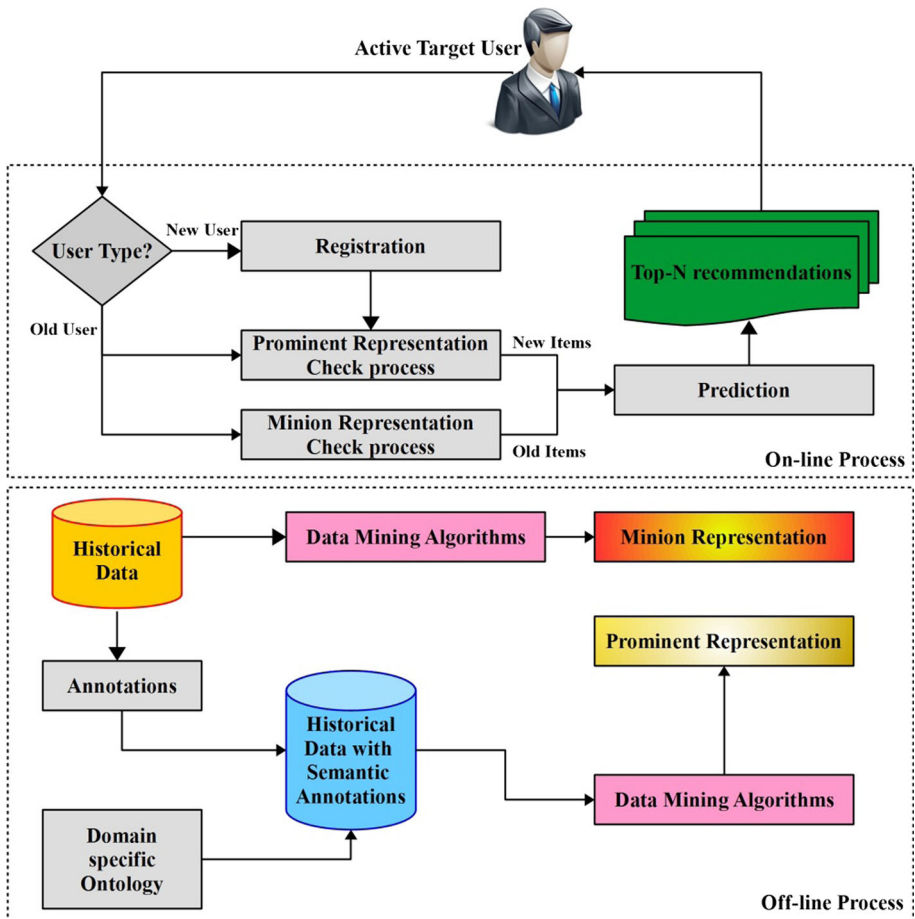


**Fig. 1** Proposed recommendation framework

specific items for the recommendations generations in the normal situations. The utilization of the minion representation for the recommendation generation does not require semantic annotations. On the other hand, the prominent representation model correlates the users' types with the item types based on the provided ratings. The users and the items are generally classified and clustered using the developed domain specific ontology and Adaptive KNN algorithm. The prominent representation is utilized to the cold-start users, to avoid the issue of unknown preferences and ratings. The prominent representation does not require ratings for the new users or items to provide recommendations to the users.

The prominent and minion representations are used by the online process of the proposed system during the generation of recommendations. The proposed system generates recommendations only to the registered users, as the basic information about the user is required to check user representation models. There are two possible ways to make recommendations to the users based on their types. The first type of the recommendation generation is for the new user who does not have rated any items. The second user type is regarding the old users who are already part of the system and rated some items. For the new users, the prominent representation enough for the recommendation generation and for the old users, both prominent and minion representation is required predict recommendations. The online process provides recommendations with both new and old items for the both new and old users in a better way. The minion representation is incapable of making recommendations with the newly added items to the system as it doesn't have users' ratings. On the other hand, the prominent representation provides recommendations with new items due to its capability to classify and cluster the similar items based their features. The characteristics of the items are shared with the other items that have ratings in the prediction model, which is used in the recommendation generation process for the new items.

The new users to the system are requested to register with their basic information to acquire semantic information of them. The semantic information is the very much required to classify them based on the ontology structures. As the new users had not provided any ratings to the items, it is hard to estimate the preferences and behavior of the user to make recommendations list. Hence the recommendation generation process is only feasible with the prominent representation model. The ontology of the prominent representation model comprises of patterns that are used to relate the users and items based on their feature and characteristics. The highly relevant items are alone recommended to the active target user. For the old user in the system who had already rated the items, to make recommendations the both minion and prominent representations are utilized. The adaptation of off-line user model building process has a greater advantage of addressing the scalability problems as the time taken for the model selection does not make any impact on the user response time. Getting speedy feedback from the RSs is the key aspect and this is considered to be the significant positive feature for our proposed recommendation model. The scalability issues of the traditional collaborative filtering based RSs are solved with our model based RS with the lesser time taken for the recommendation generation.

## 3.1 Domain Specific Ontology for Enhanced Recommendation

A domain specific ontology comprises of relationships and classes (abstract concepts) which may be shared and reused in future requirements. The web data can be classified based on the specific ontology through considering instances of ontology entities. To provide more meaningful knowledge to the users, web mining approaches are utilized to the instances of ontology entities [27]. To enhance the recommendations in our work and

overcome traditional problems, we use ontology. The developed knowledge-based domain-specific ontology is utilized to annotate semantic metadata to the MovieLens data. We have adopted public movies ontology from the TONES Ontology repository of the University of Manchester. To visualize the public movie ontology we have used Protégé editor developed by the Stanford University researchers and Fig. 2 portrays the representation of TONES movie ontology. As some user information is not available in the MoiveLens database to avoid the privacy concerns, the TONES ontology can be simplified for the better annotation processing. MovieLens dataset comprises of users' ratings for the movies and their appropriate demographic information. Based on the users' characteristics the ontology can be further re-organized through utilizing age, occupation, and gender. We have simplified the data-consideration for the movie recommendation as for user (User_ID, Gender, Age, Zip, Occupation), rating (User_ID, Movie_ID, Rating_bin, Score), and Movie (Movie_ID, title, genre). The simplified ontology is shown in Fig. 3. Through utilizing the simplified TONES ontology, high models are built by web mining approaches. We have developed a knowledge-based domain specific ontology based on the reviews and opinions of the movie. Figure 4 presents the proposed ontological model comprising of the relationships between the domains. The presented model is highly capable enough to be adaptable to the Linked Open Data.

Scalability issue mainly affects the performance of the collaborative filtering based RS which has been used with the large-scale databases. To address the sparsity problem, we utilized associative classification approaches with AKNN algorithm for clustering to enhance the precision over other recommendation models while using highly sparse data. Though model based RSs have advantages in the reduction of scalability and sparsity, yet many existing approaches doesn't prove to be good prediction models. We have conducted experiments on our proposed recommendation model with the MovieLens data and the
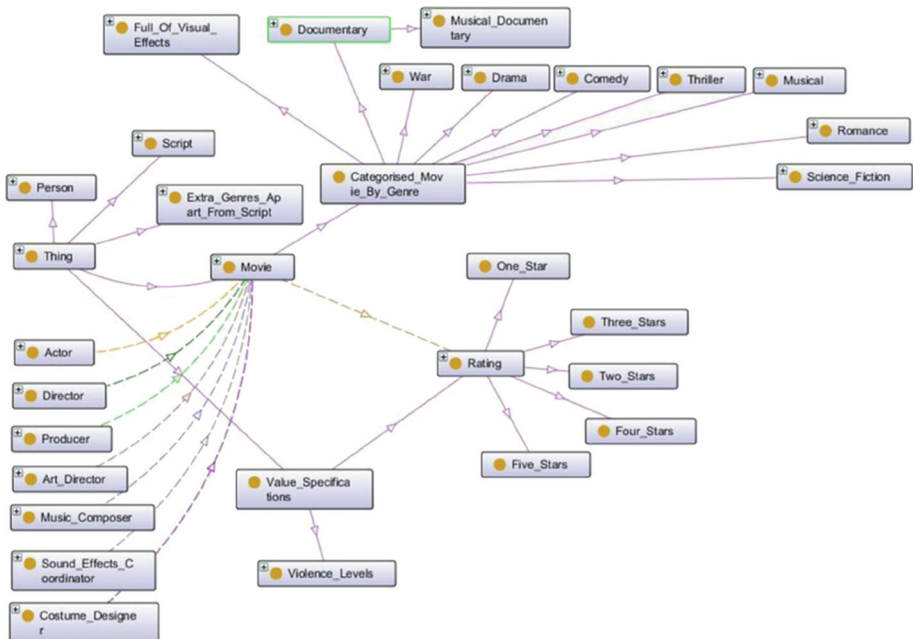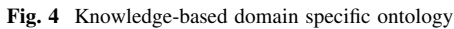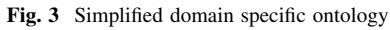


**Fig. 2** TONES movie ontology

**Fig. 3** Simplified domain specific ontology



**Fig. 4** Knowledge-based domain specific ontology

results had proved its sensitivity towards the scalability and sparsity problems. The upcoming section describes the proposed recommendation algorithms and ontology used for the estimation of users' preferences.

## 4 Adaptive KNN for Better Recommendations

In this section, we describe our algorithms used for the generation of recommendations in our proposed recommendation framework. As an initiative, we have modified traditional KNN algorithm as Adaptive KNN algorithm for the user and item clustering based on the similarity features. We adopted KNN algorithm due to its better performance among

nearest neighbor algorithms. To make top-n recommendations based on the clustering from the AKNN algorithm we have presented a new collaborative filtering algorithm.

## 4.1 AKNN Algorithm

We present a new variant of KNN algorithm as Adaptive KNN (AKNN) algorithm for the clustering process to make better predictions of unknown ratings for the active target user. The presented algorithm clusters users or items with the similar users or items until the criterion limit is reached. In this adaptive clustering approach, the clustering process is done for the target item with the similar item. For the MovieLens data, the movies are arranged based on the popularity and then it is considered for the clustering process. If the number of ratings is above the criterion limit, then the clustering is not made. Instead, it uses Each Item approach. If the number of ratings is smaller than the criterion limit than the item is clustered with the similar items until the cluster size reaches to the criterion maximum limit. Then the presented approach builds ratings prediction models to make appropriate ratings as a prediction. The Adaptive KNN for the clustering process is presented as follows:

**Algorithm**: AKNN

**Input**: *I* (Items), *U* (Users), *kR* (Known Ratings)

**Output**: *pR* (Predicted Ratings)

Specify *iVar* (independent variables) related to *I* and *U;*

Derive *dVar* (derived variables) related to *I* and *U* from *tVar* (transactional variables);

    If variables are of real type

        {

        Eliminate units' effects by subtracting mean;

        Divide with standard deviation;

        }

    If (*Number_of_data < check_criterion*)

        {

        //Cluster items with nearest neighbors

        Compute distance between the items with the following formula:

$$Dist_{x,y} = \sqrt{(H_{s1} - H_{y1})^2 + (H_{s2} - H_{y2})^2 ... + (H_{sk} - H_{yk})^2}$$

        Arrange items based on the computed distance;

        Cluster the item *s* to the k- nearest neighbors until the *size_of_the_cluster = check_criterion*;

        }

Build ratings prediction model utilizing data mining algorithms for items based on *iVar*;

Predict *pR*;

Return *pR*;

The three main characteristics of the above AKNN algorithm (1) it provides only necessary data, (2) it builds ratings prediction model for every item (3) it may cluster one item into multiple groups based on the distance and computed similarity.

## 4.2 AKNN Based CF Algorithm

We extend our AKNN algorithm as AKNN based CF algorithm for the generation of the top-n recommendation list. The prediction process of our proposed recommendation model utilizes the AKNN based CF algorithm to provide top-n recommendation list to the active target user. Our presented collaborative filtering algorithm uses bi-objective optimization to generate optimized item recommendations. The objectives for the generation of similar items to the active target user are the popularity and the closeness. We adopt the scalar optimization methodology to convert multiple objectives into the single objective [41]. In other words, the scalar optimization converts the multiple objectives into a single aggregate objective function. The below algorithm illustrates the proposed AKNN based CF approach.

**Algorithm**: AKNN based CF algorithm

**Input**: *ATU* (Active Target User), *pR* (Predicted Ratings)

**Output:** *TnRec_List* (Top-N recommendation list)

*Rec_List* ← Null;

*Expert_User_Set* ← *discover_sim_exp_user*(*ATU*, *Exp_Users*);

for each *Expert_User* ∈ *Expert_User_Set* do,

   *item_set* ← {item:$I_{Expert\_User}$| item∉ $I_{ATU}$}

   *closeness*$_{ATU, Expert\_User}$← *max*(*compute_sim_NN*($NN_{ATU}$, *item_set*))

   *aggre_set*[*Expert_User*] ← *computeagg*(*item_set*$_{ATU, Expert\_User}$ ,*closeness*$_{ATU, Expert\_User}$)

end for

*Rec_List*$_{ATU}$← *Generate_Rec*(*ATU*, *aggre_set*);

*TnRec_List*← *Sort*(*Rec_List*$_{ATU}$);

Return *TnRec_List*;

The inputs of the above algorithm include the Active Target User data and the ratings predicted by the proposed AKNN algorithm. The presented CF variant algorithm exploits the sum method to utilize the scalar optimization. After initializing the top-n recommendations list as null, the expert users with the higher similarity are determined for the active target user. Then items of the expert users are organized as the items set and these items are verified to be not in the rated items of the active target user. Then to have better precision, the closeness measure is used to compute the items with the nearest neighbors of the active target user. The items with the maximum closeness are added to the close items. Then the aggregate function is employed to aggregate the item set and closeness set. After aggregation process is completed, the recommendations are generated and finally, the generated recommendations are sorted as the top-n recommendation.

## 5 Experimental Evaluations and Discussions

In this section, the proposed recommendation algorithms are experimentally evaluated for the efficiency, performance and the recommendations effectiveness with the existing approaches. Experiments are carried out in Python 3.6.0 on Intel core i7-5500U @ 3.00 GHz system with 16 GB of memory running 64-bit Windows operating system. Apart from this, the domain-specific ontology is developed using Protégé 5.1.0, an open-source ontology editor by Stanford University.

### 5.1 MovieLens Dataset

MovieLens datasets are considered to be standard datasets in the evaluation of the rec-ommendation approaches. MovieLens dataset comprises of the users' ratings and appro-priate demographic information which is collected in a span time of seven months. The MovieLens ratings are collected on the 5-point rating scale and 5-star rating represents highly liked and 1-star rating represents most hated or disliked. The dataset consists of data from 943 users about 1682 movies with 100,000 ratings. We have used this dataset to evaluate our proposed recommendation approach. To reduce the complexity of classifi-cation to determine whether the movie is to be recommended or not, we had introduced a ratings based attribute to make the decision. When the ratings are from 3 to 5, the movie is consisted to be "recommended". When the rating is 1 or 2, the movie is considered to "Not to be recommended". This classification process avoids further transformation of the data to make recommendation decision. The dataset consists of users' demographic information such as age, gender, zip code, and occupation. Regarding the movies list of the dataset, it consists of movie genre (19 types of category), date of release and title of the movie.

### 5.2 Evaluation Metrics

The main aim of the conducted experiments is to evaluate the performance of the proposed recommendation algorithms for its movie recommendations. Experiments are conducted on MovieLens dataset. The presented domain specific ontology is exploited to make movies as recommendations. We use four evaluation metrics precision, recall, f-measure, and Accuracy to evaluate the generated recommendations.

#### 5.2.1 Precision

The commonly known positive predictive value is also known as precision. Precision is the percentage of recommended movies relevant to the user and it is defined as follows:

$$Precision = \frac{|Reco\_Movie(user) \cap Relevant\_Movie(user)|}{|Reco\_Movie(user)|}$$

Here, *user* represents the target user in the test data, *Reco_Movie(user)* is the list of recommended movies and *Relevant_Movie(user)* is the list of movies pertinent to the target user in the test set.

### 5.2.2 Recall

The percentage relevant movies that are recommended is known as recall. Recall is also known as sensitivity and it is defined as follows:

$$Recall = \frac{|Reco\_Movie(user) \cap Relevant\_Movie(user)|}{|Relevant\_Movie(user)|}$$

Here, *user* represents the target user in the test data, *Reco_Movie(user)* is the list of recommended movies and *Relevant_Movie(user)* is the list of movies pertinent to the target user in the test set.

### 5.2.3 F-Measure

The f-measure metric is the harmonic mean of recall and precision computed and is defined as:

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

### 5.2.4 Accuracy

Accuracy metric is a decision support metric, which is based on the selection made with highly matching items from the set of all available items. The binary operation of the prediction process estimates the matching scores of the item to the active target user as "good" or "bad". The accuracy rate is generally computed from the obtained recommendation using the following definition:

$$Accuracy = \frac{\sum True\,Positive + \sum True\,Negative}{\sum True\,Positive + \sum False\,Positive + \sum False\,Negative + \sum True\,Negative}$$

## 5.3 Results and Discussions

The main intention of this research work is to predict interesting products for the active target user in an efficient way. Making recommendations to the old users of the system is already studied in many existing works and has produced enough satisfying results. Hence we focus on the recommendation generation problem for the new users or cold-start problem. To address the sparsity problem, we have employed associative classification technique to deal the sparse data in a better manner. When the system is capable of handling the sparse data in the better way, the generated recommendations are found to be more efficient and relevant to user's interests. We have used associative classification algorithms such as CMAR, CBR, CPAR and FOIL for the generation of effective recommendations. We have also used J48 classifier for the purpose of experimental evaluation. The obtained experimental results are depicted in Table 1. The Figs. 5, 6, 7 and 8 portrays the comparisons of Precision, Recall, F-Measure, and Accuracy respectively. Table 2 presents the comparison of the execution time taken for the different ontologies used for the recommendation generation.

**Table 1** Experimental comparison of data mining algorithms over various ontologies

| Ontology | Algorithms | Precision | Recall | F-measure | Accuracy (%) |
|---|---|---|---|---|---|
| TONES public ontology | J48 | 0.6235 | 0.6227 | 0.6231 | 72.77 |
| | CBA | 0.6842 | 0.6451 | 0.6641 | 83.54 |
| | CMAR | 0.7573 | 0.6682 | 0.7100 | 87.34 |
| | FOIL | 0.6735 | 0.6087 | 0.6395 | 81.61 |
| | CPAR | 0.5429 | 0.5343 | 0.5386 | 46.39 |
| | AKNN | 0.7753 | 0.6794 | 0.7242 | 89.12 |
| Simplified domain specific ontology | J48 | 0.6442 | 0.6332 | 0.6387 | 75.34 |
| | CBA | 0.6947 | 0.6655 | 0.6798 | 86.94 |
| | CMAR | 0.7782 | 0.6789 | 0.7252 | 90.06 |
| | FOIL | 0.6837 | 0.6292 | 0.6553 | 84.78 |
| | CPAR | 0.5582 | 0.5549 | 0.5565 | 49.84 |
| | AKNN | 0.7861 | 0.6892 | 0.7345 | 91.56 |
| Knowledge-based domain specific ontology | J48 | 0.6636 | 0.6436 | 0.6534 | 77.49 |
| | CBA | 0.7044 | 0.6858 | 0.6950 | 88.85 |
| | CMAR | 0.7983 | 0.6887 | 0.7395 | 92.72 |
| | FOIL | 0.6941 | 0.6495 | 0.6711 | 87.14 |
| | CPAR | 0.5687 | 0.5745 | 0.5716 | 51.97 |
| | AKNN | 0.8012 | 0.7026 | 0.7487 | 93.87 |
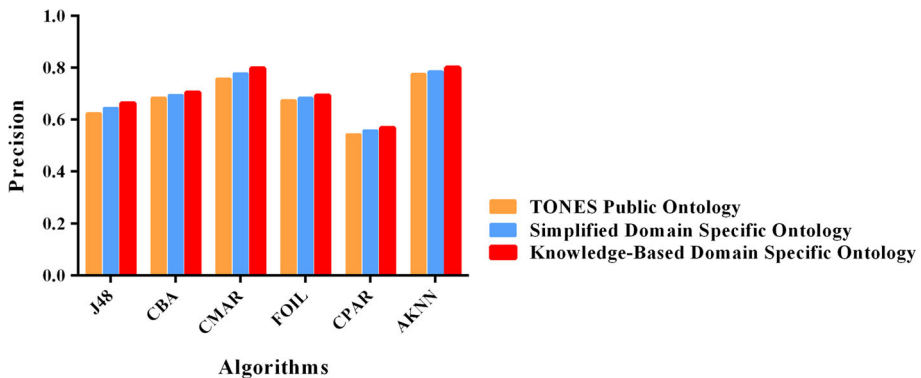


**Fig. 5** Comparison of precision

Regarding the conducted experiments, we have adopted 10-fold cross validation. For the associative classification algorithms, the confidence threshold is set to 80% and support threshold is fixed to 20%. Among the associative classification algorithms, CMAR performs best and followed by CBA and FOIL. CMAR has achieved the maximum accuracy of 92.72% with the knowledge based-based domain specific ontology. The achieved accuracy is to be considered as a significant result as the built ontology based recommendation models utilize very less information to generate more precise recommendations with highly sparse data. Compared to other algorithms, the time taken for the execution of the associative classification algorithms are bit high. CMAR is highly costly algorithm
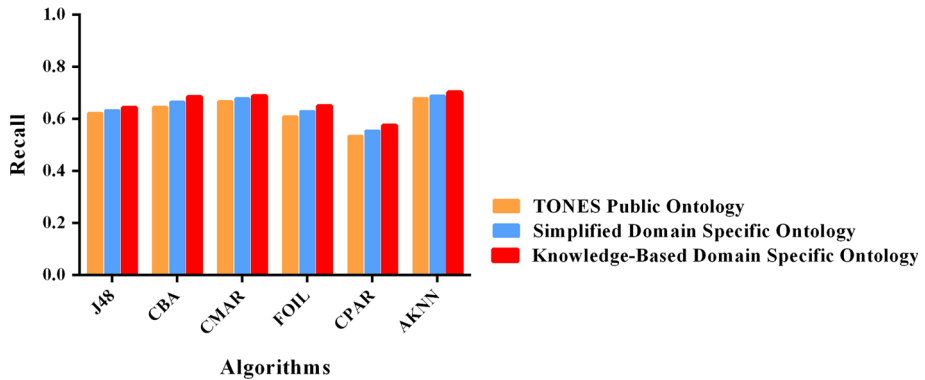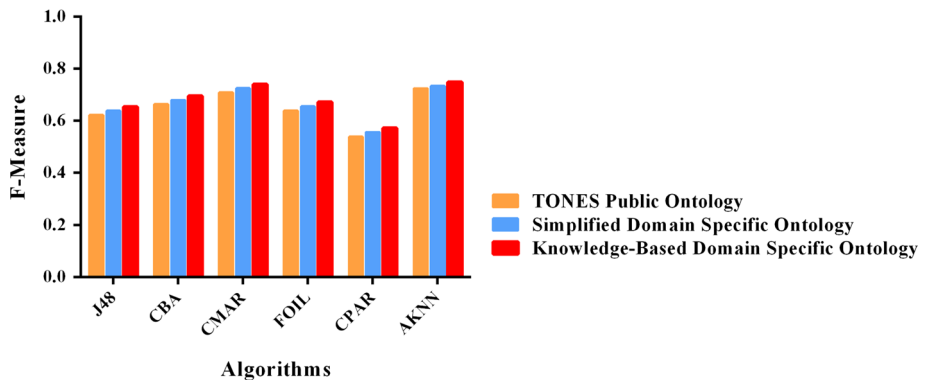
**Fig. 6** Comparison of recall
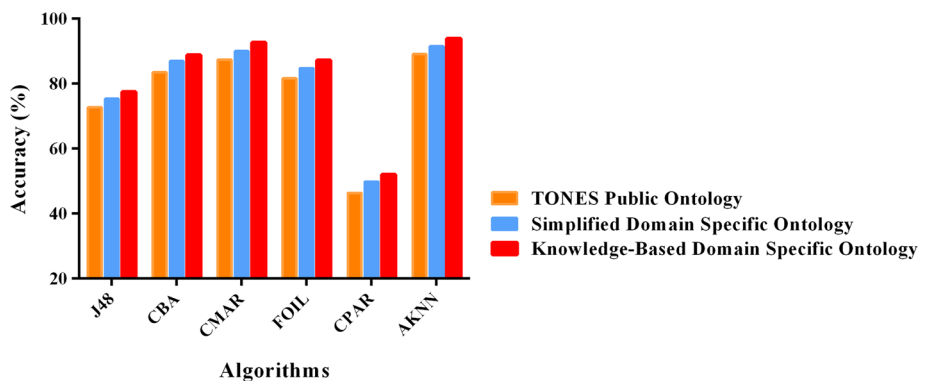


**Fig. 7** Comparison of F-Measure



**Fig. 8** Comparison of accuracy

compared to other algorithms. FOIL has produced a better recommendation in considerably lesser time. CMAR has taken almost 30 min with the knowledge-based domain specific ontology.

**Table 2** Execution time of data mining algorithms (in seconds)

| Algorithms | TONES public ontology | Simplified domain specific ontology | Knowledge-based domain specific ontology |
|---|---|---|---|
| J48 | 10.06 | 8.0 | 16.23 |
| CBA | 22.14 | 17.02 | 29.34 |
| CMAR | 1746.32 | 1519.09 | 1905.44 |
| FOIL | 3.52 | 1.42 | 6.52 |
| CPAR | 0.54 | 0.18 | 1.55 |
| AKNN | 75.35 | 75.12 | 75.54 |

CPAR algorithm is the worst performer among other algorithms taken for consideration. Though CMAR has taken higher time taken for execution, it doesn't affect our proposed recommendation framework, as our recommendation system uses offline processing module to make computations for the recommendations generation. CBA can be also used as the alternative to the CMAR algorithm as it has also produced comparatively equivalent results in a nominal execution time. CBA can be forcibly employed over CMAR for the recommendation problems that require updates in the frequent time intervals. For the conducted experiments, our proposed AKNN algorithm is an overall performer compared to other algorithms. In terms of Precision, Recall, and F-Measure, our AKNN algorithm has outperformed other algorithms. In terms of accuracy, AKNN has achieved 93.87, which is the maximum accuracy among all algorithms taken for consideration. The comparison of the time taken to execute the algorithms with reference to the various ontologies has been presented in Table 2. Among the algorithms considered for the prediction of user's preferences and the proposed AKNN has produced better recommendations. AKNN performs better with our proposed knowledge-based domain specific ontology for the generation of personalized recommendations. Concerning the execution time taken, it has taken an average of a minute, which is not a big issue with the model-based recommender systems. Compared to the advantages of our proposed AKNN algorithm, the time taken for execution is a negligible factor. The experimental results had proved the performance of our proposed AKNN algorithm over other algorithms with the highly sparse data taken for the recommendation generation.

## 6 Conclusions

Recommender Systems has attracted the focus of many researchers in the recent years. Yet still, there are several limitations and shortcomings such as sparsity, scalability, and cold-start are needed to be addressed. In this paper, we have developed a recommendation framework to address the above-said problems in as a collective way. We have developed an AKNN based collaborative filtering recommendation model with the utilization of domain specific ontology. To enhance the recommendations, we have used two different models namely, minion representation and prominent representation specifically for new and old users of the systems. The usage of two different models for the different types of users in the system had improved the performance of the recommendation and user satisfaction. The developed models correlate the item types with the profiles of the users which avoid the cold-start problem in a fair way. The developed knowledge-based domain specific ontology has performed as a better tool to annotate semantic metadata with user

historical information. Our proposed AKNN algorithm has performed better over CMAR, CPAR, J48, FOIL and CBA algorithms through all evaluation metrics. AKNN algorithm has produced good recommendations through addressing the scalability, sparsity and cold-start problems. With the help of MovieLens dataset, the newly developed, knowledge-based domain specific ontology has been validated. The AKNN algorithm has better adaptability with knowledge-based domain specific ontology and the conducted experiments had proved it. As the future work, we intend to extend our recommender system as a cross domain recommendation model to perform better on various application domains.

# References

1. Acha, B., Serrano, C., Fondón, I., & Gómez-Cía, T. (2013). Burn depth analysis using multidimensional scaling applied to psychophysical experiment data. *IEEE Transactions on Medical Imaging, 32,* 1111–1120.
2. Altinçay, H. (2007). Ensembling evidential k-nearest neighbor classifiers through multi-modal perturbation. *Applied Soft Computing, 7,* 1072–1083.
3. Argentini, A., & Blanzieri, E. (2010). About neighborhood counting measure metric and minimum risk metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32,* 763–765.
4. Barragáns-Martínez, A. B., Costa-Montenegro, E., Burguillo-Rial, J. C., Rey-López, M., Mikic-Fonte, F. A., & Peleteiro-Ramallo, A. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences, 180,* 4290–4311.
5. Bhatia, N., & Vandana, A. (2010). Survey of nearest neighbor techniques. *International Journal of Computer Science and Information Security, 8,* 302–305.
6. Billsus, D. & Pazzani, M. (1998). Learning collaborative information filters. In *15th international conference on machine learning* (pp. 46–54).
7. Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the fourteenth annual conference on uncertainty in artificial intelligence, Madison, Wisconsin, USA* (pp. 43–52).
8. Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning, 51,* 785–799.
9. Cevikalp, H., Triggs, B., & Polikar, R. (2008). Nearest hyperdisk methods for high-dimensional classification. In W. W. Cohen, A. McCallum & S. T. Roweis (Eds.), *ICML* (pp. 120–127). ACM.
10. Cheng, S. T., Chou, C. L., & Horng, G. J. (2013). The adaptive ontology-based personalized recommender system. *Wireless Personal Communications, 72*(4), 1801–1826.
11. Chien, J.-T., & Wu, C.-C. (2002). Discriminant waveletfaces and nearest feature classifiers for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24,* 1644–1649.
12. Cho, Y. H., Kim, J. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications, 23,* 329–342.
13. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems, August.*
14. Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13,* 21–27.
15. Derrac, J., Triguero, I., García, S., & Herrera, F. (2012). Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 42,* 1383–1397.
16. Díez, J., del Coz, J. J., Luaces, O., & Bahamonde, A. (2008). Clustering people according to their preference criteria. *Expert Systems with Applications, 34,* 1274–1284.

17. Domeniconi, C. & Yan, B. (2004). Nearest neighbor ensemble. ICPR (1) (pp. 228–231). IEEE Computer Society. ISBN: 0-7695-2128-2.

18. Eirinaki, M., Lampos, C., Paulakis, S., & Vazirgiannis, M. (2004). Web personalization integrating content semantics and navigational patterns. In *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management* (pp. 72–79). New York, NY, USA: ACM Press.

19. Gao, Q.-B., & Wang, Z.-Z. (2007). Center-based nearest neighbor classifier. *Pattern Recognition, 40,* 346–349.

20. Garcin, F., Dimitrakakis, C., & Faltings, B. (2013). Personalized news recommendation with context trees. In Q. Yang, I. King, Q. Li, P. Pu & G. Karypis (Eds.), *RecSys* (pp. 105–112). ACM.

21. Guo, H. (1997). Soap: Live recommendations through social agents. In *Proceedings of the fifth DELOS workshop on filtering and collaborative filtering, Budapest.*

22. Hernández-Rodríguez, S., Trinidad, J. F. M., & Carrasco-Ochoa, J. A. (2010). Fast k most similar neighbor classifier for mixed data (tree k-MSN). *Pattern Recognition, 43,* 873–886.

23. Hu, Q., Zhu, P., Yang, Y., & Yu, D. (2011). Large-margin nearest neighbor classifiers via sample weight learning. *Neurocomputing, 74,* 656–660.

24. Khribi, M. K., Jemni, M., & Nasraoui, O. (2009). Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. *Educational Technology and Society, 12,* 30–42.

25. Lee, C.-H., Kim, Y.-H., & Rhee, P.-K. (2001). Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems with Applications, 21,* 131–137.

26. Li, B., Chen, Y. W., & Chen, Y. Q. (2008). The Nearest Neighbor Algorithm of Local Probability Centers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 38,* 141–154.

27. Lim, E.-P. & Sun, A. (2005). Web mining—The ontology approach. In *Proceedings of the international advanced digital library conference, IADLC'2005.*

28. Liu, C.-L., & Nakagawa, M. (2001). Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition. *Pattern Recognition, 34,* 601–615.

29. Liu, H., Xing, H., & Zhang, F. (2012). Web personalized recommendation algorithm incorporated with user interest change. *Journal of Computational Information Systems, 8,* 1383–1390.

30. Lucas, J. P., Segrera, S., & García, M. N. M. (2012). Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems. *Expert Systems with Applications, 39,* 1273–1283.

31. Majumdar, A., & Ward, R. K. (2010). Robust Classifiers for Data Reduced via Random Projections. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 40,* 1359–1371.

32. Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2001). Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd international workshop on Web information and data management (WIDM '01)* (pp. 9–15). New York, NY, USA: ACM.

33. Moreno, M. N., Segrera, S., Batista, V. F. L., Vicente, M. D. M., & Sánchez, A. L. (2016). Web mining based framework for solving usual problems in recommender systems. A case study for movies' recommendation. *Neurocomputing, 176,* 72–80.

34. Nasraoui, O. & Petenes, C. (2003). An intelligent web recommendation engine based on fuzzy approximate reasoning. In *Proceedings of the IEEE international conference on fuzzy systems—Special track on fuzzy logic and the internet.*

35. Otebolaku, A. M., & Andrade, M. T. (2016). Context-aware personalization using neighborhood-based context similarity. *Wireless Personal Communications*, *94,* 1–24.

36. Parthasarathy, G., & Chatterji, B. N. (1990). A class of new KNN methods for low sample problems. *IEEE Transactions on Systems, Man, and Cybernetics, 20,* 715–718.

37. Ravi, L., & Vairavasundaram, S. (2016). A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. *Computational Intelligence and Neuroscience, 2016,* 1291358:1–1291358:28.

38. Sáez, J. A., Luengo, J., & Herrera, F. (2013). Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. *Pattern Recognition, 46,* 355–364.

39. Sarwar, B. M., Karypis, G., Konstan, J. A., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *World Wide Web* (pp. 285–295).

40. Schafer, B. J., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery, 5,* 115–153.

41. Soloklo, H. N., & Farsangi, M. M. (2013). Multi-objective weighted sum approach model reduction by Routh-Pade approximation using harmony search. *Turkish Journal of Electrical Engineering and Computer Sciences, 21*(Sup. 2), 2283–2293.

42. Triguero, I., García, S., & Herrera, F. (2011). Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification. *Pattern Recognition, 44,* 901–916.

43. Vairavasundaram, S., Varadharajan, V., Vairavasundaram, I., & Ravi, L. (2015). Data mining-based tag recommendation system: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5,* 87–112.
44. Vincent, P. & Bengio, Y. (2001). K-local hyperplane and convex distance nearest neighbor algorithms. In T. G. Dietterich, S. Becker & Z. Ghahramani (Eds.), *NIPS* (pp. 985–992). MIT Press.
45. Wang, F.-H., & Shao, H.-M. (2004). Effective personalized recommendation based on time-framed navigation clustering and association mining. *Expert Systems with Applications, 27,* 365–377.
46. Wang, Z., Yu, Z., Zhou, X., Chen, C., & Guo, B. (2016). Towards context-aware mobile web browsing. *Wireless Personal Communications, 91,* 187–203.
47. Xu, J. A. & Araki, K. (2006). A SVM-based personal recommendation system for TV programs. In *Proceedings of the international conference on multi-media modeling conference, Beijing, China* (pp. 401–404).
48. Yang, J.-M., Yu, P.-T., & Kuo, B.-C. (2010). A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. *IEEE Transactions on Geoscience and Remote Sensing, 48,* 1279–1293.
49. Zheng, W., Zhao, L., & Zou, C. (2004). Locally nearest neighbor classifiers for pattern classification. *Pattern Recognition, 37,* 1307–1309.
50. Zhou, B., Hui, S., & Chang, K. (2004). An intelligent recommender system using sequential web access patterns. In *IEEE conference on cybernetics and intelligent systems* (pp. 393–398).
51. Zhou, Z.-H., & Yu, Y. (2005). Ensembling local learners Through Multimodal perturbation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 35,* 725–735.

**V. Subramaniyaswamy** received the B.E. degree in Computer Science and Engineering and M.Tech. degree in Information Technology from Bharathidasan University, India and Sathyabama Universiy, India respectively. He received his Ph.D degree from Anna University, India and currently, working as a Senior Assistant Professor in the School of Computing, SASTRA University, India. His research interests include Data Mining, Recommender Systems, and Big Data Analytics.



**R. Logesh** received the B.Tech. degree in Computer Science and Engineering and M.Tech. degree in Networking from Pondicherry University, India. Currently, he is a Junior Research Fellow and pursuing the Ph.D. degree in the School of Computing, SASTRA University, India. His research interests include Recommender Systems, Information Retrieval, and Human–Computer Interaction.