Soft Comput
DOI 10.1007/s00500-014-1475-6

METHODOLOGIES AND APPLICATION

A multi-criteria recommendation system using dimensionality reduction and Neuro-Fuzzy techniques

Mehrbakhsh Nilashi · Othman Bin Ibrahim · Norafida Ithnin · Rozana Zakaria

Abstract Multi-criteria collaborative filtering (MC-CF) presents a possibility to provide accurate recommendations by considering the user preferences in multiple aspects of items. However, scalability and sparsity are two main problems in MC-CF which this paper attempts to solve them using dimensionality reduction and Neuro-Fuzzy techniques. Considering the user behavior about items' features which is frequently vague, imprecise and subjective, we solve the sparsity problem using Neuro-Fuzzy technique. For the scalability problem, higher order singular value decomposition along with supervised learning (classification) methods is used. Thus, the objective of this paper is to propose a new recommendation model to improve the recommendation quality and predictive accuracy of MC-CF and solve the scalability and alleviate the sparsity problems in the MC-CF. The experimental results of applying these approaches on Yahoo!Movies and TripAdvisor datasets with several comparisons are presented to show the enhancement of MC-CF recommendation quality and predictive accuracy. The experimental results demonstrate that SVM dominates the K-NN and FBNN in improving the MC-CF predictive accuracy evaluated by most broadly popular measurement metrics, F1 and mean absolute error. In addition, the experimental results also demonstrate that the combination of Neuro-Fuzzy and dimensionality reduction techniques remarkably improves the recommendation quality and predictive accuracy of MC-CF in relation to the previous recommendation techniques based on multi-criteria ratings.

Keywords Adaptive Neuro-Fuzzy inference systems · Higher order singular value decomposition · Multi-criteria collaborative filtering · Predictive accuracy · Classification

1 Introduction

Recommender systems assist users in finding relevant information by actively suggesting new content to them which is pooled from a whole community of users, thus relieving them of the need to browse or search at all. In this sense, they bridge the gap between searching and sharing technologies. They are techniques and tools for making personalized decisions in suggesting items to a user by considering user's preferences. The suggestions provided are aimed to support users in various decision-making processes (Jannach 2008; Jannach et al. 2010). Technically, recommender systems have their origins in different fields such as information retrieval, text classification, machine learning and decision support systems. A recommender system is used to identify sets of items that are likely to be of interest to a certain user, exploiting a variety of information sources related to both the user and the content items (Adomavicius and Tuzhilin 2005). In contrast to information filtering, recommender systems actively predict which items the user might be interested in and add them to the information flowing to the user, whereas information filtering technology is aimed at removing items from the information stream (Hanani et al. 2001). In addition, rec-

Communicated by V. Loia.

M. Nilashi (✉) · O. B. Ibrahim (✉) · N. Ithnin
Faculty of Computing, Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia
e-mail: nilashidotnet@hotmail.com

O. B. Ibrahim
e-mail: othmanibrahim@utm.my

N. Ithnin
e-mail: afida@utm.my

R. Zakaria
Construction Research Alliance, Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia
e-mail: rozana@utm.my

Published online: 21 October 2014

ommender system researches have been conducted in both of the theoretical contribution and with powerful emphasis on practical application and aim at improving commercial recommender systems.

One of the most successful algorithms with this type of information for item recommendation is Collaborative Filtering (CF) (Cechinel et al. 2013; Bobadilla et al. 2010, Bobadilla et al. 2011, 2012; Tsai and Hung 2012) which has been implemented in services provided by corporations such as Netflix, TiVo and Amazon (Linden et al. 2003). CF aims at recommending items to a certain user based on her past actions (purchase of a certain product, consumption of certain music tracks, explicit rating of certain items) and past actions of other, similar users. To be more particular, people utilize CF techniques for discovering new products they might like, getting recommendation on particular products and linking to other customers with same references. CF is the process of evaluating information using the opinion of other people. The ratings provided by users for items are the key input to CF recommender systems. Amazon is a useful example of a popular collaborative recommender system. Amazon supplements its typical search interface with a recommender system which gives the user a number of complimentary item discovery options. CF algorithms can be divided into two categories: memory-based algorithms and model-based algorithms (Adomavicius and Tuzhilin 2005; Deshpande and Karypis 2004).

Memory-based (also called neighborhood-based) algorithms are also known as lazy recommendation algorithms, because they defer the actual computational effort of predicting a user's interest in an item to the moment a user requests a set of recommendations. The training phase of a memory-based algorithm consists of simply storing all the user ratings into memory. The idea of memory-based methods is that the rating predictions for a user directly depend on his/her similar users' ratings on similar items. Since the entire rating database is kept in memory, new ratings can immediately be taken into account as they become available (Symeonidis et al. 2008a). The major drawback with memory-based CF techniques is as already noted that they tend to scale very poorly, larger rating databases require more memory and more calculations which slow down the recommendation process. There are two variants of memory-based recommendation and both are based on the K-Nearest Neighbor (K-NN) algorithm from the field of machine learning: user-based filtering (Breese et al. 1998; Herlocker et al. 1999) and item-based filtering (Sarwar et al. 2001). User-based CF has been the most popular and commonly used (memory-based) CF strategy (Konstan et al. 1997). It is based on the premise that similar users will like similar items. A user profile is collected and maintained for each user which records the items that he has consumed over time, and usually a corresponding set of ratings that judge how much he liked or disliked each item. In this manner, a model of the user's preferences for different types of items is constructed. Sometimes other types of information pertaining to the user, such as demographical information (Pazzani 1999) may also be collected in the user profile. One of the core challenges for user-based CF is the accurate identification of similarities between users based on their shared preferences.

Item-based CF was first proposed by (Sarwar et al. 2001) as an alternative style of CF that avoids the scalability bottleneck associated with the traditional user-based algorithm. The bottleneck arises from the search for neighbors in a population of users that is continuously growing. The authors argued that in an electronic commerce environment the set of items is often more static than the set of users which changes quite often. In item-based CF, similarities are calculated between items rather than between users, the intuition being that a user will be interested in items which are similar to items he has liked in the past. Two of the most popular approaches to computing similarities between users and items are the Pearson correlation coefficient and cosine-based coefficients. However, one of the problems in the recommender systems especially CF is known as the sparsity problem. Thus, these approaches make poor computation similarity when rating information is insufficient and with considering this problem, system produces the poor recommendation (Kim et al. 2010; Park and Chang 2009; Jeong et al. 2009). Furthermore, memory-based CF approaches suffer from the scalability problem. Therefore, scaling up these systems on real datasets is one of the main challenges that many studies have been provided to overcome it (Tsai and Hung 2012; Chen et al. 2009; Zhang and Chang 2006).

Compared with memory-based algorithms, model-based algorithms usually scale better in terms of their resource requirements (memory and computing time) and do not require keeping actual user profiles for predictions (Georgiou and Tsapatsoulis 2010; Adomavicius and Tuzhilin 2005; Sarwar et al. 2001). Model-based methods, such as Bayesian networks and clustering models (Bilge and Polat 2013; Li et al. 2007), address the problem from a probabilistic perspective to find the best item for a given user profile, and need to only keep the resulting model in memory while the algorithm is running. Model-based CF, for example, the work of Breese et al. (1998), can often offer significant advantages over memory-based algorithms in terms of efficiency but have not offered the same level of accuracy. It adopts an eager learning strategy, taking a probabilistic approach to predicting or recommending content, where a model of the data, i.e., the users, items and their ratings for those items, is pre-computed (Rennie and Srebro 2005; Sarwar et al. 2000; Breese et al. 1998). Indeed Breese et al. (1998) found that their model-based algorithms were four times faster than their memory-based algorithms at generating recommendations

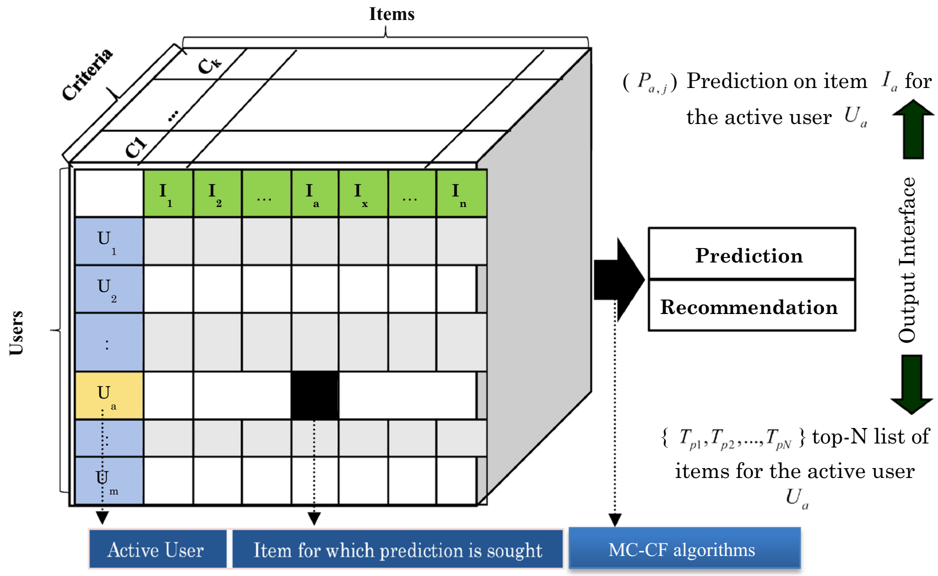


Fig. 1 The MC-CF processes

in terms of runtime. Some researchers have suggested that model-based CF can also produce better predictive accuracy than memory-based CF, using more sophisticated techniques such as matrix factorization and dimensionality reduction (Luo et al. 2012; Ahmad and Khokhar 2007; Rennie and Srebro 2005).

2 Multi-criteria collaborative filtering (MC-CF)

Adomavicius and Kwon (2007) stated that single-rating CF recommenders are indicated as systems that attempt to estimate a rating function R that has the form user $\times$ item $\rightarrow$ R_0 for predicting a rating for any given user-item pair. R_0 is totally ordered set, typically composed of real-valued numbers inside a certain range. They further disclosed that, in multi-criteria recommender systems, in comparison, the rating function R_0 gets the form user $\times$ item $\rightarrow$ $R_0 \times \mathrm{R}_1 \times \cdots \times \mathrm{R}_k$. Therefore, an overall rating R_0 must be predicted in addition to the k additional criteria ratings. The overall rating shows how well the user likes the item overall, and the criteria ratings provide the insight and explain which aspects of the item he or she likes.

Multi-criteria CF (MC-CF) recommender systems predict the overall rating for an item based on past ratings regarding both the item overall and individual criteria, and recommend to the user the item with the best overall score. Thus, the algorithm for a MC-CF recommender system can be extended from a single-rating recommender system.

In MC-CF problem, there are m users, n items and $k$criteria in addition to the overall rating. Users have provided a number of explicit ratings for items; a general rating R_0must be predicted in addition to k additional criteria ratings $(R_1, \ldots, R_k)$. It can be configured to push new items to users in two ways, either by producing a Top-N list of recommendations for a given target, or by predicting the target user's likely utility (or rating) for a particular unseen item. We will refer to these as the recommendation task and the rating prediction task in MC-CF, respectively. Figure 1 demonstrates the MC-CF problem in case of prediction and recommendation tasks for an active user U_a and item I_a.

Recommendation is a list of N products, $TP = \{T_{p1},\ T_{p2}, \ldots, T_{pN}\}$, that the active user will like the most. The recommended list usually consists of the products not already purchased by the active customer. This output interface of MC-CF algorithms is also known as Top-N recommendation. Figure 1 shows the schematic diagram of the MC-CF process. MC-CF algorithms represent the entire $m \times n \times k$ user-item criteria data as a tensor of ratings, $\underline{A}$. Each entry $a_{i,j}$ in tensor $\underline{A}$ as shown in Fig. 1 represents the preference score (ratings) of the ith user on the jth item as overall preference in addition to criteria ratings in the 3rd dimension. Each overall and criteria rating is within a numerical scale and it can as well be 0, indicating that the user has not yet rated that product.

2.1 The problems and our contributions

In the context of personalization applications, traditional single-rating CF has been highly successful; however, the research area regarding the CF with multi-criteria ratings for items has been rarely touched and fairly this issue is largely unexplored. According to Adomavicius and Kwon (2007), the problem of multi-criteria recommendations with a single and overall rating is still considered an optimization problem.

In the MC-CF, users' behavior on features of items is imprecise, subjective and vague that consequently leads to uncertainty in reasoning on perception of user on items' features. Concerning the user behavior and perception, for example, interest, the uncertainty is connected to how user interest precisely can be represented and measured. Thus, this problem in MC-CF has to be taken into consideration. We believe that fuzzy set can better solve this problem instead of relying solely on clustering methods, dimensionality reduction techniques, and regression approaches. Fuzzy methods can better predict the user preference based on items' features in MC-CF. That means, they can better solve the sparsity problem. In addition, in practical applications and situations, customers are interested to rate the items or to express their preferences in linguistic terms, such as {low interest}, {high interest} or {no interest} for the feature of items. Therefore, for MC-CF, fuzzy set is more appropriate in human linguistic reasoning with imprecise concepts instead of crisp approaches. In fuzzy approaches, linguistic terms are more suitable and accurate rather than numerical values in assessing qualitative information, which is usually related to human perceptions, opinions and tastes. With this consideration, predicting the user' preferences can be more accurate.

Furthermore, another issue in MC-CF is scalability problem. It is obvious that the MC-CF recommender systems deal with high dimensional data and therefore, applying the traditional dimensionality reduction techniques such as Singular value Decomposition (SVD) cannot appropriately solve the scalability problem in these systems. Traditional dimensionality reduction methods can be applied to two-dimensional datasets and they are failing on high-dimensional datasets. Thus, overcoming the scalability problem induced from high-dimensional data is one of the important issues that in the prior researches rarely have been explored.

Moreover, in this paper, we look at the following key questions:

- How the predictive accuracy of MC-CF can be improved using Neuro-Fuzzy and dimensionality reduction techniques?
- Does the robustness of machine learning-based classifiers impact on the MC-CF predictive accuracy and recommendation quality?

To answer above questions, in this research, we apply techniques, Adaptive Neuro-Fuzzy Inference Systems (ANFIS) with subtractive clustering, Higher Order SVD (HOSVD) as a multi-linear dimensionality reduction technique and classification methods, to alleviate sparsity, improve scalability and recommendation quality and predictive accuracy of MC-CF recommender systems. The HOSVD and classifiers incorporated in the recommendation model are considered as dimensionality reduction techniques. Using HOSVD, dimensionality reduction is performed converting data of very high dimensionality into data of much lower dimensionality. Using data in the lower dimensions, the high-quality clusters are obtained. The aim of clustering by this method is to provide a model of similar users and items for extracting fuzzy rules with high accuracy using Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and then predicting overall ratings in user- and item-based models based on the criteria ratings. In addition, in the proposed model, ANFIS aims to extract knowledge (fuzzy rules) from user rating in multi aspect to be used in rating prediction task. The extracted rules are employed for predicting unknown ratings and also revealing real level of user preferences on items' features (criteria). ANFIS provides a flexible structure of defined problem that is suitable for generating stipulated input–output pairs using a set of induced fuzzy IF–THEN rules with appropriate and varied Membership Functions (MFs). The produced Fuzzy Inference System (FIS) is served to predict user overall with proper training. The principle elements of this model are a fuzzy set, a neural network and data clustering. Moreover, non-stochastic uncertainty emerged from vagueness and imprecision is handled using ANFIS where the MFs produced by ANFIS is used for representation and reasoning users' behavior for providing rating based on their perception about item features. The MFs formed by ANFIS are continues which are more accurate in representing the features of items and user feedbacks. Thus, to prevent the problem of overfitting discussed in the previous researches (Jannach et al. 2012; Sen et al. 2009), checking data are used to minimize and solve an overheating problem in the training data.

In addition to the applying HOSVD and ANFIS methods, in this research, we examine the performance of several powerful classification and regression methods on the recommendation quality and predictive accuracy of MC-CF. K-NN classifier, Feedforward Backpropagation Neural Network (FBNN) and Support Vector Machine (SVM) as classification methods and *nu*- SVR and epsilon-SVR are applied to regression task, and then the effectiveness of these methods is compared to recommendation quality and predictive accuracy of MC-CF.

In this paper, in-depth experimental evaluations are performed on the real-world datasets of user rating in multi aspects provided by Yahoo! Movies network and TripAdvisor website.

3 Related works

In this section, we give a brief overview of fuzzy methods, Machine learning-based classifiers and multi-linear dimensionality reduction techniques that have been used in recommender systems. However, after discussing the related works using fuzzy approaches, we will have discussion on the previous works which belong to the MC-CF as we are improving the accuracy of these systems. In addition, according to Adomavicius et al. 2011 that introduced classification for multi-criteria recommender systems, we also focus on multi-criteria rating recommenders, in which users are allowed to specify their preferences (ratings) for individual products along different dimensions.

3.1 Fuzzy logic in recommender systems

The fuzzy logic field has grown considerably in a number of applications across a wide variety of domains like in the semantic music recommendation system (Lesaffre and Leman 2007), movie recommendation (Nilashi et al. 2014a) and product recommendations (Cao and Li 2007; Henrik et al. 2006). Castellano et al. (2007) developed a Neuro-Fuzzy strategy combined with soft computing approaches for recommending URLs to the active users. They used fuzzy clustering for creating a user profile considering the similar browsing behavior. Campos et al. (2008) proposed a model by combining Bayesian network for governing the relationships between the users and fuzzy set theory for presenting the vagueness in the description of users' ratings. A conceptual framework based on fuzzy logic-based was proposed by Yager (2003) to represent and then justify the recommendation rules. In the proposed framework, an internal description of the items was used that relied solely on the preferences of the active user. Carbo and Molina (2004) developed an algorithm based on CF where ratings and recommendations were considered as linguistic labels by using fuzzy sets. A model proposed by Pinto et al. (2012) combined fuzzy numbers, product positioning (from marketing theory) and item-based CF.

3.2 Machine learning-based classifiers in recommender systems

Machine learning-based classifiers are considered as supervised dimensionality reduction techniques in that there is a clear objective of discovering a reduced representation of the data where the classes are well separated (Villalba and Cunningham 2007). They have been widely used for personalized online services and item recommendations (Park et al. 2012). A more elaborate, and far more popular, instance-based classifier is the K-NN. It is one of the most common approaches to CF and therefore to designing recommender systems (Amatriain et al. 2011; Park et al. 2012). SVMs are another popular classification technique backing in statistical theory. It has been applied with great success in various text applications like text classification (Yang and Liu 1999; Joachims 1998a; Drucker and Shahrary 2001). Grcar et al. (2006) confronted the K-NN algorithm with SVM in the CF framework. They found that K-NN is dominant on datasets with relatively low sparsity. In addition, they demonstrated that on dataset with high to extremely high level of sparsity, K-NN starts failing as it is unable to form reliable neighborhoods. In such case, it is best to use a model-based approach, such as SVM classifier or SVM regression. Another strong argument for using the SVM approaches on highly sparse data is the ability to predict more ratings than with the variants of the memory-based approach. Joachims (1998b) demonstrated that on a Reuters-21578 dataset, SVMs performed better than K-NN (86.4 % accuracy vs. 82.6 %). Xia et al. (2006) proposed a heuristic method based on Smoothing SVM (SSVM) method from Lee and Mangasarian (2001) to overcome the problem caused by the sparsity of user-item matrix. They compared the heuristic method with item-based CF proposed by Zhang and Iyengar (2002) and user-based CF proposed by Breese et al. (1998).

NNs also have been successful in recommendation system implementation (Postorino and Sarne 2011; Gong and Ye 2009; Gao and Wu 2009). Lee et al. (2002a) proposed a recommender system which combines CF with Self-Organizing Map (SOM) NN. Christakou et al. (2007) proposed a recommendation system based on content and CF for recommendations concerning movies. The content filtering part of the system was based on trained NNs representing individual user preferences. They evaluated the hybrid system on the MovieLens data. Postorino and Sarne (2011) proposed a NN hybrid recommender system which was able to provide customers, associated with XML-based personal agents within a multi-agent system called MARF, with suggestions about flights' purchases. For solving data sparsity for CF, a personalized recommendation approach based on Backpropagation NNs (BPNNs) and item-based CF was presented by Gong and Ye (2009). In the method, they used the BPNNs to fill the null values in user-item matrix of ratings and item-based CF to form nearest neighborhood.

3.3 Multi-linear dimensionality reduction techniques in recommender systems

In the area of personalized web search, Sun et al. (2005) proposed Cube singular value decomposition (CubeSVD) to improve Web Search. Based on their CubeSVD analysis, which also used HOSVD technique, web search activities are carried out more efficiently. They evaluated the method on MSN search engine data. In the field of recommender systems, some recommendation models, which use third-order tensors for recommending music, objects and tags, have been proposed. Recommender models, using HOSVD for dimension reduction, have been proposed for recommending personalized music (Symeonidis et al. 2008b), movie (Nilashi et al. 2014b) and tags (Symeonidis et al. 2008c). Symenonidis et al. 2008c introduced a recommender based on HOSVD where each tagging activity for a given item from a particular user is represented by value 1 in the initial tensor, all other cases are represented with 0. Xu et al. (2006) used HOSVD to provide item recommendations. Further, their work was compared with a standard CF algorithm, without focusing on tag recommendations. Leginus and Zemaitis (2011) utilized clustering techniques for reducing tag space that improves the quality of recommendations and the execution time of the factorization and decreases the memory demands. Their proposed method was adaptable with HOSVD. They also introduced a heuristic method to speed up parameter tuning process for HOSVD recommenders.

3.4 Related work of multi-criteria rating recommender systems

In case of MC-CF, few researches has been conducted to develop the similarity calculation of the traditional memory-based CF approach to investigate multi-criteria rating (Tang and McCalla 2009; Manouselis and Costopoulou 2007; Adomavicius and Kwon 2007) that the similarities between users are estimated through aggregating traditional similarities from individual criteria or applying multidimensional distance metrics. To develop the idea of Adomavicius and Kwon (2007), Sahoo et al. (2006, 2011) extended the Flexible Mixture Model (FMM) developed by Si and Jin (2003) to multi-criteria recommender systems. The aim of their paper was to exploit context information about the user as well as multi-criteria ratings in the recommendation process. Li et al. (2008) presented a multi-criteria rating approach to improve personalized services in mobile commerce using Multi-linear SVD (MSVD). The aim of their paper was to exploit context information about the user as well as multi-criteria ratings in the recommendation process. Liu et al. (2011) presented a multi-criteria recommendation approach which is based on the clustering of users. Their idea is that for each user one of the criteria is "dominant" and users are grouped according to their criteria preferences. They applied linear least squares regression, assigning each user to one cluster, and evaluating different schemes for the generation of predictions. They evaluated methods on hotel domain dataset with five criteria, Value, Location, Rooms, Service, and Cleanliness. Jannach et al. (2012) further developed the accuracy of MC-CF by proposing a method using SVR for automatically detecting the existing relationships between detailed item ratings and the overall ratings. In addition, the learning process of SV regression models was per item and user and lastly combined the individual predictions in a weighted approach.

According to the literature on multi-criteria rating recommender systems, it is important to conduct this research as the research area regarding the CF with multi-criteria ratings for items has been rarely touched. With regarding to the research findings by Adomavicius and Kwon (2007), MC-CF recommender systems have many advantages over the pure CF and they are more accurate in items' prediction; thus, it is worthy to improve the accuracy and efficiency of these systems with sophisticated machine learning techniques. Accordingly, in this research, we propose a recommendation model for MC-CF using classification methods, HOSVD and ANFIS and investigate the impact of classifiers' robustness on MC-CF recommendation accuracy and quality.

Hence, in comparison with research efforts found in the literature for MC-CF, our work has the following differences. In this research:

- A hybrid recommendation model using dimensionality reduction (supervised and unsupervised) and ANFIS techniques is proposed for improving the predictive accuracy and recommendation quality of MC-CF.
- The impact of classifiers' robustness is examined on MC-CF recommendation accuracy and quality.

The remainder of this paper is organized as follows: ANFIS, subtractive clustering, HOSVD, classification and regression methods are introduced in separate subsections in the Sect. 4. Sect. 5 provides an overview of research method. Sect. 6 presents the result and discussion and finally, conclusions and future work in Sect. 7.

4 Preliminaries

The theoretical foundations of the methods used in this study are given in the following subsections.

4.1 ANFIS

Soft computing techniques are known for their efficiency in dealing with complicated problems when conventional analytical methods are infeasible or too expensive, with only sets of operational data available.

Fuzzy Logic (FL) and Fuzzy Inference Systems (FIS), first proposed by Zadeh (1965), provide a solution for making decisions based on vague, ambiguous, imprecise or missing data. FL represents models or knowledge using IF–THEN rules.

NNs learn system behavior using system input–output data. Neural networks have good generalization capabilities. The learning and generalization capabilities of neural networks enable it to more effectively address real-world problems. Thus, neural networks can solve many problems that are either unsolved or inefficiently solved by existing techniques, including fuzzy logic.

Both fuzzy logic and neural networks have been very successful in solving many real-world problems. However, both technologies have some limitations as well which have prevented them from providing efficient solutions for MC-CF problems. In fuzzy logic, it is usually difficult to determine the correct set of rules and membership functions from the users' preferences in MC-CF. Moreover, fine-tuning a fuzzy solution is even more difficult and takes longer. In neural networks, it is difficult to understand the "Black Box," i.e., it is incomplete compared to a fuzzy rule-based system description.

An appropriate combination of these two technologies (Neuro-Fuzzy) can effectively solve the problems of fuzzy logic and neural networks and, thus, can more effectively address the MC-CF problems. A Neuro-Fuzzy approach was used to take advantage of the neural network's ability to learn, and membership degrees and functions of fuzzy logic. The weights of the neural networks are mapped to fuzzy logic rules and member functions. Expressing the weights of the neural network by fuzzy rules also provides a better understanding of the "Black Box" and thus helps the better design of the neural network itself. Thus, while the learning of neural network is parameterized by the variation in input data, the learning of ANFIS is fixed by the rules and membership function values that we define. A Neuro-Fuzzy system is functionally equivalent to a FIS. A FIS mimics a human reasoning process by implementing fuzzy sets and approximate reasoning mechanism that uses numerical values instead of logical values (Nilashi and Ibrahim 2013b). A FIS requires a domain expert to define the MFs and determine the associated parameters in both the MFs, and the reasoning section. However, there is no standard for the knowledge acquisition process and thus the results may be different if a different knowledge engineer is at work in acquiring the knowledge from experts.

A Neuro-Fuzzy system can replace the knowledge acquisition process by humans using a training process with a set of input–output training dataset. Thus instead of depending on human experts the Neuro-Fuzzy system will determine the parameters associated with the Neuro-Fuzzy system through a training process, by minimizing an error criterion. A popular Neuro-Fuzzy system is called an ANFIS. ANFIS is a fuzzy system that uses artificial neural network theory to determine its properties (fuzzy sets and fuzzy rules) (Nilashi et al. 2011a, b; Cetişli and Barkana 2010; Buragohain and Mahanta 2008; Petrovic-Lazarevic et al. 2004; Sugeno 1985).

4.2 Subtractive clustering

The idea in Takagi–Sugeno–Kang (TSK) model, which is considered as a type of fuzzy model, is that each rule in a rule base indicates an area for a model, which can be linear (Jang 1993). The TSK rule structure in a basic shape is as follows:

$$\begin{aligned}&Iff(x_1\ is\ A_1, x_2\ is\ A_2, \ldots,\ x_k\ is\ A_k)\\&\quad then\quad y = g(x_1, x_2, \ldots)\end{aligned} \tag{1}$$

where sentences forming the condition are connected through the logical function f. The output y is obtained by g that is a function of the inputs x_i.

To establish an effective TSK model of a process, using subtractive clustering for generating clusters of data that can be candidates the linear area is constructive. The main goal of using subtractive clustering as a cluster analyzer is to partition the dataset into a number of homogeneous and natural subsets. The subtractive clustering method assumes that each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define the cluster center, based on the density of surrounding data points. Using it, the quantity of the calculation is in proportion to the number of data points and is foreign to the dimensions of the problem. However, while the actual cluster centers are not necessarily located at one of the data points, in most cases it is a good approximation, especially with the reduced computation this approach requires (Chiu 1994). In this method, a data point with the highest potential, which is a function of the distance measure, is considered as a cluster center. The data points that are close to the new cluster center are penalized to facilitate the emergence of new cluster centers (Bouchachia and Pedrycz 2006).

4.3 Higher Order SVD (HOSVD)

To represent and recognize high-dimensional data effectively, the dimensionality reduction is conducted on the original dataset for low-dimensional representation (Zhang and Ye 2011). Visualizing, comparing and decreasing processing time of data are the main advantages of dimensionality reduction techniques. SVD and HOSVD are two powerful techniques of the dimensionality reduction for matrix and tensor decomposition, respectively.

The SVD of a real matrix $A \in R^{n\times n}$ is a very powerful computation tool for solving many problems in numerical linear algebra. From the theoretical point of view, it is also used in numerical analysis as a decomposition that reveals important information about the matrix and the problem it is involved with. HOSVD proposed by Lathauwer at al. (2000) is a generalization of the SVD that can be applied on tensors. In many applications, involving tensor data the objective is to compute low-rank approximations of the data for modeling, information retrieval and explanatory purposes. These approximations are usually expressed in terms of tensor decompositions. In the following, we explain the tensor decomposition for HOSVD in third- and fourth-order tensors.

Definition 1 (*Unfolding*) The mode-n unfolding of tensor $\underline{A} \in R^{I_1\times I_2\times\cdots\times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers into columns of a matrix. More specifically, a tensor element $(i_{1,}i_2, \ldots, i_N)$ maps onto a matrix element (i_n, j), where

$$j = 1 + \sum_{p \neq n} (i_p - 1) j_p, \quad with$$

$$J_p = \begin{cases} 1, & if\, p = 1 \; or \; if \;\; p = 2 \; and \; n = 1, \\ \prod_{m \neq n}^{p-1} I_m & otherwise \end{cases} \tag{2}$$

For a tensor of order $N = 3$, we have 3 modes $n = 1, 2, 3$, and for a tensor of order $N = 4$ we have 4 modes and so on. The n-mode fibers are the columns of the n-mode unfolded matrix. The flattening of a tensor $\underline{A}$ in its n-mode is symbolized by the matrix $\mathbf{X}(n)$. For example, for a tensor of order $N = 3$, by fixing the n-th index to some value j, there exist three matrix unfolding as (Lathauwer 2004):

$$\begin{aligned} &\mathbf{Mode-1}: \; j = i_2 + (i_3 - 1) I_3, \\ &\mathbf{Mode-2}: \; j = i_3 + (i_1 - 1) I_1, \\ &\mathbf{Mode-3}: \; j = i_1 + (i_2 - 1) I_2. \end{aligned} \tag{3}$$

Definition 2 (*Frobenius-norm*) The Frobenius-norm of a tensor $\underline{A}$ of size $I \times J \times K$ is defined by Eq. (4).

$$\left\| \underline{A} \right\|_F = \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \left| t_{ijk} \right|^2 \right)^{1/2} \tag{4}$$

The Frobenius-norm can be interpreted as a measure for the "size" of the tensor. The square of this norm can be seen as the "energy" in the tensor.

Definition 3 (*Rank*) The rank of a tensor can be defined as the minimal number of terms when expressing a tensor as a sum of rank-one tensors. The second way to define a rank of a tensor is given by the dimension of the subspaces spanned by the different n-mode vectors. Given an order n tensor $\underline{A}$ we write

$$\text{Rank}(\underline{A}) = (r_1, \ldots, r_n) \; r_i = dim(span(A^{(i)}))) \tag{5}$$

where $\mathbf{A}(i)$ is the matricization of $\underline{A}$ along mode i. This is called the multi-linear rank of a tensor. For tensors in general the ranks r_i are different. The approximation problem of an order three tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3}$ is stated

$$\min_{\underline{B}} \left\| \underline{A} - \underline{B} \right\|, \qquad \text{Rank}\underline{B} = (r_1, r_2, r_3) \tag{6}$$

Assuming the rank constraint on $\underline{B}$, we can decompose $\underline{A} = (U, V, \mathbf{W}).\underline{S}$ where $U \in R^{I \times r_1}$, $V \in R^{J \times r_2}$, $W \in R^{K \times r_3}$ have full column rank and $\underline{S} \in R^{r_1 \times r_2 \times r_3}$.

Definition 4 Every complex $(I \times J \times K)$-tensor X can be written as the product $X = (U, V, W).\underline{S}$, where $U \in R^{I \times I}$, $V \in R^{J \times J}$, $W \in R^{K \times K}$, are orthogonal matrices, and the tensor $\underline{S} \in R^{I \times J \times K}$ is all-orthogonal and we have

$$\begin{aligned} &\left\| \underline{S}(1, :, :) \right\| \geq \left\| \underline{S}(2, :, :) \right\| \geq \cdots \geq 0, \\ &\left\| \underline{S}(:, 1, :) \right\| \geq \left\| \underline{S}(:, 2, :) \right\| \geq \cdots \geq 0, \\ &\left\| \underline{S}(:, :, 1) \right\| \geq \left\| \underline{S}(:, :, 2) \right\| \geq \cdots \geq 0, \end{aligned} \tag{7}$$

which are the 1-mode, 2-mode, and 3-mode singular values, also denoted $\sigma_i^{(1)}$, $\sigma_i^{(2)}$, $\sigma_i^{(3)}$. The tensor $\underline{S}$ in the HOSVD is in general full and not sparse or diagonal as Σ (a matrix contains of singular values) in the SVD of a matrix. But the all-orthogonality concept is still valid in matrix SVD. The HOSVD is an important result both for analysis and applications with tensors since it gives an ordering of the basis vectors in $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$. In fact truncating the HOSVD, i.e., taking the first r_1, r_2 and r_3 columns from $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$, respectively, and correspondingly truncating $\underline{S}$, will give a good approximation of $\underline{A}$.

Similarly, for the HOSVD represents a 4th-order tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3 \times I_4}$ as a product of another fourth-order tensor with four unitary matrices of sizes $I_j \times I_j$, respectively. In this case, the decomposition of the fourth-order tensor is given by

$$\begin{aligned} &\underline{A} = \underline{S} \times {}_1\mathbf{U}^{(1)} \times {}_2\mathbf{U}^{(2)} \times {}_3\mathbf{U}^{(3)} \times {}_4\mathbf{U}^{(4)} \\ &\Rightarrow A_{ijkl} = \sum_{m=1}^{I_1} \sum_{n=1}^{I_2} \sum_{p=1}^{I_3} \sum_{q=1}^{I_4} S_{mnpq} U_{im}^{(1)} U_{jn}^{(1)} U_{kp}^{(1)} U_{lq}^{(1)} \end{aligned} \tag{8}$$

The matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$ and $\mathbf{U}^{(4)}$ are the matrices containing the left singular vectors of the four matrices that one can obtain by attending the tensor $\underline{A}$ (Kolda and Bader 2009). In other words, $\mathbf{U}^{(n)}$ is obtained via the SVD of $\mathbf{X}^{(n)}$, the n-mode matrix unfolding of the tensor $\underline{A}$ is defined as:

$$X^{(n)} = U^{(n)} \times \Sigma^{(n)} \times V^{(n)T} \tag{9}$$

where $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are the left and right side matrices of singular vectors, respectively. The matrix $\Sigma^{(n)}$ represents the diagonal matrix containing the singular values of $\mathbf{X}^{(n)}$. Since $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$ and $\mathbf{U}^{(4)}$ are orthogonal, the core tensor $\underline{S}$ can be easily estimated via the following expression:

$$\underline{S} = \underline{A} \times {}_1U^{(1)T} \times {}_2U^{(2)^T} \times {}_3U^{(3)^T} \times {}_4U^{(4)^T}, \tag{10}$$

where $U^{(i)T}$ denotes the complex conjugate transpose of $U^{(i)}$. The core tensor $\underline{S}$ plays a role similar to that of the matrix of singular values Σ in the SVD. In fact, one can reduce the rank of the tensor by truncating the core tensor. However, the definition of rank for tensors is not as straightforward as for matrices. There are several definitions of "rank" (Kolda and Bader 2009). The truncated HOSVD is defined as a multi-rank approximation. Or symbolically, if rank $(\mathbf{D}^{(j)}) = I_j, \quad j = 1, 2, 3, 4$, then $\underline{A}$ has rank-(I_1, I_2, I_3, I_4). The truncated HOSVD consists of the representation of the 4th-order tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3 \times I_4}$ by the product of four unitary matrices $\tilde{U}^{(1)}$, $\tilde{U}^{(2)}$, $\tilde{U}^{(3)}$ and $\tilde{U}^{(4)}$ of sizes $I_j \times R_j$, $R_j < I_j$, respectively, and a small core tensor $\underline{S} \in R^{R_1 \times R_2 \times R_3 \times R_4}$.

$$\underline{A} \approx \underline{\tilde{A}} = \underline{S} \times {}_1\tilde{U}^{(1)} \times {}_2\tilde{U}^{(2)} \times {}_3\tilde{U}^{(3)} \times {}_4\tilde{U}^{(4)} \tag{11}$$

Table 1 Computational cost for main steps in HOSVD

Step	N-dim
Unfolding the tensor $\underline{T}$	$O(I_1 I_2 \ldots I_N)$
Constructing $X^{(n)} X^{(n)^T}$	$O(I^2 I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_N)$
Determining $X^{(n)} X^{(n)^T}$ to obtain $U^{(n)}$	$O(I^3)$
Contract tensor $\underline{T}$ with matrices $U^{(n)}$'s to get tensor $\underline{S}$	$O(I^2 I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_N)$

We denote $\tilde{\underline{A}}$ the rank-reduced approximation of tensor $\underline{A}$. Clearly, the matrices $\tilde{U}^{(n)}$, $n = 1, 2, 3, 4$ in Eq. (11) contain the first R_n left singular vectors of the unfolded matrix $\mathbf{D}^{(n)}$.

With above explanations, we can define HOSVD in the following Algorithm (Kolda and Bader 2009):

Algorithm 1:

Procedure HOSVD (Input: Tensor $\underline{A}$ **)**
Begin
For [n=1 to d] **Do**
  Begin
    $\mathbf{U}^{n} \leftarrow I_n$ {Leading left singular vector of $\mathbf{X}^{(n)}$};
    $n := n+1$;
  End
$\underline{S} \leftarrow \underline{A} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \times \ldots \times_d \mathbf{U}^{(d)^T}$;
Return [Core tensor $\underline{S}$ and matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(d)}$
];
End

For HOSVD the computation cost is calculated as shown in Table 1.

In this research, we use HOSVD for MC-CF to decompose a tensor of users' ratings for obtaining the data approximation. Due to the nature of experimental dataset, we perform HOSVD on third-order tensor. However, according to the above introduction of tensor decomposition, we can extend it to be applied on fourth-order tensors and higher order tensors. This can be the main advantage for MC-CF as it is a flexible and effective approach for these systems where other traditional machine learning techniques are failing. It should be noted that using HOSVD the computation time for decomposition procedure is high when the tensor order is increased. However, it can be done in the offline phase with incremental learning for data approximation procedure in online phase.

4.4 Classification methods

In this section, we introduce three classification methods for the classification task. K-NN, FBNN and SVM are the well-known and powerful approaches in classification that are used in this research for prediction of the active user class. These methods will be used in the prediction task of classifying new user rating that belongs to which class generated by HOSVD. Later, we will see that the performance of classification techniques significantly influences the MC-CF prediction accuracy.

4.4.1 K-nearest neighbor (K-NN) classifier

K-NN classifier is a well-known and powerful instance-based machine learning technique for classification data (Desterecke 2012; Yazdani et al. 2009). By learning from all sorted training instances, K-NN simply can be applied to get results from training instances. The K-NN algorithm consists of two phases: training phase and classification phase. In the training phase, the training examples are vectors (each with a class label) in a multidimensional feature space. In this phase, the feature vectors and class labels of training samples are stored. In the classification phase, K is a user-defined constant, a query or test point (unlabeled vector) is classified by assigning a label, which is the most recurrent among the K training samples nearest to that query point. In other words, the K-NN method compares the query point or an input feature vector with a library of reference vectors, and the query point is labeled with the nearest class of library feature vector. This way of categorizing query points based on their distance to points in a training dataset is a simple, yet an effective way of classifying new points. One of the advantages of the K-NN method in classifying the objects is that it requires only few parameters to tune: K and the distance metric, for achieving sufficiently high classification accuracy. Thus, in K-NN based implementations, the best choice of K and distance metric for computing the nearest distance is a critical task.

In K-NN classifier, the distance function usually is considered Euclidean distance presented in Eq. (12) (Nilashi et al. 2011c) when the input vectors and outputs are real numbers and discrete classes, respectively. In this study, we use Euclidean, City-Block and Correlation distance metrics type for distance metric.

Assume $x_1, x_2, \ldots, x_{mx}$ indicates the first row vectors and $y_1, y_2, \ldots, y_{my}$ indicates the second row vectors, the various distance metrics for measuring distance between x_s and y_t are defined as follows:

$$d_{\text{st}} = \sqrt{\sum\nolimits_{j=1}^{n} (x_{sj} - y_{tj})^2} \tag{12}$$

$$d_{\text{st}} = \sum\nolimits_{j=1}^{n} |x_{sj} - y_{tj}) \tag{13}$$

$$d_{\text{st}} = 1 - \frac{(x_s - \bar{x}_s)(y_t - \bar{x}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(y_t - \bar{y}_t)(y_t - \bar{y}_t)'}}$$

$$\bar{x}_s = \frac{1}{n}\sum\nolimits_j x_{sj} \;\; and \;\; \bar{y}_t = \frac{1}{n}\sum\nolimits_j y_{sj} \tag{14}$$

where Eqs. (12, 13) and (14) stand for Euclidean, City-Block and Correlation distance metrics, respectively.

4.4.2 Feedforward backpropagation neural network (FBNN)

A FBNN is a multilayer network that consists of an input layer, one or more hidden layers, and an output layer. All neurons in each layer are fully connected to all neurons in the successive layer. An input pattern is propagated through a hierarchy of layers in a forward direction; i.e., input, hidden, and output layers. In this research, a three-layer network consisting of an input layer, a hidden layer, and an output layer is applied for classification task. The backpropagation is the classical algorithm used for learning. It is an iterative gradient descent algorithm which is designed to minimize the mean squared error between the desired output and the generated output for each input pattern. After the input is propagated from the input layer through the output layer, an error is computed from the difference between the desired output and the generated output obtained from the output layer. If the error is not satisfied then the weights are modified while the error is propagated backward from the output layer to the input layer.

In this research, a FBNN was applied to the problem of multi-class classification. Multi-class NN classification involves building NNs that map the input feature vector to the network output containing more than two classes (Ou and Murphey 2007; Murphey and Luo 2002).

The number of hidden neurons in the hidden layer is one of the major issues to be considered in the establishment of a FBNN for classification. There are many algorithms used to determine the number of hidden neurons. Igelnik and Pao (1995) found that at least $2D$ (D denotes the dimension of the input vectors) hidden neurons can be sufficient for approximating the posteriori probability in classification problem with arbitrary accuracy. In this research, the numbers of hidden neurons are freezed by applying $2D$ hidden neurons to all the experiments. With considering $2D$ hidden neurons the FBNN classifier obtained the best classification accuracy.

4.4.3 Support vector machine (SVM)

SVM proposed by Vapnik (1995a, 1998) is a set of supervised learning methods used for classification and regression analysis. SVMs are linear non-probabilistic binary classifiers, i.e., they can classify data points as one of two classes. They are inherently two-class classifiers, however, can be extended for multi-class. In SVM, the classification is done by intersecting a hyperplane through the feature space that separates one cluster of similarly labeled training data from another. The SVM learns the parameters for this hyperplane by maximizing the margin from the hyperplane to the two training data clusters. More advanced SVMs will use soft margins that react gracefully to abnormally labeled data points and semi-overlapping data clusters that cannot be separated by a simple hyperplane. Though, the more overlap between the two clusters of data points, the worse a SVM will do. This can be mitigated by transforming the initial feature space into a higher dimensional feature space using the kernel trick.

The SVM incorporates the maximal margin strategy and the kernel method (Silva and Ribeiro 2007). The decision function of the SVM is an expansion of the kernel function. The decision function is used to predict the output for a given input. The maximal margin method is applied to improve the accuracy of the prediction. The main goal of this method is to find a hyperplane separating the data with the largest possible margin (Zhou et al. 2009).

In this research, we implemented multi-class SVM instead of binary SVM with linear, polynomial kernel of degree 2 and Radial Basis Function (RBF) kernel.

4.5 Regression methods

As a powerful machine learning technique, SVM is becoming increasingly popular. SVR is able to model complex non-linear relationships using an appropriate kernel function that maps the input matrix X onto a higher dimensional feature space and transforms the non-linear relationships into linear forms. The feature space is then used as a new input to deal with the regression problem. By introducing an ε-insensitive loss function, Vapnik extended SVM for classification to regression (Vapnik et al. 1996). As shown in Fig. 2, the data above the regression function $f(x)$ are considered class 1 data, and the data below $f(x)$ are considered class 2. In this sense, SVR transforms the regression problem into a special classification problem. Moreover, like the support vector classifier, the SVR uses soft margins to tolerate misclassification. Finally, SVR uses a tactic named ε-insensitive loss function (Schölkopf and Smola 2002) to balance the approximate accuracy and computational complexity. As shown in

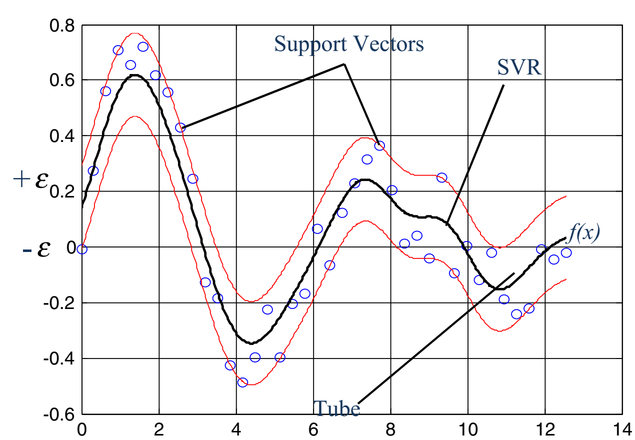


Fig. 2 Support vector regression

Fig. 2, the ε-insensitive function defines a tube with size of ε. Inside of the tube, there is no penalty on the deviation. However, outside of the tube, the penalty is imposed. The SVR approach has been completely introduced by Ferrari et al. (2010) and Bellocchio et al. (2012).

In this research, we implement the two types of SVR, ε-SVR and *nu* (υ)-SVR, that have been developed in LIBSVM (Chang and Lin 2001).

5 Research methodology

Figure 3 shows the general framework of the proposed method with a combination of dimensionality reduction techniques and ANFIS combined with subtractive clustering for discovering knowledge from user ratings and predicting the overall ratings. It can be seen that the dimensionality reduction methods are HOSVD and classification techniques. The presented framework includes two phases for item recommendation and prediction, online and offline phases. In the offline phase, firstly, we apply the HOSVD for dimensionality reduction to reveal the latent associations among its components in the third-order tensor of users' ratings. After applying HOSVD on tensors, we perform cosine-based clustering for obtaining groups of similar users and determining the labels for clusters. These class labels latter are used in the classification methods such as SVM, K-NN and FBNN for providing the prediction model of data and speediness of item recommendations. The main tasks of the dimensionality reduction process by HOSVD are reducing the dimension, obtaining best approximation of data in the tensor of user preferences about items on multi aspects, and finding users with similar preferences on items and criteria. Measuring the similarity of users based on their ratings on criteria provides the possibility of applying clustering method to clusters with similar users. After applying the clustering method that provides the classes of users with similar taste, ANFIS is used to extract knowledge (fuzzy rules) from defined clusters. As presented in Fig. 3, ANFIS is applied in clusters for inducing fuzzy rules and predicting null values in overall ratings. For increasing the accuracy of rule-based system, reducing the amount of data in any class and minimizing overfitting in the training data, subtractive clustering is combined with ANFIS.

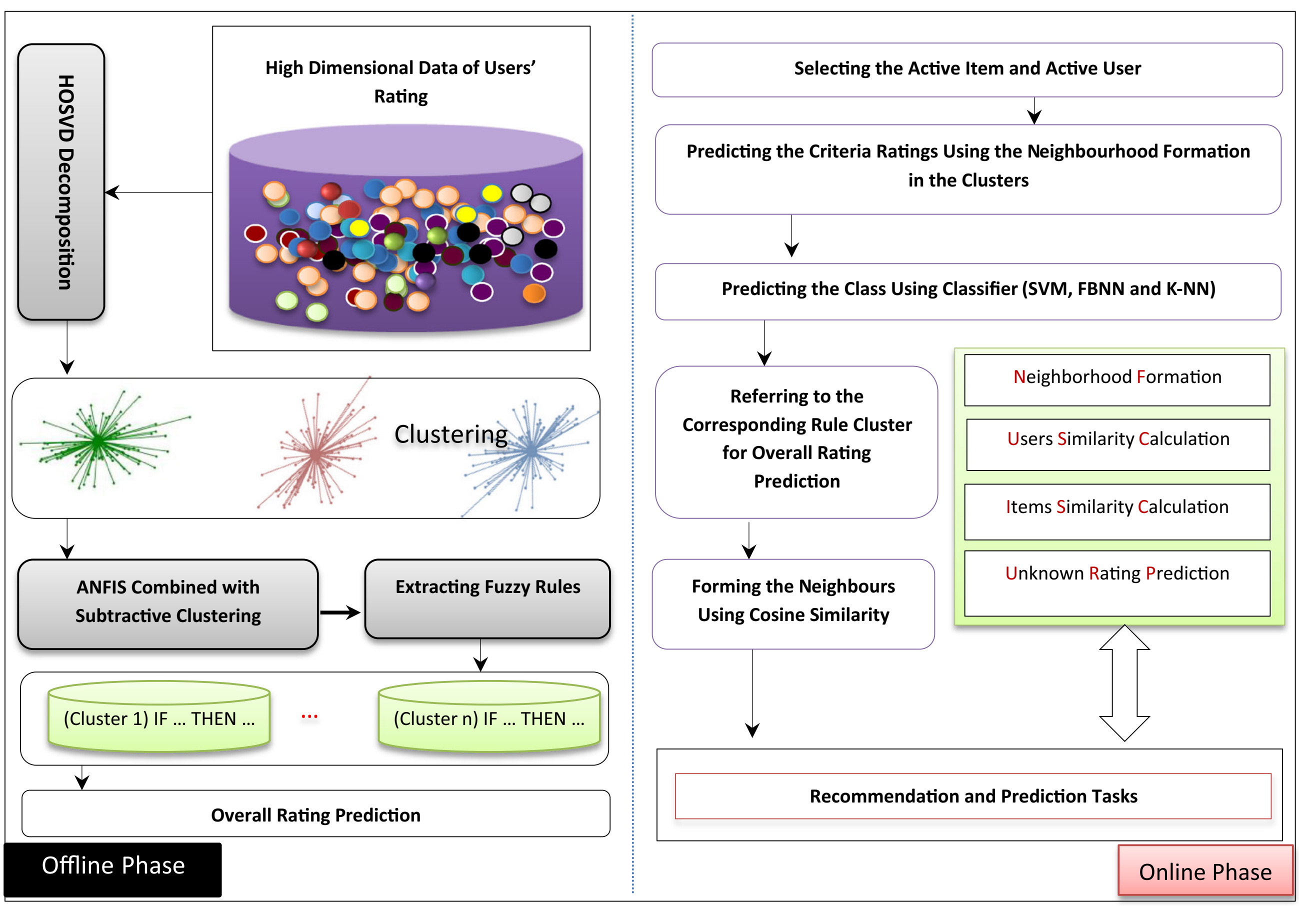


Fig. 3 Research methodology

The main steps of the proposed method for training the model in the offline phase are:

Step 1: HOSVD is applied on training data in 3-order tensor for dimensionality reduction to get the best approximation of rating information.

Step 2: The approximated data by HOSVD is used for clustering using cosine-based similarity. In fact, in this step, label for each vector of ratings is defined to be used in K-NN method in the online phase.

Step 3: ANFIS combined with subtractive clustering is used for training data in clusters obtained in the previous step for extracting fuzzy rules and forming rule clusters.

Step 4: The fuzzy rules are used for predicting existing null values of overall ratings in offline and online phases. It should be noted that for predicting the unknown overall ratings, we solved the sparsity problem in criteria using the neighborhood formation in any cluster. For predicting the unknown criteria ratings for the target item, we relied on a cosine-based similarity as a similarity measure which was performed on approximated data obtained by HOSVD.

After learning the model in the offline phase, in the online phase, the recommender system follows the recommendation and prediction tasks of MC-CF recommender systems using the three main steps:

Step 1: Selecting the active item and active user.

Step 2: Predicting criteria ratings using the neighborhood formation in any cluster. It should be noted that for predicting the unknown overall ratings, we solved the sparsity problem in criteria using the neighborhood formation in any clustering. To do this, we use the cosine similarity measure. It can be one choice for neighborhood formation.

Step 3: Using classification method (SVM, FBNN and K-NN), recommender system predicts the class label for new data.

Step 4: Recommender system refers to the corresponding fuzzy rule cluster and predicts the overall rating for active user.

Step 5: After overall rating prediction, recommender system forms the neighbors using cosine similarity presented in Eq. (15) for active user from corresponding cluster and makes predictions and Top-N recommendations.

5.1 HOSVD tensor data decomposition and clustering process

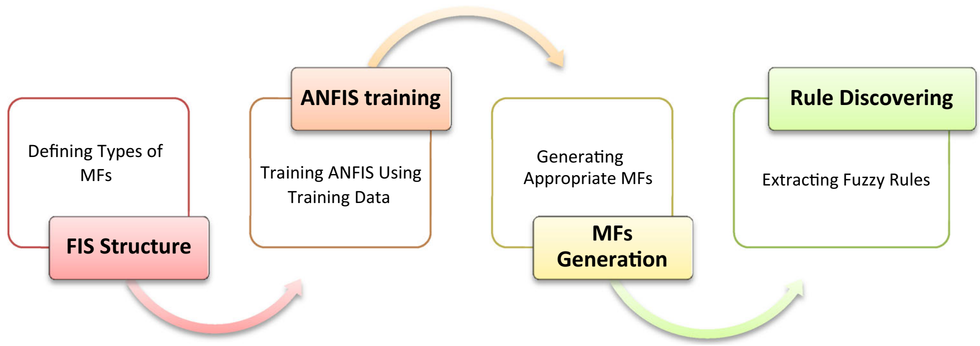
For applying HOSVD, 3-dimensional data are stored in the 3-dimensional tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3}$, whereby I_1 corresponds to the number of users, I_2 corresponds to the number of items which were rated and I_3 is the number of used criteria. Each entry of the tensor $\underline{A}$ is a number between 1 and 13. Using HOSVD, the tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3}$ which contains the user ratings of items on four criteria was decomposed into $\underline{A} \in \underline{S} \times_1 U \times_2 V \times_3 W$ in which $U \in R^{I_1 \times I_1}$, $V \in R^{I_2 \times I_2}$ and $W \in R^{I_3 \times I_3}$ are orthonormal matrices, and $\underline{S} \in R^{I_1 \times I_2 \times I_3}$ is a core tensor that satisfies all-orthogonality and ordering properties. Similar to the truncated SVD for low-rank approximation and dimensionality reduction of matrices, low-rank approximation and dimensionality reduction of higher order tensor can be done by the truncated HOSVD (but with better approximation and computation), that is, take the first r_1 columns of $\mathbf{U}$, the first r_2 columns of $\mathbf{V}$, the first r_3 columns of $\mathbf{W}$, and the top-left $r_1 \times r_2 \times r_3$ block of $\underline{S}$. Thus, for dimensionality reduction for third-dimensional dataset, HOSVD is an effective method. It is flexible to choose different r for different modes of a tensor. The size of the data goes down to $r_1 r_2 r_3 + I_1 r_1 + I_2 r_2 + I_3 r_3$ from $I_1 I_2 I_3$, and if $r_1 = r_2 = r_3$ the size of the data goes down to $r_3 + r(I_1 + I_2 + I_3)$. If we flat the tensor into a $I_1 \times I_2 I_3$ matrix, the size of the data only goes down to $r_2 + r(I_1 + I_2 I_3)$.

For MC-CF problem, in the HOSVD decomposition, the matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ show the relations between User and User, Item and Item, and Criterion and Criterion, respectively, without splitting the 3-dimensional space into pair relations.

Using this method, we can cluster data based on user and item similarity from matrix $\mathbf{U}$ and $\mathbf{V}$. Since K-NN, SVM and FBNN classifiers require supervised learning, HOSVD is selected to obtain clusters from dataset to provide the label for them. The approximated tensor using HOSVD is useful for clustering task in the MC-CF. Based on cosine similarity, we can perform clustering on users and items in an effective way. Furthermore, after tensor decomposition and tensor approximation, in the lower dimensions we can easily find the similar users and items instead of performing this task in the high-dimensional space. From the truncated matrix $\mathbf{U}$, the first row of the matrix is selected and system does cosine similarities calculation through Eq. (15) with the second row, third row and so on, until it reaches the last row. The highest value of cosine similarities is clustered with the first row. Applying this method on rows, the system obtains clusters with small number of similar users. With determining a specific number of clusters, system can combine the close clusters by calculating cosine similarity. Finally, after constructing the clusters, the system assigns the class label to the user rating vectors from the obtained cluster number. Similarly, this procedure can be performed on matrix $\mathbf{V}$ for clustering the similar items. The cosine similarity between two vectors A and B can be defined as:

$$\text{Similarity} = \cos(A, B)$$
$$= \frac{A.B}{||A|| * ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{15}$$

Fig. 4 Discovering knowledge from the users' ratings



5.2 ANFIS modeling for MC-CF

In this section, Neuro-Fuzzy system is introduced for the uncertainty problem of MC-CF for representing and reasoning users' behavior about items' features which are imprecise, subjective and vague. In MC-CF, it is important to deal with non-stochastic uncertainty problem induced from vagueness and imprecision in representing and reasoning items' features. From this perspective, ANFIS, which is one of the powerful techniques of Neuro-Fuzzy system, is used to deal with the aforementioned issues. On the one side, ANFIS was used to extract knowledge (rules) from the users' ratings in multi aspect to be used in overall rating prediction task. The extracted rules were employed for predicting unknown ratings for alleviating sparsity problem in overall rating and also revealing the real level of user preferences on items' features. The ANFIS provided flexible structure of defined problem that is suitable for generating stipulated input–output pairs using a set of induced fuzzy IF-THEN rules with appropriate and varied MFs. The produced FIS is served to predict user overall preferences about items' features with proper training. The elements of used models were a fuzzy set, a NN and data clustering. On the other side, non-stochastic uncertainty emerging from vagueness and imprecision is handled using ANFIS. The MFs produced by ANFIS are used for representation and reasoning users' behavior of providing rating according to their perception about items' features. Based on above explanations, therefore, the main tasks for discovering the knowledge from the users' ratings presented in Fig. 4 are:

- Constructing the FIS structure.
- Training the ANFIS.
- Generating the appropriate MFs.
- Extracting knowledge (Fuzzy rules).

In this study, discovering the knowledge (fuzzy rules) from users' ratings and generalizing the relationship $Y = f(X_1, X_2, ..., X_n)$ are the main goals of applying ANFIS for accurate prediction of overall ratings that accordingly leads to predictive accuracy improvement in MC-CF. In this relationship, $X_1, X_2, ..., X_n$ stands for input variables and Y stands for output variable. In the current study, overall rating or overall preference of users about items' features can be characterized as a function of items' features or criteria. Thus, we relate the variable Y to the overall rating and variables $X_1, X_2, ..., X_n$ to the criteria ratings. Predicting the relationship between inputs and output is one of the important tasks that ANFIS does. Based on the experimental dataset, the input parameters of the ANFIS model for Yahoo!Movies dataset under consideration are Acting (A), Directing (D), Story (S), and Visuals (V). Overall rating (O) stands for output that is defined as overall preference. In addition, for TripAdvisor, the input parameters of the ANFIS model for hotel dataset under consideration are Value aspect (V), Rooms aspect (R), Location aspect (L), Cleanliness aspect (C), Check in/Front desk aspect (CFE), Service aspect (S) and Business Service aspect (BS). Overall rating (O) stands for output that is defined as overall preference.

These attributes naturally are vague, imprecise, and incomplete fuzzy terms that lead to uncertainty in user interest about items' features. With this consideration, in the ANFIS, the users' interest can be introduced and expressed by fuzzy linguistic values (uncertainty modeling). In this paper, four linguistic terms, {cluster 1}, {cluster 2}, {cluster 3} and {cluster 4}, are considered that determine the domain of user interest of Acting, Directing, Story and Visuals in four regions by MFs, as is given in Fig. 5a and b for two inputs Visuals and Directing, respectively. As a general framework of ANFIS for MC-CF, Fig. 6 shows the relationship between input variables (criteria) and output (overall rating).

For Yahoo!Movies dataset, the relationship between input variables (criteria) and outputs (Overall rating) can be defined as:

$$\begin{aligned} \text{Overall rating} &= f(\text{Criteria}_1, \text{Criteria}_2, \ldots, \text{Criteria}_p) \\ &= f(\text{Acting, Visuals, Directing, Story}) \end{aligned} \tag{16}$$

In ANFIS models, the output relations are related to the inputs by mathematical relationship mapping using fuzzy rules. Fuzzy rules play the important role in the ANFIS models and are really the backbone of such systems. The shape

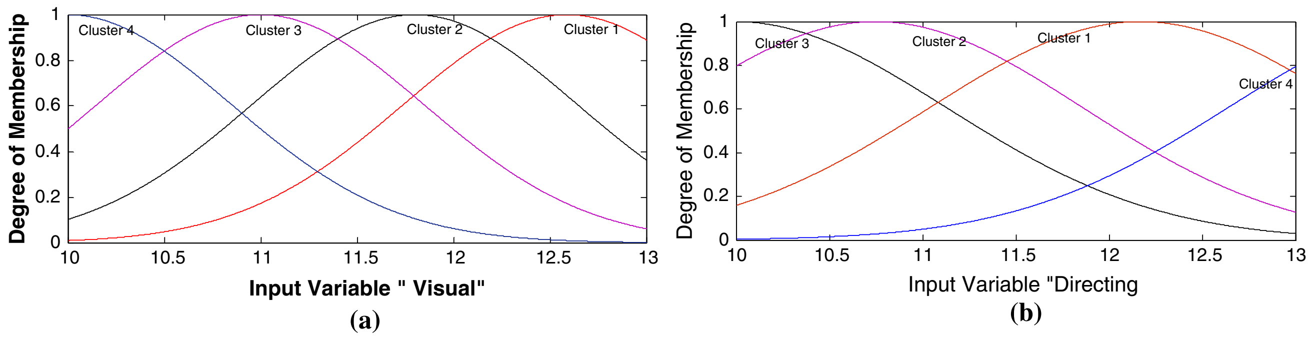


Fig. 5 Membership functions for **a** visuals **b** directing

of fuzzy rules for Yahoo! Movie and TripAdvisor datasets in ANFIS is defined as:

YahooMovie!:

Rule 1: IF A is A_1 AND D is B_1 AND S is C_1 AND V is D_1 THEN $f_1 = p_1 A + q_1 D + r_1 S + t_1 V + \pi_1$

Rule 2: IF A is A_2 AND D is B_2 AND S is C_2 AND V is D_2 THEN $f_2 = p_2 A + q_2 D + r_2 S + t_2 V + \pi_2$

....

Rule n: IF A is A_n AND D is B_n AND S is C_n AND V is D_n THEN $f_n = p_n A + q_n D + r_n S + t_n V + \pi_n$

TripAdvisor:

Rule 1:

IF V is A_1 AND R is B_1 AND L is C_1 AND C is D_1 AND CEF is E_1 AND S is F_1 AND BS is G_1
THEN $f_1 = p_1 V + q_1 R + r_1 L + t_1 C + m_1 CEF + n_1 S + h_1 BS + \pi_1$

Rule 2:

IF V is A_2 AND R is B_2 AND L is C_2 AND C is D_2 AND CEF is E_2 AND S is F_2 AND BS is G_2
THEN $f_2 = p_2 V + q_2 R + r_2 L + t_2 C + m_2 CEF + n_2 S + h_2 BS + \pi_2$

....

Rule n:

IF V is A_n AND R is B_n AND L is C_n AND C is D_n AND CEF is E_n AND S is F_n AND BS is G_n
THEN $f_n = p_n V + q_n R + r_n L + t_n C + m_n CEF + n_n S + h_n BS + \pi_n$

For example, in this study based on users' ratings to movies, ANFIS by training ratings' vectors extracts the fuzzy rules. The following example shows one of the IF-THEN fuzzy rules generated by ANFIS.

" **IF** *Acting of movie is cluster1* **AND** *Directing is cluster1* **AND** *Story is cluster1* **AND** *Visuals is cluster1* **THEN** *Overall rating is out1cluster1* ".

According to the extracted fuzzy rules by ANFIS, the out1cluster1 for the overall rating is obtained from the value of 4 inputs from their MFs. Also, using subtractive clustering in ANFIS, the system improves the precision of extracted fuzzy rules obtained from users' ratings and minimizes the overfitting in the training data. Subtractive clustering helps MC-CF recommender system to reveal accurately the users' preferences about items' features in the soft clusters. It divides the user preferences on items in fuzzy clusters that system can predict the relationship between each criteria and overall rating.

5.2.1 Training the ANFIS and model validation using checking and testing dataset

Three sets of data are used for ANFIS based modeling: training data, checking data and test data. ANFIS uses training data for constructing the model of the target system. The rows of training data are desired inputs with outputs' pair of the

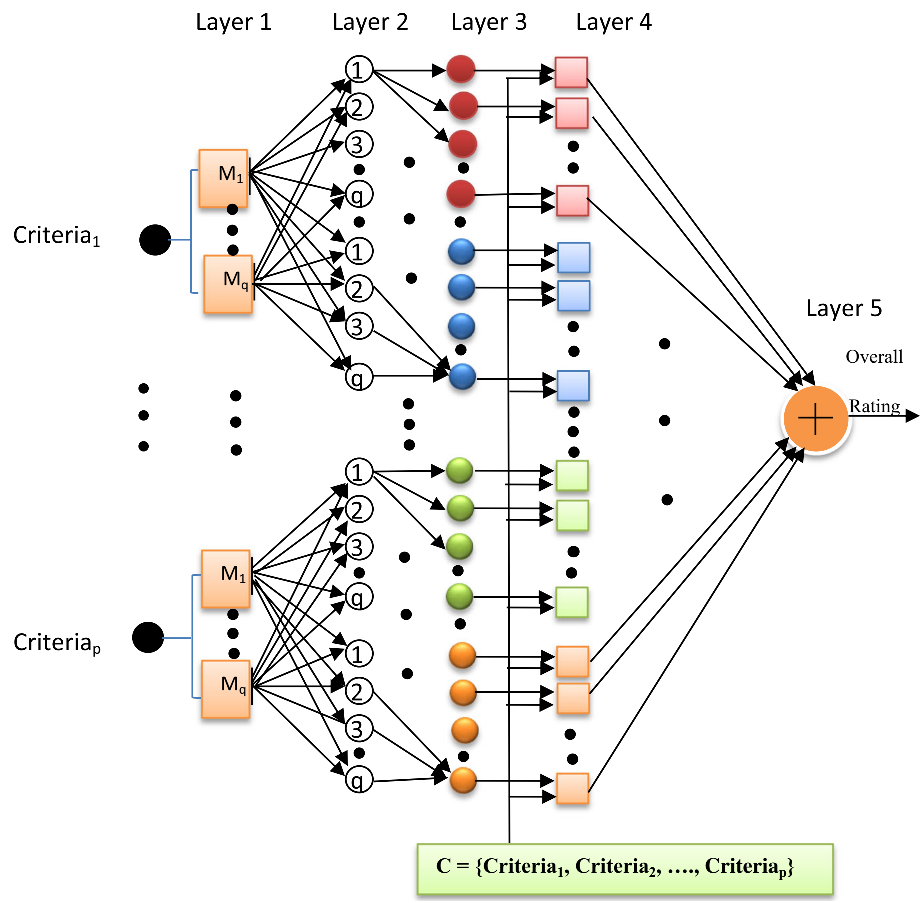


Fig. 6 The relationship between input variables (criteria) and output (overall rating) in ANFIS model

Table 2 A sample of the multi-criteria rating of Yahoo!Movies dataset

Initial data form							Final data form						
User ID	Overall rating	C_1	C_2	C_3	C_4	Movie ID	User ID	Overall rating	C_1	C_2	C_3	C_4	Movie ID
1	A+	A+	A	A	A−	1	1	13	13	12	12	11	1
	B+	B+	A+	B	A+	4		10	10	13	9	13	4
	B	B	A-	B	A+	25		9	9	11	9	13	25
	…	…	…	…	…	…		…	…	…	…	…	…
	…	…	…	…	…	…		…	…	…	…	…	…
2	A	A+	A−	A−	A+	9	2	12	13	11	11	13	9
	B+	B+	B	B	B	18		10	10	10	9	9	18
	B+	A−	A+	A+	B	2		10	11	11	13	9	2
	…	…	…	…	…	…		…	…	…	…	…	…
	…	…	…	…	…	…		…	…	…	…	…	…

target system to be modeled. For the testing generalization capability of the FIS at each epoch that prevents overfitting networks and verifies the identified ANFIS, checking data are used. The testing data are used to evaluate the model performance. The same format of training data is defined for the checking and testing data but generally their elements are different from those of the training data.

In this research, the clusters obtained by cosine similarity and HOSVD methods were divided into three groups of data: training, testing and checking including 80, 10 and 10 % of data, respectively.

6 Result and discussion

6.1 Experimental dataset

To analyze the effectiveness of the proposed method, several experiments were conducted on Yahoo!Movies and Tri-

Table 3 Statistics of movie rating on four criteria and overall rating

	Acting	Directing	Story	Visuals	Overall Rating
N	257,317	257,317	257,317	257,317	257,317
Mean	7.843	7.570	7.691	7.760	10.061
Minimum	0	0	0	0	0
Maximum	13.0	13.0	13.0	13.0	13.0

pAdvisor datasets that have been provided by Yahoo!Movies network (http://movies.yahoo.com) and TripAdvisor website (http://www.tripadvisor.com). However, Yahoo!Movies is no longer providing multi-criteria ratings. On the Yahoo!Movies network, users could rate movies in 4 dimensions: Acting, Directing, Story and Visuals and also assign an Overall Rating. A 13-level rating scale is used for ratings. The four criteria for any movie are considered as: C_1 = Acting, C_2 = Directing, C_3 = Story and C_4 = Visuals and for overall ratings O = Overall Rating. In the Yahoo!Movies dataset, all user ratings are measured on a value between 1 and 13 in quantitative scale. The ratings are on a 13-point scale ($A+$, A, $A-$, $B+$, B, $B-$, $C+$, C, $C-$, $D+$, D, $D-$, and F). All values are measured in a 13-fold qualitative scale with F denoting the worst evaluation grade and $A+$ declaring the most preferred value. For processing purposes, we replaced letters with numbers, so as 1 corresponded to the worst value, formerly denoted as F and 13 to best value, $A+$. An example of the initial and the final, after our transformations rating scheme, is shown in Table 2. The left side shows a typical raw data form, while in the right side the same data are presented in a ready to process form (final data form).

TripAdvisor represents the world largest and most successful social networking and community site in tourism (O'Connor 2008). The platform facilitates the reviewing of hotels around the world and brings together individuals in discussion forums and provides users with independent travel reviews and comments. In TripAdvisor website users can rate a hotel according to seven different dimensions: Value aspect, Rooms aspect, Location aspect, Cleanliness aspect, Check in/front desk aspect, Service aspect and Business Service aspect. In addition, users provide overall ratings on hotels. Ratings ranges from zero to five stars, and -1 indicates this aspect rating is missing in the original html file.

Generally, the information of ratings is presented in fourfold $<$ userID; itemID; Overall rating; Criteria rating $>$. This data can easily be converted into a three-dimensional rating tensor, whereas the first dimension spans the number of users and the second dimension spans the number of items and the third dimension spans the number of criteria ($\underline{A}_{m\times n\times k}$). Table 3 presents the statistics of raw movie dataset for four criteria and overall rating. As can be seen in Table 3, there are 257317 tuples of ratings where the range of ratings is between 0 and 13 on criteria and overall ratings. The average evaluation grade is 7.843, 7.570, 7.691, 7.760, and 10.061 for the Acting, Directing, Story, Visuals, and Overall rating, respectively.

In Table 3, although, there are 257,317 tuples of rating in the original dataset, however, from users who have rated very few movies are not useful for MC-CF, since, we cannot reliably know the preferences of a user from only a few of his ratings. The sparsity rate is:

$$\text{fill Rate} = \frac{\text{num Ratings} \times 100\,\%}{\text{num Users} \times \text{num Items}} = \frac{257{,}317 * 100\,\%}{127{,}829 \times 8{,}272} \gg 2.43$$

That means, not even 2.43 % of all entries in the rating tensor are filled. For TripAdvisor, the fill rate was calculated about 1.583.

Tables 4 and 5 present the sample and statistics of raw TripAdvisor dataset for seven criteria and overall rating, respectively. In Table 4, there are 28,500 tuples of ratings where the range of ratings is between -1 and 5 on criteria and overall ratings. The average evaluation grade is 2.69, 2.20,

Table 4 A sample of the multi-criteria rating of TripAdvisor dataset

User ID	Overall rating	Value aspect	Rooms aspect	Location aspect	Cleanliness aspect	Check in/front desk aspect	Service aspect	Business service	Hotel ID
18	5	5	4	−1	5	−1	5	−1	hotel_565550
14	5	5	5	5	5	5	5	5	hotel_566077
20	3	3	4	2	5	1	1	2	hotel_566077
	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.
22	4	3	3	3	4	3	4	3	hotel_570888
45	5	5	5	5	5	3	3	5	hotel_570888
11	5	−1	5	5	5	5	5	5	hotel_572859

Table 5 Statistics of hotel rating on seven criteria and overall rating

	Overall rating	Value aspect	Rooms aspect	Location aspect	Cleanliness aspect	Check in/front desk aspect	Service aspect	Business service
N	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Mean	2.69	2.20	2.70	1.45	2.84	1.36	2.72	0.46
Min.	0	−1	−1	−1	−1	−1	−1	−1
Max.	5	5	5	5	5	5	5	5

Table 6 Information of Yahoo!Movies dataset for YM-20-20, YM-10-10 and YM-5-5

Name	#Users	#Items	#Overall ratings
YM-20-20	429	491	18,504
YM-10-10	1,827	1,471	48,026
YM-5-5	5,978	3,079	82,599

Table 7 Rule of thumb for the interpretation of the Silhouette coefficient

Range	Interpretation
>0.70	Strong structure has been found
0.50–0.70	Reasonable structure has been found
0.25–0.50	The structure is weak and could be artificial
<0.25	No substantial structure has been found

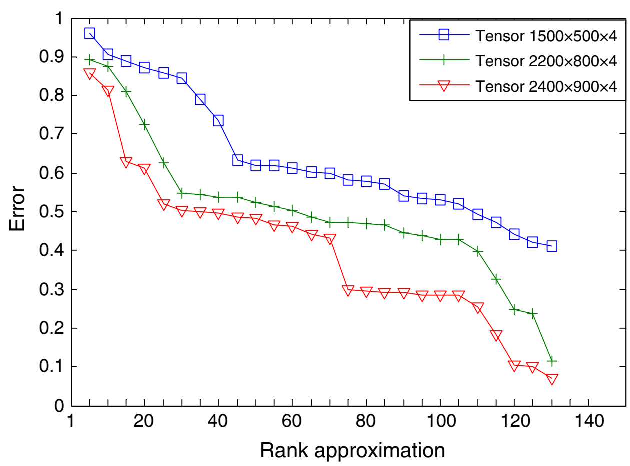


Fig. 7 Rank approximation error for three tensors of training data for Yahoo!Movies dataset

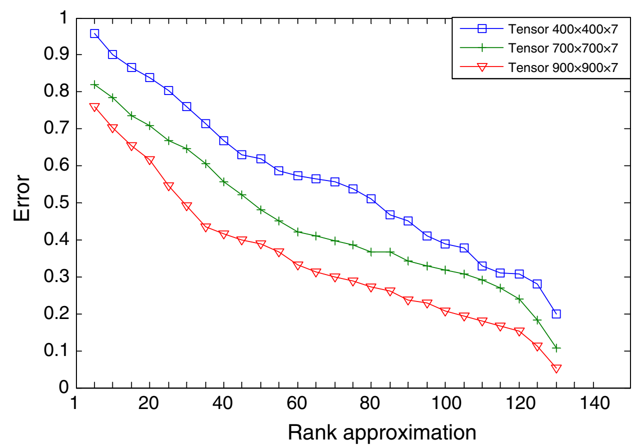


Fig. 8 Rank approximation error for three tensors of training data of TripAdvisor dataset

2.70, 1.45, 2.84, 1.36, 2.72 and 0.46 for Overall rating, Value aspect, Rooms aspect, Location aspect, Cleanliness aspect, Check in/front desk aspect, Service aspect and Business Service aspect, respectively.

Also, we need enough ratings per individual to both train and test the models. Therefore, we have retained only those records that contain items which have at least 5, 10 and 20 ratings. We pre-processed the datasets and created the test datasets with different density and quality levels and applied the proposed methods on YM-20-20 (each movie has at least 20 ratings), YM-10-10 (each movie has at least 10 ratings), YM-5-5 (each movie has at least 5 ratings). Table 6 presents the movie dataset in groups of YM-5-5, YM-10-10 and YM-20-20 (Jannach et al. 2012).

6.2 Performance of HOSVD clustering

Due to having fast calculation of HOSVD, it was performed for the training tensors $\underline{A} \in R^{1,500\times500\times4}$, $\underline{A} \in R^{2,200\times500\times4}$ and $\underline{A} \in R^{2,400\times900\times4}$ from YM-5-5 which corresponds to the training sets. For TripAdvisor dataset, tensors $\underline{A} \in R^{400\times400\times7}$, $\underline{A} \in R^{700\times700\times7}$ and $\underline{A} \in R^{900\times900\times7}$ were selected. As a result, an approximation $\tilde{A}_{(a1,a2,a3)}$ is retained for these tensors which the values set of a_1, a_2 and a_3 determine the dimensions of the core tensor. As results an approximation, we calculate the error approximation using Eq. (4).

Figure 7 shows the approximation error (Frobenius-norm) for different ranks of training tensors $\underline{A} \in R^{1,500\times500\times4}$, $\underline{A} \in R^{2,200\times500\times4}$ and $\underline{A} \in R^{2,400\times900\times4}$. As can be seen in Fig. 7, the approximation error is decreased with increasing the rank

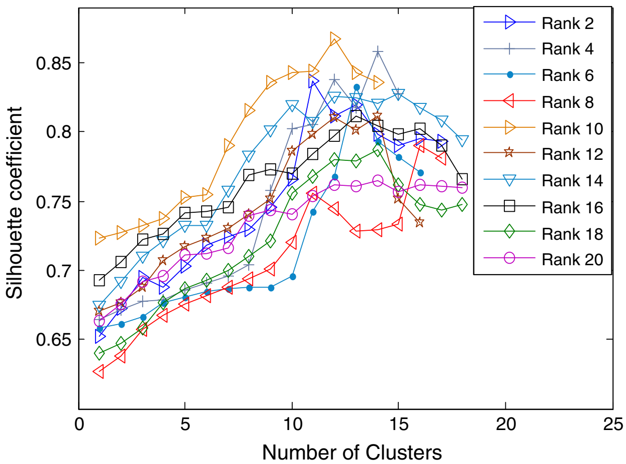


Fig. 9 Silhouette coefficient for HOSVD clustering for different approximation ranks

Table 8 Setting the parameters for SVM kernels

Kernel parameters for SVM	
Degree (poly)	2.00
Gamma in kernel function (RBF)	0.5
Coef0 in kernel function (poly/sigmoid)	2/0
Tolerance of termination criteria (eps)	0.001
C (complexity cost)	10

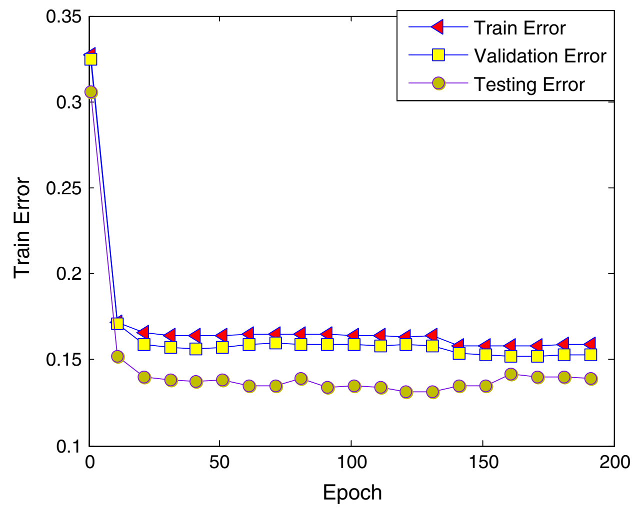


Fig. 10 Training, validation and testing error for 200 epochs in FBNN

of tensors. Further, the estimated error is lower for large-sized tensor than small-sized tensor. This confirms for MC-CF that the sufficient rating criteria and overall ratings in the tensors influence the quality of tensor approximation using HOSVD.

In this paper, we select aforementioned tensors of training data with approximation ranks 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20 for clustering task. For estimating performance of clustering from these ranks, we adopt Silhouette coefficient (Kaufinan and Rousseeuw 1990) value as the standard measure for clustering quality and use it to determine the best cluster formation. The Silhouette coefficient can assess the quality of a clustering. It is an internal index that measures how good the clustering fits the original data based on statistical properties of the clustered data. The Silhouette coefficient takes values between -1 and 1. The closer to 1, the better the clustering fits the data. Table 7 lists a general rule of thumb on how to interpret the Silhouette coefficient.

In Fig. 8, the approximation errors are presented for different ranks of training tensors $\underline{A} \in R^{400\times400\times7}$, $\underline{A} \in R^{700\times700\times7}$ and $\underline{A} \in R^{900\times900\times7}$.

Figure 9 shows the average Silhouette coefficient for HOSVD clustering for approximation ranks 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20 of tensor $\underline{A} \in R^{1{,}500\times500\times4}$. According to Fig. 9, the highest average Silhouette coefficient for clustering is obtained about 0.867 for rank 10 approximation. For tensors $\underline{A} \in R^{2{,}200\times500\times4}$ and $\underline{A} \in R^{2{,}400\times900\times4}$, the average Silhouette coefficient was obtained about 0.887 and 0.892 for ranks 16 and 18 approximation, respectively. For the $\underline{A} \in R^{400\times400\times7}$, $\underline{A} \in R^{700\times700\times7}$ and $\underline{A} \in R^{900\times900\times7}$,the highest average Silhouette coefficient was calculated 0.889, 0.901 and 0.912 for ranks 8, 12 and 18, respectively. These accuracy percentages are reasonably good. Based on observation, lower ranking approximation values do better than the higher approximation values in any tensors. This supports our claim that truncated HOSVD gives good results of data approximation for MC-CF.

6.3 Evaluating the classifiers' performance

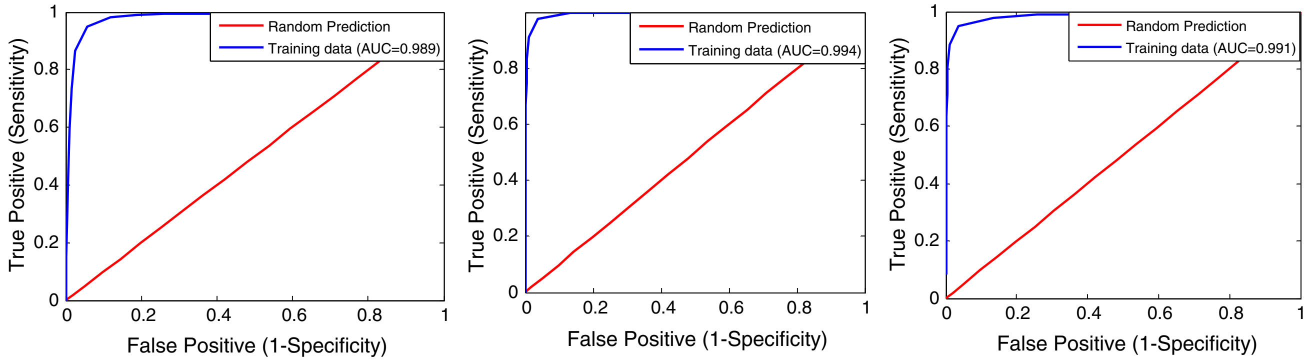
In this section, we present in detail the experimental results of classifiers on data of tensors $\underline{A} \in R^{1{,}500\times500\times4}$ and $\underline{A} \in R^{400\times400\times7}$. For evaluating the classifiers performance, we applied the three classification approaches on dataset with multi-class assigned using clustering from the approximated data. Fivefold cross-validation for 50 trials was used for all approaches to assess the classifiers' performance.

For K-NN, using fivefold cross-validation, the values $K = 1$, $K = 3$, $K = 5$ and $K = 7$ with three different methods of distance metric, the nearest distance (Euclidean), Correlation and City-Block, were tested.

For FBNN, we set the learning rate (η) to 0.5, number of neurons to 8 and number of epochs to 200. The FBNN was tested on dataset for 50 iterations using 5-fold cross-validation.

For SVM, we apply the Polynomial, RBF, Sigmoid and Linear. The parameters for this kernel were set as presented in Table 8.

In $\underline{A} \in R^{1{,}500\times500\times4}$, the results of applying K-NN method indicated that for K-NN ($K = 5$) using Euclidean



(a) K-NN **(b)** SVM **(c)** FBNN

Fig. 11 Fivefold cross-validation error rate for three classification approaches

Table 9 Computation time and error rate for different types of kernel in multi-class SVM for Yahoo!Movies dataset

Kernel	Computation time (ms)	Error rate
Polynomial	20,748	0.4503
RBF	26,786	0.1375
Sigmoid	38,751	0.1732
Linear	10,827	0.1653

Table 10 Computation time and error rate for different types of kernel in multi-class SVM for TripAdvisor dataset

Kernel	Computation time (ms)	Error rate
Polynomial	1,980	0.3870
RBF	2,245	0.1213
Sigmoid	3,145	0.1653
Linear	934	0.1521

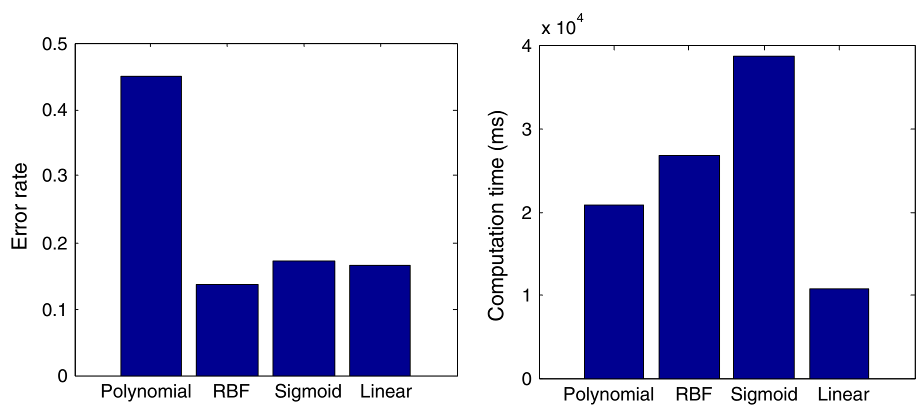
distance metric, the highest averaged classification accuracy (measured by Area Under Curve (AUC) in the Operator Characteristic Curve (ROC)) was obtained about 98.91 % in comparison to City-Block (95.89 %) and Correlation distance metric (96.76 %). In $\underline{A} \in R^{400\times400\times7}$, the AUC for Euclidean distance, City-Block and Correlation distance metric were obtained about 99.01, 96.21 and 97.14 %, respectively.

For FBNN, the training, validation and testing error are shown in Fig. 10.

In $\underline{A} \in R^{1,500\times500\times4}$, the minimum error for training, validation and testing was obtained about 0.158, 0.1524 and 0.131, respectively. For $\underline{A} \in R^{1,500\times500\times4}$ and $\underline{A} \in R^{400\times400\times7}$, the classification accuracy for FBNN was obtained about 99.11 and 99.23 %, respectively. In addition, for $\underline{A} \in R^{1,500\times500\times4}$, the accuracy was calculated about 99.41 % by SVM through the RBF kernel, 99.21 % using the linear kernel, 98.24 % using polynomial kernel with degree 2 and 98.44 % using Sigmoid kernel. The accuracy for $\underline{A} \in R^{400\times400\times7}$ was calculated about 99.52 % through the RBF kernel, 99.35 % using the linear kernel, 98.39 % using polynomial kernel with degree 2 and 98.76 % using Sigmoid kernel. Figure 11 shows the accuracy of classification methods evaluated from ROC for K-NN (using Euclidean distance), SVM (using the RBF kernel) and FBNN. As can be seen in this figure, the difference between them is not significant but the SVM using RBF kernel significantly outperforms the K-NN and FBNN approaches.

In Tables 9 and 10, the computation time and error rate of different types of SVM kernels are presented. As presented

Fig. 12 Computation time and error rate for different types of kernel in multi-class SVM



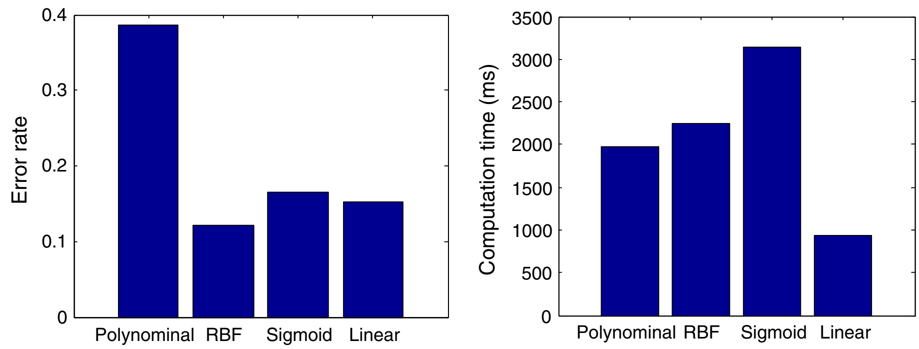


Fig. 13 Computation time and error rate for different types of kernel in multi-class SVM

Table 11 Eigenvalues and individual and cumulative variance for dataset with considering overall ratings

Tensor	Eigenvalue	Difference	Proportion (%)	Cumulative (%)
1	4.344068	4.106465	86.88	86.88
2	0.237604	0.055293	4.75	91.63
3	0.182311	0.052143	3.65	95.28
4	0.130168	0.024318	2.60	97.88
5	0.105850	–	2.12	100.00
Tot.	5.000000	–	–	–

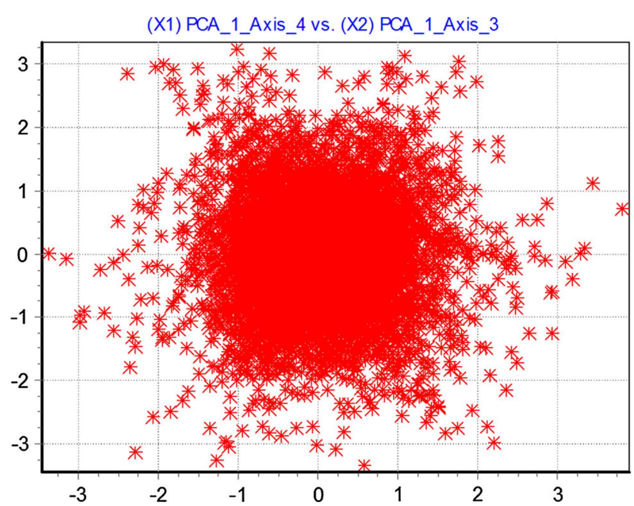


Fig. 14 Scatter plots of two PCs from the PCA result

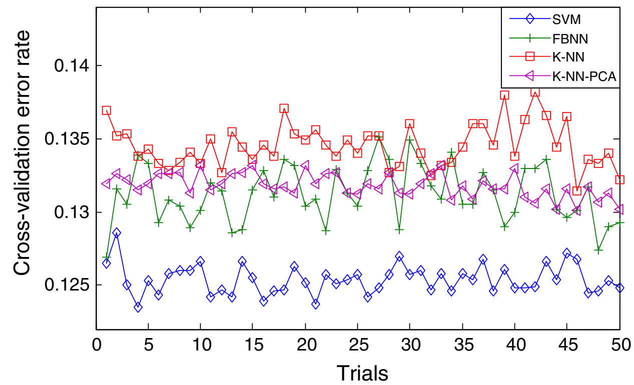


Fig. 15 Fivefold cross-validation error rate for three classification approaches

Table 12 The overall fivefold cross-validation error rate of classifiers

Tensor	Classifier	Overall fivefold cross-validation error
$\underline{A} \in R^{1,500\times500\times4}$	SVM	0.1254
	FBNN	0.1311
	K-NN	0.1345
	K-NN-PCA	0.13182
$\underline{A} \in R^{2,200\times500\times4}$	SVM	0.1118
	FBNN	0.1262
	K-NN	0.1298
	K-NN-PCA	0.1287
$\underline{A} \in R^{2,400\times900\times4}$	SVM	0.1078
	FBNN	0.1123
	K-NN	0.1169
	K-NN-PCA	0.1134

in this table and Figs. 12 and 13, the linear and RBF kernels give the best performance in case of computation time and accuracy, respectively. In addition, the worst classification accuracy is obtained using polynomial kernel and computation time of sigmoid kernel is higher than other kernels.

The main disadvantage of the K-NN algorithm is that it is a lazy learner, i.e., it does not learn anything from the training data and simply uses the training data itself for classification. Another disadvantage of this approach is that the algorithm does not learn anything from the training data, which can result in the algorithm not generalizing well and also not being robust to noisy data and especially for datasets with many features.

For solving these problems, in this paper, the Principal Component Analysis (PCA) was applied prior to the K-NN predictor on the original data of Yahoo!Movies and TripAdvisors for denoising and projecting out the components of the bottom eigenvectors that reduce K-NN error rate. PCA was also used to accelerate neighbor nearest computations in the experimental dataset where the linear preprocessing from PCA significantly reduced the amount of computation either by explicitly reducing the dimensionality of the inputs and

Table 13 The overall fivefold cross-validation error rate of classifiers

Tensor	Classifier	Overall fivefold cross-validation error
$\underline{A} \in R^{400\times400\times7}$	SVM	0.1088
	FBNN	0.1254
	K-NN	0.1316
	K-NN-PCA	0.1266
$\underline{A} \in R^{700\times700\times7}$	SVM	0.0974
	FBNN	0.119
	K-NN	0.1254
	K-NN-PCA	0.1215
$\underline{A} \in R^{900\times900\times7}$	SVM	0.084
	FBNN	0.0942
	K-NN	0.1076
	K-NN-PCA	0.1005

Table 14 Formation of induced rules by ANFIS

Rule #	Extracted fuzzy rules
1	**IF** (acting is cluster1) **AND** (directing is cluster1) **AND** (story is cluster1) **AND** (visuals is cluster1) **THEN** (overalRating is out1cluster1)(1)
2	**IF** (acting is cluster2) **AND** (directing is cluster2) **AND** (story is cluster2) **AND** (visuals is cluster2) **THEN** (overalRating is out1cluster2)(1)
3	**IF** (acting is cluster3) **AND** (directing is cluster3) **AND** (story is cluster3) **AND** (visuals is cluster3) **THEN** (overalRating is out1cluster3)(1)
4	**IF** (acting is cluster4) **AND** (directing is cluster4) **AND** (story is cluster4) **AND** (visuals is cluster4) **THEN** (overalRating is out1cluster4)(1)

re-ordering the input coordinates in terms of their variance. As can be seen in Table 11, for $\underline{A} \in R^{1,500\times500\times4}$, the eigenvalue ($\lambda_2 = 0.2376$) associated with the 2nd factor is high. It corresponds to 91.63 % of the variance. Thus, we selected two PCs as inputs for K-NN.

In Fig. 14, using a scatter plot, we project the observations in the two dimensions 3 and 4 and set the first dimension for the horizontal axis and the second dimension for the vertical one.

Finally, for $\underline{A} \in R^{1,500\times500\times4}$, the overall fivefold cross-validation error rates of SVM, FBNN, K-NN and K-NN-PCA for 50 trials were obtained about 0.1254, 0.1311, 0.1345 and 0.13182, respectively. The error rates for 50 trials are shown in Fig. 15. In Table 12, for tensors $\underline{A} \in R^{1,500\times500\times4}$, $\underline{A} \in R^{2,200\times500\times4}$ and $\underline{A} \in R^{2,400\times900\times4}$, we present the overall fivefold cross-validation error rate of SVM, FBNN, K-NN and K-NN-PCA. For $\underline{A} \in R^{400\times400\times7}$, the overall fivefold cross-validation error rates of SVM, FBNN, K-NN and K-NN-PCA for 50 trials were obtained about 0.1198, 0.1271, 0.1315 and 0.1292, respectively. In Table 13, for tensors $\underline{A} \in R^{400\times400\times7}$, $\underline{A} \in R^{700\times700\times7}$ and $\underline{A} \in R^{900\times900\times7}$, we present the overall fivefold cross-validation error rate of SVM, FBNN, K-NN and K-NN-PCA.

6.4 Evaluating the ANFIS model of MC-CF

After clustering the dataset, the ANFIS models were developed for the clusters with maximum Silhouette Coefficient. The ANFIS model for Yahoo!Movies and TripAdvisor datasets is shown in Fig. 16. In this section, we present the experimental results of ANFIS model on clusters with maximum Silhouette coefficient obtained from tensor $\underline{A} \in R^{1,500\times500\times4}$ (third cluster for rank approximation 12).

Using ANFIS, four fuzzy clusters have been determined for the 190 users' ratings for selected cluster. The number of fuzzy rule set was equal to the number of cluster centers, each representing the characteristic of the cluster as given in Table 14.

For evaluating the ANFIS model, several measures of accuracy were used to determine the model capability for predicting the overall rating. For this reason, four estimators Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), MAE, and coefficient of determination (R^2) evaluated the models. These estimators are determined by:

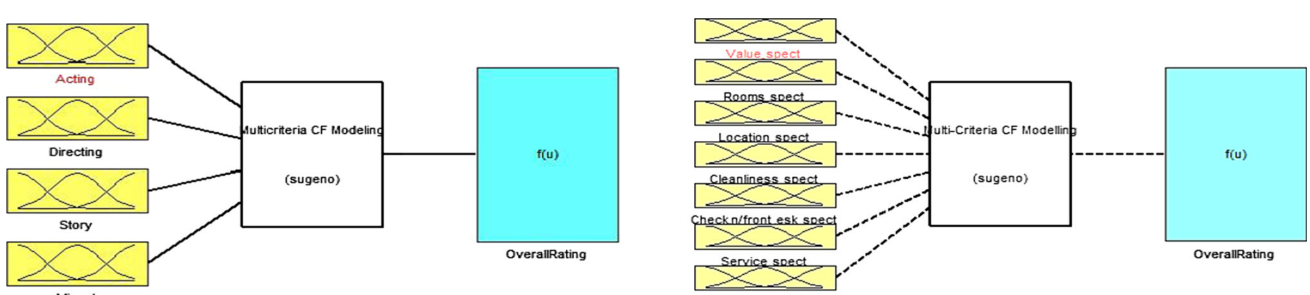


Fig. 16 Modeling MC-CF by ANFIS using Yahoo!Movies and TripAdvisor datasets

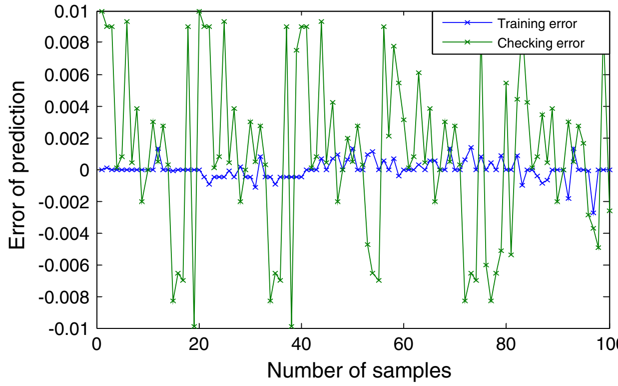


Fig. 17 Training and checking error for 100 training and checking data

$$\text{MSE} = \frac{\sum_{O=1}^{n} (\text{actual}(O) - \text{predicted}(O))^2}{n} \tag{17}$$

$$\text{MAPE} = \frac{\sum_{O=1}^{n} (\text{actual}(O) - \text{predicted}(O))/\text{actual}(O)}{n} \tag{18}$$

$$R^2 = 1 - \frac{\sum_{O=1}^{n} (\text{actual}(O) - \text{predicted}(O))^2}{\sum_{O=1}^{n} (\text{actual}(O) - \overline{\text{actual}(O)})^2} \tag{19}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{O=1}^{n} (\text{actual}(O) - \text{predicted}(O))^2}{n}} \tag{20}$$

where actual (O) indicates the real overall rating provided by user, prediction (O) denotes the predicted overall rating value, and n corresponds the number of used users' ratings. Usually, in the training process RMSE and MSE measure are used to test the prediction model, however, in this study, other performance measures were used to investigate for a more effective performance evaluation that are coefficient of determination R^2 and MAPE. The coefficient of determination R^2 provides a value between [0, 1] about the training of the proposed network. A value closer to 1 stands for the success of learning. In addition, in this study, MAPE was used that accurately identifies the model deviations.

Figure 17 demonstrates errors for a part of training and checking dataset. From the errors of 100 checking and training data, ANFIS model has been trained effectively using training data with a lowest error.

For subtractive clustering, the parameters have been set by a trial and error approach to: accept ratio to 0.5, reject ratio to 0.15 and 0.5 and squash factor to 1.25. However, we could test the effect of the two variables r_a and r_b that represent a radius of the neighborhood on the training, checking and test data for overall rating prediction error. The error was estimated in lowest value for the $r_b = 1.5r_a$ and the results of varying r_a from 0.3 and 0.8 for the radius of the neighborhood.

The average error for training, testing and checking data is presented in Tables 15 and 16 for RMSE, MSE, MAPE and R^2. As presented in these tables, the tensors with higher dimensions provide lower prediction error and higher prediction accuracy in relation to the tensors with lower dimensions.

Illustrating the interdependency between criteria and overall rating is helpful in revealing actual level of users' interest on items' features. Figure 18 illustrates the interdependency of criteria and overall rating of Yahoo!Movies dataset through the control surface obtained from the fuzzy rules generated by ANFIS combined with subtractive clustering. The level of overall rating can be depicted as a continuous function of its input parameters as Acting, Directing, Story, and Visuals. This figure depicts the variation of overall rating based on identified rules. This figure depicts the interdependency of overall rating on {Directing–Acting} and {Directing–Story}, respectively. These surface plots exactly show the users' perception and behaviors on any two features of items in one group of users with similar preferences. These results shown in the surface plots are useful

Table 15 Average RMSE, MSE, MAPE and R^2 for ANFIS models of Yahoo!Movies dataset

Tensor	Checking				Training				Testing			
	RMSE	MSE	MAPE	R^2	RMSE	MSE	MAPE	R^2	RMSE	MSE	MAPE	R^2
$\underline{A} \in R^{1,500\times500\times4}$	0.02144	0.00912	0.1823	0.8246	0.01272	0.00912	0.1823	0.9946	0.01951	0.00949	0.1023	0.9115
$\underline{A} \in R^{2,200\times500\times4}$	0.02011	0.00893	0.1711	0.8315	0.01244	0.00898	0.1773	0.9951	0.01894	0.00924	0.0903	0.9185
$\underline{A} \in R^{2,400\times900\times4}$	0.01981	0.00843	0.1691	0.8425	0.01221	0.00883	0.1727	0.9958	0.01821	0.00901	0.0889	0.9221

Table 16 Average RMSE, MSE, MAPE and R^2 for ANFIS models of TripAdvisor dataset

Tensor	RMSE	MSE	MAPE	R^2	RMSE	MSE	MAPE	R^2	RMSE	MSE	MAPE	R^2
$\underline{A} \in R^{400\times400\times7}$	0.02144	0.00912	0.1823	0.8246	0.01272	0.00912	0.1823	0.9946	0.01951	0.00949	0.1023	0.9115
$\underline{A} \in R^{700\times700\times7}$	0.02011	0.00893	0.1711	0.8315	0.01244	0.00898	0.1773	0.9951	0.01894	0.00924	0.0903	0.9185
$\underline{A} \in R^{900\times900\times7}$	0.01981	0.00843	0.1691	0.8425	0.01221	0.00883	0.1727	0.9958	0.01821	0.00901	0.0889	0.9221

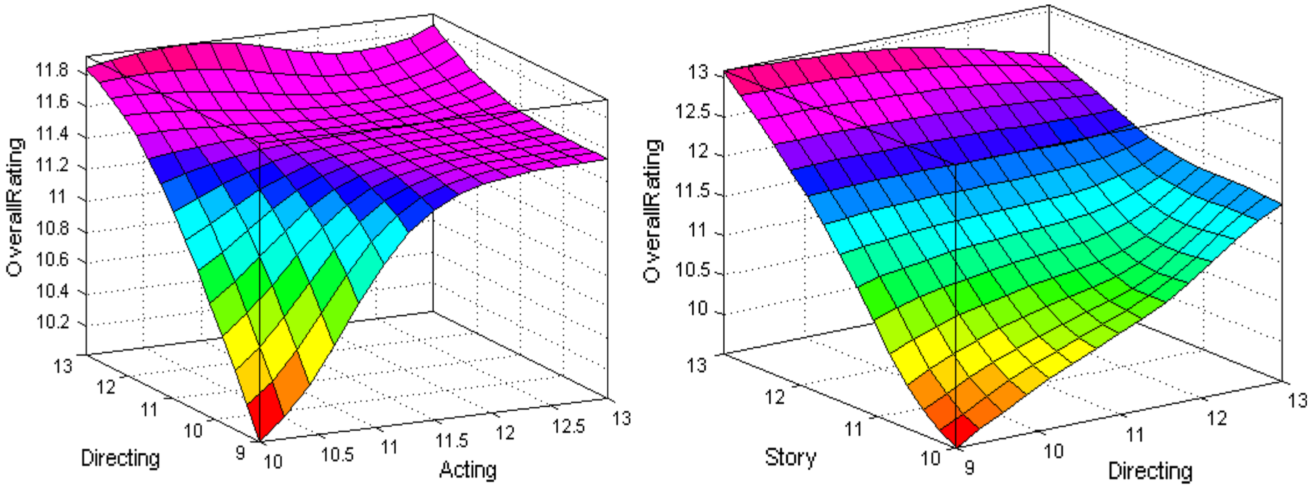


Fig. 18 Interdependency of Overall rating on some criteria of Yahoo!Movies dataset

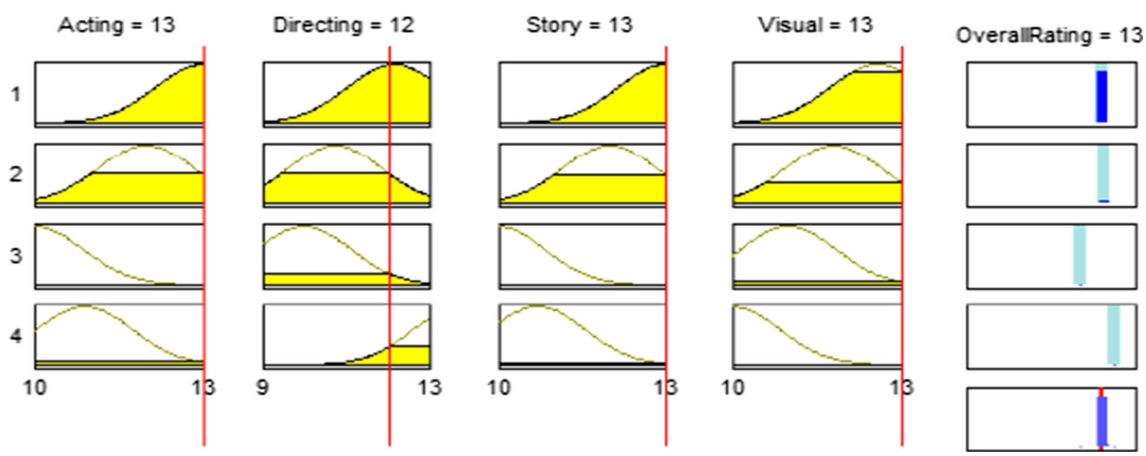


Fig. 19 Fuzzy rule viewer for input and output variables of ANFIS model

to detect the user' behavior in giving ratings to the items' features in MC-CF. Thus system can recognize which item' feature (criteria) in which level is tailored to their preferences. The fuzzy rule viewer of the established model is shown in Fig. 19 for indicating the behavior of users over the change in values of all four inputs for an overall rating. From the fuzzy rule viewer above, for example, we can see when the input parameters of Acting are at 13, Directing at 12, Story at 13, and Visuals at 13, an output of an Overall rating at 13 is obtained. Figure 20 demonstrates the interdependency of criteria and overall rating of TripAdvisor dataset through the control surface obtained from the fuzzy rules as well. The level of overall rating can be depicted as a continuous function of its input parameters as Value aspect, Rooms aspect, Location aspect, Cleanliness aspect, Check in/front desk aspect, Service aspect and Business Service aspect. The behavior of users over the change in values of all seven inputs for an overall rating is demonstrated in Fig. 21.

6.5 Evaluating the prediction accuracy of regression methods

As discussed earlier, in this research also, we implemented two types of SVR regression approaches, ε-SVR and *nu*-SVR and then evaluated them on the experimental datasets for overall rating prediction. The slack variable is ε in ε-SVR and *nu* in *nu*-SVR which requires to be selected appropriately. We applied these two types of regression approaches on the experimental datasets as training and testing data for regression model evaluation. In addition, linear and RBF kernels were considered for them. Therefore, the element kernel is considered linear without parameters and RBF with parameter gamma. The element types were considered for *nu*-SVR are *nu* and C (Complexity Cost) and ε and $C$for ε-SVR.

The results of *nu*-SVR and ε-SVR on data of tensors $\underline{A} \in R^{1,500\times500\times4}$, $\underline{A} \in R^{2,200\times500\times4}$, $\underline{A} \in R^{2,400\times900\times4}$, $\underline{A} \in R^{400\times400\times7}$, $\underline{A} \in R^{700\times700\times7}$ and $\underline{A} \in R^{900\times900\times7}$ are

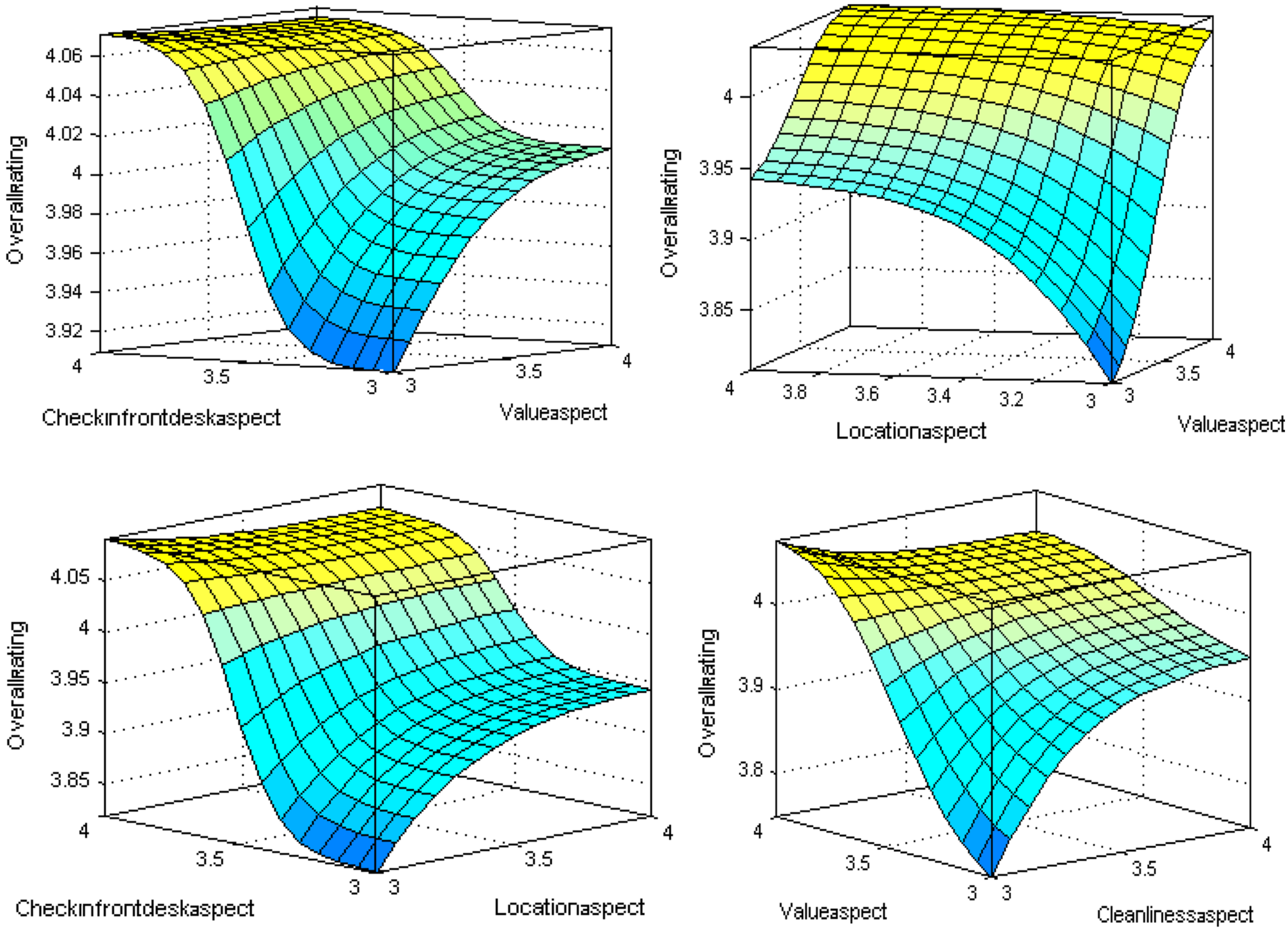


Fig. 20 Interdependency of Overall rating on some criteria of TripAdvisor dataset

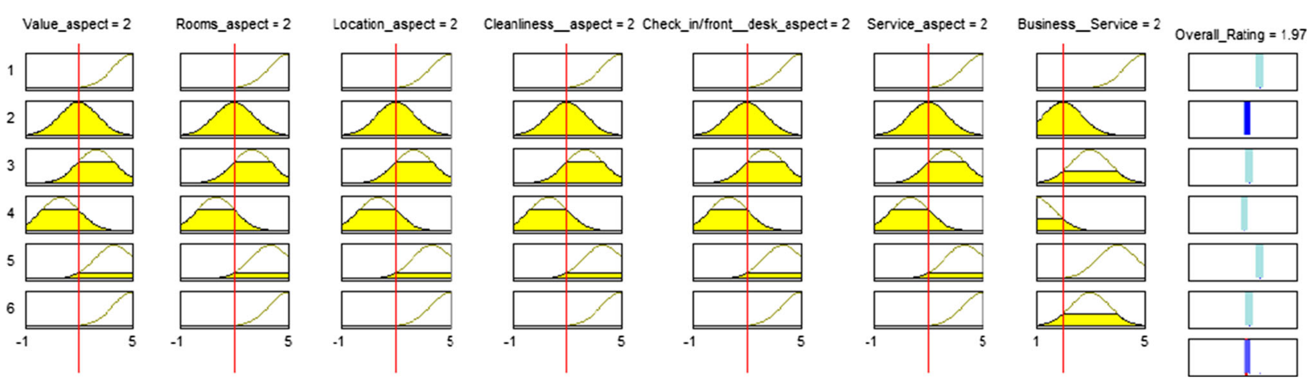


Fig. 21 Fuzzy rule viewer for input and output variables of ANFIS model for TripAdvisor dataset

presented in Tables 17 and 18. The results of SVR regression in Table 17 demonstrate that the ε-SVR using RBF kernel can better predict the overall ratings in relation to the *nu*-SVR with linear and RBF kernels. In addition, the results also show that the accuracy for tensors with higher dimensions is better than lower ones. Also, as presented in Table 18, for $\underline{A} \in R^{400\times400\times7}$, $\underline{A} \in R^{700\times700\times7}$ and $\underline{A} \in R^{900\times900\times7}$, the experimental results show that the ε-SVR using RBF kernel has better predicted the overall ratings in relation to the *nu*-SVR with linear and RBF kernels. Thus, in this paper, we compare the impact of predictive accuracy of ε-SVR using RBF kernel on the MC-CF predictive accuracy in relation to the prediction accuracy of the ANFIS prediction model of the MC-CF predictive accuracy.

Table 17 Results of nu-SVR and epsilon-SVR on Yahoo!Movies dataset

Nu-SVR Kernel: LINEAR		ε−SVR Kernel: LINEAR	
Type	*nu*-SVR	Type	Epsilon-SVR
nu	0.5	Epsilon	0.1
Kernel type	LINEAR	Kernel type	LINEAR
C (complexity cost)	1	*C* (complexity cost)	1
Tensor		$\underline{A} \in R^{1,500\times500\times4}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8714	Pseudo-R^2 (1-error/total)	0.8733
Tensor		$\underline{A} \in R^{2,200\times500\times4}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8745	Pseudo-R^2 (1-error/total)	0.8786
Tensor		$\underline{A} \in R^{2,400\times900\times4}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8767	Pseudo-R^2 (1-error/total)	0.8821
Kernel: RBF		Kernel: RBF	
Type	*nu*-SVR	Type	Epsilon-SVR
nu	0.5	Epsilon	0.1
Kernel type	RBF	Kernel type	RBF
Gamma in kernel function	0	Gamma in kernel function	0
C (complexity cost)	1	*C* (complexity cost)	1
Tensor		$\underline{A} \in R^{1,500\times500\times4}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8811	Pseudo-R^2 (1-error/total)	0.8851
Tensor		$\underline{A} \in R^{2,200\times500\times4}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8823	Pseudo-R^2 (1-error/total)	0.8863
Tensor		$\underline{A} \in R^{2,200\times500\times4}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8868	Pseudo-R^2 (1-error/total)	0.8911

6.6 MC-CF recommender system evaluation

Recommender systems have been evaluated in many, often incomparable ways. Some evaluation metrics assess how close the ratings predicted by a recommender system are to the actual ratings provided by the users. Evaluating a recommender system and determining the accuracy of an algorithm can be performed using statistical or decision support metrics (Adomavicius and Tuzhilin 2005). Statistical accuracy metrics compare estimated utilities by the recommender system against the actual utilities collected from the user (typically ratings). Thus, they are commonly used to evaluate rating prediction task' results. Examples of such metrics are MAE, RMSE, and correlation between the predictions and ratings (Adomavicius and Tuzhilin 2005).

We evaluate the proposed method for MC-CF on data of tensors $\underline{A} \in R^{1,500\times500\times4}$ and $\underline{A} \in R^{400\times400\times7}$. For this evaluations, we consider the classification methods for the RMSE, coverage and precision of the proposed model for Yahoo!Movies and TripAdvisor datasets. Then, we provide several comparisons between the results of K-NN with SVM and BFNN in cases of precision, MAE and F1 measure. In addition, MAE, precision at Top 5 and Top 7 of the proposed method are presented for tensors with different sizes for Yahoo!Movies dataset.

We determined the precision and recall of the Top-N list of each element in the test set and build the arithmetic mean of these values. The recommenders' prediction accuracy is measured by RMSE (Gunawardana and Meek 2009), which is a widely used metric for evaluating the statistical accuracy of the recommendation algorithms, given by:

Table 18 Results of nu-SVR and epsilon-SVR on TripAdvisor dataset

Nu-SVR Kernel: LINEAR		ε−SVR Kernel: LINEAR	
Type	*nu*-SVR	Type	Epsilon-SVR
nu	0.5	Epsilon	0.1
Kernel type	LINEAR	Kernel type	LINEAR
C(complexity cost)	1	C (complexity cost)	1
Tensor		$\underline{A} \in R^{400\times400\times7}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8810	Pseudo-R^2 (1-error/total)	0.8812
Tensor		$\underline{A} \in R^{700\times700\times7}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8825	Pseudo-R^2 (1-error/total)	0.8891
Tensor		$\underline{A} \in R^{900\times900\times7}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8845	Pseudo-R^2 (1-error/total)	0.8953
Kernel: RBF		Kernel: RBF	
Type	*nu*-SVR	Type	Epsilon-SVR
nu	0.5	Epsilon	0.1
Kernel type	RBF	Kernel type	RBF
Gamma in kernel function	0	Gamma in kernel function	0
C (complexity cost)	1	C (complexity cost)	1
Tensor		$\underline{A} \in R^{400\times400\times7}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8928	Pseudo-R^2 (1-error/total)	0.8933
Tensor		$\underline{A} \in R^{700\times700\times7}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.8934	Pseudo-R^2 (1-error/total)	0.8957
Tensor		$\underline{A} \in R^{900\times900\times7}$	
Analysis of variance sum of squares			
Pseudo-R^2 (1-error/total)	0.9016	Pseudo-R^2 (1-error/total)	0.9126

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\Omega|}\sum\nolimits_{u_i,o_j\in\Omega}|a_{ij}-p_{ij}|^2} \tag{21}$$

where $\Omega = \{(u_i, o_j)|u_i$ had rated o_j in the probe set$\}$ and a_{ij} and p_{ij} are the actual and predicted ratings, respectively. A lower value of RMSE indicates a higher accuracy of the recommendation system.

The recommendation quality of proposed model is also evaluated using coverage measures. Coverage measures how many items a recommender system can make recommendation for. Its value can be given in terms of a percentage on either the total number of items, or the number of items in which a user may have some interest. Systems with lower coverage may be less valuable for users, since they are limited in the decisions they are able to help with (Herlocker et al. 1999). Coverage is an important metric, as many modern e-commerce services contain millions of items in the catalog, which should be recommended to customers. The coverage can be defined by following formula:

$$\mathrm{Coverage} = \frac{P_{\mathrm{successful}}}{P_{\mathrm{successful}} + P_{\mathrm{failed}}} \tag{22}$$

where $P_{\mathrm{successful}}$ is the number of successful predictions (as in a prediction was possible to make) evaluated. P_{failed} is the number of failed predictions (as in no prediction could be made).

For YM-5-5 (each movie has at least five ratings), YM-10-10 (each movie has at least 10) ratings and YM-20-20 (each movie has at least 20 ratings), RMSE and coverage obtained from proposed approach are presented in Table 19. From this

Table 19 Coverage and RMSE of proposed method with classifiers for YM-5-5, YM-10-10, and YM-20-20

Size of neighborhoods	Datasets																	
	Coverage and RMSE for proposed model with K-NN						Coverage and RMSE for proposed model with FBNN						Coverage and RMSE for proposed model with SVM					
	YM-5-5		YM-10-10		YM-20-20		YM-5-5		YM-10-10		YM-20-20		YM-5-5		YM-10-10		YM-20-20	
	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage
5	0.55	0.99	0.53	0.99	0.53	0.99	0.55	0.99	0.52	1	0.51	1	0.54	1	0.51	1	0.5	1
15	0.54	0.99	0.53	0.99	0.52	0.99	0.53	0.99	0.52	1	0.51	1	0.52	1	0.51	1	0.5	1
25	0.54	0.99	0.52	0.99	0.52	0.99	0.53	1	0.52	1	0.51	1	0.52	1	0.51	1	0.5	1
35	0.53	0.99	0.52	0.99	0.51	0.99	0.52	1	0.52	1	0.5	1	0.51	1	0.5	1	0.49	1
45	0.53	0.99	0.51	1	0.51	1	0.51	1	0.51	1	0.5	1	0.5	1	0.5	1	0.49	1
55	0.52	1	0.51	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.49	1	0.48	1

table, it is clear that in our proposed method the neighborhood size strongly affects the coverage of MC-CF recommendation. In addition, the recommendation coverage and RMSE are better using SVM in relation to the other classifiers such as K-NN and FBNN. Furthermore, the experimental results on $\underline{A} \in R^{400\times400\times7}$ presented in Table 20 confirm that the recommendation coverage and RMSE are better using SVM in relation to the other classifiers.

To compare the proposed method with the HOSVD, truncated SVD and some state-of-the-art approaches in MC-CF, we employed the decision support accuracy metrics such as recall, precision and F1 metrics, which are widely used in recommender systems to evaluate the quality of recommendations (Bagherifard et al. 2013; Nilashi et al. 2013a; Billsus and Pazzani 2000). These metrics take into consideration how many relevant items are recommended by the recommender system. Usually, the ranking of recommended items is also considered in the metric values. Thus, these metrics are better suited in cases where it is important to test if lists of recommended items contain items that are valuable for the user.

Recall indicates the ability of a system to present all relevant items (Sarwar et al. 2000). In reality, it may not be possible to retrieve all the relevant items from a collection, especially when the collection is large. A system may be able to retrieve a proportion of the total relevant items. Thus, the performance of a system is often measured by the recall ratio, which denotes the percentage of the relevant items retrieved in a given situation. Precision implies the ability of a system to present only the relevant items. This relates to its ability to not retrieve non-relevant item (Sarwar et al. 2000). This factor, that is how far the system is able to withhold unwanted items in a given situation, is measured in terms of precision ratio. Table 21 shows Contingency table for computing precision and recall. These precision and recall measures are presented by Eq. (23) and Eq. (24), respectively.

$$Precision(Reclist) = \frac{|\{relevant\ items\} \cap \{top-N\ items\}|}{|\{top-N\ items\}|} = \frac{N_{SR}}{N_{ST}} \quad (23)$$

$$Recall(Reclist) = \frac{|\{relevant\ items\} \cap \{top-N\ items\}|}{|\{relevant\ items\}|} = \frac{N_{SR}}{N_{TR}} \quad (24)$$

Several approaches have been taken to combine Precision and Recall into a single metric. One approach is the F1 metric as presented in Eq. (25), which combines precision and recall into a single number with a free parameter beta β, as a harmonic mean (Gedikli and Jannach 2013; Jannach et al. 2012; Billsus and Pazzani 2000; Cho et al. 2007).

$$F_\beta = \frac{(\beta^2+1)Precision \times Recall}{\beta^2 \times \Pr ecision + Recall} \quad (25)$$

Table 20 Coverage and RMSE of proposed method with classifiers for TripAdvisor dataset

Size of neighborhoods	Methods					
	Coverage and RMSE for proposed model with K-NN		Coverage and RMSE for proposed model with FBNN		Coverage and RMSE for proposed model with SVM	
	RMSE	Coverage	RMSE	Coverage	RMSE	Coverage
5	0.53	0.99	0.51	1	0.50	1
15	0.50	1	0.49	1	0.47	1
25	0.49	1	0.48	1	0.45	1
35	0.47	1	0.46	1	0.43	1
45	0.46	1	0.43	1	0.41	1
55	0.45	1	0.42	1	0.40	1

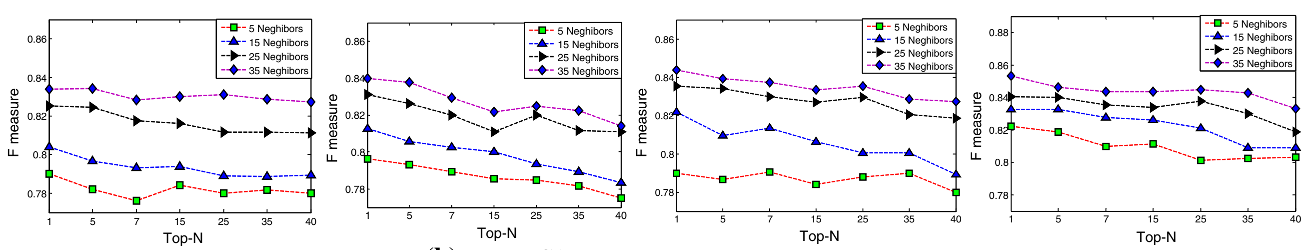
Table 21 Contingency table for computing precision and recall

	Selected	Unselected	Total (selected unselected)
Relevant	N_{SR}	N_{UR}	N_{TR}
Irrelevant	N_{SI}	N_{UI}	N_{TI}
Total	N_{ST}	N_{UT}	N_{TT}

where parameter $\beta \in [0, 1]$ determines the relative influence of both metrics (the value $\beta = 1$ is commonly used).

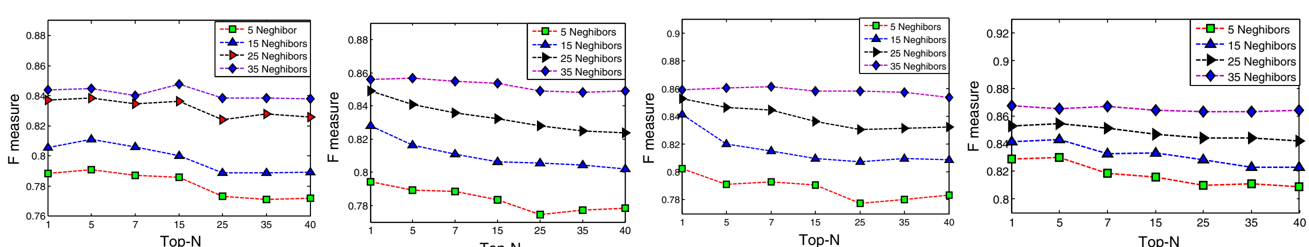
We run the experiments on datasets YM-10-10 and YM-20-20 datasets for N equal 1, 5, 7, 15, 25, 35, and 40, where N is the number of items to be recommended by the Top-N recommender systems. We compare the result of MC-CF recommendation quality using F1-metric for K-NN, K-NN-PCA, SVM and FBNN approaches. These comparisons are made for YM-10-10 and YM-20-20 for different Top-N recommendations (N = 1, 5, 7, 15, 25, 35 and 40). From all the F1 curves in Figs. 22 and 23, we can notice that high accuracy is obtained by the proposed method for all classification approaches when the size of neighbors is increased versus the Top-N recommendations.

It should be noted, according to the rank 12 approximation defined for HOSVD decomposition and the number of clusters determined for recommendation process, we present the precision of some clusters. For rank 12 approximation, the defined number of clusters is changed iteratively: starting from the three clusters, after each iteration number of clusters is increased by 3 until 12. These clusters are selected for testing the precision of MC-CF recommendation. Precision@5 and Precision@7 are considered for this test for YM-10-10 and YM-20-20. Figure 24 illustrates the precision value for Precision@5 and Precision@7 versus number of clusters.



(a) **K-NN** **(b)** **K-NN-PCA** **(c)** **FBNN** **(d)** **SVM**

Fig. 22 F1 measure and Top-N recommendation of YM-10-10 for tensor $\underline{A} \in R^{1{,}000\times 800\times 4}$



(a) **K-NN** **(b)** **K-NN-PCA** **(c)** **FBNN** **(d)** **SVM**

Fig. 23 F1 measure and Top-N recommendation of YM-20-20 YM-10-10 for tensor $\underline{A} \in R^{300\times 200\times 4}$

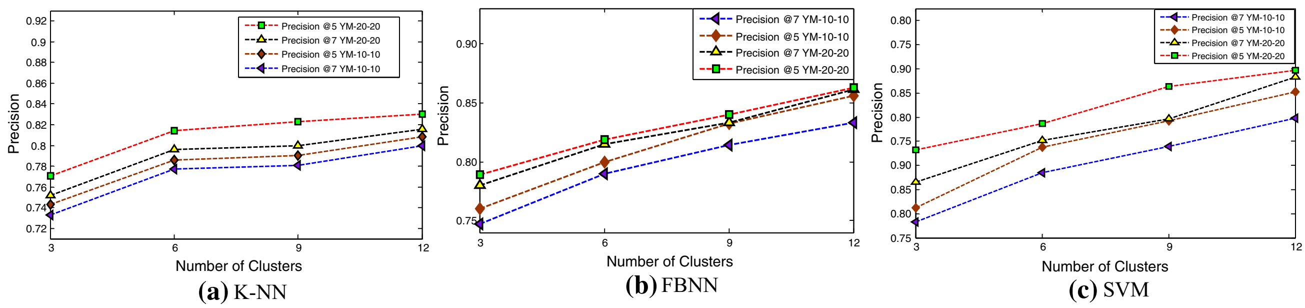


Fig. 24 Precision versus number of clusters of YM-10-10 and YM-20-20 for proposed method with K-NN, FBNN and SVM classifiers

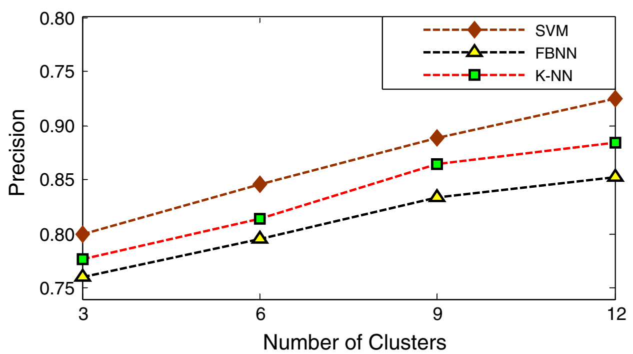


Fig. 25 Precision versus number of clusters in $\underline{A} \in R^{400\times400\times7}$ for proposed method with K-NN, FBNN and SVM classifiers

In addition, as can be seen in Fig. 24, the worst precision is obtained for YM-10-10 at precision@5 in the third cluster and best precision is obtained for YM-20-20 at Precision@5 in the twelfth cluster. The result of this evaluation demonstrates that for YM-10-10 and YM-20-20 the precision is increased with increasing the number of clusters. However, the recommendation quality is slightly better for our proposed method using SVM. Furthermore, the results clearly reveal that the proposed method gives better recommendation quality for YM-20-20. Similarly, TripAdvisor dataset, the results are demonstrated in Fig. 25.

We also evaluated the efficiency of the proposed method. Given the scalability challenge, in this paper, HOSVD assisted to (1) factorize large tensors efficiently using much less time than standard methods, while at the same time (2) obtain low-rank factors that preserve the main variance of the tensor of data. Thus, due to the dimensionality reduction, we could better form and pre-compute the neighborhood where the prediction generation was much faster. This means that forming neighborhoods in the low-dimensional eigenspace provided better quality and performance. In addition, after decomposing the tensor by HOSVD, data has effectively been clustered using cosine-based similarity was performed in an effective way, which caused a good performance is obtained in MC-CF.

To experimentally show the effectiveness of clustering using HOSVD and cosine-based similarity, we performed the experiments on similarity-based approach developed by Adomavicius and Kwon (2007) and compared with the proposed method. They proposed different potential ways to calculate the similarity between users based on their criteria ratings. It should be noted that Chebyshev distance metric performed best among their similarity-based approaches.

Figure 26 presents the performance results of our experiments for proposed method and similarity-based approach using Chebyshev distance metric. The throughput is plotted as a function of the cluster size demonstrated in Fig. 26. In addition, for the proposed method, the throughput was tested for different classification methods used in the online

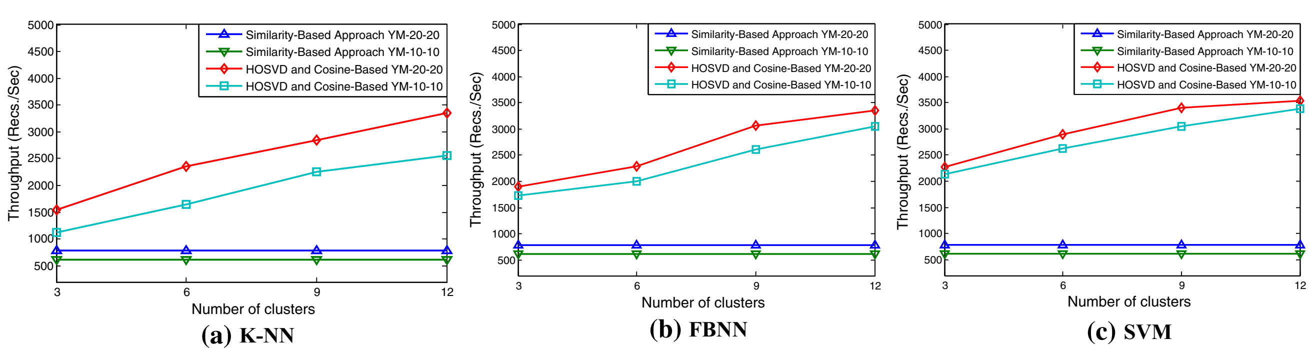


Fig. 26 Throughput of proposed method with K-NN, FBNN and SVM vs. similarity-based approach

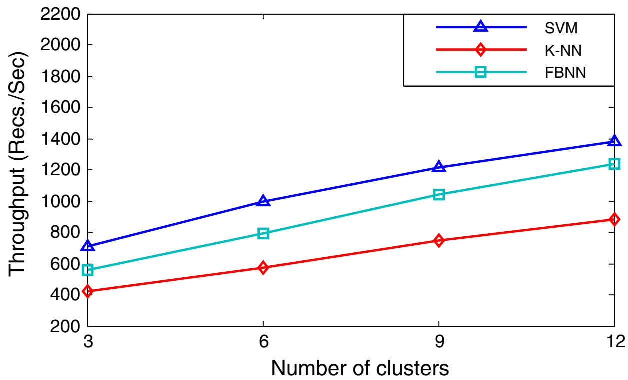


Fig. 27 Throughput of proposed method with K-NN, FBNN and SVM on $\underline{A} \in R^{400\times400\times7}$

phase. We define throughput of a MC-CF recommender system as the number of recommendations for k users (k = 5) generated per second. From the curves in this plot, we see that using the HOSVD and cosine-based approaches for clustering the high-dimensional data, the throughput is substantially higher than the MC-CF based on similarity-based approach. This is due to the reason that with the clustered approach using HOSVD and cosine-based similarity the prediction algorithm uses a fraction of neighbors. The throughput of multi-criteria recommender system increases rapidly with the increase in the number of clusters with the small sizes. Since the MC-CF based on similarity approach has to scan through all the neighbors, the number of clusters does not impact the throughput. These results show that the proposed method is more efficient than similarity-based approach using Chebyshev distance metric. Figure 27 shows the performance results of the experiments for proposed method on TripAdvisor dataset.

To proposed method be comparable with previous works, we also evaluated our approach on the YM-10-10 using Precision@5, Precision@7 and MAE on different sizes of tensors of users, item and criteria.

MAE presented in Eq. (26) measures the average absolute deviation between a predicted rating and the user's true rating. The MAE is computed by first summing these absolute errors of the corresponding N rating predictions for all the M users, and then averaging the sum by the total number of users. The lower the MAE, the more accurately the recommender predicts ratings.

$$\mathrm{MAE}(\mathrm{pred}, \mathrm{act}) = \sum_{i=1}^{N} \left| \frac{\mathrm{pred}_{u,i} - \mathrm{act}_{u,i}}{N} \right| \quad (26)$$

where N is the number of items that user u has expressed an opinion.

In Tables 22 and 23, we report Precision@5 and Precision@7 values as well as the MAE for all approaches implemented in this paper. As can be seen in these tables, the evaluation was also performed for SVD and HOSVD techniques without using ANFIS with subtractive clustering on YM-10-10 and YM-20-20 datasets. Table 24 presents the MAE, precision of proposed method, HOSVD and truncated SVD for $\underline{A} \in R^{400\times400\times7}$, $\underline{A} \in R^{700\times700\times7}$ and $\underline{A} \in R^{900\times900\times7}$.

From the results presented in Tables 22, 23 and 24, we can find that the precision at Top-5 and Top-7 of the proposed method is higher than algorithms that solely use approaches such as truncated SVD and HOSVD for YM-10-10. In addition, the MAE is relatively low for tensors with higher ranks. Moreover, according to the experiment results presented in the above tables, the combination of HOSVD, ANFIS, subtractive clustering and SVM provides the best predictive accuracy for MC-CF with lowest MAE.

Table 22 MAE, precision at Top 5 and Top 7 of proposed method, HOSVD and truncated SVD for YM-10-10

Tensors	$\underline{A} \in R^{500\times400\times4}$			$\underline{A} \in R^{800\times700\times4}$			$\underline{A} \in R^{1000\times800\times4}$		
Algorithm	Precision@5	Precision@7	MAE	Precision@5	Precision@7	MAE	Precision@5	Precision@7	MAE
HOSVD	75.34	72.85	1.17	75.82	73	1.136	76.4	73.27	1.105
Truncated SVD	74.03	72.19	1.75	74.73	72.73	1.7	75.6	73.36	1.669
HOSVD, SVM and SVR	79.1	78.78	1.13	79.56	79.29	0.978	81.14	79.89	0.895
HOSVD, ANFIS, subtractive clustering and K-NN	81.44	80.78	0.96	81.6	81.2	0.908	82.27	81.36	0.875
HOSVD, ANFIS, subtractive clustering and K-NN-PCA	81.89	81.02	0.94	82.27	81.27	0.908	82.64	81.55	0.875
HOSVD, ANFIS, subtractive clustering and FBNN	82.23	81.56	0.93	82.55	81.73	0.9	83.2	82.7	0.859
HOSVD, ANFIS, subtractive clustering and SVM	85.11	84.18	0.89	85.6	84.4	0.867	86.27	85.26	0.818

Table 23 MAE, precision at Top 5 and Top 7 of proposed method, HOSVD and truncated SVD for YM-20-20

Tensors	$\underline{A} \in R^{150\times100\times4}$			$\underline{A} \in R^{250\times150\times4}$			$\underline{A} \in R^{300\times200\times4}$		
Algorithm	Precision@5	Precision@7	MAE	Precision@5	Precision@7	MAE	Precision@5	Precision@7	MAE
HOSVD	78.57	76.43	0.95	79.2	78.90	0.95	79.5	78.19	0.88
Truncated SVD	75.12	73.21	1.45	75.9	74.7	1.45	77.4	76.79	1.28
HOSVD, SVM and SVR	82.98	80.27	0.98	83.45	81.12	0.923	83.89	83.13	0.845
HOSVD, ANFIS, subtractive custering and K-NN	83.34	81.32	0.91	83.8	81.67	0.87	84.4	83.87	0.829
HOSVD, ANFIS, subtractive custering and K-NN-PCA	84.71	82.82	0.89	85.9	84.56	0.89	86.2	85.56	0.8
HOSVD, ANFIS, subtractive custering and FBNN	85.17	83.64	0.88	86.8	85.41	0.88	87.3	86.87	0.778
HOSVD, ANFIS, subtractive custering and SVM	89.25	86.13	0.84	90.1	89.11	0.84	90.8	90.1	0.749

Table 24 MAE, precision of proposed method, HOSVD and truncated SVD for TripAdvisor dataset

Tensors	$\underline{A} \in R^{150\times100\times4}$		$\underline{A} \in R^{700\times700\times7}$		$\underline{A} \in R^{900\times900\times7}$	
Algorithm	Precision	MAE	Precision	MAE	Precision	MAE
HOSVD	79.21	0.93	80.22	0.92	82.5	0.86
Truncated SVD	76.16	1.34	76.72	1.23	79.4	1.13
HOSVD, SVM and SVR	83.12	0.93	85.32	0.89	85.75	0.82
HOSVD, ANFIS, Subtractive Clustering and K-NN	84.21	0.90	86.32	0.87	87.76	0.814
HOSVD, ANFIS, Subtractive Clustering and K-NN-PCA	85.61	0.88	87.45	0.86	88.29	0.79
HOSVD, ANFIS, Subtractive Clustering and FBNN	86.21	0.86	89.67	0.82	89.76	0.761
HOSVD, ANFIS, Subtractive Clustering and SVM	90.75	0.82	92.15	0.81	93.8	0.721

7 Conclusions and future work

In this paper, a new method was proposed for MC-CF recommendation using dimensionality reduction, Neuro-Fuzzy and classification techniques. The main goals of the paper were defined to improve the MC-CF predictive accuracy and overcome some shortcomings such as sparsity, scalability and uncertainty of representing and reasoning user behavior and perception about items' features. We applied HOSVD, ANFIS with subtractive clustering and classification methods for dimensionality reduction, knowledge discovery and classification tasks, respectively. The approaches in the proposed method were used in two phases, offline and online. In the offline phase, we provided a model for MC-CF using HOSVD and ANFIS with subtractive clustering. For predicting the new user class in the online phase, SVM, K-NN and FBNN were used.

HOSVD reduced the noise of high-dimensional data effectively and improved the scalability problem. Also, using HOSVD, all factors in the third-order tensor of users, items and criteria were considered all together to reveal latent relationships between them. The result of applying the HOSVD method on the dataset assisted system to form high-quality clusters using the cosine-based similarity. Furthermore, tensor decomposition using HOSVD on the dataset demonstrated its advantages in case of dimensionality reduction in more than 2 dimensions for obtaining a favorable approximation of the information.

The experimental results on Yahoo!Movies and TripAdvisor datasets clearly demonstrated the capability of ANFIS modeling by MFs and rules without the human expert intervention in MC-CF. Besides, the model of ANFIS combined with subtractive was used to extract knowledge from user ratings and preferences on items' features. This was done by incorporating the element of training into the existing Neuro-Fuzzy system. The results also indicated that Neuro-Fuzzy provides a better and more reliable model for overall ratings prediction. In addition, by the training element of ANFIS the rules and the MFs were properly tuned to predict the overall ratings that had a better advantage in terms of the simplicity of the algorithm and the speed of the training convergence. In addition, users' ratings on items in MC-CF are accumulated overtime and fuzzy rules can be amended and maintained in a rules database for prediction task. The advantage of this method is its flexibility and extendibility which can be developed for datasets with many features.

We analyzed the predictive accuracy of our proposed method in the domain of movie and hotel recommendation based on a real-world datasets. In addition, we evaluated the proposed method in cases of RMSE, MAE and precision metrics to be comparable in relation to the previous works. Our experiments confirmed that the combination of HOSVD technique, ANFIS combined with subtractive clustering and classifiers outperforms the previous methods and significantly leads to the improvement in predictive accuracy of MC-CF measured in terms of standard accuracy metrics. Furthermore, the predictive accuracy of the proposed method with combination of HOSVD, ANFIS with subtractive clustering and SVM classification was better than methods developed in this paper such as HOSVD and ANFIS with K-NN and FBNN approaches and even HOSVD and SVR.

In the proposed method, ANFIS models of items and users are updated in the offline; however, in the MC-CF recommenders, data are dramatically updated and therefore incremental learning approaches are needed to consider new arrived ratings. In addition, in conducting this research, lack of multi-criteria dataset was one of the main barriers in evaluating the proposed model and incorporated methods. And the methods and recommendation model were evaluated solely on one multi-criteria dataset without incorporating tags and content of users and items. Thus, for the future work, the performance of the hybrid methods will be examined on large and different types of multi-criteria datasets for further validating the predictive accuracy and recommendation quality. Furthermore, future studies will focus on further improvement of the MC-CF recommendation accuracy by incorporating incremental techniques and other resource to the tensor ratings such as tag and content of users and items. Moreover, we will focus on developing incremental tensor decomposition and Neuro-Fuzzy as it is a shortcoming in this and the previous researches. The followings are the recommendations for future study in this area:

- Although multi-criteria ratings can be a good choice for pure CF recommendation, the accuracy of MC-CF can be improved more with incorporating other resources such as tags and content of users and items to the tensor of ratings. With incorporating these resources, the fuzzy semantic techniques can be applied to better alleviate sparsity problem and enhance the MC-CF recommendation accuracy.
- The proposed multi-criteria recommendation model can be extended to the incremental-based recommendation. Therefore, future studies will focus on further improvement of the MC-CF recommendation accuracy and efficiency by incorporating incremental techniques using incremental HOSVD and incremental ANFIS.

Acknowledgments The authors would like to acknowledge the financial support from Fundamental Research Grant Scheme (FRGS) (No. FRGS/1/2014/ICT07/UTM/02/1; vote no R.J130000.7806.4F466) of the Ministry of Education (MOE), Malaysia. The authors also acknowledge the Research Management Centre (RMC) of the Universiti Teknologi Malaysia (UTM) for providing excellent research environment to complete this work. Appreciation also goes to the anonymous reviewers whose comments helped us to improve the manuscript.

References

Adomavicius G, Kwon Y (2007) New recommendation techniques for multicriteria rating systems. Intell Syst IEEE 22(3):48–55

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowl Data Eng IEEE Trans 17(3):734–749

Ahmad W, Khokhar A (2007) An architecture for privacy preserving collaborative filtering on web portals. In: information assurance and security. IAS 2007. Third international symposium on, 2007. IEEE, pp 273–278

Amatriain X, Jaimes A, Oliver N, Pujol JM (2011) Data mining methods for recommender systems. In: recommender systems handbook. Springer, pp 39–71

Avci E (2008) Comparison of wavelet families for texture classification by using wavelet packet entropy adaptive network based fuzzy inference system. Appl Soft Comput 8(1):225–231

Bagherifard K, Nilashi M, Ibrahim O, Ithnin N, Nojeem LA (2013) Measuring semantic similarity in grids using ontology. Int J Innov Appl Stud 2(3):230–237

Bellocchio F, Ferrari S, Piuri V, Borghese NA (2012) Hierarchical approach for multiscale support vector regression. Neural Netw Learn Syst IEEE Trans 23(5):1448–1460

Bilge A, Polat H (2013) A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. Inf Process Manag 49(4):912–927

Billsus D, Pazzani MJ (2000) Learning collaborative information filters. In: Proceedings of the fifteenth international conference on machine learning, p 48

Bobadilla J, Ortega F, Hernando A (2012) A collaborative filtering similarity measure based on singularities. Inf Process Manag 48(2):204–217

Bobadilla J, Ortega F, Hernando A, Alcalá J (2011) Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowl Based Syst 24(8):1310–1316

Bouchachia A, Pedrycz W (2006) Enhancement of fuzzy clustering by mechanisms of partial supervision. Fuzzy Sets Syst 157(9):1733–1759

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the fourteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., pp 43–52

Buragohain M, Mahanta C (2008) A novel approach for ANFIS modelling based on full factorial design. App Soft Comput 8(1):609–625

Cao Y, Li Y (2007) An intelligent fuzzy-based recommendation system for consumer electronic products. Expert syst Appl 33(1):230–240

Carbo J, Molina JM (2004) Agent-based collaborative filtering based on fuzzy recommendations. Int J Web Eng Technol 1(4):414–426

Castellano G, Fanelli A, Torsello M (2007) A neuro-fuzzy collaborative filtering approach for web recommendation. Int J Comput Sci 1(1):27–29

Cechinel C, Sicilia M-Á, Sánchez-Alonso S, García-Barriocanal E (2013) Evaluating collaborative filtering recommendations inside large learning object repositories. Inf Process Manag 49(1):34–50

Cetişli B, Barkana A (2010) Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training. Soft Comput 14(4):365–378

Chang C-C, Lin C-J (2001) LIBSVM: a library for support vector machines, software, www.csie.ntu.edu.tw/~cjlin/libsvm

Chen G, Wang F, Zhang C (2009) Collaborative filtering using orthogonal nonnegative matrix tri-factorization. Inf Process Manag 45(3):368–379

Chiu SL (1994) Fuzzy model identification based on cluster estimation. J Intell Fuzzy Syst 2(3):267–278

Cho J, Kwon K, Park Y (2007) Collaborative filtering using dual information sources. Intell Syst IEEE 22(3):30–38

Christakou C, Vrettos S, Stafylopatis A (2007) A hybrid movie recommender system based on neural networks. Int J Artif Intell Tools 16(05):771–792

de Campos LM, Fernández-Luna JM, Huete JF (2008) A collaborative recommender system based on probabilistic inference from fuzzy observations. Fuzzy Sets Syst 159(8):1554–1576

De Lathauwer L, De Moor B, Vandewalle J (2000) On the best rank-1 and rank-(R 1, R 2,., Rn) approximation of higher-order tensors. SIAM J Matrix Anal Appl 21(4):1324–1342

De Lathauwer L (2004) First-order perturbation analysis of the best rank-(R1, R2, R3) approximation in multilinear algebra. J Chem 18(1):2–11

Deshpande M, Karypis G (2004) Item-based top-n recommendation algorithms. ACM Trans Inf Syst TOIS 22(1):143–177

Destercke S (2012) A k-nearest neighbours method based on imprecise probabilities. Soft Comput 16(2):833–844

Drucker H, Shahrary B, Gibbon DC (2001) Relevance feedback using support vector machines. In: ICML, pp 122–129

Ferrari S, Bellocchio F, Piuri V, Borghese NA (2010) Multi-scale support vector regression. In: Neural Networks (IJCNN), The 2010 international joint conference on IEEE, pp 1–7

Gao M, Wu Z (2009) Personalized context-aware collaborative filtering based on neural network and slope one. In: cooperative design, visualization, and engineering. Springer, pp 109–116

Gedikli F, Jannach D (2013) Improving recommendation accuracy based on item-specific tag preferences. ACM Trans Intell Syst Technol TIST 4(1):11

Georgiou O, Tsapatsoulis N (2010) Improving the scalability of recommender systems by clustering using genetic algorithms. In: artificial neural networks-ICANN 2010. Springer, pp 442–449

Gong S, Ye H (2009) An item based collaborative filtering using bp neural networks prediction. In: industrial and information systems. IIS'09. International conference on, 2009. IEEE, pp 146–148

Grcar M, Fortuna B, MladeniF D, Grobelnik M (2006) kNN versus SVM in the collaborative filtering framework. In: data science and classification. Springer, pp 251–260

Gunawardana A, Meek C (2009) A unified approach to building hybrid recommender systems. In: Proceedings of the third ACM conference on Recommender systems. ACM, pp 117–124

Hanani U, Shapira B, Shoval P (2001) Information filtering: overview of issues, research and systems. User Model User Adapt Interact 11(3):203–259

Hayajneh MT, Hassan AM, Al-Wedyan F (2010) Monitoring defects of ceramic tiles using fuzzy subtractive clustering-based system identification method. Soft Comput 14(3):615–626

Stormer H, Werro N, Risch D (2006) Recommending products with a fuzzy classification. Europe, CollECTeR

Herlocker JL, Konstan JA, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 230–237

Igelnik B, Pao Y-H (1995) Estimation of size of hidden layer on basis of bound of generalization error. In: Proceedings of neural networks. IEEE international conference on, 1995. IEEE, pp 1923–1927

Jang J-S (1993) ANFIS: Adaptive-network-based fuzzy inference system. Syst Man Cybern IEEE Trans 23(3):665–685

Jannach D (2008) Finding preferred query relaxations in content-based recommenders. In: intelligent techniques and tools for novel system architectures. Springer, pp 81–97

Jannach D, Karakaya Z, Gedikli F (2012) Accuracy improvements for multi-criteria recommender systems. In: Proceedings of the 13th ACM conference on electronic commerce. ACM, pp 674–689

Jannach D, Zanker M, Felfernig A, Friedrich G (2010) Recommender systems: an introduction. Cambridge University Press

Jeong B, Lee J, Cho H (2009) An iterative semi-explicit rating method for building collaborative recommender systems. Expert Syst Appl 36(3):6181–6186

Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. Springer, Berlin, pp 137–142

Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In: Proceedings of the European conference on machine learning, Berlin

Kaufinan L, Rousseeuw PJ (1990) Finding groups in data: an introduction to Cluster analysis, Wiley

Kim H-N, Ji A-T, Ha I, Jo G-S (2010) Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. Electron Commer Res Appl 9(1):73–83

Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500

Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J (1997) GroupLens: applying collaborative filtering to Usenet news. Commun ACM 40(3):77–87

Lee M, Choi P, Woo Y (2002) A hybrid recommender system combining collaborative filtering with neural network. In adaptive hypermedia and adaptive web-based systems. Springer, Berlin, pp 531–534

Lee PY, Hui SC, Fong ACM (2002) Neural networks for web content filtering. Intell Syst IEEE 17(2):48–57

Lee Y-J, Mangasarian OL (2001) SSVM: a smooth support vector machine for classification. Comput Optim Appl 20(1):5–22

Leginus M, Zemaitis V (2011) Speeding up tensor based recommenders with clustered tag space and improving quality of recommendations with non-negative tensor factorization. Master's thesis, Aalborg University

Lesaffre M, Leman M (2007) Using fuzzy logic to handle the users' semantic descriptions in a music retrieval system. In: theoretical advances and applications of fuzzy logic and soft computing. Springer, pp 89–98

Li Q, Myaeng SH, Kim BM (2007) A probabilistic music recommender considering user opinions and audio features. Inf Process Manag 43(2):473–487

Li Q, Wang C, Geng G (2008) Improving personalized services in mobile commerce by a novel multicriteria rating approach. In: Proceedings of the 17th international conference on World Wide Web, pp 1235–1236

Linden G, Smith B, York J (2003) Amazon. com recommendations: item-to-item collaborative filtering. Int Comput IEEE 7(1):76–80

Liu L, Mehandjiev N, Xu D-L (2011) Multi-criteria service recommendation based on user criteria preferences. In: Proceedings of the fifth ACM conference on recommender systems. ACM, pp 77–84

Luo X, Xia Y, Zhu Q (2012) Incremental collaborative filtering recommender based on regularized matrix factorization. Knowl Based Syst 27:271–280

Manouselis N, Costopoulou C (2007) Experimental analysis of design choices in multiattribute utility collaborative filtering. Int J Pattern Recognit Artif Intell 21(02):311–331

Murphey YL, Luo Y (2002) Feature extraction for a multiple pattern classification neural network system. In: Proceedings of pattern recognition. 16th international conference on, 2002. IEEE, pp 220–223

Nauck D (1997) Neuro-fuzzy systems: review and prospects. In: Proceedings of fifth European congress on intelligent techniques and soft computing (EUFIT'97). pp 1044–1053

Nilashi M, Ibrahim O, Bagherifard K, Janahmadi N, Barisami M (2011c) Application of k-nearest neighbour predictor for classifying online customer trust. J Theor Appl Inf Technol 36(1):18–25

Nilashi M, Ibrahim OB (2013b) A model for detecting customer level intentions to purchase in B2C Websites using TOPSIS and fuzzy logic rule-based system. Arab J Sci Eng, pp 1–16

Nilashi M, Bagherifard K, Ibrahim O, Alizadeh H, Nojeem LA, Roozegar N (2013a) Collaborative filtering recommender systems. Res J Appl Sci Eng Technol 5(12):4168–4182

Nilashi M, Bagherifard K, Ibrahim O, Janahmadi N, Barisami M (2011a) An application expert system for evaluating effective factors on trust in B2C Websites. Engineering 3:7

Nilashi M, Fathian M, Gholamian MR, Ibrahim OB, Talebi A, Ithnin N (2011b) A comparative study of adaptive neuro fuzzy inferences system (ANFIS) and fuzzy inference system (FIS) approach for trust in B2C electronic commerce websites. JCIT 6(5):25–43

Nilashi M, Ibrahim O, Ithnin N (2014) Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system. Knowl Based Syst 60:82–101

Nilashi M, Ibrahim OB, Ithnin N (2014) Hybrid recommendation approaches for multi-criteria collaborative filtering. Expert Syst Appl 41(8):3879–3900

O'Connor P (2008) User-generated content and travel: a case study on tripadvisor. com. In: information and communication technologies in tourism 2008. Springer, pp 47–58

Ou G, Murphey YL (2007) Multi-class pattern classification using neural networks. Pattern Recognit 40(1):4–18

Park DH, Kim HK, Choi IY, Kim JK (2012) A literature review and classification of recommender systems research. Expert Syst Appl 39(7):10059–10072

Park Y-J, Chang K-N (2009) Individual and group behavior-based customer profile model for personalized product recommendation. Expert Syst Appl 36(2):1932–1939

Pazzani MJ (1999) A framework for collaborative, content-based and demographic filtering. Artif Intell Rev 13(5–6):393–408

Petrovic-Lazarevic S, Coghill K, Abraham A (2004) Neuro-fuzzy modelling in support of knowledge management in social regulation of access to cigarettes by minors. Knowl Based Syst 17(1):57–60

Pinto MA, Tanscheit R, Vellasco M (2012) Hybrid recommendation system based on collaborative filtering and fuzzy numbers. In: fuzzy systems (FUZZ-IEEE), IEEE international conference on, 2012. IEEE, pp 1–6

Postorino MN, Sarne GM (2011) A neural network hybrid recommender system. In: neural nets WIRN10. In: Proceedings of the 20th Italian workshop on neural nets, 2011. IOS Press, p 180

Rennie JD, Srebro N (2005) Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd international conference on machine learning. ACM, pp 713–719

Sahoo N, Krishnan R, Duncan G, Callan J (2011) Research note-the halo effect in multicomponent ratings and its implications for recommender systems: the case of Yahoo!. Movies. Inf Syst Res 23(1):231–246

Sahoo N, Krishnan, R, Duncan, G, Callan JP (2006) Collaborative filtering with multi-component rating for recommender systems. In: Proceedings of the sixteenth workshop on information technologies and systems

Sarwar B, Karypis G, Konstan J, Riedl J (2000) Application of dimensionality reduction in recommender system-a case study. DTIC document

Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. ACM, pp 285–295

Schölkopf B, Smola A (2002) Learning with kernels. MIT Press, Cambridge

Sen S, Vig J, Riedl J (2009) Tagommenders: connecting users to items through tags. In: Proceedings of the 18th international conference on World wide web. ACM, pp 671–680

Si L, Jin R (2003) Flexible mixture model for collaborative filtering. In: Proceedings of the 20th international conference on machine learning. D.C, Washington, p 704

Silva C, Ribeiro B (2007) On text-based mining with active learning and background knowledge using SVM. Soft Comput 11(3):519–530

Sugeno M (1985) Industrial applications of fuzzy control. Elsevier

Sun J-T, Zeng H-J, Liu H, Lu Y, Chen Z (2005) CubeSVD: a novel approach to personalized web search. In: Proceedings of the 14th international conference on World Wide Web. ACM, pp 382–390

Symeonidis P, Nanopoulos A, Manolopoulos Y (2008a) Providing justifications in recommender systems. Syst Man Cybern Part A Syst Humans IEEE Trans 38(3):1262–1272

Symeonidis P, Nanopoulos A, Manolopoulos Y (2008) Tag recommendations based on tensor dimensionality reduction. In: Proceedings of the 2008 ACM conference on recommender systems. ACM, pp 43–50

Symeonidis P, Ruxanda M, Nanopoulos A, Manolopoulos Y (2008) Ternary semantic analysis of social tags for personalized music recommendation. In: ISMIR'08: Proceedings of the 9th international conference on music information retrieval. Citeseer, pp 219–224

Tang TY, McCalla G (2009) The pedagogical value of papers: a collaborative-filtering based paper recommender. J Digit Inf 10(2)

Tsai C-F, Hung C (2012) Cluster ensembles in collaborative filtering recommendation. Appl Soft Comput 12(4):1417–1425

Vapnik V (1995) The nature of statistical learning theory. Springer, Berlin

Vapnik V (1998) Statistical Learning Theory. Wiley, New York

Vapnik V, Golowich S, Smola A (1996) Support vector method for function approximation, regression estimation, and signal processing. Adv Neural Inf Process Syst 9:281–287

Vapnik V (1995) The nature of statistical learning theory. Springer, New York

Villalba SD, Cunningham P (2007) An evaluation of dimension reduction techniques for one-class classification. Artif Intell Rev 27(4):273–294

Xia Z, Dong Y, Xing G (2006) Support vector machines for collaborative filtering. In: Proceedings of the 44th annual Southeast regional conference. ACM, pp 169–174

Xu Y, Zhang L, Liu W (2006) Cubic analysis of social bookmarking for personalized recommendation. In: frontiers of WWW research and development-APWeb 2006. Springer, pp 733–738

Yager RR (2003) Fuzzy logic methods in recommender systems. Fuzzy Sets Syst 136(2):133–149

Yang Y, Liu X (1999) A re-examination of text categorization methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 42–49

Yazdani A, Ebrahimi T, Hoffmann U (2009) Classification of EEG signals using Dempster Shafer theory and a k-nearest neighbor classifier. In: neural engineering. NER'09. 4th international IEEE/EMBS conference on, 2009. IEEE, pp 327–330

Zadeh LA (1965) Fuzzy Sets. Inf Control 8(3):338–353

Zhang F, Chang H-Y (2006) A collaborative filtering algorithm employing genetic clustering to ameliorate the scalability issue.

In: e-Business Engineering. ICEBE'06. IEEE international conference on, 2006. IEEE, pp 331–338
Zhang T, Iyengar VS (2002) Recommender systems using linear classifiers. J Mach Learn Res 2:313–334
Zhang Z, Ye N (2011) Learning a tensor subspace for semi-supervised dimensionality reduction. Soft Comput 15(2):383–395
Zhou L, Lai KK, Yu L (2009) Credit scoring using support vector machines with direct search for parameters selection. Soft Comput 13(2):149–155