Iranian Journal of Fuzzy Systems

Volume 18, Number 6, (2021), pp. 83-99

Original paper

University of Sistan and Baluchestan

A fuzzy approach to review-based recommendation: Design and optimization of a fuzzy classification scheme based on implicit features of textual reviews

S. Hasanzadeh[1], S. M. Fakhrahmad[2] and M. Taheri[3]

[1,2,3] *Department of Computer Science and Engineering, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran*

s.hasanzadeh@shirazu.ac.ir, fakhrahmad@shirazu.ac.ir, motaheri@shirazu.ac.ir

Abstract

In the design of recommender systems, it is believed that the set of reviews written by a user can somehow reveal his/her interests, and the content of an item can also be implied from its corresponding reviews. The present study attempts to model both the users and the items via extracting key information from the existing textual reviews. Based on this information, a fuzzy rule-based classifier is designed and tuned, which aims to predict whether a typical user will be interested in a typical item or not. For this purpose, the set of all reviews belonging to a user are mapped to a vector representing the user's interests. Similarly, the set of reviews written by different users over an item are merged and mapped to a vector representing the item. By conjoining these two vectors, a longer vector is obtained which will be used as the input of the classifier. To optimize the classifier, an adaptive approach is suggested and rule-weight learning is carried out, accordingly. The performance of the proposed fuzzy recommender system was evaluated on the Amazon dataset. Experimental results narrate from the promising classification ability of the proposed recommender system compared to state of the art.

Keywords: Recommender systems, review-based, fuzzy classification, rule weighting.

1 Introduction

Recommender systems help users to find the most relevant information that they need by spending less time and less energy. Recommender systems are classified in two main categories: content-based and collaborative filtering [3, 30]. The content-based systems recommend based on item features. In this approach, items similar to the ones in the user's history of selections are recommended to her. On the other hand, Collaborative Filtering (CF) recommender systems work based on the similarity between the users. This similarity is usually measured based on the common interests of the users implied from items in their selection history. Thus, these systems suggest an item to a user, if it has been chosen by a similar one[9, 22, 35, 37].

A weakness in most of CF techniques is modeling the users and items only based on the numeric rating scores provided by the users and regardless of the valuable information existed in the textual reviews. Moreover, in many situations, there is no explicit information such as user profiles and product catalogs from which users' interests and item contents can be directly revealed. One of the approaches employed to address this lack of data is using the information included in the text of reviews. In many commercial systems, besides the numeric rating scores, users can write reviews for the products they have used. In fact, the users explain the reasons and the underlying dimensions behind their ratings in the text reviews. Thus, a review is much more expressive than a single rating score. The reviews contain information that somehow shows the interests of the user and the content of the item. This information is useful to overcome the sparsity problem. Effective making use of the information included in textual reviews may lead to a good solution especially in absence of rating scores or for situations where rating scores are incomplete in most cases.

Corresponding Author: S. M. Fakhrahmad

Received: August 2020; Revised: April 2021; Accepted: June 2021.

There are some statistics showing the importance of textual reviews from the users' viewpoint. More than 32% of consumers rate a product online, over 33% of people write reviews, and about 88% trust the online written reviews [30]. Thus, the reviews which include the preferences and sentiments of the users play an important role in affecting the sales of an item.
Several machine learning as well as natural language processing methods have already been proposed for the task of rating prediction. However, the computational complexity of applying these methods on large data is a challenge. The method proposed in this paper tries to model the users and the items based on their associated set of reviews and using a fuzzy classification approach. It represents the users' preferences as well as the items' content using contextual information given in reviews. The proposed method performs the recommendation task straight forward, i.e., without estimating the rating score. This one phase recommendation is achieved by the classifier developed in this study, which predicts the interest of the user to make a decision for recommending an item (or not), accordingly.
The proposed model works based on the similarity between the users as well as the similarity between items using a fuzzy approach. Considering fuzzy similarity among the set of users and items based on their textual elements, rather than focusing on histories of rating scores, can almost solve the cold-start problem, which is one of the advantages of the proposed model. However, this interesting feature is reachable only if a textual profile exists for a new coming user since he/she has not written any review, yet. Similarly, for a new item, a textual catalog is necessary to be used for modeling the item.
The rest of the paper is organized as follows. Section 2 is devoted to the related work. In Section 3, the details of the proposed recommender system which includes a fuzzy rule-based classifier and a rule-weight learning method are illustrated. Experimental results are given in Section 4. Finally, the paper is concluded in Section 5.

2 Related work

The related work aimed to be addressed in this section, is divided into two main parts. The first part includes the studies which leverages fuzzy approaches and the second part introduces those which use textual reviews in the design of recommender systems.

2.1 Fuzzy recommender systems

In recent decades, several studies have made use of fuzzy logic in design of recommender systems. The main differences of the fuzzy approaches existing in the literature refer to their key features, their evaluation strategies, the datasets they have employed, and the application areas of the developed systems. A taxonomy of the fuzzy tools and approaches which exist in the literature is presented in Figure 1.

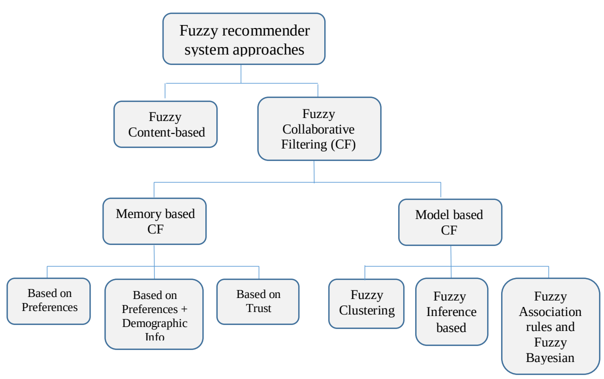


Figure 1: A taxonomy of fuzzy recommender systems already proposed in the literature

Some of the studies that have made use of fuzzy logic in developing recommender systems are content-based systems [2, 5, 7, 29, 41, 50, 62, 68, 76]. Content-based recommender systems generally include two main phases; namely, the profiling (user/item) and the matching process. Fuzzy logic has been employed by different studies, in both of these phases. For example, Zenebe and Norcio [76] proposed a representation method for the items and user feedbacks using fuzzy sets, and developed a content-based recommendation system. They applied fuzzy versions of different similarity metrics, including Jaccard, Cosine, and Correlation similarity. They also used fuzzy aggregation methods for measuring the confidence scores of recommendations. Another use of fuzzy logic in that work is considering a membership degree for each movie in each genre, for a more effective comparison of items.

As one of the shortcomings of the existing fuzzy content-based systems, most of them do not incorporate experimental evaluation, notably limiting the novelty and the scope of the fuzzy logic-supported approaches. Moreover, they are mostly focused on the fuzzy modeling of items' contents, rather than the users' preferences, as one of the major sources of uncertainty in recommender systems.

The second category of the studies includes collaborative filtering approaches. In this category, the similarity function to compare users/items has an important role. For this purpose, most of the fuzzy CF approaches use fuzzy sets to fuzzify users' preferences as the main source of uncertainty. Most of the methods of this category have used the rating scores given by the user and fuzzified them with the defined fuzzy sets such as *Strongly Interested*, *More Interested*, *Interested*, *Less Interested*, and *Not Interested*. The set of fuzzy collaborative filtering works, in the literature, can be divided into two subsets: memory-based and model-based. There are totally three groups of studies in the category of memory-based collaborative filtering: studies that use only the rating values [11, 28, 36, 43, 57, 73], researches using demographic information as well as items' attributes [55, 70, 71, 75], and trust-based methods [8, 40]. Many studies in the category of collaborative filtering assume the availability of additional information beyond the preference values for building fuzzy-supported similarity measures. However, such information is not always available.

In the category of model-based CF, there are plenty of studies proposing unsupervised learning schemes to cluster users or/and items. Many of them use the well-known fuzzy c-means algorithm [19, 34, 64]. There are also studies based on other clustering algorithms such as relational fuzzy subtractive clustering, co-clustering, picture fuzzy clustering, folksonomy-focused intuitionistic fuzzy agglomerative hierarchical clustering, fuzzy geographical clustering, linear fuzzy clustering, and other fuzzy clustering algorithms [17, 23, 26, 32, 56, 60, 72]. Most of them applies a general-purpose fuzzy clustering algorithm to the problem and do not focus on the particularities of the nature of recommender systems data and its possible effect on the development of the clustering approach.

The other category of fuzzy approaches to the recommendation task is fuzzy inference based systems. There are a number of inference-based and rule-based methods already proposed [44, 45, 46, 47, 61]. It should be noted that the success of fuzzy rules and fuzzy inference-based systems in recommender systems is highly dependent on the effective modeling of users' preferences and items' attributes. But a small fraction of the studies belonging to this category, are directly related to the management of preference values. It is one of the facts aimed to be addressed effectively in the present study.

Some other studies also exist in the literature, which have focused on the use of fuzzy association rule mining for developing the recommendation system, for example, the studies performed by Chen and Tai [13], Pinho Lucas et al. [51], Leung et al. [38], and Teng et al. [59]. A collaborative Filtering approach was proposed, by Leung et al. [38], based on fuzzy association rules and multiple-level similarity. For this purpose, they have fuzzified numeric rating scores and map them to three sets, i.e., *Like*, *Neutral* and *Dislike*. They uses some metrics such as fuzzy support and fuzzy confidence to weight the rules and measured how interesting a rule could be. Campos et al. [15] proposed a recommender system that combines probabilistic inference and fuzzy observations. Their proposed system consists of three components. The first component maps the fuzzy rating values (input) to a probabilistic distribution. The second component performs probabilistic reasoning to compute the probability distribution over the expected vote. Finally, the last part calculates the user's vote.

There are also some works proposed by Kant and Bharadwaj [31] and Zhang et al. [77], which integrate fuzzy logic and Bayesian concepts to design model-based recommender systems. There are a few studies based on fuzzy logic that work with reviews. They will be introduced within the next section.

2.2 Review-based recommender systems

This section is devoted to introduction of the studies that employ textual reviews. Only a few of them are based on fuzzy logic concepts. Most of the review-based recommender systems focus on sentiment analysis and opinion mining [1, 12, 18]. Some of them only aim to measure the polarity (negative or positive orientation) of the comments without need to predict the ratings. In other words, these approaches directly recommend (or reject) the items according to the polarity of corresponding provided comments. Some others implement multi-label classification over the existing comments to separately predict the rating of each item for a given user.

Compared to the models only based on the rating scores, text mining techniques can further process the users' feedbacks and may be better choices in developing content-based recommendation systems [1].

In [21], Garcia-Duran et al. have proposed a simple recommendation model using data from Yelp Business Rating Prediction Challenge and Amazon Product Data. In this model, which is based on embedding and opinion mining, the users, items, and reviews are jointly mapped to vectors. The embedding task is performed in a way that a review

embedding vector would be similar to the sum of vectors of corresponding user and item.
In [39], Leung et al. discuss the benefits of using opinion mining techniques to improve collaborative filtering models by two approaches. The first one is to measure the polarity as a feature where user ratings are not present. The second is to incorporate opinion distances among users to modify existing collaborative filtering models. In another study provided by Melville et al., a content-based recommender system was proposed by taking user comments into consideration as a bag-of-words to generate additional features [42].
There are several recommender systems based on Amazon and IMDb datasets, which recommend movies to users by employing a collaborative filtering recommender system [16], and even some recent work on using textual data for rating prediction [48]. The analysis of the style in reviews can be applied to detect which users are more "useful", i.e., with more informative reviews [49]. Other approaches go even further, by generating user profiles based on their social network [6]. The approach by Alves et al. explores the possibilities of the social graph to generate more advanced user profiles.
In [74], the authors developed a topic model-based on *Aspect and Sentiments Unification Model* (ASUM), denoted as RAS. They combined the sentiments in review texts and the rating scores to precisely learn latent factors of users and items.
Another sentiment-based model belongs to Sulthana et al. [53], who developed an Ontology and Context-based Recommendation System (OCBRS) to extract the opinion of the review. Their model includes a Neuro-Fuzzy Classification scheme, in which fuzzy rules are used to represent the context of reviews and determine the sentiments. This approach automatically classifies the reviews under the respective fuzzy rule. The developed Ontology is utilized as a repository of context.
In [58], the authors proposed a new neural method with separate Attention-LSTM blocks: one block for the user, one for the product, and one neighborhood block. A similarity distance is calculated with an NMF method between every user to create a set of 'neighbors' for each user. The reviews from all the neighbors of the user feed the neighborhood block to increase the robustness of the global model. Seo et al. proposed a full convolutional neural network (CNN), denoted as D-Attn, to perform the task of recommendation on Amazon Product and Yelp Business Rating Prediction Challenge 2013 datasets [54]. The designed network consists of two similar blocks for the user and the item, each of which is composed of two blocks having different attention layers. The local attention selects informative keywords from a local window while the global attention model (G- Attn) ignores irrelevant words from long review sequences. For modeling a typical user, the reviews of a user are concatenated and presented in a single embedding matrix. The reviews related to one item are also represented in another similar matrix, at the same time.
The model proposed in [27] is an aspect-based model and consists of three major parts, i.e., identifying references to item aspects in user reviews, classifying the aspect-level sentiment orientation of the opinions in the reviews, and exploiting the extracted aspect opinion information to enhance the recommendations. Wang et al. [65] proposed a user-personalized review rating prediction method that makes use of both ratings and textual reviews. The proposed rating prediction model consists of two main parts; the first part uses textual reviews, while the other is based on the rating values. Afterwards, the results of the two parts are integrated into the final result.
A3NCF is another recommender model proposed by Cheng et al. [14]. In their proposed scheme, a new topic model is developed which aims at the extraction of user preferences as well as item characteristics from review texts. The extracted knowledge is used to capture a user's special attention to each aspect of the target item. Wang et al. focused on the challenging problem of big data in recommendation systems [67]. They developed a movie recommendation framework based on a hybrid recommendation model and sentiment analysis on the Spark platform. In their proposed approach, which is accomplished in two phases, a preliminary recommendation list is first generated. A module of sentiment analysis is then employed to optimize the list.
MTER which stands for Multi-Task Explainable Recommendation is a complex system of recommendation developed by Wang et al. [66]. One of the major parts of the developed system is based on the hypothesis that the appearance of a feature level opinion phrase strongly depends on either the user or the item. Rather than being only a recommender system, this model can also be considered as an opinionated textual explanation system. Their proposed model can predict how a user would appreciate a particular item at the feature level.
PMI-IR (point-wise mutual information and information retrieval) which was proposed by Turing, is based on measuring sentiment orientation (SO) of adjectives that appear in the text. The semantic similarity of two terms is estimated according to SOs and synonym terms will be detected, accordingly. The main output of this work is a synonymy lexicon [63]. Qumsiyeh. And Ng. proposed a recommender system for multimedia items. They extracted a set of features from textual reviews about a specific movie or picture and developed a classifier based on the extracted features [52].
Kim et al. proposed a movie search engine based on both user ratings and reviews. The main goal of this system is query expansion and keyword recommendation while the user is typing. The main problems of this system include the overfitting problem and diversity decreasing in recommender results [33]. Ganu et al. proposed a system for restaurant

recommendation. The main task in their proposed system is the clustering of the users. This clustering is performed based on the reviews written by the users as well as the items they have liked [20]. The model proposed by Molla, performs sentiment analysis over the posts about Samsung products using a different account for Samsung Company [69]. Dmah. H. and Xiao Zheng then followed their idea and deployed sentiment analysis on Twitter posts for trusted friends related to a particular personal account [4].

In [34], a fuzzy C-means approach has been proposed for user-based Collaborative Filtering and its performance against different clustering approaches has been assessed. The MovieLens dataset was used to compare different clustering algorithms. They are evaluated in terms of recommendation accuracy, precision, and recall. The empirical results indicate that a combination of Center of Gravity defuzzified Fuzzy Clustering and Pearson correlation coefficient can yield better recommendation results, compared to other techniques. In [24], Hadad introduced a review-based recommendation approach that obtains contextual information by mining user reviews. The proposed approach relates to features obtained by analyzing textual reviews using methods developed in Natural Language Processing (NLP) and information retrieval discipline to compute a utility function over a given item. Two variations of their recommendation system have been introduced in [24], namely ACM and MCM. An item utility is a measure that shows how much it is preferred according to the user's current context. In this system, the context inference is modeled as a similarity between the user's review history and the item's review history.

In [25], another solution for review-based recommendation was proposed by the authors of the present study. In that work, we had made use of the information included in user reviews as well as the provided rating scores to develop a review-based rating prediction system. The main focus of that study was handling the uncertainty problem of the rating scores, which was accomplished by fuzzifying the rating values. Since the proposed method in that study requires a history of ratings for different users, like many of the existing recommender systems, it suffers from the cold-start problem. To handle this issue, in the proposed study, we try to do the recommendation process for a typical user directly and without any need for prediction of the rating score, before determining the relevance.

3 Proposed method

In this section, the proposed recommender system is illustrated. The major idea behind the developed system is based on the following key assumptions:

- The reviews provided by different users for a specific item can represent the key features of the item.
- The history of reviews written by a user can somehow imply the user's preferences.

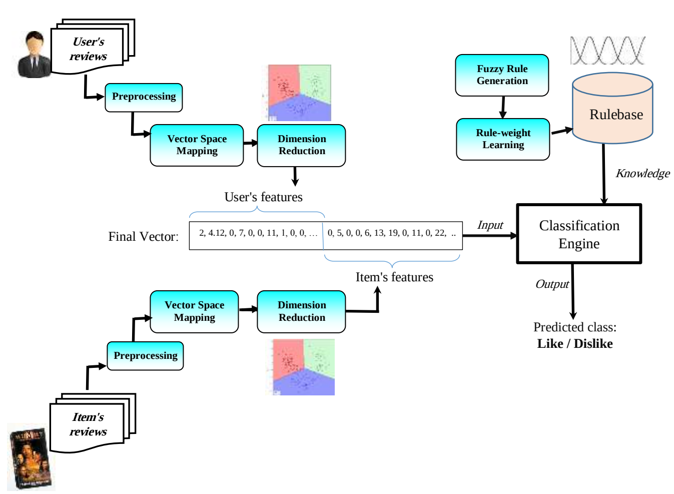


Figure 2: A schematic view of the proposed fuzzy recommender system

The set of textual reviews have been used, in this paper, as a valuable source of knowledge, which includes contextual information as well as content description statements. The focus of the present study is on recommending the movies, and consequently, the proposed scheme is applied for mining reviews of the movies' domain, included in the Amazon dataset [78]. In this dataset, each review, provided by a user, contains an overall rating as well as a textual comment. A schematic view of the proposed scheme is presented in Figure 2.

3.1 Dataset construction

As the first step, 8000 users were selected from the Amazon dataset. The selected users include those who had frequently written reviews for the items. For each user, all belonging reviews were concatenated and stored as the main representor of the user in the dataset. Then, from the set of movies being commented on by these 8000 users, 5000 movies having the highest number of provided reviews were selected. The set of reviews existing for each movie were concatenated and stored in the dataset, similarly.

To increase the reliability of the dataset, which will be used throughout the experiments, we are highly interested in detecting and excluding fake or low-value reviews. For this purpose, the helpfulness ratio aligned with each review, which is the result of other users' votes to the review, were considered. Thus, before concatenating the set of reviews for a user or an item, the reviews that have low helpfulness values, are removed from the dataset. Each concatenated set of reviews is called here, user/item profile.

Finally, a three-column dataset is built showing the existing relationships between the users and the items. Each tuple is associated with a single review-text provided along with a score. The first and second columns store the profiles of corresponding user and item. The third column is a binary feature that is 1 (or positive class) if the user had been interested in the movie, and 0 (or negative class), otherwise. It is revealed by the rating score the user had assigned to the movie. Since in the Amazon dataset, the rating scores range from one to five, the rating scores of four and five have been considered as positive class. For other cases, the class label is set to negative.

3.1.1 Preprocessing

Processing of reviews is considered as the main part of the proposed system. Since the reviews are presented in natural language form, a set of usual preprocessing tasks are needed to be accomplished before starting the major process. The preprocessing done at this stage includes Transforming all characters to lower case, detection and correction of spelling errors, removal of punctuations, stopwords, and numbers, replacing slang terms and, finally, stemming.

Spelling errors and slang terms are usually observed in informal texts, such as reviews. To prepare the text for a more effective mining process, misspelled words and slang terms should be replaced by correct and unambiguous alternatives. In this work, the noisy channel method [10] is used for spelling correction. The Twitter dictionary [79] is also used to detect and replace the slang terms.

After performing the lexical analysis, punctuations, numbers, and stopwords are deleted from the text, and finally, the available words are stemmed.

3.1.2 Mapping to vector space

In this phase, for each user, the set of all belonging reviews (which are merged in the corresponding profile), is mapped to a vector of TF-IDF values. Similarly, a TF-IDF vector is constructed for each item profile. By conjoining these two vectors, a longer vector is obtained, in the proposed model, to represent a specific pair of *user* and *item*. By doing the same task for each pair of user and item, a dataset is constructed.

All dimensions of the conjoined vectors aligned by the binary class label constitute the schema of the dataset. As mentioned before, the class label in each row of the dataset shows whether the corresponding user is interested in the associated item. The final dataset totally includes 7911600 rows, as the result of this phase.

3.1.3 Dimension reduction

The resulted dataset, in the previous phase, is high dimensional since it consists of two sets of features (to represent the user and the item, respectively), and each set is almost as large as the whole vocabulary.

To reduce the dimensions of the dataset to a reasonable number, PCA algorithm has been used. The output of PCA algorithm in this work is a dataset including $100 + 100$ features representing pairs of the user and the item.

3.2 Fuzzy classification

In this part, based on the dataset constructed in the previous steps, an accurate fuzzy classifier is aimed to be developed. The process of fuzzy classifier development includes training and test phases. The training phase includes rule-base construction and rule weighting, as will be discussed in the next subsections.

3.2.1 Rule-base construction

In this phase, a binary classifier is developed for a 2-class problem in its n-dimensional feature space, where n is 200 as described in the previous section. It was also mentioned that 7911600 labeled patterns $X_p = [x_{p,1}, x_{p,2}, \ldots, x_{p,200}]$, $p = 1, 2, \ldots, 7911600$ are included in the provided dataset, each aligned by a binary class value.
A simple approach to generating fuzzy rules is to partition the feature space by specifying k fuzzy sets on the domain interval of each attribute. More than a single k is used here. Examples of this kind of partitioning are shown in Figure 3.

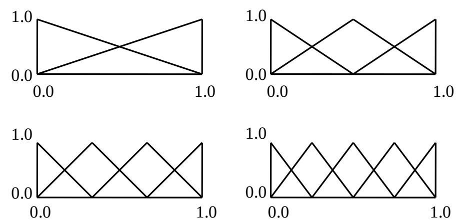


Figure 3: Examples on the partitioning of an input attribute using different numbers of fuzzy sets ($k = 2, 3, 4, 5$)

After determining a partitioning of the feature space, a usual method of rule generation is to consider all possible combinations of attributes to generate the fuzzy rules. The selection of the consequent class for a fuzzy rule can be easily determined based on the confidence of the association rule between the antecedent and each binary value of the consequent. Each fuzzy rule is written in the form of $A_j \Rightarrow$ class C_j, where, A_j is a multi-dimensional fuzzy set representing the antecedent conditions (200 antecedents), and C_j is a binary class label. Confidence (denoted by Con) of this fuzzy association rule is defined in (1).

$$\text{Con}\left(A_j \Rightarrow \text{class } C\right) = \frac{\sum_{X_p \in \text{class } C} \mu_j\left(X_p\right)}{\sum_{p=1}^{m} \mu_j\left(X_p\right)}, \tag{1}$$

where, $\mu_j(X_p)$ is the compatibility degree of the pattern X_p with the antecedent of the rule and m is the number of training data, which is 7911600 in this work. The consequent class C_j of an antecedent combination A_j is determined by choosing the class C with the higher confidence. It is expressed in (2).

$$C_j = \text{argmax}_C \, \text{Con}\left(A_j \Rightarrow \text{class } C\right). \tag{2}$$

There may be some cases where the consequent class of a fuzzy rule cannot be determined since the confidence values obtained for both classes are equal. For such a case, the fuzzy rule is not generated, and so will not be included in the rule-base.
The method used in this work for rule generation is grid partitioning and generating equi-length rules, accordingly. In this method, every rule includes all the features of the dataset on its left-hand side. Figure 4 shows an example of 2-dimensional grid partitioning, with $k = 4$. In this example, a total number of $4^2 (= 16)$ rules may be generated. Each partition in this figure is the decision area of one of these 16 rules.

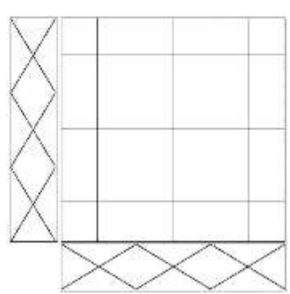


Figure 4: Grid partitioning of a 2-dimensional problem, using 4 fuzzy sets for each dimension

One of the problems in grid partitioning is that for an n-dimensional problem, k^n antecedent combinations should be considered for rule generation. In this problem, no. of dimensions after reduction is 200, which is still a large number. It is impractical to consider k^{200} antecedent combinations.

The idea used to overcome this problem is to generate a fuzzy rule only if it has at least one training pattern in its decision area. Regarding Figure 4, a, pattern is in the decision area of a rule, if all attribute values of the pattern have membership degrees greater than 0.5 in the corresponding antecedent conditions of that rule. Using this simple idea, the maximum number of generated rules will be equal to the number of training patterns (i.e., m), and the time complexity of the rule generation process will be $O(m \cdot n)$, rather than $O(k^n)$.

3.2.2 Rule weighting

In the previous section, a rule-base was constructed using the training dataset. All generated rules were initially considered equal in the term of importance. However, when utilizing the developed classifier, the rules do not have similar effects on the classification ability. Some rules may be valuable since they help the classifier for determining the correct class of the patterns. On the other hand, some rules may mislead the classifier in decision making. Thus, rule-weight learning can be carried out to tune the classifier towards receiving better accuracies. In this section, an adaptive algorithm for rule-weight learning is proposed and employed to assign a weight in the interval $[0, \infty]$ to each rule.

The proposed rule-weight learning method performs a kind of reward and punishment approach in an adaptive manner. Using the leave-one-out validation technique, for each left training pattern x, the weights of two rules are updated, based on the gradient descent method. The most compatible rule with the pattern that has the actual class of x in its consequent part is rewarded by increasing its weight. On the other hand, the most compatible but with different-class rule is punished by decreasing its weight.

To optimize the set of weights assigned to the rules, the following objective function is defined which is aimed to be minimized. The function J represents the misclassification error rate of the classifier, in the leave-one-out validation approach as presented in (3).

$$J = \frac{1}{n} \sum_{x} \text{Step}\left(\frac{w^{\neq} \cdot \mu^{\neq}(x)}{w^{=} \cdot \mu^{=}(x)}\right), \tag{3}$$

where, the step function is defined in (4).

$$\text{Step}(\tau) = \begin{cases} 1 & \text{if } \tau \geq 1 \\ 0 & \text{if } \tau < 1 \end{cases} \tag{4}$$

In (3), the pairs $\left(w^{=}, \mu^{=}(x)\right)$ and $\left(w^{\neq}, \mu^{\neq}(x)\right)$ represent the weights and the compatibility grades associated with the most compatible rules (with x) of the same and different class, respectively.

One of the problems in using the gradient descent method to minimize J is that, the function J is not differentiable and even continuous. To handle this issue, the step function is approximated by Sigmoid function, presented in (5), which is both continuous and differentiable.

$$\Phi(\tau) \;= \frac{1}{(1 + e^{\beta(1-\tau)})}. \tag{5}$$

Replacing the step function in (3) with Sigmoid, the objective function is reformed as (6).

$$J = \frac{1}{n} \sum_{x} \Phi\big(r(x)\big), \tag{6}$$

where, $r(x)$ is defined as shown in (7).

$$r(x) = \frac{w^{\neq} \cdot \mu^{\neq}(x)}{w^{=} \cdot \mu^{=}(x)}, \tag{7}$$

As mentioned above, J is going to be minimized using the gradient descent method. For this purpose, the weights of a pair of rules (the nearest same-class and the nearest different-class rules) should be changed each time. Thus, the derivative of J with respect to $w^{=}$ and $w^{\neq}$ should be computed.

To derive the gradient descent update equations, $\Phi'(\tau)$ is computed as given in (8).

$$\Phi'(\tau) = \frac{\partial \Phi}{\partial \tau} = \frac{(\beta e^{\beta(1-\tau)})}{(1 + e^{\beta(1-\tau)})^2}. \tag{8}$$

$\Phi'(\tau)$ is a function whose maximum value occurs in $\tau = 1$ and vanishes for $|\tau - 1| \gg 0$. This function approaches the Dirac delta function for large values of β. For small values of β, it is approximately constant in a wide range of its domain.

Regarding the mentioned equations, the gradient descent equation in (9) is obtained to be used for updating the rule weight of the nearest same-class rule.

$$\frac{\partial J}{\partial w^{=}} = \frac{\partial J}{\partial \Phi}\frac{\partial \Phi}{\partial r}\frac{\partial r}{\partial w^{=}} = \sum_x \Phi'\left(r\left(x\right)\right) \frac{-\,w^{\neq}\cdot\ \mu^{\neq}(x)}{\left(w^{=}\right)^2\cdot\ \mu^{=}(x)} = -\sum_x \Phi'\left(r\left(x\right)\right)\frac{r(x)}{w^{=}} \tag{9}$$

Similarly, the equation (10) is used for the nearest different-class rule:

$$\frac{\partial J}{\partial w^{\neq}} = \frac{\partial J}{\partial \Phi}\frac{\partial \Phi}{\partial r}\frac{\partial r}{\partial w^{\neq}} = \sum_x \Phi'\left(r\left(x\right)\right) \frac{\mu^{\neq}(x)}{w^{=}\cdot\ \mu^{=}(x)} = \sum_x \Phi'\left(r\left(x\right)\right)\frac{r(x)}{w^{\neq}} \tag{10}$$

Finally, the equations (11) and (12) are used for updating the two mentioned rules, which can be considered as reward and punishment tasks.

$$w^{=}_{\text{new}} = w^{=}_{\text{old}} + \eta \sum_x \Phi'\left(r\left(x\right)\right)\frac{r\left(x\right)}{w^{=}} \tag{11}$$

$$w^{\neq}_{\text{new}} = w^{\neq}_{\text{old}} - \eta \sum_x \Phi'\left(r\left(x\right)\right)\frac{r\left(x\right)}{w^{\neq}} \tag{12}$$

where η is a small positive real number to represent the learning factor of the algorithm.
The pseudo-code of the rule-weight learning algorithm is given in Figure 5.

```
Algorithm GDW(D, R, W, β, η, ε)
{
    // D: training data set; R,W: initial rule-base and weights;
    // β: sigmoid slope; η: learning factor; ε: a small constant number

    λ' = ∞;  λ = J(R,W);
    while (|λ' - λ|> ε)
    {
        λ' = λ
        for each x in D
        {
            rule^= = NearestRule_SameClass(R,W,x)
            rule^≠ = NearestRule_DiffClass(R,W,x)
            i = index(rule^=) and k = index(rule^≠)
            Compute r(x) by (7)
            z1 = Φ'(r(x), β) . r(x)/W[i]
            z2 = Φ'(r(x), β) . r(x)/W[k]
            W[i] = W[i] + η ×  z1
            W[k] = W[k] - η × z2
        }
        λ = J(R,W)
    }
    return W
}
```

Figure 5: The proposed algorithm of rule-weight learning

After running the rule weight learning algorithm, the decision areas of the rules covering many compatible patterns and few incompatible ones grow more significantly. In other words, better rules will gain more power in determining the class labels of the query patterns.
One of the parameters needed by the algorithm is η. It can take just a fixed value or may vary among different rules using some heuristics; for example, it may be proportional to the no. of compatible patterns existing in the decision area of a rule. Moreover, for smoother (but slower) convergence, it was decreased along the successive iterations of the algorithm. A generally suitable value of the learning factor for any data cannot be found since it depends on the nature and characteristics of the training data.

3.3 Justification of the proposed approach

The present study has focused on proposing a light, descriptive and editable recommender system. Each one of these properties has been discussed in following. The proposed system is based on extracting proper fuzzy rules as a light processing task in comparison with heavy learning systems such as deep neural networks. The deep networks not only have time consuming learning phases, but also require a huge amount of memory space to store the weights of the layers such that loading most of them is not practical on normal PCs. In addition, learning these models is not stable and the objective function has many local optimum solutions such that regenerating the results is usually hard and highly dependable to the learning parameters, cross-validation division, batch sizes and optimization methods. This is why; the light recommender systems are still attractive.

Even some shallow models such as Support Vector Machines (SVM) are not comparable with the proposed model from the computational point of view. The large datasets are not usually linearly separable and SVM or other linear models cannot achieve a good performance except of using non-linear kernels e.g., RBF. The kernel-based SVM is optimized in the dual model with a kernel matrix of size $m \times m$, whereas m is the number of training instances. In addition, computational complexity of solving the model is quadratically related to m. This is why; the models such as SVM or any kernel-based model cannot practically used for large dataset models regarding both memory and time complexity. The rule-based model including fuzzy systems are, in comparison with the above models, light and executable. Moreover, SVM is highly sensitive to outliers, noisy support vectors and the value of regularization parameter. Fuzzy systems, due to the uncertainty property of the rules, achieve more stability in such situations.

It should be notified that; the history of rule-based systems is lied on being descriptive. Although, in the proposed model, the original features are not preserved, the fuzzy rules, associativity of the antecedents and features, importance of the fuzzy sets and their frequency in the rules are clear. Hence, any analysis of the model (e.g., sensitive analysis or extracting adversarial examples) is easy. In addition, the experts can insert to, remove or edit any rule of the model to adapt the pretrained system for example to cover cold start cases in recommender systems.

Another key feature of the proposed method, which should be noted, is that it can overcome the cold-start problem, which is a common issue in many recommender systems. As illustrated in the previous sections, the proposed method does not work based on rating scores and it extracts and uses some implicit features from the existing textual data. Since in the training phase, the extracted features of users and items are fuzzified and represented by fuzzy rules, the new user does not need the rating history to be available. Instead, if textual information about the user (i.e., user's profile) is available, it can be used as his/her initial data to construct the representative vector. This vector will then be compared with the fuzzy rules included in the rule-base and the fuzzy reasoning process and the class assignment task will be accomplished. However, since the user's profile may include limited or incomplete information, the representative vector of the user shall be updated over time, as he/she rates the items and writes reviews for them. A similar scenario can be followed for a new item to become resistant to the cold-start problem, just if a describing catalog is available for the item.

4 Experimental results

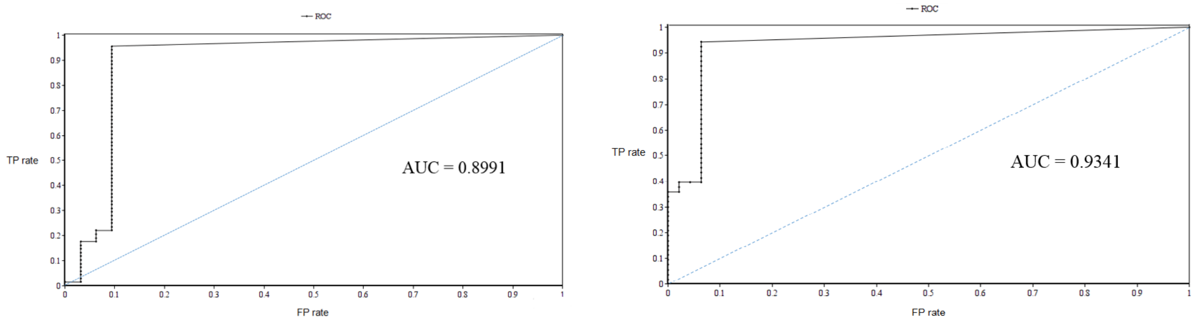
To evaluate the proposed recommender system, the dataset that was illustrated in Section 3.1, has been used. As mentioned before, the dataset includes 7911600 records and 200 attributes, all extracted from textual elements inside the reviews. The first 100 attributes in a row describe a user, and the rest describe an item. It was also mentioned that rating scores 4 and 5 were replaced by the class label $1('+')$, while the label of $0('-')$ was assigned to the others. For a test user, the final recommendation list offered by the system consists of the set of movies with positive assignments.

To evaluate the proposed classifier, we used the ten-fold cross validation method (in 10 rounds, to reduce the variability). In each round of the experiment, after shuffling the data set, it was divided into 10 partitions, each time one of the partitions was left for validation, and training of the classifier was accomplished by the remaining nine partitions. The mean and SD[1] values of the obtained results in different runs were finally computed, which will be shown, subsequently.

We performed ROC analysis to observe the diagnostic ability of the developed classifier. To evaluate the effect of the proposed rule weighting method, ROC analysis was carried out two times, once before rule weighting and once after that. AUC values were measured in each case.

Figure 6 shows the ROC curves and AUC values of the proposed recommender system, before and after applying the rule-weighting mechanism. In these ROC curves, the true positive (TP) rate of recommendations against the false positive (FP) rate of recommendations at various threshold settings has been plotted. TP is the number of positive assignments that were correct (i.e., true recommendations), and FP is the number of positive assignments that were actually negative in the dataset (i.e., false recommendations).

[1] Standard Deviation



(a) Before rule-weighting (b) After rule-weighting

Figure 6: Roc curves and AUC values for the proposed recommender system (classifier), before and after rule-weighting

It can be revealed from Figure 6 that the proposed weighting scheme has a positive effect on the classification ability. The AUC value was improved from 0.89 to 0.93 after the rule-weight learning was carried out. Comparison of these AUC values with other supervised methods will be done and reported in the subsequent set of experiments.

Table 1 is devoted to the comparison of the proposed recommender system with other fuzzy model-based collaborative filtering schemes, in terms of Accuracy, Precision, Recall, F1 score, and AUC. The reported results include the mean values aggregated from different runs of each algorithm. The proposed scheme has also been compared with similar recommender systems, most of which work on textual reviews. The results obtained from multiple runs of different algorithms are given in Table 2, in terms of the standard descriptive statistical calculation, i.e., mean ±SD.

Table 1: Comparison of the proposed recommender system with other fuzzy model-based CF systems

	Algorithm	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**
Fuzzy Clustering methods	Koohi et al. [34]	58.3	62.2	53.3	57.4	0.6101
	He et al. [26]	57.6	63.1	51.8	56.89	0.5917
	Katarya et al. [32]	62.8	63.5	61	62.22	0.6364
Fuzzy Inference methods	Nguyen et al. [44]	72.3	69.9	74.4	72.07	0.792
	Nilashi et al. [47]	75.7	73.5	**75.9**	74.68	0.8331
Fuzzy ARs/ Fuzzy Bayesian	Lucas et al. [51]	64	65.2	69.6	67.32	0.7011
	Teng et al. [59]	71.8	70.4	73.3	71.82	0.7826
	Zhang et al. [77]	64.4	62.6	65.2	63.87	0.6213
	Proposed model (- rule weighting)	84	83.2318	72.08	77.26	0.8991
	Proposed model (+ rule weighting)	**87.33**	**88**	75.1315	**81.0581**	**0.9341**

Other evaluation metrics used throughout the experiments include Accuracy, Precision, Recall, and F1 Score. Accuracy is the proportion of true label assignments ($'+'$ or $'-'$) over the total number of assignments, while Precision is the fraction of actually positive (favorite) items among the recommendation list. Recall is the fraction of favorite items that have been recommended over the total number of favorite items. The F1 score is the harmonic mean of precision and recall taking both metrics into account.

The results given in Table 1 and Table 2 show that the proposed recommender system performs better than its counterparts, which were assessed as well in this experiment. The effect of the adaptive rule weighting method on the success of the proposed classifier is also clear and considerable. It can be seen in Table 2 that before applying rule weight learning, the proposed method is not the best in terms of precision, recall, and F1 score. However, after running

Table 2: Comparison of the proposed recommender system with other review-based systems (mean ±SD)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**
KNN80 (based on reviews' *polarity*)	63.66 ± 0.22	66.25 ± 0.26	39.25 ± 0.33	49.30 ± 0.08	0.52 ± 0.0072
UPRRP	77.33 ± 0.38	79.24 ± 0.94	64.61 ± 0.42	71.18 ± 0.89	0.72 ± 0.013
A3NCF	79.33 ± 0.43	84.76 ± 0.57	65.92 ± 0.4	74.16 ± 0.92	0.82 ± 0.0102
ACM (TF-IDF)	69.00 ± 0.35	78.37 ± 0.72	42.96 ± 0.32	55.50 ± 0.49	0.53 ± 0.0044
MCM (TF-IDF)	71.00 ± 0.77	76.66 ± 0.76	51.11 ± 0.82	61.33 ± 0.23	0.62 ± 0.016
Slope one	83.00 ± 0.72	87.82 ± 0.07	74.81 ± 0.08	80.8 ± 0.07	0.90 ± 0.007
Bipolar Slope one	75.33 ± 0.68	86.74 ± 0.38	53.33 ± 0.98	66.05 ± 1.05	0.61 ± 0.0052
Biased Matrix Factorization	76.66 ± 1.24	78.26 ± 0.67	66.66 ± 0.76	72.00 ± 0.86	0.76 ± 0.0126
Factor Wise Matrix Factorization	75.00 ± 0.49	77.27 ± 0.19	62.96 ± 0.6	69.38 ± 0.94	0.65 ± 0.0055
Hernndez-Rubio method	78.33 ± 0.68	80.95 ± 1.25	65.38 ± 0.93	72.34 ± 0.37	0.74 ± 0.0077
RAS	81.33 ± 0.74	87.00 ± 0.14	66.92 ± 0.43	75.65 ± 1.08	0.84 ± 0.0057
OCBRS	81.00 ± 1.58	84.61 ± 0.73	67.69 ± 0.72	75.21 ± 0.72	0.85 ± 0.0072
User Item Baseline	73.66 ± 0.92	79.78 ± 0.6	55.55 ± 1.02	65.50 ± 0.63	0.62 ± 0.0075
Proposed model (- rule weighting)	84.00 ± 0.45	83.23 ± 1.59	72.08 ± 0.35	77.26 ± 0.73	0.89 ± 0.0038
Proposed model (+ rule weighting)	$\mathbf{87.33} \pm 0.7$	$\mathbf{88.00} \pm 0.82$	$\mathbf{75.13} \pm 0.09$	$\mathbf{81.05} \pm 0.04$	$\mathbf{0.93} \pm 0.0079$

the rule weighting method, the proposed method obtains the best results in terms of all evaluation metrics. To evaluate the significance of the observed improvements, a further statistical analysis will be presented in the following section.

4.1 Statistical analysis

To determine the significance of the observed improvements, the results obtained from multiple runs of each algorithm were statistically analyzed using IBM SPSS V.22 (SPSS Inc.). In addition to the standard descriptive statistical calculation (mean ±SD), which was presented above, for each of the evaluation metrics (i.e., Accuracy, Precision, Recall, F1 and AUC), One-way ANOVA and Tukey's HSD tests were performed to determine significant differences in given comparisons. The significance level was considered 0.05. The one-way ANOVA test is aimed to determine whether there is an overall difference among the results, while Tukey's HSD test performs a pairwise comparison, and allows us to determine between which of the various pairs of means there is a significant difference.
The results obtained from One-way ANOVA test are given in Table 3.

Table 3: The results of One-way ANOVA test on obtained values of different evaluation metrics

Investigated Measure	***f*-ratio value**	***p*-value**
Accuracy	37.95272	$< .00001$
Precision	29.71555	$< .00001$
Recall	442.2285	$< .00001$
F1	86.77443	$< .00001$
AUC	161.43198	$< .00001$

The results of One-way ANOVA test, as shown in Table 3, confirm that there are statistically significant differences among the means of compared algorithms, for all measures; Accuracy, Precision, Recall, F1 and AUC. However, for further investigation, the Tukey's HSD test also seems necessary, to perform a pairwise analysis.
Using the Tukey's HSD test, the difference significance of every pair of algorithms given in Table 2 were analyzed. However, in this part of the statistical analysis, we are only interested in analyzing the difference between the method having the best achieved results and its counterparts. Thus, in Table 4, the results of the Tukey's HSD test over only four pairs including the proposed method with rule weighting and its closest competitors are presented. For ease of presentation, in Table 4, the proposed method with and without rule weighting are denoted as $\mathrm{PM}^{+}$and PM^{-}, respectively. In this table, Q is the major measure the Tukey test revolves around, which is referred to as the Studentized range statistic, and is calculated based on the pair of mean values under investigation.
The results of the Tukey's HSD test revealed that there is a significant difference between the best results which were achieved by the proposed method with rule weighting, and the results of its counterparts, in most cases. Although, there were a few cases that the difference was not determined to be significant. These exceptions occurred for RAS and Slopeone algorithms. The Tukey's HSD test, showed that the improvement in terms of Precision achieved by

Table 4: The results of Tukey's HSD test analyzing the significance of differences observed between different measures of algorithm pairs

Pairwise Comparisons (Method1 : Method2)	$Q_{.05} = 4.2319$ $Q_{.01} = 5.2933$ Accuracy Difference	Precision Difference	Recall Difference	F1 Difference	AUC Difference
PM$^+$: PM$^-$	$Q = 8.05$ ($p = .00013$)	$Q = 12.32$ ($p = .00000$)	$Q = 16.59$ ($p = .00000$)	$Q = 12.68$ ($p = .00000$)	$Q = 13.82$ ($p = .00000$)
PM$^+$: OCBRS	$Q = 15.30$ ($p = .00000$)	$Q = 8.75$ ($p = .00004$)	$Q = 40.46$ ($p = .00000$)	$Q = 19.54$ ($p = .00000$)	$Q = 27.64$ ($p = .00000$)
PM$^+$: RAS	$Q = 14.50$ ($p = .00000$)	$Q = 2.58$ ($p = .38717$)$*$	$Q = 44.65$ ($p = .00000$)	$Q = 18.07$ ($p = .00000$)	$Q = 31.09$ ($p = .00000$)
PM$^+$: SlopeOne	$Q = 10.47$ ($p = .00000$)	$Q = 0.46$ ($p = .99726$)$*$	$Q = 1.74$ ($p = .73425$)$*$	$Q = 0.84$ ($p = .97479$)$*$	$Q = 11.05$ ($p = .00000$)

* Non-significant difference

the proposed method compared to the RAS method is not significant. A similar situation is also observed for the differences between the results of the proposed method and the SlopeOne algorithm, in terms of Precision, recall and F1, which were non-significant. Except of these few cases, all other improvements that were reported in Tables 1 and 2 are significant. The differences between the results of the proposed method before and after applying the rule weight learning scheme, were also significant, for all evaluation metrics.

5 Conclusion

In this paper, a review-based recommender system was proposed, which models the users and the items using their corresponding textual reviews. The proposed scheme does not need the history of numeric rating values for the users which are not usually precise and reliable. It is also not dependent on user profiles and item catalogs and is suitable for environments where these two important sources of information are not present. However, if these two sources of information are available, they can be used for any pair of user and item instead of their belonging textual reviews. In such cases, the system will be robust against the cold-start problem. The major part of the proposed system is a fuzzy binary classifier which receives a pair of user and item and predicts whether the user will be interested in the item or not. It was shown through the experiments against the Movies section of the AMAZON dataset that the proposed rule weight learning method improves the classification ability of the system, significantly.
The proposed system achieved the AUC value of 93.41%, which is promising when comparing to its counterparts. Comparing various supervised methods, the best results in terms of accuracy, precision, recall, F1 score and AUC were received by the proposed method after applying the rule weighting method. It implies the key role of the adaptive weighting method. That is before tuning the rule weights, the results of the proposed scheme were not the best among all counterparts. Statistical analysis using One-way ANOVA and Tukey's HSD tests over the results obtained from multiple runs of different algorithms showed that for all evaluation metrics applied, the improvements achieved after applying the rule weighting scheme are significant. Moreover, it was also proved that except of two cases, the differences between the proposed method (after rule weighting) and other existing methods are significant. Future work may include focusing on efficiency, fuzzy set tuning, imbalanced problem and noisy patterns. In addition, the fuzzy model can be integrated with neural networks to use their effectiveness but in a light structure and a white box.

References

[1] S. Aciar, D. Zhang, S. Simoff, J. Debenham, *Informed recommender: Basing recommendations on consumer product reviews*, Intelligent Systems, IEEE, **22** (2007), 39-47.

[2] M. N. M. Adnan, M. R. Chowdury, I. Taz, T. Ahmed, R. M. Rahman. *Content-based news recommendation system based on fuzzy logic*, International Conference on Informatics, Electronics and Vision (ICIEV), IEEE, (2014), 1-6, Doi: 10.1109/ICIEV.2014.6850800.

[3] P. Aggarwal, V. Tomar, A. Kathuria, *Comparing content-based and collaborative filtering in recommender systems*, International Journal of New Technology and Research (IJNTR), **3**(4) (2017), 65-67.

[4] D. Alahmadi, X. Zeng, *ISITS: Implicit social trust and sentiment based approach to recommender systems*, Expert Systems with Applications Journal, **42** (2015), 8840-8849.

[5] H. Al-Qaheri, S. Banerjee, *Design and implementation of a policy recommender system towards social innovation: An experience with hybrid machine learning*, In Intelligent Data Analysis and Applications, Springer, (2015), 237-250.

[6] D. Alves, M. Freitas, T. Moura, D. Souza, *Using social network information to identify user contexts for query personalization*, In DBKDA 2013, The Fifth International Conference on Advances in Databases, Knowledge, and Data Applications, (2013), 45-51.

[7] D. Anand, B. S. Mampilli, *Folksonomy-based fuzzy user profiling for improved recommendations*, Expert Systems with Applications, **41**(5) (2014), 2424-2436.

[8] P. Bedi, P. Vashisth, *Empowering recommender systems using trust and argumentation*, Information Sciences, **279** (2014), 569-586.

[9] R. Bell, Y. Koren, C. Volinsky, *Modeling relationships at multiple scales to improve accuracy of large recommender systems*, In Proceedings of the 13th ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining KDD, New York, NY, USA: ACM, (2007), 95-104.

[10] E. Brill, R. C. Moore, *An improved error model for noisy channel spelling correction*, In ACL 2000, Hong-Kong, (2000), 286-293.

[11] J. Castro, F. J. Quesada, I. Palomares, L. Martínez, *A consensus-driven group recommender system*, International Journal of Intelligent Systems, **30**(8) (2015), 887-906.

[12] M. Chelliah, S. Sarkar, *Product recommendations enhanced with reviews*, Proceedings of the Eleventh ACM Conference on Recommender Systems-RecSys '17, (2017), 398-399.

[13] C. Chen, W. Tai, *A user preference classification method in information recommendation system*, The Fourth International Conference on Electronic Business / Beijing (ICEB), (2004), 1091-1096.

[14] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, M. Ankanhalli, *A3NCF: An adaptive aspect attention model for rating prediction*, Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), (2018), 3748-3754.

[15] L. M. De Campos, J. M. Fernandez-Luna, J. F. Huete, *A collaborative recommender system based on probabilistic inference from fuzzy observations*, Fuzzy Sets and Systems, **159**(12) (2008), 1554-1576.

[16] S. Debnath, N. Ganguly, P. Mitra, *Feature weighting in content-based recommendation system using social network analysis*, In Proceedings of the 17th International Conference on World Wide Web WWW 2008, New York, NY, USA: ACM, (2008), 1041-1042.

[17] M. K. K. Devi, P. Venkatesh, *Smoothing approach to alleviate the meager rating problem in collaborative recommender systems*, Future Generation Computer Systems, **29**(1) (2013), 262-270.

[18] Q. Diao, M. Qiu, C. Wu, A. J. Smola, J. Jiang, C. Wang, *Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)*, In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August (2014), 193-202.

[19] M. H. Esfahani, F. K. Alhan, *New hybrid recommendation system based on c-means clustering method*, The 5th Conference on Information and Knowledge Technology (IKT), IEEE, (2013), 145-149.

[20] G. Ganu, Y. Kakadkar, A. Marian, *Improving the quality of prediction using textual information in online user reviews*, Information System Journal, **38** (2013), 1-15.

[21] A. Garcia-Durán, R. González, D. Onoro-Rubio, M. Niepert, H. Li, *TransRev: Modeling reviews as translations from users to items*, European Conference on Information Retrieval, (2018), 234-248.

[22] R. Gemulla, E. Nijkamp, P. J. Haas, Y. Sismanis, *Large-scale matrix factorization with distributed stochastic gradient descent*, In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD, New York, NY, USA: ACM, (2011), 69-77.

[23] C. Guan, K. K. F. Yuen, F. Coenen, *Towards an intuitionistic fuzzy agglomerative hierarchical clustering algorithm for music recommendation in folksonomy*, IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, (2015), 2039-2042.

[24] T. Hadad, *Review-based rating prediction*, arXiv preprint arXiv: 1607.00024v4 [cs.IR], (2016).

[25] S. Hasanzadeh, S. M. Fakhrahmad, M. Taheri, *Review-based recommender systems: A proposed rating prediction scheme using word embedding representation of reviews*, The Computer Journal, (2020), 63, Doi: 10.1093/comjnl/bxaa044.

[26] H. Q. He, Z. L. Fan, *An improved collaborative filtering recommendation algorithm based on coclustering*, In AETIE, (2015), 508-515.

[27] M. Hernndez-Rubio, I. Cantador, A. Bellogín, *A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews*, User Modeling and User-Adapted Interaction, 1-61.

[28] Y. C. Hu, Y. J. Chiu, Y. L. Liao, Q. Li, *A fuzzy similarity measure for collaborative filtering using nonadditive grey relational analysis*, Journal of Grey System, **27**(2) (2015), 93-103.

[29] H. H. Huang, H. C. Yang, E. H. C. Lu, *A fuzzy rough set based ontology for hybrid recommendation*, IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), (2015), 358-359.

[30] N. Kadam, S. Kumar, *A review of content and collaborative filtering approaches on movielens data*, International Research Journal of Engineering and Technology, **3**(3) (2016), 273-278.

[31] V. Kant, K. K. Bharadwaj, *Integrating collaborative and reclusive methods for effective recommendations: A fuzzy bayesian approach*, International Journal of Intelligent Systems, **28**(11) (2013), 1099-1123.

[32] R. Katarya, O. P. Verma, *A collaborative recommender system enhanced with particle swarm optimization technique*, Multimedia Tools and Applications, (2016), 1-15.

[33] H. Kim, K. Han, J. Cho, J. Hong, *Movie mine: Personalized movie content search by utilizing user comments*, IEEE Transactions and Consumer Electronics, **58** (2012), 1416-1426.

[34] H. Koohi, K. Kiani, *User based collaborative filtering using fuzzy c-means*, Measurement, **91** (2016), 134-139.

[35] Y. Koren, *Factor in the neighbors: Scalable and accurate collaborative filtering*, ACM Transactions on Knowledge Discovery from Data, **4**(1) (2010), 1-24.

[36] P. Ladyzynski, P. Grzegorzewski, *Vague preferences in recommender systems*, Expert Systems with Applications, **42**(24) (2015), 9402-9411.

[37] D. Lemire, A. Maclachlan, *Slope one predictors for online rating-based collaborative filtering*, In Proceedings of SIAM Data Mining (SDM'05), 2005.

[38] C. W. K. Leung, S. C. F. Chan, F. L. Chung, *A collaborative filtering framework based on fuzzy association rules and multiple-level similarity*, Knowledge and Information Systems, **10**(3) (2006), 357-381.

[39] C. W. K. Leung, S. C. F. Chan, F. Chung, *Integrating collaborative filtering and sentiment analysis: A rating inference approach*, In Proceedings of the ECAI 2006 Workshop on Recommender Systems, (2006), 62-66.

[40] S. Linda, K. K. Bharadwaj, *A fuzzy trust enhanced collaborative filtering for effective context-aware recommender systems*, In Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems, Springer, **2** (2016), 227-237.

[41] M. Mao, J. Lu, G. Zhang, J. Zhang, *A fuzzy content matching-based e-commerce recommendation approach*, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), (2015), Doi: 10.1109/FUZZ-IEEE.2015.7338036.

[42] P. Melville, R. J. Mooney, R. Nagarajan, *Content-boosted collaborative filtering for improved recommendations*, In Proceedings of the National Conference on Artificial Intelligence. Menlo Park, CA; Cambridge, MA; London: AAAI Press, MIT Press., (1999), 187-192.

[43] M. B. Menhaj, S. Jamalzehi, *Scalable user similarity estimation based on fuzzy proximity for enhancing accuracy of collaborative filtering recommendation*, 4th International Conference on Control, Instrumentation, and Automation (ICCIA), IEEE, (2016), 220-225.

[44] D. A. Nguyen, T. H. Duong, *Video recommendation using neuro-fuzzy on social tv environment*, In Advanced Computational Methods for Knowledge Engineering, Springer, (2015), 291-298.

[45] M. Nilashi, O. Bin Ibrahim, N. Ithnin, *Hybrid recommendation approaches for multi-criteria collaborative filtering*, Expert Systems with Applications, **41**(8) (2014), 3879-3900.

[46] M. Nilashi, O. Bin Ibrahim, N. Ithnin, *Multicriteria collaborative filtering with high accuracy using higher order singular value decomposition and neuro-fuzzy system*, Knowledge-Based Systems, **60** (2014), 82-101.

[47] M. Nilashi, O. Bin Ibrahim, N. Ithnin, N. H. Sarmin, *A multi-criteria collaborative filtering recommender system for the tourism domain using expectation maximization (em) and pca-anfis*, Electronic Commerce Research and Applications, **14**(6) (2015), 542-562.

[48] A. Oghina, M. Breuss, M. Tsagkias, M. Rijke, *Predicting imdb movie ratings using social media*, Advances in Information Retrieval 2012, Springer Berlin Heidelberg, **7224** of Lecture Notes in Computer Science, (2012), 503-507.

[49] J. Otterbacher, *Gender, writing and ranking in review forums: A case study of the IMDB*, Knowledge and Information Systems, **1** (2012), 1-20.

[50] I. Pardines, V. López, A. Sanmartín, M. O. de Toledo, C. Fernández, *Collaborative recommendation system for environmental activities management mobile application*, Practical Applications of Intelligent Systems (ISKE), Springer, (2014), 327-335.

[51] J. Pinho Lucas, A. Laurent, M. N. Moreno, M. Teisseire, *A fuzzy associative classification approach for recommender systems*, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, **20**(4) (2012), 579-617.

[52] R. Qumsiyeh, Y. Ng, *Predicting the rating of multimedia items for making personalized recommendations*, In the Proceeding of the 35th International ACU SIGIR Conference on Research and Development Information Retrieval, (2012), 475-484.

[53] A. Razia, *Sulthana, subburaj ramasamy, ontology and context based recommendation system using neuro-fuzzy classification*, Computers and Electrical Engineering, **74** (2019), 498-510.

[54] S. Seo, J. Huang, H. Yang, Y. Liu, *Interpretable convolutional neural networks with dual local and global attention for review rating prediction*, In Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys), 2017.

[55] L. H. Son, *HU-FCF: A hybrid user-based fuzzy collaborative filtering method in recommender systems*, Expert Systems with Applications, **41**(15) (2014), 6861-6870.

[56] L. H. Son, N. T. H. Minh, K. M. Cuong, N. V. Canh, *An application of fuzzy geographically clustering for solving the cold-start problem in recommender systems*, International Conference on Soft Computing and Pattern Recognition (SoCPaR), IEEE, (2013), 44-49.

[57] L. H. Son, N. T. Thong, *Intuitionistic fuzzy recommender systems: An effective tool for medical diagnosis*, Knowledge-Based Systems, **74** (2015), 133-150.

[58] Y. Tay, L. A. Tuan, S. C. Hui, *Multi-pointer co-attention networks for recommendation*, arXiv:1801.09251v2 [cs.CL] 21 Jun (2018).

[59] Y. Teng, L. Zhang, Y. Tian, X. Li, *A novel fahp based book recommendation method by fusing apriori rule mining*, 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), IEEE, (2015), 237-243.

[60] N. T. Thong, L. H. Son, *HIFCF: An effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis*, Expert Systems with Applications, **42**(7) (2015), 3682-3701.

[61] S. Tiwari, S. Kaushik, *Crowdsourcing based fuzzy information enrichment of tourist spot recommender systems*, International Conference on Computational Science and Its Applications (ICCSA), (2015), 559-574.

[62] C. H. Tsai, *A fuzzy-based personalized recommender system for local businesses*, Proceedings of the 27th ACM Conference on Hypertext and Social Media (ACM), (2016), 297-302.

[63] P. Turneg, *Thumbs up or thumbs down? Sentiment orientation applied to unsupervised classification of reviews*, In the Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics, (2002), 417-424.

[64] B. Veloso, B. Malheiro, J. C. Burguillo, *A multi–agent brokerage platform for media content recommendation*, International Journal of Applied Mathematics and Computer Science, **25**(3) (2015), 513-527.

[65] B. Wang, B. Chen, L. Ma, G. Zhou, *User-personalized review rating prediction method based on review text content and user-item rating matrix*, Information, **10**(1) (2019), 1-15.

[66] N. Wang, H. Wang, Y. Jia, Y. Yin, *Explainable recommendation via multi-task learning in opinionated text data*, (2018), Doi: 10.1145/3209978.3210010.

[67] Y. Wang, M. Wang, W. Xu, *A sentiment-enhanced hybrid recommender system for movie recommendation: A big data analytics framework*, Wireless Communications and Mobile Computing, **2018**, (2018), 9 pages.

[68] M. Wasid, V. Kant, *A particle swarm approach to collaborative filtering based recommender systems through fuzzy features*, Procedia Computer Science, **54** (2015), 440-448.

[69] C. Weng, S. Chi-Fai, F. Chung, *Integrating collaborative filtering and sentiment analysis: A rating inference approach*, In The Proceeding of ECAI Workshop on Recommender Systems, (2006), 62-66.

[70] D. Wu, J. Lu, G. Zhang, *A fuzzy tree matching-based personalized E-learning recommender system*, IEEE Transactions on Fuzzy Systems, **23**(6) (2015), 2412-2426.

[71] D. Wu, G. Zhang, J. Lu, *A fuzzy preference treebased recommender system for personalized business-to-business e-services*, IEEE Transactions on Fuzzy Systems, **23**(1) (2015), 29-43.

[72] S. Xu, J. Watada, *A method for hybrid personalized recommender based on clustering of fuzzy user profiles*, IEEE International Conference on Fuzzy Systems (FUZZ), (2014), 2171-2177.

[73] R. Yera Toledo, J. Castro, L. Martinez, *A fuzzy model for managing natural noise in recommender systems*, Applied Soft Computing, **40** (2016), 187-198.

[74] D. Yu, Y. Mu, Y. Jin, *Rating prediction using review texts with underlying sentiments*, Information Processing Letters, **117** (2017), 10-18.

[75] L. Zadeh, *The concept of a linguistic variable and its applications to approximatereasoning, Part I.*, Information Sciencies, **8** (1975), 199-249.

[76] A. Zenebe, A. F. Norcio, *Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems*, Fuzzy Sets and Systems, **160**(1) (2009), 76-94.

[77] S. Zhang, C. Xi, Y. Wang, W. Zhang, Y. Chen, *A new method for e-government procurement using collaborative filtering and bayesian approach*, The Scientific World Journal, (2013), Doi:10.1155/2013/129123.

[78] Amazon (2016). Amazon.com: Movies and TV. https://www.amazon.com/movies-tv-dvd-bluray/b/ref=sd_allcat_mov?ie=UTF8&node=2625373011 [Online; accessed June 2016].

[79] Twitter Dictionary (2016). Twitter Dictionary: A Guide to Understanding Twitter Lingo. http://www.webopedia.com/quick_ref/Twitter_Dictionary_Guide.asp [Online; accessed June 2016].

Iranian Journal of Fuzzy Systems
Volume 18, Number 6 (2021), pp. 207
University of Sistan and Baluchestan

A fuzzy approach to review-based recommendation: Design and optimization of a fuzzy classification scheme based on implicit features of textual reviews

S. Hasanzadeh, S. M. Fakhrahmad and M. Taheri

یک رویکرد فازی در توصیه مبتنی بر نظرات متنی: طراحی و بهینه‌سازی یک کلاسه‌بند فازی براساس ویژگی‌های ضمنی نظرات نوشتاری

چکیده. در طراحی سیستم‌های توصیه‌گر، مجموعه نظرات متنی نوشته شده توسط یک کاربر می‌تواند تا حدی علایق او را آشکار سازد، و محتوای یک قلم داده نیز ممکن است از نظرات متنی مرتبط با آن استنباط گردد. مطالعه حاضر سعی دارد از طریق استخراج اطلاعات کلیدی از مجموعه نظرات متنی موجود، به مدل‌سازی کاربرها و اقلام بپردازد. بر اساس این اطلاعات، یک کلاسه‌بند فازی مبتنی بر قوانین طراحی و بهینه خواهد شد؛ با این هدف که بتواند علاقمندی یک کاربر خاص به یک قلم داده نمونه را بیش‌بینی کند. بدین منظور، مجموعه تمام نظرات متنی که به یک کاربر تعلق دارد به یک بردار که نشان دهنده علایق آن کاربر است، نگاشت می‌شود. بطور مشابه، مجموعه نظرات متنی نوشته شده توسط کاربران متفاوت در مورد یک قلم داده نیز ادغام شده و به یک بردار که نمایش‌دهنده آن قلم داده است، نگاشت می‌شود. با وصل کردن این دو بردار، برداری بزرگتر حاصل می‌شود که به عنوان داده ورودی به کلاسه‌بند استفاده خواهد شد. برای بهینه کردن کلاسه‌بند یک رویکرد تطبیقی بیشنهاد می‌گردد و براساس آن، یادگیری وزن قوانین انجام می‌شود. کارایی سیستم توصیه‌گر فازی پیشنهادی با استفاده از مجموعه داده آمازون ارزیابی شده است. نتایج آزمایشات انجام شده نشان‌دهنده قابلیت کلاسه‌بندی امیدوارکننده سیستم توصیه‌گر پیشنهادی در مقایسه با آخرین نمونه‌های مشابه می‌باشد.