Applied Soft Computing 40 (2016) 187–198

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A fuzzy model for managing natural noise in recommender systems

Raciel Yera [a], Jorge Castro [b], Luis Martínez [c,*]

[a] University of Ciego de Ávila, Ciego de Ávila, Cuba
[b] Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
[c] Computer Science Department, University of Jaén, Jaén, Spain

ARTICLE INFO

Article history:
Received 9 April 2015
Received in revised form
14 September 2015
Accepted 1 October 2015
Available online 28 November 2015

Keywords:
Recommender systems
Natural noise
Fuzzy logic
Computing with words

ABSTRACT

E-commerce customers demand quick and easy access to products in large search spaces according to their needs and preferences. To support and facilitate this process, recommender systems (RS) based on user preferences have recently played a key role. However the elicitation of customers preferences is not always precise either correct, because of external factors such as human errors, uncertainty and vagueness proper of human beings and so on. Such a problem in RS is known as *natural noise* and can bias customers recommendations. Despite different proposals have been presented to deal with natural noise in RS none of them is able to manage properly the inherent uncertainty and vagueness of customers preferences. Hence, this paper is devoted to a new fuzzy method for managing in a flexible and adaptable way such uncertainty of natural noise in order to improve recommendation accuracy. Eventually a case study is performed to show the improvements produced by this fuzzy method regarding previous proposals.

1. Introduction

The development of the e-commerce has made available huge quantities of information and items that customers cannot filter effectively. Therefore, the facilitation of finding out quick and easily items that fits their preferences and needs is an important challenge nowadays. Recommender systems (RS) are probably the most successful tool to support personalised recommendations [1,2]. Currently, they are widely used in different scenarios like e-commerce [3], e-learning [4,5], e-government [6], tourism [7,8], web pages [9], and digital games [10].

Different approaches have been used in RS, being the content-based (CBRS) [11] and the collaborative filtering (CFRS) [1] the most widespread. CBRS methods are based on items' descriptions to generate the users' recommendations, meanwhile CFRS have performed this task just using users ratings about items.

CFRS are currently the most popular type of RS in real world because they perform very well even when items descriptions are not available. In spite of this, the necessity of customers preferences has produced some problems that limit their performance, such as *cold start* and *sparsity* [1,12], and more recently new related problems regarding the quality of the rating data [13–15]. Specifically, Ekstrand et al. [16] pointed out that the rating elicitation process is not error-free, hence the ratings can contain noise. They mentioned that such a noise, previously coined natural noise in [17], could be caused by human error, mixing of factors in the rating process, uncertainty and other factors. They stated that its detection and correction should provide more accurate recommendations.

Thereby several approaches have been introduced for managing these rating inconsistencies in recommendation scenarios depending on the information available, such as user dependent approaches [14], item-attributes dependent approaches [18], and also approaches that manage natural noise only using the rating values [15]. They all perform generally better that base algorithms, but they present an important limitation to deal with inherent uncertainty and vagueness of preferences because they solely represent and manage them by means of crisp values that may imply lack of robustness.

Therefore, this paper aims at developing a more flexible approach for dealing with natural noise in RS, that properly models such vagueness and uncertainty by means of fuzzy tools such as fuzzy sets [19], fuzzy linguistic approach [20,21] and computing with words [22,23] that have provided successful results modelling that type of uncertainty in other problems [24–26]. Hence, the main objective of this paper is to improve accuracy of RS by managing natural noise with a novel fuzzy-based method that manages the uncertainty present in natural noise by using fuzzy tools [19] that provide a greater flexibility in the characterisation process of elements in the recommendation system, leading to improvements in the recommendation accuracy.

The remainder of this paper is structured as follows. First, Section 2 provides the required background for the current research. Section 3 focuses on the novel fuzzy approach for dealing with natural noise in CFRS. Section 4 then presents an experimental procedure to evaluate the performance of the approach in relation to the previous works. Finally, Section 5 concludes the article and remarks further work.

2. Preliminaries

This section provides the required background for the current research, including basics about CFRS, a review of natural noise processing in CFRS, and a description of the fuzzy logic tools used in the proposal.

* Corresponding author. Tel.: +34 953211902.
E-mail addresses: yeratoledo@gmail.com (R. Yera), jcastro@decsai.ugr.es (J. Castro), martin@ujaen.es (L. Martínez).

http://dx.doi.org/10.1016/j.asoc.2015.10.060
1568-4946/

Table 1
Users' preferences over items, the rating matrix.

	i_1	$\cdots$	i_k	$\cdots$	i_n
u_1	r_{u_1,i_1}	$\cdots$	r_{u_1,i_k}	$\cdots$	r_{u_1,i_n}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
u_j	r_{u_j,i_1}	$\cdots$	r_{u_j,i_k}	$\cdots$	r_{u_j,i_n}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
u_m	r_{u_m,i_1}	$\cdots$	r_{u_m,i_k}	$\cdots$	r_{u_m,i_n}

2.1. Basics in collaborative filtering

A recommender system (RS) has been defined as "*any system that produces individualised recommendations as output or has the effect of guiding the user in a personalised way to interesting or useful objects in a large space of possible options*" [27]. Therefore, the main tasks of a RS are: (i) to gather information about the users' needs and interests, and (ii) to present the items that might satisfy such needs and interests. Therefore, the recommendation problem can be formally defined as finding the most suitable item, or set of items, for a target user, which maximises the rating prediction:

$$Recommendation(I, u_j) = \underset{i_k \in I}{\arg\max}[Prediction(u_j, i_k)] \quad (1)$$

being $I = \{i_1, \ldots, i_n\}$ the set of all the items and $Prediction(u_j, i_k)$ is a function to predict how satisfied would be the user u_j with the item i_k, regarding the data available about u_j and i_k.

Different approaches have been proposed in the literature to recommend, such as content-based [28,29], knowledge-based [30], or demographic-based [31]. However, collaborative filtering (CFRS) is the most widespread approach in RSs [1,2,16,32], because of its ability to provide effective recommendations only requiring minimal information [33]. This information is usually composed by explicit or implicit feedback from the users. This work is focused on CFRS with explicit feedback preferences, r_{u_j,i_k} (see Table 1), which are given by user's preference values over a subset of items.

Among the approaches for CFRS, two pioneer and yet effective methods are the user-based and item-based collaborative filtering approaches [34]. Both methods rely on the nearest neighbours algorithm. Due to the fact that, both represent key methods in the CFRS research, both of them will be used as the base for the evaluation of the natural noise fuzzy approach proposed in this paper.

Additionally, it will be also used two slope one methods, proposed by Lemire and Mclachlan [35], which represent popular and widely used approaches due to their simplicity and appropriate performance [36,37].

2.2. Natural noise treatment in recommender systems

Several authors have pointed out that the user preferences in RS could be inconsistent due to several reasons [17]. These inconsistencies have been classified in two main groups: (i) *malicious noise*, when the inconsistencies are intentionally inserted to bias the recommendation [38], or (ii) *natural noise*, when the inconsistencies appear without malicious intentions [39]. While the malicious noise received much attention since the beginning of the use of RS [40,41], natural noise has attracted less attention from researchers.

The concept of natural noise was introduced by O'Mahony et al. [17], in which the authors focus on discovering noisy ratings, both malicious and natural noisy. Amatriain et al. [39] performed a user study to obtain a better understanding about how natural noise tends to appear. Afterwards in [42] they propose strategies to correct the inconsistent preferences. More recently, Pham and Jung [14] proposed the use of item attributes, such as genre, actors, or director, in the case of movies, to detect and correct natural noisy ratings. They focus on finding a preference model based on this information for each user, and then identify as anomalous those positive ratings that do not match the model. The inconsistent ratings are then corrected using the information associated to other users identified as experts.

While the previous methods focus on the detection of noisy ratings, Li et al. [13] propose the discovery of *noisy but non-malicious users* by detecting user's *self-contradictions*, following the principle that highly-correlated items should be similarly rated. This study focuses on noise detection at user level, which could not be detailed enough in some scenarios. In this direction, Yera et al. [15] proposes a method for correcting noisy preferences that only relies on the ratings and it processes them at a rating level. This approach follows the principle that users and items have their *own tendency* giving or receiving ratings. Once the tendencies have been identified, the ratings that contradict such tendencies can be classified as possibly noisy.

All previous methods manage natural noise in CFRS in different ways, for instance O'Mahony et al. [17] and Li et al. [13] remove noisy information from the dataset. Others use additional information beyond the rating matrix, such as Pham and Jung [14] and Amatriain et al. [42]. Eventually Yera et al. [15] corrects ratings without additional information.

A common limitation of the natural noise processing methods is their rigid management of the noise. As proved in other domains [43,44], the noise cannot be properly managed just like that, due to its inherent uncertainty and vagueness. Therefore, this paper presents an approach that properly manages these features by using fuzzy tools based on fuzzy logic and fuzzy linguistic approach. These approaches have risen to facilitate the information modelling and enhance the reliability and flexibility of classical models. This approach will focus on the natural noise correction just using ratings, and leads to an improvement in the recommendations.

2.3. Fuzzy sets

There are certain real world concepts that cannot be defined in a precise way, such as *old* person, *comfortable* seat or *good* taste. Fuzzy sets and fuzzy linguistic approach [19–21] allow to model uncertain or vague information, through the definition of a fuzzy set over the universe of discourse. Fuzzy sets were proposed by Zadeh [45] to extend the notion of a set by introducing the degree of membership of elements. This establishes a correspondence between the elements of the universe of discourse X into the interval [0, 1], which is given by a membership function:

$$\mu_{\tilde{A}} : X \to [0, 1]$$

This way, a fuzzy set $\tilde{A}$ defined over the domain X is represented by a set of pairs of the element x and its membership:

$$\tilde{A} = (x, \mu_{\tilde{A}}(x)) \quad x \in X, \quad \mu_{\tilde{A}}(x) \in [0, 1] \quad (2)$$

Once defined a fuzzy set, the notions of intersection and union over traditional sets are extended to be defined for fuzzy sets [46,47].

A straightforward and natural tool to model inherent uncertainty and vagueness or characterise preferences is the use of linguistic descriptors. The fuzzy linguistic approach [19,21] based on the fuzzy sets theory [46] provides a direct way to represent and manage such an uncertainty. The use of linguistic information implies to operate with such a type of information. Hence, different computational models and methodologies have been presented in the literature to operate with linguistic information, within the fuzzy theory stands out the Computing with Words (CW) methodology whose roots are based on the fuzzy logic and fuzzy linguistic approach, and provides a direct way to accomplish linguistic

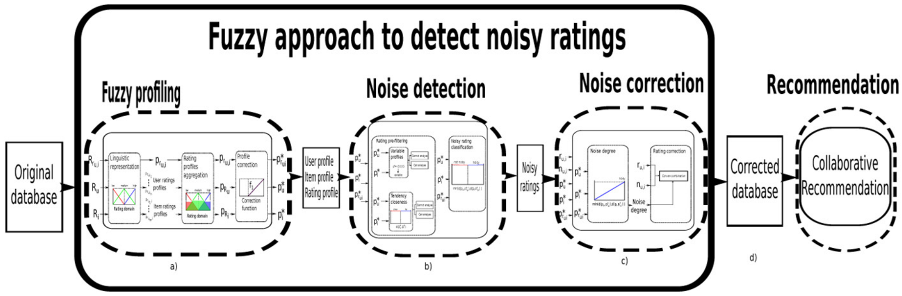


Fig. 1. General scheme of the fuzzy approach to eliminate noise in ratings database.

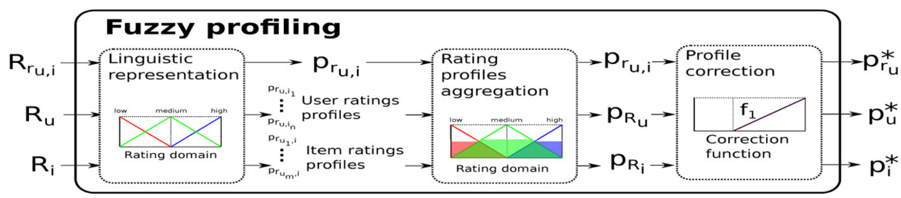


Fig. 2. Fuzzy profiling.

computational processes [22,48]; finally it enhances the reliability and flexibility of crisp models to deal with uncertainty with linguistic information [49].

This paper applies previous fuzzy tools to the natural noise management process, by applying them to user, item and rating profiling.

3. A fuzzy approach to detect noisy ratings

Natural noise in RS may bias recommendations, here it is introduced a new fuzzy approach to deal with it that unlike previous rigid proposals, this new approach uses fuzzy tools for managing in a more flexible way the inherent uncertainty of customers preferences/needs. The general scheme of this fuzzy approach for managing natural noise is depicted in Fig. 1 (amplified in Figs. 2, 5a and 6) and consists of the following steps:

(a) *Fuzzy profiling*: To manage the uncertainty of ratings, they are fuzzyfied and represented by fuzzy linguistic labels to obtain the fuzzy profiles for users, items and ratings.
(b) *Noise detection*: From the previous fuzzy profiles, the ratings are classified as noisy or not noisy, performing first a pre-filtering step that discards unclear profiles and then a noise classification process is applied to.
(c) *Noise correction*: Ratings classified as noisy are processed to be corrected if necessary.
(d) *Output*: It is a database free of natural noisy ratings which can be used in the recommendation process to improve the recommendation results.

Following sections describe in further detail previous steps. But for the sake of clarity, first it is introduced the notation used in the fuzzy approach:

- U is the set of users u in the database, $U = \{u_1, \ldots, u_m\}$.
- I is the set of items i in the database, $I = \{i_1, \ldots, i_n\}$.
- r_{ui} is the rating for user u and item i.
- $S = \{low, medium, high\}$ is the set of fuzzy labels to characterise ratings.
- R_u is the set of ratings associated to user u, also applicable to R_i for the item i, and R_{ui} for the rating r_{ui}.
- p_{R_u} is the fuzzy profile associated to user u (also applicable to p_{R_i} (for the items i), and $p_{R_{ui}}$ (for the rating r_{ui})).
- $p^*_{R_u}$ is the corrected fuzzy profile (also applicable to $p^*_{R_i}$ and $p^*_{R_{ui}}$).
- $p^*_{R_u,l}$ is the membership value of a corrected profile to a given label $l \in \{low, medium, high\}$.
- $R_{\_}$ is the set of ratings referred to a given user, item or rating.
- $p^*_{R_{\_}}$ is the corrected profile associated to a given user, item, or rating.
- $d(p^*_{R_{\_}}, p^*_{R_\bullet})$ is the distance between profiles $p^*_{R_{\_}}$ and $p^*_{R_\bullet}$.
- n_{ui} is the new rating prediction for user u and item i.
- r^*_{ui} is the finally corrected rating.

3.1. Fuzzy profiling

The main idea of our proposal is the proper managing of the vagueness of user's preferences and their possible noise in the elicitation process. The first step of the fuzzy approach consists of a fuzzy linguistic representation of ratings, producing user's, item's and rating's fuzzy profiles that will be then transformed by using CW into *modified* profiles that boost their tendencies in a flexible way. These *modified* profiles will be used to characterise ratings as noisy or not noisy (see Fig. 2).

Consequently, membership functions that characterise the ratings over its universe of discourse are defined. These functions are respectively associated to the fuzzy labels $S = \{low, medium, high\}$, whose fuzzy semantics are shown in Fig. 3.

Remark 1. The rating domain is specific to the recommendation problem, the most common one is from one to five stars, but could be easily extended to other domains.

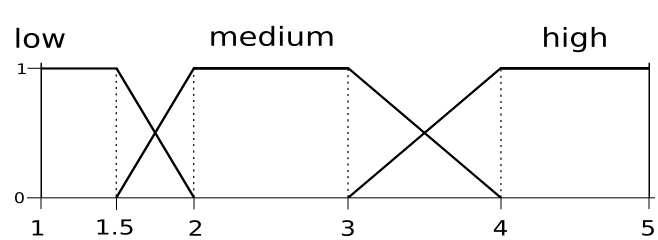


Fig. 3. Fuzzy definition of the rating domain for the one to five stars domain.

Once the ratings are represented by fuzzy linguistic terms [20], they are used to build the users', items', and ratings' profiles. The aim of the profiles is to highlight the *rating tendency* of each element and allow a deeper and more flexible analysis.

Specifically, a user's profile p_{R_u} is computed in terms of the fuzzy representation of user's ratings denoted by R_u (see Eqs. (3) and (4)).

$$\mu_{low}(R_u)=\frac{\sum_{r_{ui}\in R_u}\mu_{low}(r_{ui})}{|R_u|},\quad \mu_{medium}(R_u)=\frac{\sum_{r_{ui}\in R_u}\mu_{medium}(r_{ui})}{|R_u|},$$
$$\mu_{high}(R_u)=\frac{\sum_{r_{ui}\in R_u}\mu_{high}(r_{ui})}{|R_u|} \quad (3)$$

$$p_{R_u}=(\mu_{low}(R_u),\mu_{medium}(R_u),\mu_{high}(R_u)) \quad (4)$$

For items their profiles are built similarly to users and in the rating profile, $p_{R_{ui}}$, for rating r_{ui}. It is only used the rating r_{ui} itself, i.e., $R_{ui}=\{r_{ui}\}$.

This representation facilitates an initial tendency characterisation of the profiles. According to the terms defined in Fig. 3 and the impossibility of having high membership values for the three terms, it is proposed six different tendencies for the profiles:

1. *Low profiles*: Profiles with high membership values for $\mu_{low}(R_u)$ or $\mu_{low}(R_i)$.
2. *Medium profiles*: Profiles with high membership values for $\mu_{medium}(R_u)$ or $\mu_{medium}(R_i)$.
3. *High profiles*: Profiles with high membership values for $\mu_{high}(R_u)$ or $\mu_{high}(R_i)$.
4. *Low–medium profiles*: Profiles with high membership values for $\mu_{low}(R_u)$ and $\mu_{medium}(R_u)$ (for users), or $\mu_{low}(R_i)$ and $\mu_{medium}(R_i)$ (for items).
5. *Medium–high profiles*: Profiles with high membership values for $\mu_{medium}(R_u)$ and $\mu_{high}(R_u)$ (for users), or $\mu_{medium}(R_i)$ and $\mu_{high}(R_i)$ (for items).
6. *Low–high profiles*: Profiles with high membership values for $\mu_{low}(R_u)$ and $\mu_{high}(R_u)$ (for users), or $\mu_{low}(R_i)$ and $\mu_{high}(R_i)$ (for items).

However, in some occasions profiles present similar membership values for all terms, $p_{R_u}\approx(\frac{1}{3},\frac{1}{3},\frac{1}{3})$, $p_{R_i}\approx(\frac{1}{3},\frac{1}{3},\frac{1}{3})$, being impossible to characterise a tendency. Hence to boost tendencies and attenuate unclear cases, our approach makes use of a CW process [19,21] that transforms profiles by increasing high membership values and discarding low ones. Multiple functions can be used to model this behaviour, as depicted in Fig. 4, however, the most suitable for this modification is f_1, given that it performs a softer modification of the values above the k value. The function f_1 has the parameter k, whose value must be greater than $\frac{1}{3}$ to allow certain flexibility. In this case it is set $k=0.35$.

Once the transformation function has been defined, Eq. (5) shows how to generate the *modified* profiles through the function f_1. For the sake of clarity in the rest of the work, the membership value of a *modified* profile to a label $l\in S$ is denoted as $p^*_{R_-,l}$.

$$\begin{aligned} p^*_{R_-} &= (p^*_{R_-,low},p^*_{R_-,medium},p^*_{R_-,high}) \\ &= (f_1(\mu_{low}(R_-)),f_1(\mu_{medium}(R_-)),f_1(\mu_{high}(R_-))) \end{aligned} \quad (5)$$

Remark 2. In the case of a rating profile, the function f_1 does not alter the ratings' profiles, given that they are composed by just one rating value, and then $p^*_{R_{ui}}=p_{R_{ui}}$.

3.2. Noise detection phase

Once it has been obtained the *modified* profiles it is necessary to check if ratings r_{ui} are noisy by analysing the rating tendency of its user u and item i (see Fig. 5a).

Consequently, in the noise detection phase, the profiles $p^*_{R_u}$, $p^*_{R_i}$ and $p^*_{R_{ui}}$ are computed for the user, the item, and the rating, respectively, and used to determine whether the rating matches the user's and item's tendencies (the rating is classified as not noisy in such a case), or it lies out of the tendency that it should follow (in this case the rating may be noisy). Keeping in mind this behaviour, the noise detection phase consists of two main steps (graphically in Fig. 5b):

1. *Rating pre-filtering*: Ratings are analysed to determine whether it is eligible for the noise classification step by comparing the user's and item's profiles with a distance function, and determining if they are close enough.
2. *Noisy rating classification*: Each rating initially included in this phase compares its profile with the user's and item's profile. If the rating's profile is far enough from both of them, it is classified as noisy.

These steps are further detailed in the sections below.

3.2.1. Rating pre-filtering

For detecting noisy ratings it is analysed the rating tendency of both user, u, and item, i, associated to the rating, r_{ui} under study. Specifically, it is interesting to search for those cases in which the user's rating tendency is similar to the item's rating tendency. In these cases, searching for natural noise makes sense, since a rating that lies out of a matching user-item rating tendency is prone to be noisy. On the other hand, when the user's and item's rating tendencies do not match, there is no clear rating tendency that the rating should follow, therefore it is difficult to detect natural noise in those cases.

Therefore, when analysing the rating r_{ui}, it is compared the user's and the item's rating tendency to determine if they are similar enough by using the δ_1 threshold.

$$d(p^*_{R_u},p^*_{R_i})<\delta_1 \rightarrow r_{ui}\ \textit{is elegible for noise classification} \quad (6)$$

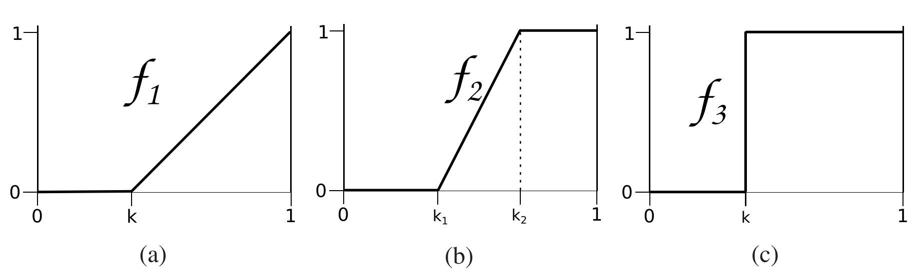


Fig. 4. Different fuzzy transformation functions.

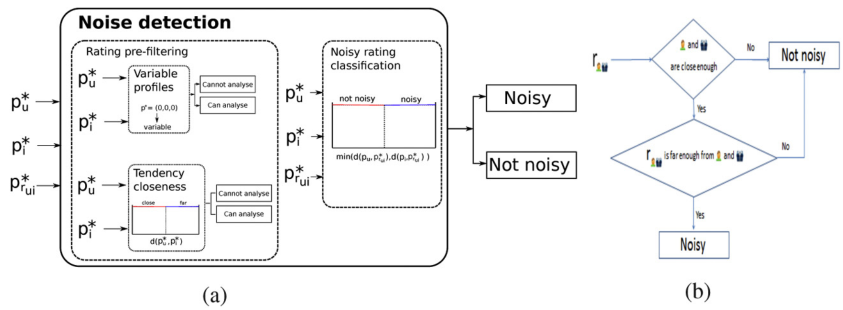


Fig. 5. Noisy detection phase: (a) noise detection and (b) noise detection steps.

There are different similarity measures.[1] In Appendix A there is an analysis about the most widespread similarity measures that justifies the use of the Manhattan distance, Eq. (7), to compare the profiles in our proposal.

$$d(p^*_{R_u}, p^*_{R_i}) = \sum_{l \in S} |p^*_{R_u,l} - p^*_{R_i,l}| < \delta_1 \tag{7}$$

It is necessary to define the δ_1 threshold value to decide at what point the user's and item's profiles are similar enough. With this aim, in Appendix B is shown that the value $d(p^*_{R_u}, p^*_{R_i})$ are always in the interval $[0, 2]$, hence it is set $\delta_1 = 1$ in order to consider similar profiles those ones whose dissimilarity is lower than the 50% of the possible distance values (see Eq. (7)).

Remark 3. Additional experiments were executed to empirically find the best value for the parameter δ_1, concluding that for $\delta_1 = 1$ the proposal obtains a similar performance to the optimal value.

In certain cases the user and/or the item do not show a clear tendency, according to the six possible tendencies presented previously. This kind of profile is called *variable* profile. Regarding the profile definition explained in Section 3.1, the fuzzy transformation function f_1 facilitates the detection of these particular profiles as those having zero membership for all the labels (see Eq. (8)). For these cases, it is difficult to make assumptions about the tendency that the rating should follow, thus the ratings associated to *variable* profiles are not candidates for the noise detection and correction, and they are not then checked according to Eq. (6).

$$p^*_{R_x} = (0, 0, 0) \rightarrow p^*_{R_x} \textit{ is variable} \tag{8}$$

3.2.2. Noisy rating classification

After the ratings having similar user's and item's profiles are detected, the next step is to analyse, just for these ratings, the rating value and determine whether it is noisy or not. In these cases, the rating's profile should be close to the corresponding user's or item's profiles to consider the rating as not-noisy. In other words, a not-noisy rating should be supported by the user's or the item's rating tendency. However, when the rating does not follow either the user's or the item's rating tendencies, it is prone to be noisy.

The user, item and rating profiles represent the rating tendencies, it is then checked how far is the closest of both user's and item's profiles to the rating's profile and compare the distance with δ_2 threshold, to decide whether the rating is noisy or not:

$$\min(d(p^*_{R_u}, p^*_{R_{ui}}),\ d(p^*_{R_i}, p^*_{R_{ui}})) \geq \delta_2 \rightarrow r_{ui} \textit{ is noisy} \tag{9}$$

[1] As the similarity between two objects can be defined from a given distance between those objects, we will use distance function as synonym of dissimilarity measure.

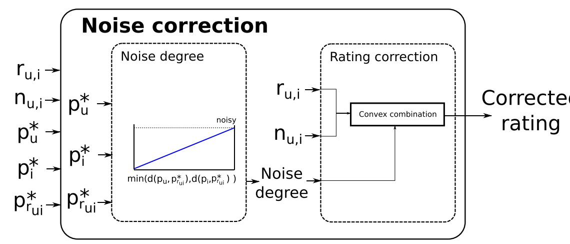


Fig. 6. Noise correction.

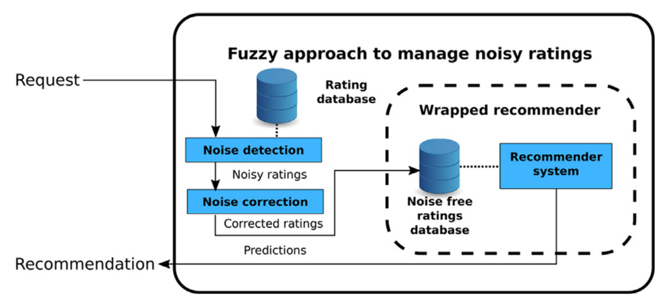


Fig. 7. Architecture of the fuzzy approach to manage noisy ratings.

Regarding the δ_2 threshold, similarly to δ_1 threshold, its value is set to one, since the rating's profile is defined similarly to user's and item's profiles. Hence, $\delta_2 = 1$.

The result of this phase is a classification of all ratings as noisy or not noisy.

3.3. Noise correction phase

The noisy ratings have been already detected, the goal is to correct them for decreasing the noise (see Fig. 6).

Our proposal not only fixed noisy ratings but also it provides a noisy degree of each rating that makes much more flexible and adaptable the rating correction. Therefore ratings with a low degree of noise should be slightly modified meanwhile ratings with a high degree of noise should be greatly modified. To define the noise degree of a noisy rating, it is needed to use the normalised Manhattan dissimilarity [50] between profiles:

$$\textit{dissimilarity}(p^*_{R_x}, p^*_{R_y}) = \frac{d(p^*_{R_x}, p^*_{R_y}) - d_{\min}}{d_{\max} - d_{\min}} = d(p^*_{R_x}, p^*_{R_y}) - 1 \tag{10}$$

where $p^*_{R_x}$ and $p^*_{R_y}$ are the two profiles being compared, and d_{min} is the minimum value distance between two profiles. As the profiles distances are compared with δ_2 threshold, $d_{min} = 1$. d_{max} is the maximum value of the distance between profiles, which in our context is $d_{max} = 2$, as shown in Appendix B.

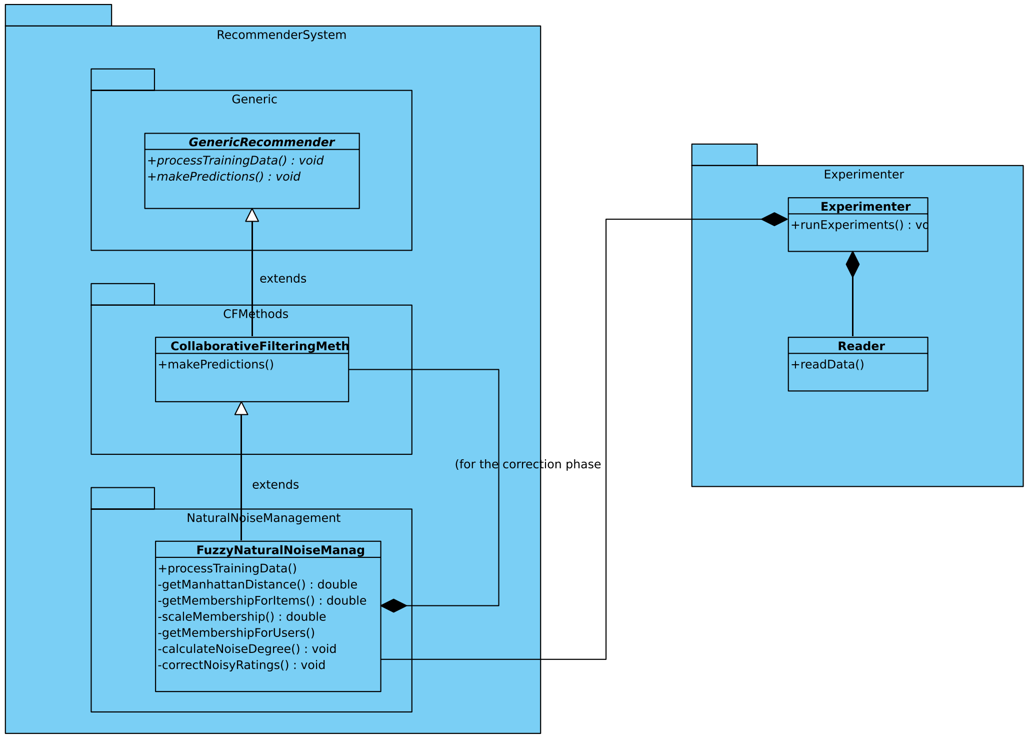


Fig. 8. UML design model.

Definition 1. Let $p^*_{R_{ui}}$, $p^*_{R_u}$ and $p^*_{R_i}$ be the rating, user and item profiles. The noisy degree of the noisy rating is defined as the dissimilarity between profiles:

$$NoiseDegree_{r_{ui}} = T(dissimilarity(p^*_{R_u}, p^*_{R_{ui}}),\ dissimilarity(p^*_{R_i}, p^*_{R_{ui}})) \quad (11)$$

being T the t-norm between the user-rating and the item-rating profiles dissimilarity.

In our context, the *minimum* is used as the t-norm [51].

Once it is known the noise degree of the rating, r_{ui}, a new rating value, n_{ui}, for the same user and item is predicted, through a traditional CFRS using only the not noisy ratings. Finally, noisy ratings are corrected as follows:

$$r^*_{ui} = r_{ui} * (1 - NoiseDegree_{r_{ui}}) + n_{ui} * NoiseDegree_{r_{ui}} \quad (12)$$

where r_{ui} is the original value of the rating, n_{ui} is the rating prediction for user u and item i by using the available data without the noisy ratings, and $NoiseDegree_{r_{ui}}$ is the noise degree of the rating being corrected.

The correction is performed to all noisy ratings providing a de-noised dataset that can be used in the RS to offer more accurate recommendations. Because of this approach does not need additional information (only the original dataset), the noise detection and correction processes can be applied as a pre-processing step, prior to compute recommendations requested to the RS.

3.4. Architecture and implementation

The previous fuzzy approach aims at improving recommendations of CFRS, but it is necessary to clarify the architecture and the implementation of this approach to develop a RS with this approach.

Fig. 7 shows the integration of the fuzzy approach managing natural noise in the general scheme of a RS. It seems reasonable to facilitate such an integration with an existing RS that the system would be wrapped by the natural noise approach. Thus, the ratings are analysed and corrected prior to the recommendation computation. It is clear that the noise correction should not be applied every time that the wrapped recommender system is used, but its application should be carried out by an off-line process from time to time in order to clean the rating database of natural noisy ratings.

The implementation of this fuzzy natural noise management method is available in the website http://sinbad2.ujaen.es/works/recommender-systems/nn-fuzzy in which all source code is accessible and licensed under the terms of the GNU General Public License.[2] The UML design model for a better understanding of the performance of this code is shown in Fig. 8, in which the different packets developed for each phase of the method are briefly described.

4. Case study

This section describes the experiments developed to evaluate the effect of our proposal on the improvement of the recommendation accuracy. Hence our proposal is specified and then compared with other methods that deal with noise in the ratings. First, the datasets (MovieLens, MovieTweeting and Netflix Tiny) in which experiments have been performed are detailed in Section 4.1. Then a specification of the fuzzy method applied in the case study is provided in Section 4.2. After that, the evaluation protocol and the experimental setup are detailed in Sections 4.3 and 4.4. The discussion over the results is included in Section 4.5, and finally a further statistical analysis about the results is presented in Section 4.6.

[2] http://www.gnu.org.

Table 2
MovieLens (ml-100k), MovieTweeting (mt) and Netflix Tiny (netflix-t) datasets.

	ml-100k	mt	netflix-t
Users	943	21,018	4427
Items	1682	12,569	1000
Ratings	100,000	140,000	56,136
Matrix ratio	0.5606	1.6722	4.4270
Sparsity	0.9369	0.9994	0.9873
Rating domain	[1, 5]	[0, 10]	[1, 5]
Rating type	Integer	Integer	Integer
Positive ratings	4	7.5	4
Fuzzy domain	Fig. 3	Fig. 9	Fig. 3

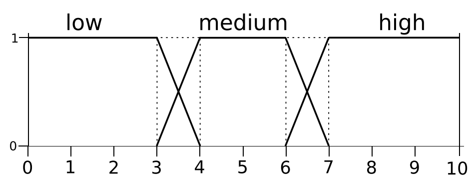


Fig. 9. Fuzzy definition of the ratings domain for MovieTweeting dataset.

4.1. Datasets

To perform a complete case study, the methods are evaluated using three well-known datasets in the field of CFRS: MovieLens[3] (ml-100k), MovieTweetings[4] (mt), and Netflix Tiny[5] (netflix-t) datasets. Each dataset has specific features (see Table 2) that affect the performance of different techniques, such as the rating domain, the sparsity or the relation between the number of users and items.

4.2. Specification of the fuzzy method

According to the general scheme introduced in Fig. 1, the specification of the fuzzy method presented for this case study is the following one:

4.2.1. Fuzzy profiling

Ratings for MovieLens 100k and Netflix Tiny datasets assessed in $[1, 5]$ are transformed to fuzzy representations as follows:

$$low(r_{ui}) = \begin{cases} 1 & if \quad 1 \leq r_{ui} \leq 1.5 \\ -2 * r_{ui} + 4 & if \quad 1.5 \leq r_{ui} \leq 2 \\ 0 & if \quad 2 \leq r_{ui} \leq 5 \end{cases} \tag{13}$$

$$med(r_{ui}) = \begin{cases} 0 & if \quad 1 \leq r_{ui} \leq 1.5 \\ 2 * r_{ui} - 3 & if \quad 1.5 \leq r_{ui} \leq 2 \\ 1 & if \quad 2 \leq r_{ui} \leq 3 \\ -r_{ui} + 4 & if \quad 3 \leq r_{ui} \leq 4 \\ 0 & if \quad 4 \leq r_{ui} \leq 5 \end{cases} \tag{14}$$

$$high(r_{ui}) = \begin{cases} 0 & if \quad 1 \leq r_{ui} \leq 3 \\ r_{ui} - 3 & if \quad 3 \leq r_{ui} \leq 4 \\ 1 & if \quad 4 \leq r_{ui} \leq 5 \end{cases} \tag{15}$$

Ratings of MovieTweeting dataset are assessed in $[0, 10]$ then:

$$low(r_{ui}) = \begin{cases} 1 & if \quad 0 \leq r_{ui} \leq 3 \\ -r_{ui} + 4 & if \quad 3 \leq r_{ui} \leq 4 \\ 0 & if \quad 4 \leq r_{ui} \leq 10 \end{cases} \tag{16}$$

$$med(r_{ui}) = \begin{cases} 0 & if \quad 0 \leq r_{ui} \leq 3 \\ r_{ui} - 3 & if \quad 3 \leq r_{ui} \leq 4 \\ 1 & if \quad 4 \leq r_{ui} \leq 6 \\ -r_{ui} + 7 & if \quad 6 \leq r_{ui} \leq 7 \\ 0 & if \quad 7 \leq r_{ui} \leq 10 \end{cases} \tag{17}$$

$$high(r_{ui}) = \begin{cases} 0 & if \quad 0 \leq r_{ui} \leq 6 \\ r_{ui} - 6 & if \quad 6 \leq r_{ui} \leq 7 \\ 1 & if \quad 7 \leq r_{ui} \leq 10 \end{cases} \tag{18}$$

Now the profiles for each user, item and rating are obtained from their set of ratings in the following way:

$$\begin{aligned} \mu_{low}(R_\bullet) &= \frac{\sum_{r_{ui} \in R_\bullet} \mu_{low}(r_{ui})}{|R_\bullet|} \\ \mu_{med}(R_\bullet) &= \frac{\sum_{r_{ui} \in R_\bullet} \mu_{med}(r_{ui})}{|R_\bullet|} \\ \mu_{high}(R_\bullet) &= \frac{\sum_{r_{ui} \in R_\bullet} \mu_{high}(r_{ui})}{|R_\bullet|} \end{aligned} \tag{19}$$

The CW process to boost tendencies and remove unclear cases are carried out by using function f_1 (see Fig. 10), obtaining the *modified* profiles, $p^*_{R_\bullet}$:

4.2.2. Noise detection

The ratings are classified as noisy or not noisy. First in the pre-filtering process profiles without a clear tendency are discarded (variable profiles).

$$p^*_{R_\bullet} = (0, 0, 0) \rightarrow p^*_{R_\bullet} \text{ is variable} \tag{20}$$

Ratings whose user and item profiles are too far, $d(p^*_{R_u}, p^*_{R_i}) > \delta_1 = 1$ (being d Manhattan distance), are discarded due to an unclear user-item rating tendency, otherwise are eligible for the noise detection phase in which they are classified as noisy or not noisy according to Eq. (9)

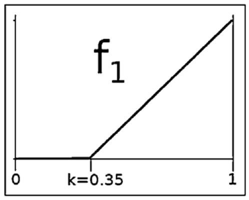
4.2.3. Noise correction

Each noisy rating obtained in the noise detection phase is corrected by computing its noisy degree according to Eqs. (10) and (11), and obtaining a corrected rating, r^*_{ui}, by Eq. (12) in which it is used a new rating n_{ui}, predicted by using UKNN predictor with: (i) neighbourhood size: 60, (ii) similarity measure: Pearson correlation coefficient and (iii) prediction method: weighted average.

4.3. Evaluation protocol

The datasets are prepared according to the procedure proposed by Gunawardana and Shani [52]. To build training and test sets, they suggest to select a group of users from the original dataset and to randomly hide n_u ratings for each user u, where n_u is also randomly selected for each user. These hidden ratings compose the test set, and the remaining ones are chosen to be the training set. To ensure that the results are not biased due to the partitions constructed, the partitioning process is performed several times and

[3] http://grouplens.org/datasets/movielens/.
[4] https://github.com/sidooms/MovieTweetings, version of November 2013.
[5] Netflix Tiny is a small version of Netflix dataset, available in the Personalised Recommendation Algorithms Toolkit (PREA) website http://www.prea.gatech.edu/.



$$f_1(x) = \begin{cases} 0 & if \quad 0 \leq r_{ui} \leq 0.35 \\ \frac{20}{13} * x - \frac{7}{13} & if \quad 0.35 \leq r_{ui} \leq 1 \end{cases} \tag{20}$$

$$p^*_{R_\bullet} = (\mu^*_{low}(R_\bullet), \mu^*_{med}(R_\bullet), \mu^*_{high}(R_\bullet)) \tag{21}$$

Fig. 10. Function used to amplify tendencies.

the results are averaged. Using the training and the test set, the methods' performance is evaluated through the following steps:

1. To apply a noise correction method over the training set, obtaining the modified training set.
2. To recommend with a given recommendation method using the modified training set, which is a de-noised version of the original training set.
3. To evaluate the recommendation results using different evaluation measures.
4. To compare different natural noise approaches based on their recommendation results.

Several evaluation metrics have been used to evaluate CFRS in the literature, highlighting different aspects of the recommendations. In this case we use two widespread metrics [52]: the Mean Absolute Error (MAE), which focuses on how well the recommendation technique predicts the hidden ratings; and the F1 score, which combines the precision and recall of the recommendations to evaluate how accurate is the recommender system at recommending liked items. The F1 score can be measured at different recommendation list sizes, in this case it is computed for the top 5 (F1@5) and top 10 (F1@10).

Our evaluation also depends on a rating prediction approach, which is used in the second step of the evaluation. In this case four widespread CFRS methods are employed, previously referred in Section 2.1: the user-based CFRS, the item-based CFRS [34], the weighted slope one, and the bi-polar slope one [35]. In the rest of the case study they will be respectively named as UKNN, IKNN, SlopeOne, and BPSlope. For UKNN and IKNN methods, Pearson's correlation coefficient is used as similarity measure. Also for UKNN and IKNN, the number of neighbours in the prediction calculation are fixed to $k = 60$.

4.4. Experimental setup

For evaluating our proposal, it is compared with previous works. Specifically, it is compared with the baseline, which computes the recommendations without performing a noise preprocessing; and with three approaches conceived to process natural noise. Therefore, five different methods for dealing with noise in recommender systems are considered:

- *Base*: The recommendation predictor results are computed with the original dataset. The Base method is included in the experiments to establish a minimum performance of the noise preprocessing methods.
- *DiffBased*: It eliminates the ratings whose difference with a new prediction for the same user and item is higher than one unit in the rating scale [17].
- *NNMU*: It eliminates the top noisiest users, which are detected by discovering contradictions when they rate similar items [13]. In the experiments, up to ten users are eliminated. In the case that a prediction is requested for an eliminated user, the global rating bias is assigned [53].
- *NN-Crisp*: It corrects ratings whose value lies out of the user-item detected category and the difference between the original and a prediction exceeds one [15].
- *NN-Fuzzy*: It corrects ratings detected as noisy by checking the distances between the user's, item's, and rating's profiles, which are defined using fuzzy sets. The prediction of new ratings n_{ui} is performed using the user-based CFRS approach.

4.5. Results

Once the experimentation has been performed, the results are shown in Tables 3–5 for MAE, F1@5 and F1@10, respectively. In the tables, the result of the best preprocessing method for each configuration are highlighted in bold.

Regarding the overall results for MAE, NN-Fuzzy achieved the best performance for all cases, proving that the NN-Fuzzy method benefits the results in prediction performance. For the MovieLens dataset, in the UKNN, IKNN, and SlopeOne cases, the second best performance was associated to NN-Crisp. However, in the case of BPSlope, NN-Crisp performs worse than the baseline, while the proposed method is still the best. On the other hand, NNMU slightly improves the baseline for UKNN and IKNN but performs worse for SlopeOne and BPSlope. An interesting observation is that the DiffBased method performs worse than the baseline for all cases in this dataset regarding this metric.

Focusing on the results on each dataset, for MovieTweeting, NN-Fuzzy also introduces improvements compared with previous works. Additionally, the MAE value associated to NNMU is worst than the baseline for UKNN and IKNN, while the MAE in DiffBased is worst for BPSlope. In the case of Netflix Tiny dataset, NNMU only outperforms the baseline for the IKNN predictor, and DiffBased has a similar or high performance in relation to the baseline for all cases. Finally, NN-Crisp also improves the baseline for all cases, although its behaviour is always under NN-Fuzzy.

Given that F1@5 and F1@10 measures are associated to the same property of the recommender, but measured at different recommendation list size, their results are analysed together. F1@5 and F1@10 results are shown in Tables 4 and 5.

Regarding the overall results, for F1@5 and F1@10 measure the NN-Fuzzy approach is the best for almost all the configurations, with the exception of BPSlope prediction on MovieLens and MovieTweeting datasets. Examining the concrete results values, it is clear that the application of the noise preprocessing methods provide a relative improvement for F1 score than for MAE, whose results improvements are tight.

Focusing on the results on each dataset, for MovieLens dataset, the best performances for BPSlope is associated to Base. In the rest of the cases (UKNN, IKNN and Slope), in this dataset the four methods for handling natural noise always outperform the baseline according to the F1@5 measure, being NNMU the approach with the worst performance. This behaviour is also related to F1@10, where NNMU is under the baseline for three of the four recommendation methods.

In the case of MovieTweeting dataset, the improvements of NN-Fuzzy are small in relation to NN-Crisp. Similarly to MAE results,

Table 3
MAE results on the different datasets.

Dataset	Predictor	Base	DiffBased	NNMU	NN-Crisp.	NN-Fuzzy
MovieLens	UKNN	0.7647	0.7662	0.7644	0.7632	**0.7608**
	IKNN	0.7705	0.7749	0.7699	0.7674	**0.7656**
	SlopeOne	0.7771	0.7801	0.7780	0.7756	**0.7728**
	BPSlope	0.7837	0.7954	0.7843	0.7840	**0.7812**
MovieTweeting	UKNN	1.2117	1.1966	1.2139	1.1971	**1.1924**
	IKNN	1.2032	1.2006	1.2055	1.1834	**1.1783**
	SlopeOne	1.3372	1.3081	1.3356	1.3027	**1.2937**
	BPSlope	1.3058	1.3244	1.3025	1.2982	**1.2915**
Netflix	UKNN	0.7867	0.7746	0.7903	0.7769	**0.7720**
	IKNN	0.8013	0.7904	0.7978	0.7874	**0.7820**
	SlopeOne	0.8161	0.8062	0.8259	0.8048	**0.8004**
	BPSlope	0.8068	0.8077	0.8127	0.7967	**0.7941**

Table 4
F1@5 results on the different datasets.

Dataset	Predictor	Base	DiffBased	NNMU	NN-Crisp.	NN-Fuzzy
MovieLens	UserKNN	0.2005	0.2143	0.2050	0.2164	**0.2230**
	ItemKNN	0.1959	0.2085	0.2003	0.2094	**0.2144**
	SlopeOne	0.1855	0.2034	0.1871	0.2022	**0.2078**
	BPSlope	**0.2278**	0.2222	0.2276	0.2264	0.2265
MovieTweeting	UserKNN	0.2412	0.2435	0.2404	0.2537	**0.2544**
	ItemKNN	0.2347	0.2383	0.2341	0.2470	**0.2479**
	SlopeOne	0.2298	0.2074	0.2321	0.2391	**0.2400**
	BPSlope	0.2305	0.1910	0.2314	**0.2321**	0.2320
Netflix	UKNN	0.2210	0.2413	0.2246	0.2358	**0.2579**
	IKNN	0.2430	0.2539	0.2339	0.2507	**0.2628**
	SlopeOne	0.2314	0.2484	0.2251	0.2432	**0.2593**
	BPSlope	0.2841	0.2813	0.2840	0.2835	**0.2855**

Table 5
F1@10 results on the different datasets.

Dataset	Predictor	Base	DiffBased	NNMU	NN-Crisp.	NN-Fuzzy
MovieLens	UserKNN	0.2406	0.2620	0.2404	0.2644	**0.2720**
	ItemKNN	0.2408	0.2597	0.2414	0.2598	**0.2674**
	SlopeOne	0.2291	0.2540	0.2254	0.2520	**0.2603**
	BPSlope	**0.2874**	0.2831	0.2817	0.2872	**0.2874**
MovieTweeting	UserKNN	0.2593	0.2644	0.2581	0.2789	**0.2793**
	ItemKNN	0.2563	0.2622	0.2552	0.2729	**0.2742**
	SlopeOne	0.2502	0.2223	0.2495	0.2618	**0.2630**
	BPSlope	0.2486	0.2004	0.2483	**0.2511**	0.2500
Netflix	UKNN	0.1878	0.2071	0.1896	0.2017	**0.2236**
	IKNN	0.2218	0.2356	0.2124	0.2303	**0.2439**
	SlopeOne	0.2149	0.2332	0.2091	0.2263	**0.2419**
	BPSlope	0.2674	0.2659	0.2652	0.2673	**0.2694**

for F1@5 score, NNMU performs worst than Base for UKNN and IKNN, and DiffBased for BPSlope. In the case of F1@10, the results for NNMU are worst than the baseline for all cases. Despite these facts, NN-Fuzzy method performs best for almost all the scenarios, with the exception of BPSlope, where NN-Crisp slightly outperforms it.

For Netflix regarding the F1@5 results, NN-Fuzzy was the only one that outperforms the baseline for the four predictors. In the case of DiffBased, it performs worse than the baseline in BPSlope, while NNMU obtains a lower value for ItemKNN and SlopeOne. At last, NN-Crisp performs worse than the baseline in BPSlope. The results associated to the F1@10, shown in Table 5, are similar to F1@5 in general term, with the exception of BPSlope, where NNMU achieved the worst performance, in contrast to the previous case where DiffBased was the method with the lowest value.

Globally, the results show certain interesting behaviours. The most important are those related to the NNMU, which tends to achieve bad results, and a relative low performance for BPSlope with the natural noise handling methods. It is important to remark that NNMU approach is conceived to detect noisy-but-non malicious users. This implies that, rather than making a rating level analysis to detect noise, it performs a user level analysis. Although NNMU improves Base in some cases, it seems that a loss of valuable information is produced, which affects the recommendations. This suggests that completely removing users is not the best approach to manage noise, thus a lower level analysis should be done, at a rating-level.

In relation to the results of the methods using the BPSlope recommender, its performance is particularly affected by the noise preprocessing methods, resulting in a smaller performance improvement. The reason for this is that the natural noise handling methods tend to convert low ratings into higher ones [15]. This, in conjunction with the fact that BPSlope is strongly based on the polarity of the ratings, implying that it relies more on negative ratings than the rest of the recommendation methods, produces a loss of information which the BPSlope recommender considers

Table 6
Friedman *p*-values for the three evaluation metrics.

	p-Value
MAE	3.0828024066842374E−7
F1@5	7.510660850740258E−5
F1@10	4.94787648741557E−7

Table 7
Holm table for MAE an NN-Fuzzy as the control method.

i	Algorithm	*p*
4	NNMU	1.2027508945921403E−7
3	DiffBased	2.0416942840373563E−5
2	Base	6.278731906212546E−5
1	NN-Crisp	0.05280751141611364

Table 8
Holm table for F1@5 an NN-Fuzzy as the control method.

i	Algorithm	*p*
4	NNMU	3.6090232367496604E−5
3	DiffBased	3.6090232367496604E−5
2	Base	0.0012488309880884449
1	NN-Crisp	0.03886710381241722

Table 9
Holm table for F1@10 an NN-Fuzzy as the control method.

i	Algorithm	*p*
4	NNMU	2.8359172119178996E−8
3	DiffBased	6.278731906212546E−5
2	Base	0.0012488309880884449
1	NN-Crisp	0.03886710381241729

important. The results on F1@5 and F1@10 for MovieLens dataset, in which the Base method performs better than the others, confirm that.

4.6. Statistical analysis

To perform a deeper analysis of the results, we apply statistical tests to verify the significance of the results [54]. At first, the Friedman test ($p = 0.05$) is used to determine whether the difference of ranks is significant. Once the null hypothesis of the Friedman test has been rejected, the Holm test ($p = 0.05$) is applied, by using as the control algorithm the one with the best ranking in the Friedman test.

Tables 6–9 present the details of these tests performed over the recommendation results. They were execute through the five compared methods, over 12 problem cases (the three datasets times the four rating predictors) to check statistical significance. For the Holm test, we will analyse the results for significance levels of $p = 0.05$, given that this value are widely used for this specific case [55–57].

In the case of the Friedman's, all the tests rejected the null hypothesis of equality for the three evaluation measures, for a significance level of $p = 0.05$. This fact implies that there are significant differences between the five compared methods.

Afterwards, in the case of the Holm tests, NN-Fuzzy was the control method for the three metrics MAE, F1@5 and F1@10. The tests were able to reject all the null hypotheses that the NN-Fuzzy is statistically equal to each other technique, for a significance level of $p = 0.05$ in the three evaluation measures, with the exception of the Holm test over MAE results, in which the null hypothesis of equality between NN-Fuzzy and the method ranked in the second place ($p = 0.052806$) could not be rejected at $p = 0.05$.

5. Concluding remarks

Natural noise is denominated to the noise unintentionally introduced by human beings when they are eliciting preferences. Its management in recommender systems has attracted the attention of many researchers and several proposals have been done. However, such proposals deal with natural noise in a rigid way such an small variation can make change the classification of ratings as noisy or not noisy.

Therefore, to avoid such a strict managing process, this paper has introduced a more flexible process to deal with natural noise based on fuzzy tools that makes much more adaptable the classification and correction of noisy ratings in RS. Eventually, the outperformance of the fuzzy method proposed here regarding previous ones has been shown in an extensive case study that validates with the results that the fuzzy management of natural noise provides better accuracy in RS than rigid methods.

Our future research will be focused on study the managing of natural noise on other recommendation scenarios, such as context-awareness, group recommendations and hybrid systems.

Acknowledgements

This research work was partially supported by the Research Project TIN-2012-31263, the Eureka SD Project (agreement number 2013-2591), which is supported by the Erasmus Mundus Programme of the European Union, and also the Spanish Ministry of Education, Culture and Sport FPU fellowship (FPU13/01151).

Appendix A. Similarity measures in the natural noise scenario

This appendix discusses about the suitability of the different similarity measures in our natural noise scenario.

An important group of works focus on the study of similarity measures [58–60]. In this context, Cha [58] presented an extensive review of the similarity measures and provided a classification of them into several families such as the Minkowski family, the intersection family, the inner product family, and the Shannon's entropy family. More recently, Chiclana et al. [59] presented a comparative study of five similarity measures (Manhattan, Euclidean, Cosine, Dice and Jaccard) in a consensus scenario of group decision making. Taking into account the needs of ourscenario for comparing profiles, the most widespread similarity measures has been analysed as follows:

- Cosine and inner product similarities [60]. They are used when the vectors are sparse and have similarity of lacking characteristics. However, in our scenario it is expected to compare aligned and orthogonal vectors, giving as result only the values one or zero, respectively. For these cases, the cosine similarity cannot properly compare them. This fact discards its use and the use of other inner product similarity measures, which share a similar behaviour.
- Pearson's correlation coefficient. Recently Grabusts [61] have recalled that this measure tends to detect the difference in shapes rather than determining the magnitude of difference between two objects, thus it does not reflect properly the differences in our current scenario.
- Chebyshev distance. It belongs to Minkowski distance family (see Eq. (A.1)). Minkowski distances directly reflect the equivalence between homologous categories of the profiles, assuming that two profiles are similar when their memberships to *low*, *medium* and *high* are close. However, for Chebyshev distance, only the dimension with the largest difference is taken into account and

the rest of them do not influence the distance value. Therefore, it is not the optimal choice as the similarity measure.

$$d_{Minkowski}(x,y)=\sqrt[\lambda]{\sum_k |x_k-y_k|^{\lambda}} \tag{A.1}$$

- Euclidean and $\lambda>1$ Minkowski distances. Although these distances consider all the dimensions, i.e., a variation on any dimension influences the distance value, they place two vectors closer if the differences are split equally among the dimensions and further if the differences are concentrated in one dimension. In our case, this is not the correct behaviour, since the fact that the differences are concentrated in a single dimension does not implies a higher difference between profiles.
- Manhattan distance. It reflects the differences between the dimensions without giving importance to how these differences are distributed among the dimensions. Also the distance reflects in a direct way the differences between the memberships. These properties make the Manhattan distance the best choice for the profile comparison context.

$$d_{Manhattan}(x,y)=\sum_k |x_k-y_k| \tag{A.2}$$

Appendix B. Proof of distance between profiles

This appendix is focused on proving that $d(p^*_{R_u}, p^*_{R_i}) \in [0,2]$. This result is used to assign the initial values to parameters δ_1 and δ_2, which are used in the natural noise treatment method proposed.

The triangular inequality in the form $|a-b|\le a+b$ (for $a\ge 0$ and $b\ge 0$) directly implies:

$$|p^*_{R_u,low}-p^*_{R_i,low}| \le p^*_{R_u,low}+p^*_{R_i,low} \tag{B.1}$$

$$|p^*_{R_u,medium}-p^*_{R_i,medium}| \le p^*_{R_u,medium}+p^*_{R_i,medium} \tag{B.2}$$

$$|p^*_{R_u,high}-p^*_{R_i,high}| \le p^*_{R_u,high}+p^*_{R_i,high} \tag{B.3}$$

Adding Eqs. (B.1)–(B.3), it remains:

$$\begin{aligned}&|p^*_{R_u,low}-p^*_{R_i,low}|+|p^*_{R_u,medium}-p^*_{R_i,medium}|+|p^*_{R_u,high}-p^*_{R_i,high}|\\ &\le p^*_{R_u,low}+p^*_{R_u,medium}+p^*_{R_u,high}+p^*_{R_i,low}+p^*_{R_i,medium}+p^*_{R_i,high}\end{aligned} \tag{B.4}$$

However, Eqs. (3) and (5) imply that:

$$p^*_{R_u,low}+p^*_{R_u,medium}+p^*_{R_u,high}\le 1 \tag{B.5}$$

$$p^*_{R_i,low}+p^*_{R_i,medium}+p^*_{R_i,high}\le 1 \tag{B.6}$$

Putting Eqs. (B.5) and (B.6) in Eq. (B.4), it remains:

$$|p^*_{R_u,low}-p^*_{R_i,low}|+|p^*_{R_u,medium}-p^*_{R_i,medium}|+|p^*_{R_u,high}-p^*_{R_i,high}|\le 2 \tag{B.7}$$

The left member of this equation is $d(p^*_{R_u}, p^*_{R_i})$, so it directly concludes that $d(p^*_{R_u}, p^*_{R_i})\le 2$. Finally, the fact that it is defined through the sum of absolute values (Eq. (7)) guarantees that $d(p^*_{R_u}, p^*_{R_i})\ge 0$. Therefore, $d(p^*_{R_u}, p^*_{R_i}) \in [0,2]$.

References

[1] G. Adomavicius, A.T. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.

[2] J.A. Konstan, J. Riedl, Recommender systems: from algorithms to user experience, User Model. User-Adapt. Interact. 22 (1–2) (2012) 101–123.

[3] B. Xiao, I. Benbasat, E-commerce product recommendation agents: use, characteristics, and impact, Manag. Inf. Syst. Q. 31 (1) (2007) 137–209.

[4] C. De Maio, G. Fenza, M. Gaeta, V. Loia, F. Orciuoli, S. Senatore, RSS-based e-learning recommendations exploiting fuzzy FCA for knowledge modeling, Appl. Soft Comput. 12 (1) (2012) 113–124.

[5] R.Y. Toledo, Y.C. Mota, An e-learning collaborative filtering approach to suggest problems to solve in programming online judges, Int. J. Distance Educ. Technol. 12 (2) (2014) 51–65.

[6] X. Guo, J. Lu, Intelligent e-government services with personalized recommendation techniques, Int. J. Intell. Syst. 22 (5) (2007) 401–417.

[7] J. Noguera, M. Barranco, R. Segura, L. Martínez, A mobile 3D-GIS hybrid recommender system for tourism, Inf. Sci. 215 (2012) 37–52.

[8] S. Schiaffino, A. Amandi, Building an expert travel agent as a software agent, Expert Syst. Appl. 36 (2, Part 1) (2009) 1291–1299.

[9] G. Castellano, A.M. Fanelli, M.A. Torsello, Newer: a system for neuro-fuzzy web recommendation, Appl. Soft Comput. 11 (1) (2011) 793–806.

[10] P.H. Abreu, D.C. Silva, F. Almeida, J. Mendes-Moreira, Improving a simulated soccer team's performance through a memory-based collaborative filtering approach, Appl. Soft Comput. 23 (2014) 180–193.

[11] M. Pazzani, D. Billsus, Content-based recommendation systems, Adapt. Web 4321 (2007) 325–341.

[12] R. Rodríguez, M. Espinilla, P. Sánchez, L. Martínez, Using linguistic incomplete preference relations to cold start recommendations, Internet Res. 20 (3) (2010) 296–315.

[13] B. Li, L. Chen, X. Zhu, C. Zhang, Noisy but non-malicious user detection in social recommender systems, World Wide Web 16 (5–6) (2013) 677–699.

[14] H.X. Pham, J.J. Jung, Preference-based user rating correction process for interactive recommendation systems, Multimed. Tools Appl. 65 (1) (2013) 119–132.

[15] R. Yera, Y. Caballero, L. Martínez, Correcting noisy ratings in collaborative recommender systems, Knowl.-Based Syst. 76 (2015) 96–108.

[16] M.D. Ekstrand, J.T. Riedl, J.A. Konstan, Collaborative filtering recommender systems, Found. Trends Hum.-Comput. Interact. 4 (2) (2011) 81–173.

[17] M.P. O'Mahony, N.J. Hurley, G. Silvestre, Detecting noise in recommender system databases, in: Proceedings of the 11th International Conference on Intelligent User Interfaces, ACM, 2006, pp. 109–115.

[18] X. Amatriain, A. Jaimes, N. Oliver, J.M. Pujol, Data mining methods for recommender systems, in: Recommender Systems Handbook, Springer US, 2011, pp. 39–71, Ch. 2.

[19] R.R. Yager, L.A. Zadeh, An Introduction to Fuzzy Logic Applications in Intelligent Systems, vol. 165, Springer Science & Business Media, 2012.

[20] R. Rodríguez, L. Martínez, An analysis of symbolic linguistic computing models in decision making, Int. J. Gen. Syst. 42 (1) (2013) 121–136.

[21] L.A. Zadeh, Computing With Words: Principal Concepts and Ideas. Studies in Fuzziness and Soft Computing, Springer, Berlin, 2012.

[22] L. Martínez, D. Ruan, F. Herrera, Computing with words in decision support systems: an overview on models and applications, Int. J. Comput. Intell. Syst. 3 (4) (2010) 382–395.

[23] J. Mendel, L. Zadeh, R. Yager, J. Lawry, H. Hagras, S. Guadarrama, What computing with words means to me, IEEE Comput. Intell. Mag. 5 (1) (2010) 20–26.

[24] F. Estrella, M. Espinilla, L. Martínez, Fuzzy linguistic olive oil sensory evaluation model based on unbalanced linguistic scales, J. Multiple-Valued Logic Soft Comput. 22 (2014) 501–520.

[25] P. Sánchez, L. Martínez, C. García, F. Herrera, E. Herrera-Viedma, A fuzzy model to evaluate the suitability of installing an ERP system., Inf. Sci. 179 (14) (2009) 2333–2341.

[26] M. Espinilla, J. Montero, J. Rodríguez, Computational intelligence in decision making, Int. J. Comput. Intell. Syst. 7 (Suppl. 1) (2014) 1–5.

[27] R. Burke, Hybrid recommender systems: survey and experiments, User Model. User-Adapt. Interact. 12 (4) (2002) 331–370.

[28] L. Martínez, L.G. Pérez, M. Barranco, A multigranular linguistic content-based recommendation model: research articles, Int. J. Intell. Syst. 22 (5) (2007) 419–434.

[29] J. Castro, R.M. Rodríguez, M.J. Barranco, Weighting of features in content-based filtering with entropy and dependence measures, Int. J. Comput. Intell. Syst. 7 (1) (2014) 80–89.

[30] L. Martínez, M.J. Barranco, L.G. Pérez, M. Espinilla, A knowledge based recommender system with multigranular linguistic information, Int. J. Comput. Intell. Syst. 1 (3) (2008) 225–236.

[31] M.G. Vozalis, K.G. Margaritis, Using SVD and demographic data for the enhancement of generalized collaborative filtering, Inf. Sci. 177 (15) (2007) 3017–3037.

[32] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.

[33] I. Pilászy, D. Tikk, Recommending new movies: even a few ratings are more valuable than metadata, in: Proceedings of the Third ACM Conference on Recommender Systems, ACM, 2009, pp. 93–100.

[34] C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, in: Recommender Systems Handbook, Springer US, 2011, pp. 107–144, Ch. 4.

[35] D. Lemire, A. Maclachlan, Slope one predictors for online rating-based collaborative filtering SDM, vol. 5, SIAM, 2005, pp. 1–5.

[36] C. Mao, J. Chen, QoS prediction for web services based on similarity-aware slope one collaborative filtering, Informatica 37 (2) (2013) 139–148.

[37] Z. Mi, C. Xu, A recommendation algorithm combining clustering method and slope one scheme, in: Bio-Inspired Computing and Applications, Springer, 2012, pp. 160–167.
[38] C. Dellarocas, Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior, in: Proceedings of the 2nd ACM Conference on Electronic Commerce, ACM, 2000, pp. 150–157.
[39] X. Amatriain, J.M. Pujol, N. Oliver, I like it. . .i like it not: evaluating user ratings noise in recommender systems, in: User Modeling, Adaptation, and Personalization, Springer, 2009, pp. 247–258.
[40] I. Gunes, C. Kaleli, A. Bilge, H. Polat, Shilling attacks against recommender systems: a comprehensive survey, Artif. Intell. Rev. 42 (4) (2014) 767–799.
[41] B. Mobasher, R. Burke, R. Bhaumik, C. Williams, Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness, ACM Trans. Internet Technol. 7 (4) (2007), Article 23.
[42] X. Amatriain, N. Lathia, J.M. Pujol, H. Kwak, N. Oliver, The wisdom of the few: a collaborative filtering approach based on expert opinions from the web, in: Proceedings of the 32nd International ACM SIGIR Conference, ACM, New York, NY, USA, 2009, pp. 532–539.
[43] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. 250 (2013) 113–141.
[44] S. Vaishali, K. Rao, G. Rao, A review on noise reduction methods for brain MRI images, in: 2015 International Conference on Signal Processing and Communication Engineering Systems (SPACES), 2015, pp. 363–365, http://dx.doi.org/10.1109/SPACES.2015.7058284.
[45] L. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[46] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic – Theory and Applications, Prentice Hall, 1995.
[47] H.-J. Zimmermann, Fuzzy Set Theory and Its Applications, Springer Science & Business Media, 2001.
[48] L. Martínez, F. Herrera, An overview on the 2-tuple linguistic model for computing with words in decision making: extensions, applications and challenges, Inf. Sci. 207 (1) (2012) 1–18.
[49] L. Martínez, D. Ruan, F. Herrera, E. Herrera-Viedma, P. Wang, Linguistic decision making: tools and applications, Inf. Sci. 179 (14) (2009) 2297–2298.
[50] L. Candillier, F. Meyer, F. Fessant, Designing specific weighted similarity measures to improve collaborative filtering systems, in: Advances in Data Mining. Medical Applications, E-Commerce, Marketing, and Theoretical Aspects, Springer, 2008, pp. 242–255.
[51] E.P. Klement, R. Mesiar, E. Pap, Triangular Norms, Springer, 2000.
[52] A. Gunawardana, G. Shani, A survey of accuracy evaluation metrics of recommendation tasks, J. Mach. Learn. Res. 10 (2009) 2935–2962.
[53] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[54] J. Mandel, The Statistical Analysis of Experimental Data, Courier Corporation, 2012.
[55] Y. Caises, A. González, E. Leyva, R. Pérez, Combining instance selection methods based on data characterization: an approach to increase their effectiveness, Inf. Sci. 181 (20) (2011) 4780–4798.
[56] R. Nicolas, A. Sancho-Asensio, E. Golobardes, A. Fornells, A. Orriols-Puig, Multilabel classification based on analog reasoning, Expert Syst. Appl. 40 (15) (2013) 5924–5931.
[57] Y.B. Ruiz-Blanco, Y. Marrero-Ponce, P.J. Prieto, J. Salgado, Y. García, C.M. Sotomayor-Torres, A Hooke's law-based approach to protein folding rate, J. Theor. Biol. 364 (2015) 407–417.
[58] S.-H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, City 1 (2) (2007) 1.
[59] F. Chiclana, J.T. Garcia, M.J. del Moral, E. Herrera-Viedma, A statistical comparative study of different similarity measures of consensus in group decision making, Inf. Sci. 221 (2013) 110–123.
[60] M.M. Deza, E. Deza, Encyclopedia of Distances, Springer Berlin Heidelberg, 2009.
[61] P. Grabusts, Distance metrics selection validity in cluster analysis, Sci. J. Riga Tech. Univ. Comput. Sci. 45 (1) (2011) 72–77.