Int. J. Appl. Math. Comput. Sci., 2019, Vol. 29, No. 3, 595–610
DOI: 10.2478/amcs-2019-0044

ON EXPLAINABLE FUZZY RECOMMENDERS AND THEIR PERFORMANCE EVALUATION

TOMASZ RUTKOWSKI [a,c,*], KRYSTIAN ŁAPA [b], RADOSŁAW NIELEK [c]

[a] Senfino, 1412 Broadway 21st floor, New York City, NY 10018, USA
e-mail: XAI@senfino.com

[b] Institute of Computational Intelligence
Częstochowa University of Technology, Al. Armii Krajowej 36, 42-200 Częstochowa, Poland

[c] Faculty of Information Technology
Polish-Japanese Academy of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland

This paper presents a novel approach to the design of explainable recommender systems. It is based on the Wang–Mendel algorithm of fuzzy rule generation. A method for the learning and reduction of the fuzzy recommender is proposed along with feature encoding. Three criteria, including the Akaike information criterion, are used for evaluating an optimal balance between recommender accuracy and interpretability. Simulation results verify the effectiveness of the presented recommender system and illustrate its performance on the MovieLens 10M dataset.

Keywords: recommender systems, explainable recommendations, fuzzy systems, Akaike information criterion.

1. Introduction

In the literature, several types of recommender systems have been presented (see, e.g., Lops *et al.*, 2011; Wei *et al.*, 2017; Zhang *et al.*, 2018), and the most popular techniques are known under the names “content-based filtering” and “collaborative filtering.”

In the case of content-based filtering, recommender systems suggest to a user items (e.g., movies or books) characterized by features similar to those ones that the user preferred in the past. In this scenario, recommendations are based on the content of a given item. In collaborative filtering, items are recommended to a user by similarities to other users (similar users’ preferences). In that scenario, recommendations are based on other users’ rates concerning the items and, e.g., the weighted average of the rates.

Modern recommender systems are designed with the use of machine learning algorithms. For an excellent survey, a reader is referred to the work of Portugal *et al.* (2018). The major drawback of the existing techniques is the lack of explainability. In many applications, e.g., medical diagnosis or venture capital investment recommendations, it is essential to explain the rationale behind a specific recommendation. Motivated by this fact, in this paper we propose a novel approach to design explainable recommenders.

In order to create an explainable recommendation system, fuzzy IF-THEN rules are employed to represent knowledge about users’ preferences. Then, the inference based on fuzzy logic is applied in the Mamdani fuzzy system (see, e.g., Rutkowska, 2002) to produce the recommendations. The method of fuzzy rule generation, proposed by Wang and Mendel (1991), is now combined with a technique of rule reduction. This approach is enhanced by three criteria, including the Akaike information criterion (see, e.g., Söderström and Stoica, 1989), which allow evaluating an optimal balance in terms of the recommender’s accuracy and interpretability.

The explainability of the proposed recommender is assured due to the following:

1. Interpretable fuzzy rules with fuzzy sets as linguistic values of attributes describing items and users’ preferences (fuzzy sets with semantic meanings).

*Corresponding author

2. Incorporation of rule weights into the fuzzy system. The weights can be interpreted with regard to rule importance.

3. The reduction of the fuzzy rules makes the rule base simpler, and thus easier to produce explainable recommenders.

It is worth emphasizing that this approach leads to a moderate number of interpretable fuzzy rules and, in consequence, greatly facilities the explanation of the recommender system.

Moreover, the use of the Akaike information criterion, as well as the final prediction error and the Schwartz criterion (see Section 3.3), allows solving the problem of the compromise between the system error and the number of rules.

The presented approach greatly improves our previous attempts (Rutkowski *et al.*, 2018a; 2018b) to the design of explainable recommender systems. The paper is organized as follows. Section 2 presents a short description of the Wang–Mendel rule generation method for construction of the Mamdani system with rule weighting factors. This is the background and our starting point to develop explainable fuzzy recommenders. The main part of this contribution is given in Section 3 by describing the proposed method (Algorithm 1) and feature encoding along with model evaluation criteria. Simulation results illustrating the performance of the system are described in Section 4. Finally, Section 5 contains the main conclusions and remarks.

2. Interpretable fuzzy rules

In order to employ fuzzy IF-THEN rules in the recommender system proposed in this paper, we recall the idea of the Wang–Mendel algorithm (see Wang and Mendel, 1991) for fuzzy rule generation.

It should be emphasized that the material which we present below is perhaps the first mathematically precise description of that method in the multidimensional case. Now, we introduce the notation that will be useful in describing a recommendation system and its features' encoding (see Section 3.2).

We consider a fuzzy system with n inputs and one output. Let $x_1, x_2, \ldots, x_n$ and y be linguistic variables corresponding to input and output variables, respectively, of the fuzzy system.

The input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ in the space $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \cdots \times \mathbf{X}_n$, as well as $y \in Y$, can take crisp values, denoted as $\overline{\mathbf{x}} = [\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n]^T$ and $\overline{y}$, respectively; in this case, each universe of discourse can be the space of real numbers.

With regard to fuzzy sets in fuzzy IF-THEN rules, it should be noted that generally different numbers of linguistic values (fuzzy sets) can be defined for particular linguistic variables. Denote by N_i the number of linguistic fuzzy sets for $x_i,\ i = 1, 2, \ldots, n$, and by N_y the number of fuzzy sets for y.

Assume that the input fuzzy sets, denoted by A_{il}, are defined by membership functions $\mu_{A_{il}}(x_i),\ i = 1, 2, \ldots, n$, and $l = 1, 2, \ldots, N_i$, in the universes of discourse $X_1, X_2, \ldots, X_n$, and analogously the output fuzzy sets, B_l, are defined by membership functions $\mu_{B_l}(y)$, $l = 1, 2, \ldots, N_y$, in Y.

We can assign linguistic values for the inputs and output as follows:

$$
\begin{aligned}
x_1 &: A_{11}, A_{12}, \ldots, A_{1N_1},\\
x_2 &: A_{21}, A_{22}, \ldots, A_{2N_2},\\
&\vdots\\
x_n &: A_{n1}, A_{n2}, \ldots, A_{nN_n},\\
y &: B_1, B_2, \ldots, B_{N_y}.
\end{aligned}
$$

Suppose that we recorded M input-output data pairs $\overline{\mathbf{x}}^{(j)}, \overline{y}^{(j)}$, where $j = 1, 2, \ldots, M$, and $\overline{\mathbf{x}}^{(j)} = [\overline{x}_1{}^{(j)}, \overline{x}_2{}^{(j)}, \ldots, \overline{x}_n{}^{(j)}]^T$.

The task is to generate a set of fuzzy IF-THEN rules from the input-output data. The first step of the Wang–Mendel method is to divide the input and output spaces into fuzzy regions. Then, in the second step, fuzzy rules are generated from given data pairs. The number of fuzzy regions refers to the numbers of linguistic values, N_i, for $i = 1, 2, \ldots, n$, and N_y.

Having M data pairs $\overline{\mathbf{x}}^{(j)}, y^{(j)}$, for $j = 1, 2, \ldots, M$, degrees of membership in different regions are determined, and every data pair is put into the region of maximal degree, according to the equations

$$\mu_{A_i^j}(\overline{x}_i^{(j)}) = \max_{l=1,2,\ldots,N_i} \left\{ \mu_{A_{il}}(\overline{x}_i{}^{(j)}) \right\}, \tag{1}$$

for $i = 1, 2, \ldots, n$, and

$$\mu_{B^j}(\overline{y}^{(j)}) = \max_{l=1,2,\ldots,N_y} \left\{ \mu_{B_l}(\overline{y}^{(j)}) \right\}, \tag{2}$$

Finally, we obtain one rule from one input-output data pair. In this way, from the second step, we get M fuzzy IF-THEN rules, R^j, for $j = 1, 2, \ldots, M$, of the following form:

$$\textbf{IF } x_1 \text{ is } A_1^j \text{ AND } x_2 \text{ is } A_2^j \text{ AND} \ldots \text{ AND } x_n \text{ is } A_n^j$$

$$\textbf{THEN } y \text{ is } B^j. \tag{3}$$

This means that A_i^j, $i = 1, 2, \ldots, n$, are the fuzzy sets chosen from A_{il}, where $l = 1, 2, \ldots, N_i$, which satisfies Eqn. (1), while B^j equals the fuzzy set from B_l, where $l = 1, 2, \ldots, N_y$, which satisfies Eqn. (2).

In the third step, a degree is assigned to each of the M rules, with the use of (1) and (2), as follows:

$$D(R^j) = \mu_{A_1^j}(\overline{x}_1^{(j)})\mu_{A_2^j}(\overline{x}_2^{(j)})\dots\mu_{A_n^j}(\overline{x}_n^{(j)})\mu_{B^j}(\overline{y}^{(j)}). \quad (4)$$

As a matter of fact, the degree is obtained calculating the value of the rule firing level for the data pair that generated this rule, multiplied by the membership of the consequent fuzzy set.

The antecedent matching degree, also called the degree of activation of the rule R^j or the rule firing level, is expressed by

$$\tau_j = \prod_{i=1}^{n} \mu_{A_i^j}(\overline{x}_i) \quad (5)$$

for $j = 1, 2, \dots, M$.

Equations (4) and (5) can also be expressed in a more general form, by using a T-norm:

$$D(R^j) = T\Big\{\mu_{A_1^j}(\overline{x}_1^{(j)}), \mu_{A_2^j}(\overline{x}_2^{(j)}), \dots, \mu_{A_n^j}(\overline{x}_n^{(j)}), \mu_{B^j}(\overline{y}^{(j)})\Big\} \quad (6)$$

and

$$\tau_j = T\left\{\mu_{A_1^j}(\overline{x}_1^{(j)}), \mu_{A_2^j}(\overline{x}_2^{(j)}), \dots, \mu_{A_n^j}(\overline{x}_n^{(j)})\right\}. \quad (7)$$

The product, and the min operation are the most often used examples of t-norm functions.

The degree, $D(R^j)$, assigned to each rule R^j, $j = 1, 2, \dots, M$, allows reducing the number of the rules, including into the rule base of a fuzzy system only the rules with the maximal degree in particular regions. In this way, the problem of conflicting rules, i.e., those that have the same IF part but different THEN parts, is solved, resulting in the reduction of the initial number of rules.

Thus, starting from M rules generated by M data pairs, the Wang–Mendel algorithm produces the rule base of a fuzzy system composed of the reduced number of N rules, where $N \leq M$.

The Mamdani type of a fuzzy system with inference based on N fuzzy IF-THEN rules, generated by the Wang–Mendel method, can be described by the following mathematical model:

$$\overline{y} = \frac{\sum_{k=1}^{N} \overline{y}^k \tau_k}{\sum_{k=1}^{N} \tau_k}, \quad (8)$$

where $\overline{y}^k$, $k = 1, 2, \dots, N$, is is the point in which the membership function $\mu_{B^k}(y)$ takes the maximal value, and τ_k is given by (5), where j is replaced by k and M by N.

It should be noted that $\tau_1, \tau_2, \dots, \tau_N$ in Eqn. (8) are special cases of those determined using (7).

Introducing to the antecedents of the rules (8) their importance weights, we get

$$\overline{y} = \frac{\sum_{k=1}^{N} w_k \overline{y}^k \tau_k}{\sum_{k=1}^{N} w_k \tau_k} \quad (9)$$

which can describe the fuzzy recommender studied in this paper.

Remark 1. Although the formula (9) has been applied in the literature in various problems of classification and modeling (see, e.g., Alvarez-Estevez and Moret-Bonillo, 2018; Ishibuchi and Nakashima, 2001; Ishibuchi and Yamamoto, 2005; Nauck and Kruse, 1998; Rutkowski, 2004), for the first time it will be used in the context of designing fuzzy explainable recommenders. Moreover, our approach allows significantly reducing the number of rules.

Despite the fact that the usefulness of rule weighs is discussed in the literature (see, e.g., Simiński, 2010; Nauck and Kruse, 1998), we find this issue interesting from the interpretability and explainability point of view with regard to recommender systems. In general, the weights can be interpreted as the rule importance in the sense of expressing the number of data items in a dataset described by this rule. The more data items match the rule, the more important it is.

3. Description of the proposed method

This section presents the main algorithm proposed for the design of explainable recommender systems, including the procedure of rule reduction. The idea of the method is described in Section 3.1. Then, in Section 3.2, the procedure of feature encoding, introduced in order to apply the recommender to the MovieLens data, is presented in detail. In Section 3.3, three criteria are used for evaluating an optimal balance between the recommender's accuracy and interpretability. Finally, a short summary is included in Section 3.4.

3.1. Idea of the proposed method. There are various methods of improving the performance of fuzzy systems, from introducing weights of the fuzzy rules (see, e.g., Ishibuchi and Nakashima, 2001), through reducing the number of the fuzzy rules (see, e.g., Cpalka, 2017), to optimizing a fuzzy system, usually by modifying fuzzy set parameters (see, e.g., Jin, 2000) or rule consolidation (see, e.g., Riid and Preden, 2017) and using the collaborative fuzzy clustering (see, e.g., Prasad *et al.*, 2017).

In this paper, a hybrid solution is proposed in which system optimization is applied after successive reductions of fuzzy rules (see Algorithm 1). The optimization of the system, in this case, refers to optimized values of the weights. The rule reduction procedure should increase the accuracy of the system.

In order to reduce the rules, the method of removing successively the least beneficial fuzzy rules was used. To find the least beneficial fuzzy rule, particular rules are turned off and on in the system one by one, and the system error is calculated after each change. The least beneficial rule is the one whose RMSE was the lowest after reduction.

As explained above, during the optimization of the system, we assume that only the weights (initialized equally by default) of the rules can be modified. For this purpose, we use the standard evolutionary strategy ES ($\mu + \lambda$) (see, e.g., Rutkowski, 2008). However, any optimization algorithm can be applied instead. This approach allows keeping a semantically interpretable form of fuzzy sets and increasing system accuracy.

Moreover, this approach is more legible than the interpretation of systems in which fuzzy sets become uninterpretable after the optimization of their parameters (see, e.g., Jin, 2000). The optimized system will be labeled as WO in the remainder of this paper.

Two approaches will be considered: C1, in which values of the weights are reset each time the fuzzy rule is removed, and C2, in which the values of the weights are stored after the optimization and do not reset.

The performance of the system in which the values of the weights were rounded (to one decimal place) after the optimization was also tested. The goal was to increase the transparency of the fuzzy rule notation. The approach with the rounded weight values is marked in this paper as WR, while WO denotes the system with optimized weights (not rounded values).

The idea of the presented approaches is shown in Algorithm 1. System optimization and evaluation are repeated after the reduction of each fuzzy rule, with the goal of finding the best reduction level for which the weight values give the best system performance. The evaluation is illustrated in Section 4.

It is worth adding that, as part of the tests, the results for a given number of rules and a specific reduction level were saved and then averaged. This is due to the fact that, in recommendation systems, the datasets for each user are different so a different number of fuzzy rules is created by using the WM (Wang–Mendel) method.

Algorithm 1. Design and reduction of explainable recommender systems.

```
for all users do
  Create N fuzzy rules using the WM method
  Set system weights to equal values
  while N > 3 do
    Evaluate system (WM)
    Optimize system weights using ES
    Store system weights (only for variant C2)
    Evaluate system (WO)
    Round system weights
    Evaluate system (WR)
    Reset system weights (only for variant C1)
    for k = 1 to N do
      Temporarily remove k-th fuzzy rule
      Evaluate the system and store the error
      Include the removed fuzzy rule
    end for
    Remove the least beneficial fuzzy rule
  end while
end for
```

3.2. Feature encoding. Objects (items) that can be recommended are described by attributes (features). In recommender systems, attribute values can be of different type, not necessarily crisp (numerical). In general, an object is characterized by attributes whose values can be numerical or categorical (nominal), and other data types can be considered as well.

Denote by $\mathbf{a} = [a_1, a_2, \dots, a_n]^T$ a vector of attributes describing an object $\mathbf{o}_j$ that belongs to $\mathbf{O}$ that is a space of the objects. The vector $\mathbf{a}$ corresponds to the vector of linguistic variables $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, considered in Section 2.

Now, let us focus our attention on attribute values. In the case of categorical (nominal, linguistic) values, we can use symbols N_i, for $i = 1, 2, \dots, n$, and N_y concerning both the number of values of attributes (input variables) and output (decision) variables, respectively. Fuzzy sets as linguistic values can be viewed as special cases of categorical values.

Let $\mathbf{v} = [v_1, v_2, \dots, v_n]^T$ denote a vector of values of particular attributes a_i, for $i = 1, 2, \dots, n$, in general, and let $\mathbf{v}^j = [v_1{}^j, v_2{}^j, \dots, v_n{}^j]^T$ represent this vector for the object $\mathbf{o}_j$, where $j = 1, 2, \dots, M$. It is worth noticing that the vector $\mathbf{v}^j$ corresponds to the vector $\overline{\mathbf{x}}^{(j)} = [\overline{x}_1{}^{(j)}, \overline{x}_2{}^{(j)}, \dots, \overline{x}_n{}^{(j)}]^T$, considered in Section 2.

However, with regard to recommender systems, objects may be characterized by attributes that take more than only one value, and not necessarily values belonging to the set of reals.

For example, when movies are considered, their attributes such as genre can be described by categorical values, e.g., action, comedy, drama, and there are movies characterized, e.g., as comedy and drama.

Let $V_i = \{V_{i1}, V_{i2}, \dots, V_{iN_i}\}$ be a set of values of the attribute a_i, for $i = 1, 2, \dots, n$. The notation of the attribute values corresponds to fuzzy linguistic values, A_{il}, for $i = 1, 2, \dots, n$ and $l = 1, 2, \dots, N_i$, presented in Section 2.

For the above-mentioned example, a_i can be $genre$ and $V_i = \{action, comedy, drama, \dots\}$. Of course, each object $\mathbf{o}_j$, $j = 1, 2, \dots, M$, which is a movie in this case, is characterized by attributes with values taken from the same set of the attribute values.

Let d_j, $j = 1, 2, \ldots, M$, be the score (rating) of the object $\mathbf{o}_j$, given by a user. With regard to the movies, users assess chosen movies by using grades from the set $\{2, 3, 4, 5\}$ of the user's preference values that express how much the user likes a movie.

The rating values, in a movie dataset (e.g., *MovieLens*), correspond to the output values $\overline{y}^{(j)}$ in the input-output data pairs, considered in Section 2, with regard to the output variable $y \in Y$. The number of output values, N_y, in the case of discrete values of the grades $\{2, 3, 4, 5\}$ equals 4.

Let us define the set of matrices $\mathbf{C}^{(i)} = [c_{jl}{}^{(i)}]_{M \times N_i}$, for $i = 1, 2, \ldots, n$, where $c_{jl}{}^{(i)}$, for $j = 1, 2, \ldots, M$ and $l = 1, 2, \ldots, N_i$, are determined according to the following equation:

$$c_{jl}{}^{(i)} = \begin{cases} d_j & \text{if } V_{il} \text{ is in object } \mathbf{o}_j, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

It should be explained that each matrix $\mathbf{C}^{(i)}$, $i = 1, 2, \ldots, n$, includes M rows corresponding to M objects (items), e.g., movies, and N_i columns labeled by the values $V_{i1}, V_{i2}, \ldots, V_{iN_i}$ of the attribute a_i. The elements $c_{jl}{}^{(i)}$ of the matrix $\mathbf{C}^{(i)}$, for $j = 1, 2, \ldots, M$ and $l = 1, 2, \ldots, N_i$, equal d_j if the attribute value V_{il} occurs in an object $\mathbf{o}_j$, and zero otherwise. With regard to the example of the movie characterized as comedy and drama, every element in the row assigned to this movie equals zero except for those referring to comedy and drama.

In order to apply the fuzzy system described in Section 2 as the recommender, we need M input-output data pairs $\overline{\mathbf{x}}^{(j)}, \overline{y}^{(j)}$, $j = 1, 2, \ldots, M$, and $\overline{\mathbf{x}}^{(j)} = [\overline{x}_1{}^{(j)}, \overline{x}_2{}^{(j)}, \ldots, \overline{x}_n{}^{(j)}]^T$. As mentioned earlier, this corresponds to the pairs $\mathbf{v}^j$, d_j, where $\mathbf{v}^j = [v_1{}^j, v_2{}^j, \ldots, v_n{}^j]^T$ represents single numerical values referring to particular attributes a_i^j, $i = 1, 2, \ldots, n$, of an object $\mathbf{o}_j$, $j = 1, 2, \ldots, M$.

We call the single values that characterize the attributes the attribute preferences, and determine them based on the attribute value preferences in the following way:

$$\psi(V_{il}) = \frac{\sum_{j=1}^{M} c_{jl}{}^{(i)}}{m_l{}^{(i)}}, \tag{11}$$

where ψ means the preference and $m_l{}^{(i)}$ denotes the number of non-zero elements in the l-th column of the matrix $\mathbf{C}^{(i)}$, $i = 1, 2, \ldots, n$.

Then, the attribute preference, $v_i{}^j$, is determined as

$$v_i{}^j = \frac{\sum_{l=1}^{\alpha_j{}^{(i)}} \psi^*(V_{il})}{\alpha_j{}^{(i)}}, \tag{12}$$

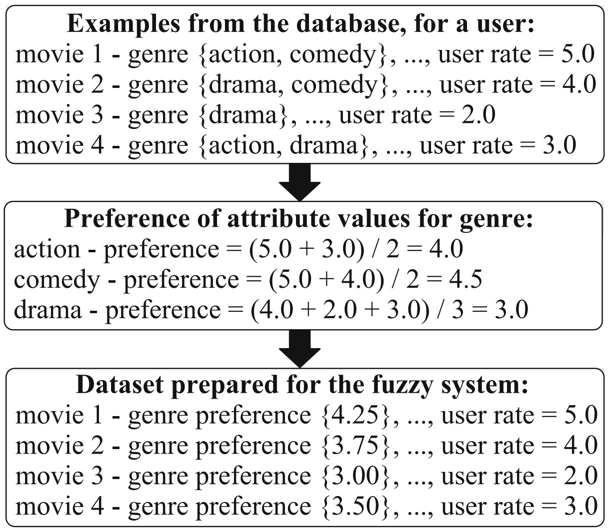


Fig. 1. Example of preparation of the dataset for a user.

for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, M$, where ψ^* denotes the preference of only those attribute values V_{il}, $l = 1, 2, \ldots, N_i$, that are included as values of the object $\mathbf{o}_j$, and $\alpha_j{}^{(i)}$ is the number of $\psi^*(V_{il})$.

Figure 1 illustrates the above-described process of determining the attribute value preferences, according to Eqn. (11), and the results of the attribute preference obtained by use of the formula (12). This simple example refers to $M = 4$ movies as the objects $\mathbf{o}_j$, $j = 1, 2, 3, 4$, assessed by a user with the ratings 5, 4, 2, and 3, respectively. This means that $d_1 = 5$, $d_2 = 4$, $d_3 = 2$, $d_4 = 3$. The attribute $genre$ is considered, and $\mathbf{o}_1$ is characterized by two attribute values, i.e., *action* and *comedy*, object $\mathbf{o}_2$ by *drama* and *comedy*, object $\mathbf{o}_3$ by only one value, i.e., *drama*, and $\mathbf{o}_4$ by *action* and *drama*. Therefore, the attribute value preferences, ψ, are calculated for *action*, *comedy*, and *drama* as V_{il}, for $l = 1, 2, 3$ and i referring to the $genre$ as the attribute.

It is very easy to present matrix $\mathbf{C}^{(i)}$ for this simple example. The following $genre$ preference values were obtained in this case: 4.25 for $\mathbf{o}_1$, 3.75 for $\mathbf{o}_2$, 3.00 for $\mathbf{o}_3$, and 3.50 for $\mathbf{o}_4$. In the same way, we calculate the preference values for other attributes, and all of them are included in the data pairs $\mathbf{v}^j$, d_j, for the fuzzy recommender. It should be emphasized that this example concerns only one user with regard to the recommendation system. The same procedure is repeated for other users.

3.3. Solution evaluation. Since the rates of objects are numerical values, the rules obtained by the WM method are used in this paper as the base for the Mamdani type fuzzy system for regression tasks (see, e.g., Cpalka, 2017) with the goal of user rate prediction. This allows calculating the RMSE (commonly used error definition, known as the root mean square error), and also obtaining more accurate classification of recommended objects.

In addition to the RMSE, the accuracy (ACC) of predictions of the exact user rate were calculated. To make this possible, the system's output values were rounded to the possible values of the object's rate and in this way the standard classification error (ACC) could be calculated (see, e.g., Kuncheva, 2000). There was also a classification error (YES/NO), checking whether or not an item (object) should be recommended (if the value of its recommendation is more than a half of possible values). This approach is considered in some other papers.

Because the proposed method allows obtaining a different system accuracy for a different reduction level of fuzzy rules, choosing the optimal balance in terms of accuracy–interpretability is not a trivial task. In this paper, the use of isocriterial lines and criteria for model evaluation with regard to their complexity is proposed (see, e.g., Söderström and Stoica, 1989).

The first criterion considered is the Akaike information criterion (AIC), which is defined as follows:

$$\text{AIC} = M \cdot \ln Q + 2 \cdot p, \tag{13}$$

where M stands for the number of dataset samples (in this paper this value was set to the average number of the dataset samples generated for all users), Q denotes the system error, p means parameters that are optimized in the system (equal to the number of weights and analogously the number of fuzzy rules).

The second criterion used is the final prediction error (FPE), defined as follows:

$$\text{FPE} = Q \cdot \left(\frac{M \cdot n + p}{M \cdot n - p} \right), \tag{14}$$

where n denotes the number of system inputs.

The last criterion considered is the Schwarz criterion which is expressed by the following equation:

$$S = M \cdot \ln Q + p \cdot \ln M. \tag{15}$$

The isocriterial lines are lines representing fixed values of the criteria with different values of system errors and the number of system parameters. This approach allows solving the problem of a compromise between the system error and the number of optimized parameters describing the system. The points located on the isocriterial lines, with the smallest criterion values, characterize systems that are called sub-optimal. These provide the smallest values of statistical criteria within the examined structures (in this case, differing in the number of system rules).

3.4. Summary of the proposed method.

The proposed approach (a) is based on the WM method, which allows generating interpretable and simple fuzzy rules, (b) uses weights' optimization and does not modify fuzzy sets parameters in order to keep interpretable fuzzy sets, (c) applies fuzzy rule reduction not only to simplify the system but also to decrease the system error, (d) allows checking how the optimization of system weights works with a different level of rule reduction, (e) employs the standard ES for weight optimization, (f) includes a version that rounds fuzzy rules weights to increase the system's interpretability, and (g) uses isocriterial lines in order to keep the optimal balance between the system interpretability and system error.

4. System performance evaluation

Simulations that illustrate the performance of the system proposed for recommendation are described in this section. The results of the simulations verify the effectiveness of the recommender system for the MovieLens 10M dataset. The datasets are available on the Internet as versions of different sizes (see Harper and Konstan, 2015). These datasets contain user ratings for movies.

4.1. Description of computer simulations.

In the simulations, the cases described in Section 3.1 and presented in Table 1 were tested. In case C1, the weight values are reset after each reduction of a fuzzy rule, while in case C2, weight, the values are stored after optimization and do not reset (see Algorithm 1).

Table 1. Simulation cases.

Case	Variant	Weight values
WM	–	not optimized
WO-C1	C1	optimized and reset
WR-C1	C1	optimized, rounded, and reset
WO-C2	C2	optimized and do not reset
WR-C2	C2	optimized, rounded, do not reset

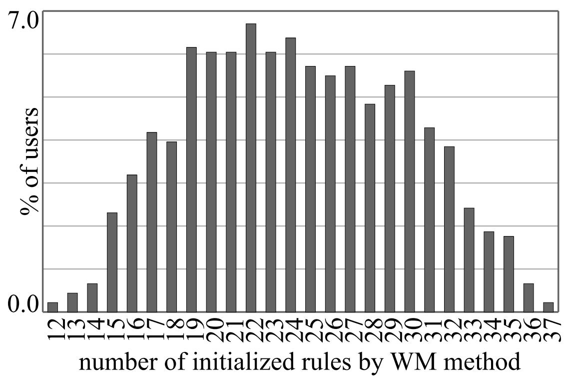


Fig. 2. Histogram of the initial number of rules generated with the use of the WM method.

The number of fuzzy sets (linguistic values), N_i, for each attribute corresponding to linguistic variables x_i, $i = 1, 2, \ldots, n$, was chosen as Gaussian-type fuzzy sets, for $N_i = 5$. The following parameters of the ES algorithm were set: population size = 32, number of iterations = 100, evaluation function = RMSE.

For the simulations, the MovieLens 10M database is used, and three inputs, $n = 3$, are considered: genre preference (explained in Section 3.2), year (numeric values), and keywords preference. Moreover, datasets were prepared for the first 100 users who rated more than 30 movies from the database.

As a testing method, 10-fold cross-validation was applied, and only learning dataset samples were taken for creating fuzzy rules with the WM method (the remaining dataset samples are used as testing samples).

Since rating predictions for unknown samples is important in recommendation systems, the error values for testing samples are crucial while comparing the results. Therefore, most of the conclusions in this paper are focused on this analysis (see Tables 2–4).

4.2. Simulation results. The histogram of the initial number of rules generated by using the WM method is shown in Fig. 2. Depending on a user and the number of rated objects, the WM generates different numbers of fuzzy rules, with an average of 24 rules for the proposed feature encoding. The number of rules for particular users differs from 12 to 37.

Detailed simulation results are presented in Tables 2 (RMSE), 3 (ACC) and 4 (YES/NO). The values of the criteria are included in Tables 5 and 6. The isolines, for the AIC, FPE and Schwarz criteria, are illustrated in Figs. 4 and 5.

A comparison of the WM, WO-C1, WR-C1, WO-C2 and WR-C2 cases is shown in Figs. 6, 7 and 8. The optimization process of weights for different numbers of fuzzy rules is presented in Fig. 9, and the histogram of accuracy obtained for different users is portrayed in Fig. 3. In addition, examples of fuzzy sets and fuzzy rules obtained for the presented methods are shown in Tables 7 (according to the reduction level of fuzzy rules considering obtained accuracy) and 8 (according to the reduction in fuzzy rules for the AIC, FPE and Schwarz criteria).

Analyzing Tables 2 and 3, we see that 32%–36% of the rules generated by the WM method can be removed, resulting in better performance of the system on both learning and testing data samples, with regard to the average RMSE and ACC, although in Table 4, where the YES/NO classification error is considered, the same result is obtained only for the learning samples but not for the testing samples. It is worth noticing that the value of the YES/NO error in the case of 36% rule reduction does not differ much from the value of this error for the case of 0%, i.e., no rule reduction. This means that 36% of the rules can be removed, but not more in all the cases of the WM system.

Similar results were obtained for the WO-C2 system, which is the best case compared with WO-C1, WR-C1,

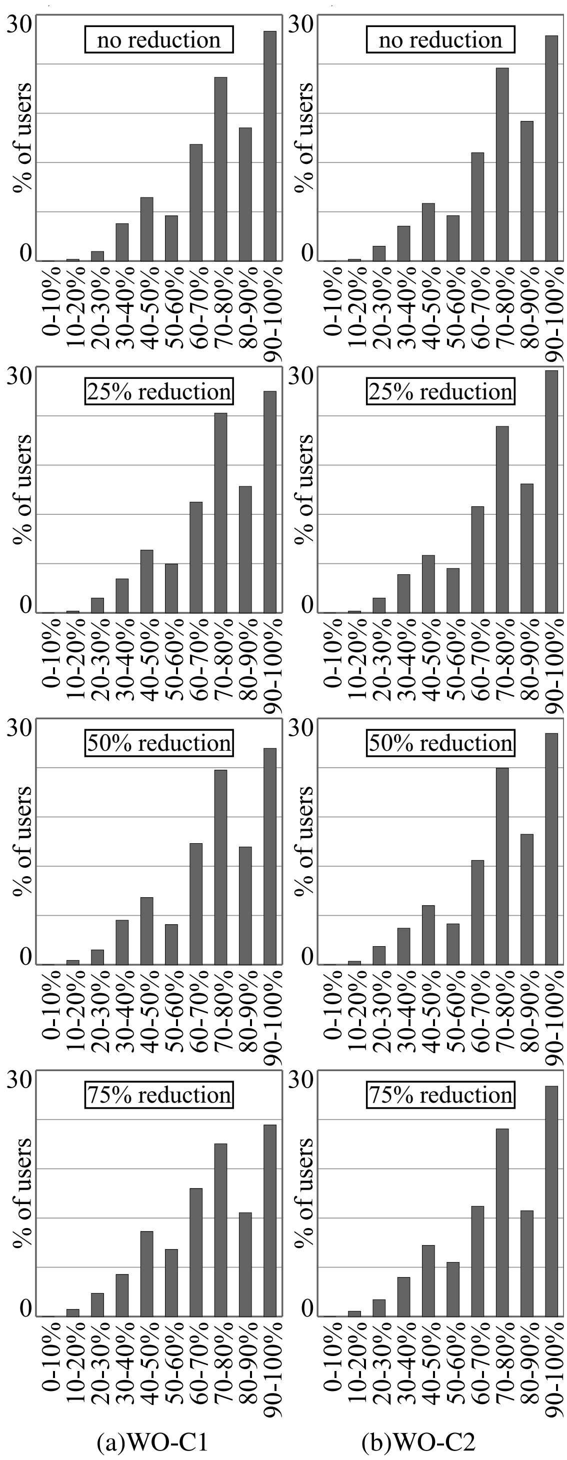
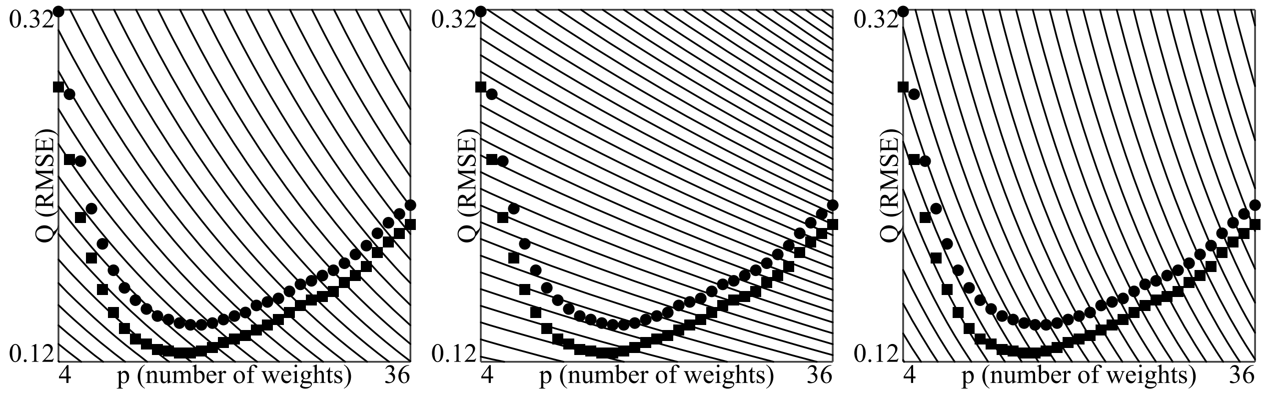
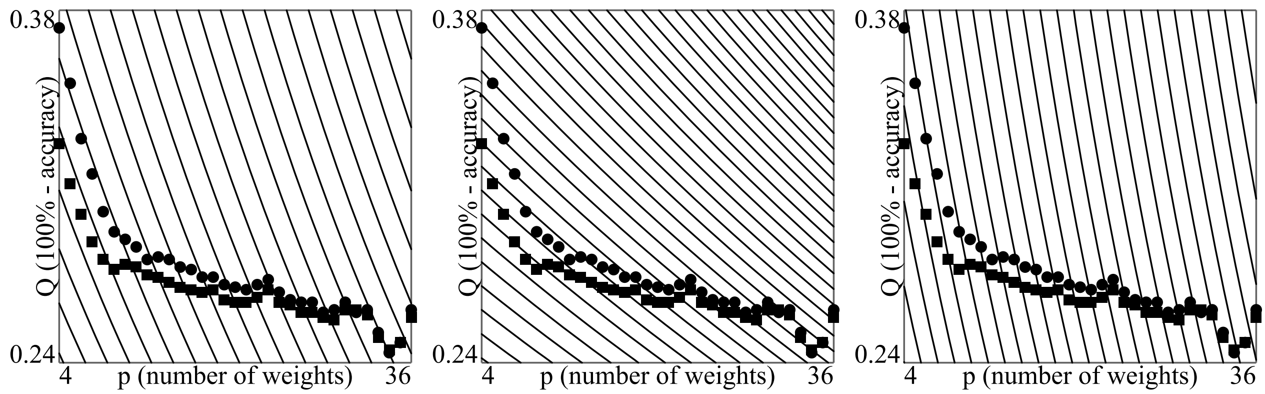


Fig. 3. Histogram of accuracy (ACC) obtained for different users.



(a) Akaike (AIC) (b) FPE (c) Schwarz (S)

Fig. 4. Isolines lines for the learning samples, where circles and squares represent the WO-C1 and WO-C2 systems, respectively.



(a) Akaike (AIC) (b) FPE (c) Schwarz (S)

Fig. 5. Isolines lines for the testing samples, where circles and squares represent the WO-C1 and WO-C2 systems, respectively.

and WR-C2. Figures 4 and 5, as well as Figs. 6, 7 and 8, show the best performance of the WO-C2 system. In addition, Fig. 3 illustrates better accuracy of the WO-C2 system than WO-C1, and the best for 25% rule reduction (or probably more) but less than 50%.

Figure 9 also portrays better performance of the WO-C2 system than that of WO-C1. However, with regard to the RMSE, it shows that about 20 rules give a good result for both learning and testing data samples but more (30 rules) are worse for the learning samples, but best for the testing data samples. It seems obvious that it is possible to obtain better results by reducing the number of rules for learning data, yet the reduced number of rules may not always be sufficient for testing data.

Looking at Fig. 4 and Table 5, we see that the value of parameter p, which corresponds to the number of weights (and hence the number of fuzzy rules), equals 11 for the AIC, 14 for the FPE, and 9 for the Schwarz criterion. Thus, these criteria indicate the number of rules that can be removed with an optimal balance in terms of system accuracy and interpretability.

The obtained fuzzy rules are semantically interpretable. For example, rule number 5 from Table 7: *"IF genre pref. is high and year is medium and keywords pref. is high THEN user rate is high"* can be used as an explanation, as the antecedents of the rule are direct reasons for the recommendation. With this approach, it is also possible to refer the user to the visualization of the particular fuzzy set describing the meaning of the following terms: *"high genre pref."*, *"medium year"* and *"high keywords pref."*. The number of rules is low, and sufficient for good accuracy of the recommender system. Moreover, such a rule base allows explaining the performance of the system with regard to the recommendations (see Tables 7 and 8, with Figs. 10 and 11, which illustrate the fuzzy rules for two users).

5. Conclusions and final remarks

The most important issue in this paper is the explainable fuzzy recommender. The explainability is realized by using semantically interpretable fuzzy IF-THEN rules. Examples of such rules are presented in Tables 7 and 8,

Table 2. Average RMSE for all users in terms of the percentage of reduced rules (RR).

% of	learning samples					testing samples				
RR	WM	WO-C1	WR-C1	WO-C2	WR-C2	WM	WO-C1	WR-C1	WO-C2	WR-C2
0%	0.2199	0.1425	**0.1485**	0.1423	**0.1472**	0.3096	0.2670	0.2688	0.2677	0.2701
4%	0.2302	0.1605	0.1637	0.1513	0.1580	0.3023	**0.2641**	**0.2669**	0.2619	**0.2648**
8%	0.2106	0.1518	0.1582	0.1408	0.1560	0.2980	0.2674	0.2698	0.2658	0.2719
12%	0.1915	0.1395	0.1470	0.1283	0.1465	0.2910	0.2670	0.2706	0.2635	0.2724
16%	0.1908	0.1448	0.1511	0.1316	0.1534	0.2890	0.2688	0.2713	0.2638	0.2734
20%	0.1876	0.1462	0.1535	0.1329	0.1595	0.2876	0.2693	0.2713	0.2662	0.2765
24%	0.1807	**0.1421**	0.1497	0.1282	0.1576	0.2858	0.2669	0.2701	**0.2600**	0.2731
28%	0.1801	0.1429	0.1527	0.1279	0.1638	0.2932	0.2755	0.2791	0.2682	0.2863
32%	0.1789	0.1438	0.1531	0.1290	0.1684	0.2914	0.2748	0.2777	0.2683	0.2906
36%	**0.1775**	0.1435	0.1540	0.1285	0.1716	**0.2843**	0.2698	0.2741	0.2637	0.2848
40%	0.1801	0.1463	0.1580	0.1299	0.1804	0.2896	0.2750	0.2795	0.2692	0.2937
44%	0.1803	0.1465	0.1583	0.1299	0.1794	0.2949	0.2779	0.2823	0.2713	0.2956
48%	0.1797	0.1455	0.1594	**0.1278**	0.1869	0.2917	0.2739	0.2782	0.2683	0.2958
52%	0.1892	0.1519	0.1663	0.1325	0.1910	0.2971	0.2776	0.2826	0.2721	0.2994
56%	0.1956	0.1561	0.1741	0.1353	0.2002	0.3026	0.2793	0.2858	0.2738	0.3039
60%	0.1993	0.1592	0.1830	0.1375	0.2120	0.3048	0.2845	0.2935	0.2747	0.3057
64%	0.2143	0.1708	0.1955	0.1467	0.2240	0.3209	0.2941	0.3042	0.2802	0.3110
68%	0.2292	0.1826	0.2139	0.1570	0.2429	0.3272	0.2968	0.3067	0.2806	0.3189
72%	0.2442	0.1963	0.2321	0.1691	0.2582	0.3291	0.2980	0.3127	0.2813	0.3280
76%	0.2691	0.2179	0.2585	0.1886	0.2902	0.3532	0.3192	0.3364	0.2932	0.3448
80%	0.3025	0.2453	0.2967	0.2168	0.3166	0.3715	0.3305	0.3532	0.2978	0.3538
84%	0.3450	0.2826	0.3513	0.2465	0.3504	0.3892	0.3415	0.3758	0.3024	0.3638
88%	0.3945	0.3289	0.3950	0.2815	0.3776	0.4229	0.3680	0.4089	0.3233	0.3830
92%	0.4406	0.3642	0.4229	0.3045	0.4021	0.4685	0.3943	0.4311	0.3344	0.4035

Table 3. Average ACC for all users in terms of the percentage of reduced rules (RR).

% of	learning samples					testing samples				
RR	WM	WO-C1	WR-C1	WO-C2	WR-C2	WM	WO-C1	WR-C1	WO-C2	WR-C2
0%	77.699	91.091	90.600	91.019	90.607	67.149	**76.721**	**76.469**	76.463	75.903
4%	76.523	89.049	88.595	90.213	89.533	67.095	75.701	75.783	76.087	75.969
8%	79.674	89.838	89.337	91.279	90.114	68.622	75.684	75.424	76.433	75.760
12%	82.977	**91.313**	**90.892**	92.679	**91.370**	70.778	76.154	75.830	76.810	75.908
16%	83.371	90.870	90.444	92.481	90.734	71.440	75.601	75.407	76.254	75.303
20%	83.562	90.565	90.018	92.221	90.126	71.341	75.424	75.047	76.408	75.142
24%	84.641	90.913	90.450	92.629	90.446	71.924	76.007	75.488	**77.361**	**76.089**
28%	85.113	91.018	90.458	**92.871**	90.215	72.212	75.628	75.451	76.992	75.490
32%	**85.344**	90.832	90.217	92.753	89.784	72.393	75.514	75.222	76.691	74.724
36%	85.340	90.644	89.970	92.654	89.499	**72.715**	75.504	75.011	76.682	74.880
40%	84.921	90.408	89.694	92.568	88.833	72.458	75.258	74.652	76.504	74.453
44%	84.882	90.415	89.615	92.602	88.959	71.478	74.953	74.275	76.147	73.934
48%	84.744	90.372	89.512	92.774	88.731	72.012	75.177	74.604	76.137	73.853
52%	83.361	89.594	88.683	92.266	88.184	71.875	75.049	74.671	76.568	74.182
56%	82.466	89.009	87.846	91.903	87.524	70.927	75.035	74.588	76.575	74.082
60%	81.833	88.428	87.055	91.445	86.700	70.369	74.356	73.855	76.099	73.787
64%	80.098	87.106	85.554	90.440	85.515	68.854	73.377	72.607	75.521	72.940
68%	78.280	86.052	84.154	89.407	83.852	68.518	73.879	73.221	75.819	73.069
72%	76.838	84.565	82.632	88.170	82.539	68.159	73.075	72.258	75.639	72.642
76%	74.454	82.450	80.187	86.261	80.290	65.625	71.480	70.541	74.488	71.682
80%	71.005	79.897	77.270	83.562	78.153	64.340	70.986	69.363	74.530	71.608
84%	67.335	75.982	72.789	80.602	74.940	62.052	68.947	67.010	73.520	70.176
88%	62.994	71.200	68.071	77.023	71.904	58.543	66.360	64.570	71.274	67.985
92%	59.315	68.213	65.502	74.907	70.161	54.622	64.517	62.078	70.704	66.941

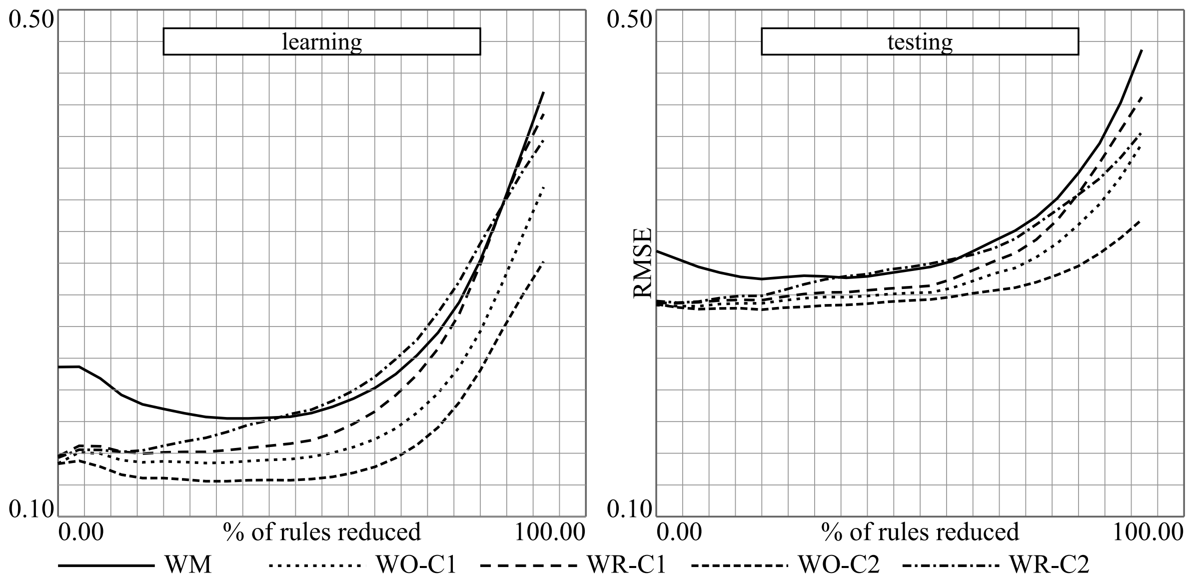


Fig. 6. Comparison of simulation cases in terms of the RMSE and the percentage of reduced rules.

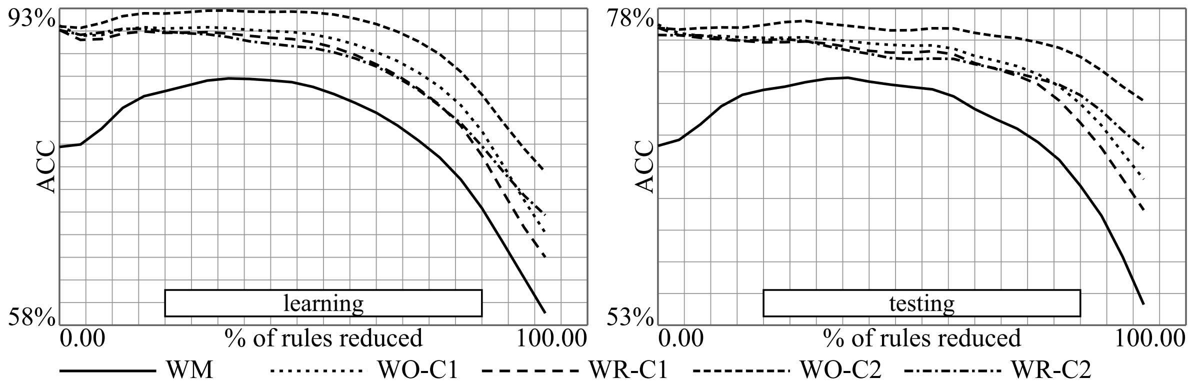


Fig. 7. Comparison of simulation cases in terms of ACC and the percentage of reduced rules.

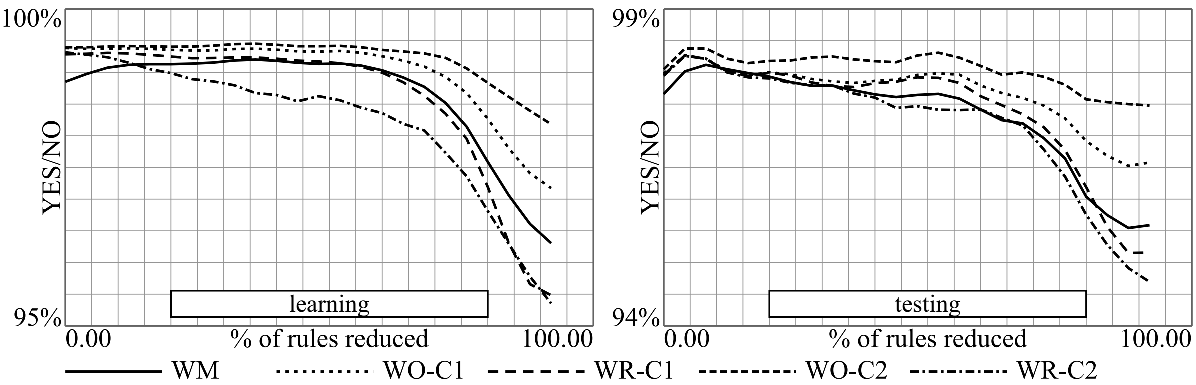


Fig. 8. Comparison of simulation cases in terms of YES/NO and the percentage of reduced rules.

Table 4. Average YES/NO for all users in terms of the percentage of reduced rules (RR).

% of	learning samples					testing samples				
RR	WM	WO-C1	WR-C1	WO-C2	WR-C2	WM	WO-C1	WR-C1	WO-C2	WR-C2
0%	98.852	99.387	99.286	99.397	**99.316**	97.659	97.981	97.961	98.049	97.948
4%	98.988	99.358	99.309	99.370	99.301	**98.242**	**98.541**	**98.552**	**98.705**	**98.649**
8%	99.085	99.368	99.288	99.422	99.224	98.146	98.291	98.293	98.382	98.184
12%	99.155	99.399	**99.329**	99.431	99.193	97.971	97.807	97.789	98.046	97.814
16%	99.121	99.371	99.284	99.399	99.043	98.036	97.986	97.968	98.222	98.005
20%	99.115	99.346	99.217	99.403	98.924	97.962	98.063	98.079	98.170	97.945
24%	99.152	99.350	99.252	99.412	98.998	97.806	97.960	97.940	98.149	97.768
28%	99.147	99.344	99.210	99.414	98.763	97.772	97.884	97.838	98.250	97.794
32%	99.166	99.349	99.224	99.437	98.817	97.792	97.844	97.750	98.310	97.828
36%	**99.256**	**99.419**	99.278	**99.489**	98.809	97.808	97.847	97.769	98.189	97.749
40%	99.188	99.355	99.205	99.430	98.398	97.554	97.811	97.788	98.120	97.424
44%	99.106	99.301	99.167	99.392	98.727	97.599	97.940	97.935	98.224	97.635
48%	99.164	99.345	99.184	99.419	98.511	97.681	97.910	97.839	98.143	97.259
52%	99.133	99.341	99.165	99.428	98.640	97.655	98.036	97.996	98.430	97.504
56%	99.129	99.328	99.088	99.410	98.529	97.647	98.009	97.905	98.364	97.471
60%	99.062	99.264	99.020	99.356	98.163	97.453	97.846	97.595	97.909	97.238
64%	98.907	99.171	98.887	99.302	98.379	97.139	97.525	97.379	98.037	97.544
68%	98.792	99.120	98.624	99.337	98.022	97.152	97.654	97.472	97.948	97.065
72%	98.618	98.980	98.377	99.262	97.851	97.291	97.610	97.165	98.012	96.877
76%	98.150	98.694	98.052	99.091	97.331	96.439	97.159	96.767	97.828	96.400
80%	97.663	98.352	97.428	98.842	96.898	96.199	97.057	96.393	97.573	95.790
84%	96.953	97.760	96.104	98.576	96.211	95.478	96.538	95.407	97.324	95.066
88%	96.563	97.287	95.385	98.422	95.765	95.564	96.459	94.891	97.697	94.976
92%	96.306	97.175	95.482	98.182	95.356	95.588	96.577	95.155	97.478	94.695

as well as Figs. 10 and 11. The fuzzy sets in these rules are semantically interpretable as *very low, low, medium, high, very high*.

The Wang–Mendel method (WM) was applied in order to generate the rules of this type from the dataset. However, these rules can be modified by a learning procedure to achieve better performance of the system; see the work of Rutkowski *et al.* (2018b), where a neuro-fuzzy recommender is presented. It should be emphasized that there is a trade-off between the interpretability and accuracy of the system. The learning procedure changes the fuzzy sets, making them less interpretable. Therefore, in this paper, we do not want to modify the rules in this way. Instead, the weights are assigned to fuzzy IF-THEN rules.

Applying the weights causes modification of decision boundaries in the attribute space, as Ishibuchi and Nakashima (2001) explained in their paper. Thus, the rules are better adjusted to the data, resulting in better performance of the system. It is worth adding that Nauck and Kruse (1998) show that by introducing the weights we obtain the same effect as by modification of fuzzy sets in the rules. As Ishibuchi and Nakashima (2001) illustrated, the rule weights also change the decision boundaries to achieve better performance of a classifier.

It should be emphasized that, when a rule is missing in the Wang–Mendel table, the decision boundaries are also different, as Ishibuchi and Nakashima (2001) showed. Thus, removing a rule from the rule base changes the performance of the system. Therefore, in Algorithm 1 proposed in this paper, the recommender is evaluated after the removal of the least beneficial fuzzy rule.

With regard to Algorithm 1, several variants of the system are considered: WM, WO-C1, WR-C1, WO-C2, WR-C2 (see Tables 1–4). The first one (WM) is a system with the rules generated by the WM method, without the rule weights. Analyzing the results included in these tables, we observe that the systems with the weights perform better than the WM recommender.

Compared the different variants of the system with the weights, we see, as concluded in Section 4.2, that, generally, WO-C2 gives better results. This is a system with optimized weights (and not rounded) in the case when the weights have not been reset. This means that the procedure of reducing the rules always concerns the same system (not changed by resetting the values of the weights).

It also seems obvious that better performance is obtained for the system with optimized weights and not rounded values. However, this is also a compromise between accuracy and interpretability.

The approach proposed in this paper, as mentioned in Section 1, is enhanced by the criteria (AIC, FPE, Szwarz) for evaluating an optimal balance in terms of the recommender's accuracy and interpretability.

In future work, it would be worth applying other

Table 5. AIC, FPE and Schwarz criteria for different numbers of fuzzy rules (p is equal to the number of fuzzy rules and thus the number of weights, Q stands for the RMSE of **learning** samples for the WO method).

p	WO-C1				WO-C2			
	Q	AIC	FPE	Szwarz	Q	AIC	FPE	Szwarz
36	0.209	-7.927	0.337	61.619	0.198	-10.650	0.319	58.896
35	0.204	-11.108	0.325	56.506	0.193	-14.007	0.307	53.607
34	0.199	-14.378	0.312	51.304	0.188	-17.250	0.295	48.432
33	0.193	-17.810	0.300	45.940	0.182	-21.027	0.281	42.723
32	0.186	-21.873	0.284	39.946	0.174	-25.159	0.266	36.659
31	0.181	-25.184	0.273	34.702	0.169	-28.662	0.255	31.225
30	0.176	-28.475	0.263	29.480	0.165	-31.851	0.246	26.104
29	0.172	-31.919	0.252	24.104	0.160	-35.370	0.235	20.653
28	0.169	-34.766	0.244	19.325	0.157	-38.293	0.228	15.798
27	0.166	-37.447	0.238	14.712	0.155	-41.215	0.221	10.944
26	0.164	-40.245	0.231	9.982	0.152	-44.120	0.214	6.108
25	0.160	-43.477	0.222	4.818	0.148	-47.511	0.206	0.784
24	0.156	-46.688	0.214	-0.324	0.144	-51.000	0.197	-4.636
23	0.154	-49.434	0.208	-5.002	0.141	-53.811	0.191	-9.379
22	0.152	-52.208	0.203	-9.707	0.138	-56.835	0.185	-14.335
21	0.148	-55.310	0.196	-14.742	0.135	-59.978	0.178	-19.410
20	0.146	-58.138	0.190	-19.501	0.133	-62.931	0.173	-24.295
19	0.144	-60.752	0.185	-24.047	0.131	-65.756	0.168	-29.051
18	0.142	-63.454	0.180	-28.681	0.128	-68.664	0.163	-33.891
17	0.141	-65.943	0.176	-33.102	0.126	-71.619	0.158	-38.778
16	**0.141**	-67.811	0.174	-36.902	0.125	-73.909	0.155	-43.000
15	0.142	-69.412	0.173	-40.435	**0.125**	-75.875	0.153	-46.898
14	0.144	-70.893	**0.173**	-43.848	0.126	-77.662	**0.151**	-50.617
13	0.146	-71.969	0.174	-46.855	0.127	-79.091	0.151	-53.977
12	0.150	-72.770	0.175	-49.588	0.130	-80.194	0.152	-57.012
11	0.155	**-73.184**	0.179	-51.934	0.133	**-80.806**	0.154	-59.556
10	0.162	-72.892	0.184	-53.574	0.139	-80.701	0.158	-61.383
9	0.172	-71.723	0.194	**-54.336**	0.148	-79.608	0.166	**-62.221**
8	0.187	-69.512	0.208	-54.057	0.161	-77.127	0.179	-61.673
7	0.207	-66.410	0.226	-52.887	0.179	-73.725	0.196	-60.203
6	0.234	-62.152	0.253	-50.561	0.202	-69.562	0.219	-57.971
5	0.272	-56.470	0.290	-46.811	0.235	-63.937	0.250	-54.278
4	0.319	-50.329	0.336	-42.601	0.276	-57.706	0.291	-49.979

rule-based techniques (see, e.g., Bologna and Hayashi, 2017; Liu *et al.*, 2017) to design explainable recommender systems.

Acknowledgment

This research was supported by the Polish National Science Center under the grant 2015/19/B/ST6/03179.

References

Alvarez-Estevez, D., Moret-Bonillo, V. (2018). Revisiting the Wang–Mendel algorithm for fuzzy classification, *Expert Systems* **35**(4): 35:312268.

Bologna, G. and Hayashi, Y. (2017). Characterization of symbolic rules embedded in deep DIMLP networks: A challenge to transparency of deep learning, *Journal of Artificial Intelligence and Soft Computing Research* **7**(4): 265–286.

Cpalka, K. (2017). *Design of Interpretable Fuzzy Systems*, Studies in Computational Intelligence 684, Springer Verlag, Cham.

Harper, F.M. and Konstan, J.A. (2015). The MovieLens datasets: History and context, *ACM Transactions on Interactive Intelligent Systems* **5**(4):19:1–19:19.

Ishibuchi H. and T. Nakashima (2001). Effect of rule weights in fuzzy rule-based classification systems, *IEEE Transactions on Fuzzy Systems* **9**(4): 506–515.

Ishibuchi H. and T. Yamamoto (2005). Rule weight specification in fuzzy rule-based classification systems, *IEEE Transactions on Fuzzy Systems* **13**(4): 428–435.

Jin, Y. (2000). Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement, *IEEE Transactions on Fuzzy Systems* **8**(2): 212–221.

Kuncheva, L. (2000). *Fuzzy Classifier Design*, Studies in Fuzziness and Soft Computing, Vol. 49, Springer Verlag, New York, NY.

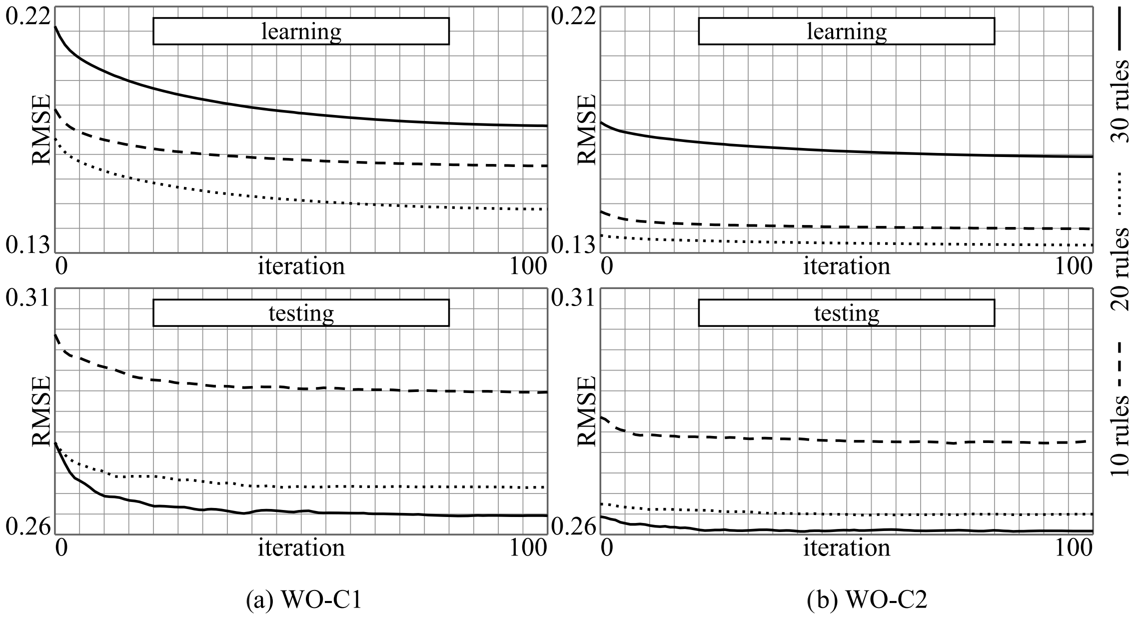


Fig. 9. Optimization of weights for different numbers of fuzzy rules.

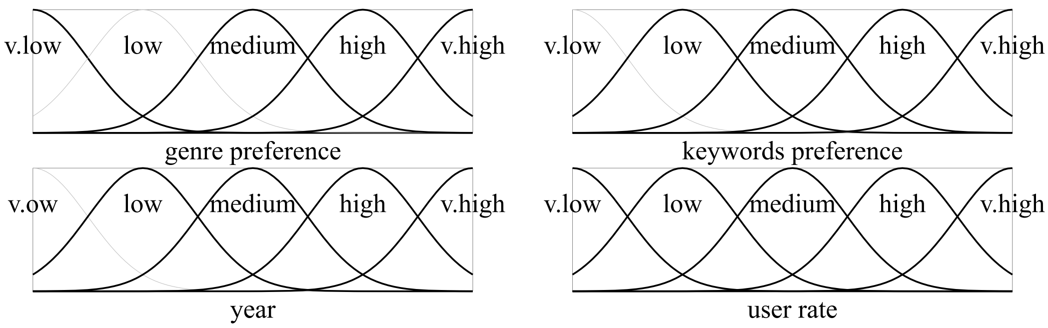


Fig. 10. Fuzzy sets in fuzzy rules from Table 7.

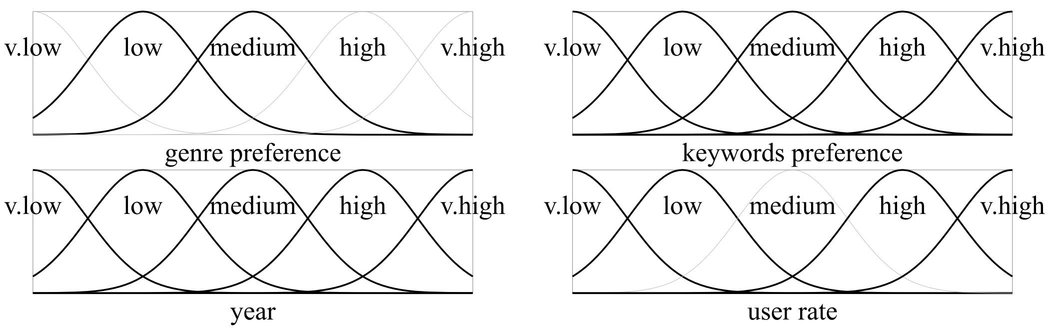


Fig. 11. Fuzzy sets in fuzzy rules from Table 8.

Table 6. AIC, FPE and Schwarz criteria for different numbers of fuzzy rules (p is equal to number of fuzzy rules and thus the number of weights, Q stands for the RMSE of **testing** samples for the WO method).

p	WO-C1				WO-C2			
	Q	AIC	FPE	Szwarz	Q	AIC	FPE	Szwarz
36	0.261	3.529	0.422	73.075	0.258	2.977	0.417	72.523
35	0.248	-1.097	0.395	66.517	0.248	-1.205	0.394	66.409
34	**0.244**	-3.937	0.383	61.745	**0.245**	-3.678	0.385	62.004
33	0.252	-4.352	0.390	59.398	0.250	-4.720	0.387	59.031
32	0.261	-4.448	0.399	57.370	0.259	-4.969	0.395	56.850
31	0.260	-6.671	0.392	53.215	0.261	-6.593	0.393	53.293
30	0.264	-7.955	0.393	50.000	0.261	-8.564	0.388	49.391
29	0.261	-10.490	0.383	45.533	0.257	-11.338	0.377	44.685
28	0.260	-12.640	0.377	41.451	0.258	-13.185	0.373	40.906
27	0.264	-13.922	0.377	38.237	0.260	-14.710	0.371	37.449
26	0.264	-15.828	0.373	34.399	0.260	-16.672	0.367	33.556
25	0.265	-17.701	0.369	30.595	0.263	-18.119	0.366	30.176
24	0.268	-19.075	0.368	27.289	0.264	-19.891	0.362	26.473
23	0.273	-20.241	0.369	24.191	0.269	-20.931	0.364	23.501
22	0.271	-22.547	0.362	19.953	0.266	-23.616	0.355	18.884
21	0.269	-25.057	0.354	15.511	0.264	-25.933	0.348	14.636
20	0.270	-26.846	0.351	11.791	0.264	-27.898	0.344	10.739
19	0.271	-28.534	0.348	8.171	0.265	-29.773	0.340	6.932
18	0.274	-29.968	0.347	4.804	0.269	-30.939	0.341	3.834
17	0.274	-32.094	0.342	0.747	0.268	-33.112	0.335	-0.271
16	0.277	-33.530	0.341	-2.621	0.269	-35.015	0.332	-4.106
15	0.278	-35.233	0.339	-6.255	0.270	-36.817	0.328	-7.840
14	0.281	-36.777	0.337	-9.731	0.272	-38.441	0.327	-11.396
13	0.282	-38.590	0.334	-13.476	0.274	-40.013	0.325	-14.899
12	0.281	-40.805	**0.328**	-17.623	0.275	-41.839	0.322	-18.657
11	0.286	-41.903	0.330	-20.653	0.278	-43.284	0.321	-22.034
10	0.289	-43.327	0.329	-24.009	0.279	-45.068	0.318	-25.750
9	0.292	-44.838	0.328	-27.452	0.277	-47.388	0.312	-30.001
8	0.300	**-45.332**	0.334	-29.878	0.281	-48.673	**0.312**	-33.218
7	0.315	-44.962	0.345	-31.440	0.288	-49.446	0.316	-35.923
6	0.329	-44.724	0.356	-33.133	0.299	**-49.577**	0.323	-37.986
5	0.351	-43.439	0.374	-33.780	0.311	-49.639	0.332	-39.980
4	0.373	-42.362	0.393	**-34.635**	0.327	-49.043	0.344	**-41.316**

Liu, H., Gegov, A. and Cocea, M. (2017). Rule based networks: An efficient and interpretable representation of computational models, *Journal of Artificial Intelligence and Soft Computing Research* **7**(2): 111–123.

Lops, P., Gemmis, M. and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends, *in* F. Ricci *et al.* (Eds), *Recommender Systems Handbook*, Springer, New York, NY, pp.73–105.

Nauck, D. and R. Kruse (1998). How the learning of rule weights affects the interpretability of fuzzy systems, *IEEE International Conference on Fuzzy Systems 1998 (FUZZ-IEEE'98), Ancorage, AK, USA*, pp.1235–1240.

Portugal, I., Paulo S.C. Alencar and Cowan, D.D. (2018). The use of machine learning algorithms in recommender systems: A systematic review, *Expert System Applications* **97**: 205–227.

Prasad, M., Liu, Y.-T., Li, D.-L., Lin, C.-T., Shah, R.R. and Kaiwartya, O.P. (2017). A new mechanism for data visualization with TSK-type preprocessed collaborative fuzzy rule based system, *Journal of Artificial Intelligence and Soft Computing Research* **7**(1): 33–46.

Riid, A. and Preden, J.-S. (2017). Design of fuzzy rule-based classifiers through granulation and consolidation, *Journal of Artificial Intelligence and Soft Computing Research* **7**(2): 137–147.

Rutkowska, D. (2002). *Neuro-Fuzzy Architectures and Hybrid Learning*, Studies in Fuzziness and Soft Computing, Springer Verlag, New York, NY.

Rutkowski, L. (2004). *Flexible Neuro-Fuzzy Systems: Structures, Learning and Performance Evaluation*, Kluwer Academic Publishers, Boston, MA/Dordrecht/London.

Rutkowski, L. (2008). *Computational Intelligence: Methods and Techniques*, Springer, Berlin.

Rutkowski, T., Romanowski, J., Woldan, P., Staszewski, P. and Nielek, R. (2018). Towards interpretability of the movie recommender based on a neuro-fuzzy approach, *17th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2018, Zakopane, Poland,* pp. 752–762.

Table 7. Examples of fuzzy rules for the user with id = 7 and optimal reduction of fuzzy rules considering the obtained accuracy.

k	IF			THEN	rule weight		
	genre pref. is	year is	keywords pref. is	user rate is	WM	WO-C2	WR-C2
1	high	v.high	medium	medium	0.500	0.543	0.500
2	high	high	v.high	v.high	0.500	0.076	0.100
3	v.high	v.high	v.high	v.high	0.500	0.917	0.900
4	high	medium	v.high	v.high	0.500	0.604	0.600
5	high	medium	high	high	0.500	0.228	0.200
6	high	low	v.high	v.high	0.500	0.620	0.600
7	medium	v.high	low	v.low	0.500	0.591	0.600
8	v.high	v.high	high	high	0.500	0.465	0.500
9	high	v.high	low	v.low	0.500	0.580	0.600
10	high	low	medium	low	0.500	0.443	0.400
11	high	medium	medium	medium	0.500	0.296	0.300
12	high	high	low	v.low	0.500	0.149	0.100
13	high	low	high	high	0.500	0.516	0.500
14	medium	v.high	v.high	v.high	0.500	1.000	1.000
15	v.low	v.high	low	low	0.500	0.642	0.600
16	medium	v.high	medium	medium	0.500	0.345	0.300
system errors			RMSE	learning	0.247	0.210	0.212
				testing	0.344	0.330	0.328
			ACC	learning	75.1	82.0	80.1
				testing	57.1	65.8	61.9
			YES/NO	learning	99.4	99.4	99.4
				testing	100.0	100.0	100.0

Table 8. Examples of fuzzy rules for the user with id = 2 and optimal reduction of fuzzy rules considering the AIC, FPE and Schwarz criteria.

k	IF			THEN	rule weight		
	genre pref. is	year is	keywords pref. is	user rate is	WM	WO-C2	WR-C2
1	medium	medium	v.high	v.high	0.500	0.310	0.300
2	medium	v.high	medium	low	0.500	0.156	0.200
3	low	v.high	v.low	v.low	0.500	1.000	1.000
4	low	v.high	low	v.low	0.500	0.073	0.100
5	medium	v.high	v.high	v.high	0.500	1.000	1.000
6	medium	high	v.high	v.high	0.500	1.000	1.000
7	low	high	medium	low	0.500	0.108	0.100
8	low	v.low	v.high	v.high	0.500	0.450	0.500
9	medium	v.high	high	high	0.500	0.266	0.300
10	low	medium	high	high	0.500	0.050	0.100
11	low	v.high	medium	high	0.500	0.273	0.300
12	medium	low	low	low	0.500	0.056	0.100
system errors			RMSE	learning	0.171	0.113	0.141
				testing	0.188	0.173	0.159
			ACC	learning	86.7	88.9	88.9
				testing	100.0	100.0	100.0
			YES/NO	learning	97.7	100.0	100.0
				testing	100.0	100.0	100.0

Rutkowski, T., Romanowski, J., Woldan, P., Staszewski, P., Nielek, R. and Rutkowski, L. (2018). A content-based recommendation system using neuro-fuzzy approach, *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2018), Piscataway, NJ, USA*, pp. 1–8, DOI:10.1109/FUZZ-IEEE.2018.8491543.

Simiński, K. (2010). Rule weights in a neuro-fuzzy system with a hierarchical domain partition, *International Journal of Applied Mathematics and Computer Science* **20**(2): 337–347, DOI: 10.2478/v10006-010-0025-3.

Söderström, T. and Stoica, P. (1989). *System Identification*, Prentice Hall International, Upper Saddle River, NJ.

Wang, L.-X. and Mendel J.M. (1992). Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **22**(6): 1414–1427.

Wei, J., He, J., Chen, K., Zhou, Y. and Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items, *Expert Systems with Applications* **69**: 29–39.

Zhang, S., Yao L., Sun A. and Y. Tay (2018). Deep learning based recommender system: A survey and new perspectives, *ACM Computing Surveys* **52**(1), Article No. 5.

Tomasz Rutkowski received his MSc degree in computer engineering from the Polish-Japanese Academy of Information Technology and an MBA degree from Kozminski University in Warsaw, Poland. He is currently working on his PhD in computer science and management. His research interests concern AI explainability, transparency and the human-in-the-loop approach. He is also a co-CEO at Senfino, an explainable AI company based in New York City.

Krystian Łapa is an associate professor in the Institute of Computational Intelligence at the Częstochowa University of Technology, where he received his PhD degree in 2015. His research interests include fuzzy systems, interpretability, population-based algorithms, nonlinear modelling and, in the broad sense, computational intelligence.

Radosław Nielek is an associate professor at the Polish-Japanese Academy of Information Technology, where he has received his PhD degree. His research interests include co-design of ICTs for the elderly focused on improving life quality, web credibility and social informatics. He has published over 70 papers on web mining, social informatics and social simulation.

Received: 5 March 2019
Revised: 13 April 2019
Accepted: 29 April 2019