On Explainable Recommender Systems Based on Fuzzy Rule Generation Techniques

Tomasz Rutkowski[1,3], Krystian Łapa[2(✉)], Robert Nowicki[2], Radosław Nielek[3], and Konrad Grzanek[4,5]

[1] Senfino, 1412 Broadway 21st floor, New York City, NY 10018, USA
XAI@senfino.com
[2] Institute of Computational Intelligence, Czestochowa University of Technology, Częstochowa, Poland
{krystian.lapa,robert.nowicki}@iisi.pcz.pl
[3] Polish-Japanese Academy of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland
nielek@pjatk.edu.pl
[4] Information Technology Institute, University of Social Sciences, 90-113 Łódź, Poland
kgrzanek@san.edu.pl
[5] Clark University, Worcester, MA 01610, USA

Abstract. This paper presents an application of the Zero-Order Takagi-Sugeno-Kang method to explainable recommender systems. The method is based on the Wang-Mendel and the Nozaki-Ishibuchi-Tanaka techniques for the generation of fuzzy rules, and it is best suited to predict users' ratings. The model can be optimized using the Grey Wolf Optimizer without affecting the interpretability. The performance of the methods has been shown using the MovieLens 10M dataset.

Keywords: Explainable AI · Recommender systems · Wang-Mendel and Nozaki-Ishibuchi-Tanaka methods · Grey Wolf Optimizer · Interpretability

1 Introduction

In recent years, there have been many attempts to add explainability and transparency to machine learning models [1,5]. However, not so many papers have been published regarding explainability in recommender systems.

A recommender (or recommendation) system is any system that offers items in a personalized way to a specific user or guides him to the product best suited to his profile. There are three general types of recommender systems - collaborative filtering, content-based and hybrid approach [2,13,19].

Collaborative filtering systems recommend items by identifying other users with similar taste disregarding attributes of considered objects. Content-based

L. Rutkowski et al. (Eds.): ICAISC 2019, LNAI 11508, pp. 358–372, 2019.
https://doi.org/10.1007/978-3-030-20912-4_34

recommender systems analyze the attributes of considered objects, therefore they do not need any information about other users' preferences. Hybrid approach combines many different recommendation methods. For the purpose of this paper, we focus on content-based recommender systems. Collaborative filtering uses similarity to other people and has two downsides. One is that it requires many ratings from different users, otherwise, the algorithm would face so-called cold-start problem. The other problem is that from the explainability perspective it is not enough to say that a recommender system recommends something because other people like it too. With the content-based approach, the challenge is to analyze items that user rated before, prepare the profile of preferences and recommend new items based on this knowledge. The goal of our research is to provide not only recommendations but also explanations. We believe that to achieve truly explainable system it has to be interpretable and transparent [9].

As presented in previous work [17], it is possible to use rule-based [10,14] recommender systems to achieve explainability without losing too much accuracy. Such a system is usually based on fuzzy logic [3,16]. As rules are by definition interpretable by humans, they can be used to generate explanations. However, it is not trivial to generate rules from examples, reduce them and optimize [15]. In this paper, we propose a new method derived from Wang-Mendel and Nozaki-Ishibuchi-Tanaka methods, combined with the Zero-Order Takagi-Sugeno-Kang fuzzy system. It allows to deal with singleton outputs, which can be further optimized using Wolf Grey Optimizer. All experiments use the MovieLens 10M dataset.

Structure of the paper is as follows: Sect. 2 contains a description of rule generation methods, Sect. 3 presents a proposed approach, Sect. 4 shows the simulation results and the conclusions are drawn in Sect. 5.

2 Rule Generation Methods

In this section, the basic methods of generating fuzzy rules from data have been presented.

2.1 Wang-Mendel Method

In the Wang-Mendel method fuzzy rules are created as follows:

$$\begin{aligned} R_j : \text{IF } x_1 \text{ IS } A_{1,inp_{j,1}} \text{ AND ... AND} \\ x_n \text{ IS } A_{n,inp_{j,n}} \text{ THEN } y \text{ IS } B_{out_j}, \end{aligned} \tag{1}$$

where R_j stands for j-th fuzzy rule, j is fuzzy rule index ($j = 1, ..., M$), M is the number of fuzzy rules (initial number of fuzzy rules is equal to the number of data set samples), x_i stands for fuzzy system inputs ($i = 1, ..., n$), n is the number of fuzzy system inputs, y is fuzzy system output, $A_{i,l}$ stand for input fuzzy sets where i indicates the index of system input and l indicates the index of fuzzy set in i-th input ($l = 1, ..., m$), m is the number of fuzzy partitions,

B_l stands for output fuzzy sets, and indexes $inp_{j,i}$ and out_j indicates the fuzzy sets from corresponding inputs and outputs, which are selected as follows:

$$\mu_{A_{i,inp_{j,i}}}\left(\bar{x}_{j,i}\right) = \max_{l=1,\ldots,m}\left\{\mu_{A_{i,l}}\left(\bar{x}_{j,i}\right)\right\}, \tag{2}$$

$$\mu_{B_{out_j}}\left(\bar{y}_j\right) = \max_{l=1,\ldots,m}\left\{\mu_{B_l}\left(\bar{y}_j\right)\right\}, \tag{3}$$

where $\bar{x}_{j,i}$ stands for data set input values, $\bar{y}_j$ stand for data set output values, $\mu_A(\cdot)$ and $\mu_B(\cdot)$ are membership functions of corresponding fuzzy sets.

The initial fuzzy rule base in form of (1) is subject to reduction. For this process for each j-th fuzzy rule an importance degree is calculated:

$$\lambda_j\left(\bar{\mathbf{x}}_j, \bar{y}_j\right) = T\left\{\mu_{B_{out_j}}\left(\bar{y}_j\right), \mu_{A_{1,inp_{j,1}}}\left(\bar{x}_{j,1}\right), \ldots, \mu_{A_{n,inp_{j,n}}}\left(\bar{x}_{j,n}\right)\right\}, \tag{4}$$

where T stands for algebraic t-norm operator. The final rule base is obtained by reducing conflicting rules (with equal values of $inp_{j,i}$ and different values of out_j) and identical rules (with equal values of $inp_{j,i}$ and out_j). During the reduction, only rules with the highest value of (4) are kept and thus the final rule base contain K fuzzy rules ($K \leq M$).

2.2 Nozaki-Ishibuchi-Tanaka Method

In the Nozaki-Ishibuchi-Tanaka method fuzzy rules also have form of (1). However, instead of calculating out_j indexes by Eq. (3) for each fuzzy rule singletons s_j are calculated as follows:

$$s_j = \frac{\sum_{j=1}^{M} \tau_j\left(\bar{\mathbf{x}}_j\right)^{\alpha} \cdot y_j}{\sum_{j=1}^{M} \tau_j\left(\bar{\mathbf{x}}_j\right)^{\alpha}}, \tag{5}$$

where $\alpha > 0$ is a parameter of the method (in this its value its selected as 1) and the $\tau_j\left(\bar{\mathbf{x}}_j\right)$ is the activation level of j-th fuzzy rule calculated as follows:

$$\tau_j\left(\bar{\mathbf{x}}_j\right) = T\left\{\mu_{A_{1,inp_{j,1}}}\left(\bar{x}_{j,1}\right), \ldots, \mu_{A_{n,inp_{j,n}}}\left(\bar{x}_{j,n}\right)\right\}. \tag{6}$$

The final rule base is obtained by reducing fuzzy rules with identical inputs (with equal values of $inp_{j,i}$ and s_k) after which K fuzzy rules remain ($K \leq M$). Finally, the out_k indexes are selected as follows:

$$\mu_{B_{out_k}}\left(s_k\right) = \max_{l=1,\ldots,m}\left\{\mu_{B_l}\left(s_k\right)\right\}. \tag{7}$$

3 Proposed Method Description

The proposed method is based on the fuzzy rule generation methods with equally (uniform) spaced partitions. Such an approach allows to obtain clear and readable fuzzy sets (see e.g. Fig. 1). The two methods are considered as a base methods: Wang-Mendel (WM) and Nozaki-Ishibuchi-Tanaka (NIT). On the basis of WM and NIT methods, Mamdani fuzzy systems can be efficiently created.

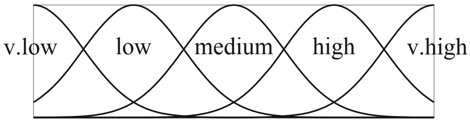


Fig. 1. Example of equally spaced partitions

3.1 Zero-Order Takagi-Sugeno-Kang Fuzzy System

In this paper, we propose to use fuzzy system resulting from WM and NIT methods in the form of Zero-Order Takagi-Sugeno-Kang (ZO-TSK) fuzzy system. In such a system the output is calculated as follows:

$$y\left(\bar{\mathbf{x}}\right) = \frac{\sum\limits_{k=1}^{K} v_k \cdot \tau_k\left(\bar{\mathbf{x}}\right)}{\sum\limits_{k=1}^{K} \tau_k\left(\bar{\mathbf{x}}\right)}, \tag{8}$$

v_k are singleton positions ($k = 1, ..., K$), K stands for the number of fuzzy rules, $\mu_k(\cdot)$ stands for rule activation level calculated as in Eq. (6). The singletons are numeric values as opposed to output fuzzy sets. Such an approach changes the form of fuzzy rules to the following:

$$\begin{array}{c} R_k : \text{IF } x_1 \text{ IS } A_{1,inp_{k,1}} \text{ AND ... AND} \\ x_n \text{ IS } A_{n,inp_{k,n}} \text{ THEN } y = v_k \end{array}. \tag{9}$$

The use of numerical values in a fuzzy rule changes the way they can be interpreted. However, the authors think that this is beneficial for a recommendation systems in which the output value of the system is usually a numerical value (e.g. movie rate).

In this paper, three ways to create a ZO-TSK system were used: WM-T (where the singleton values are set as centers of output fuzzy sets of corresponding rules from WM Mamdani type fuzzy system), NIT-T (where the singleton values are set as centers of output fuzzy sets of corresponding rules from NIT Mamdani type fuzzy system) and NIT-S (where the values of singletons as set directly to s_k values from NIT method) - see Fig. 2. Without any further optimization, the WM-T system will behave identically to WM, and the NIT-T system to NIT. In this paper, further modifications are considered, which is why the names of these systems are distinguished.

3.2 System Optimization

The core of this paper is the assumption that singleton values can be optimized without loss of the system interpretation. In order to get it, the ranges of singletons for each fuzzy rule are limited individually. The limitation results from

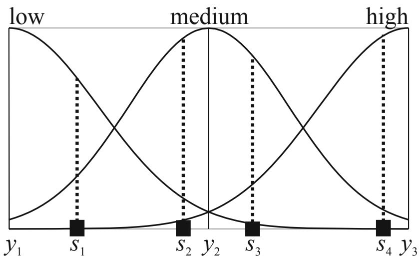


Fig. 2. Example of $\mathbf{s}$ values generated by NIT method that are used as singletons values in NIT-S case and $\mathbf{y}$ values (centers of output fuzzy sets) used as singletons in NIT-T case.

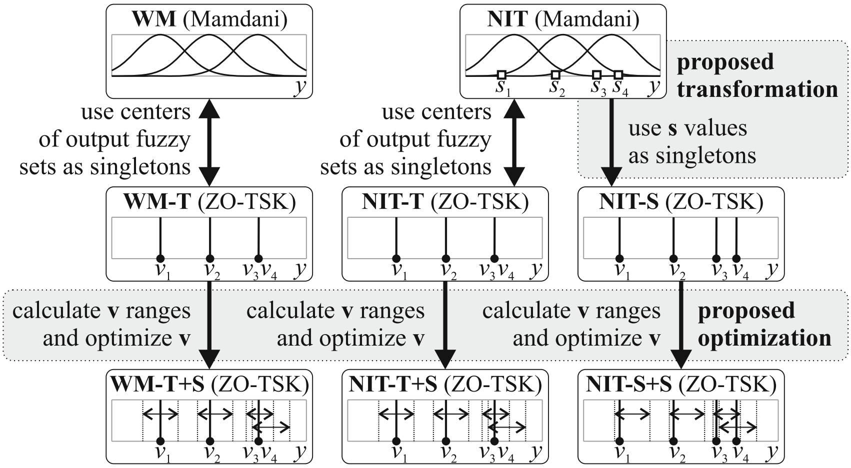


Fig. 3. Idea of the method proposed in this paper. It is worth noting that from one output fuzzy set a different number of singletons can be created (equal to the number of rules that are connected to a given set), double-sided arrows indicate the calculated ranges $< v_{k,min}; v_{k,max} >$ in which singletons can be optimized. The systems WM-T and NIT-T are identical without further modifications of systems WM and NIT.

data set outputs of data set samples for which the highest activation level for a specified fuzzy rule was achieved. The limitations are calculated as follows:

$$v_{k,min} = \min_{\left\{j\in\{1,\ldots,M\}:\mu_k(\bar{\mathbf{x}}_z)=\max\limits_{l=1,\ldots,K}\left\{\mu_{R_l}(\bar{\mathbf{x}}_j)\right\}\right\}} \{\bar{y}_j\}, \tag{10}$$

$$v_{k,max} = \max_{\left\{j\in\{1,\ldots,M\}:\mu_k(\bar{\mathbf{x}}_j)=\max\limits_{l=1,\ldots,K}\left\{\mu_{R_l}(\bar{\mathbf{x}}_j)\right\}\right\}} \{\bar{y}_j\}. \tag{11}$$

With this assumption, the ranges of values for each singleton are different ($v_{k,min}$ and $v_{k,max}$ are calculated for each k-th fuzzy rule). In addition, narrowing the ranges to values resulting from the data should not cause loss of trust in the system's prediction.

The optimization of singleton values can be performed by any optimization algorithm (see e.g. [4,18]). In this paper, a GWO is used to optimize WM-T, NIT-T, and NIT-S systems and the optimized systems will be referred accordingly as WM-T+S, NIT-T+S and NIT-S+S. The Grey Wolf Optimizer (GWO) is a meta-heuristics inspired by leadership hierarchy and hunting procedure of grey wolves in nature [11]. It has been successfully applied for solving various optimization problems (see e.g. [8,12,20]). In this algorithm, the three best individuals (wolves) are called in sequence alpha (α), beta (β) and delta (δ). The rest of the wolves are called omega (ω).

The modification (called hunting) of individuals parameters is performed only for ω wolves. It is assumed that α, β and δ wolves have better knowledge about the potential location of optimum (called prey). The hunting is performed as follows:

$$\boldsymbol{D}_{\alpha/\beta/\delta} = \left|\boldsymbol{C}_{\alpha/\beta/\delta} \cdot \boldsymbol{X}_{\alpha/\beta/\delta} - \boldsymbol{X}\right|, \tag{12}$$

$$\boldsymbol{X}_{1/2/3} = \boldsymbol{X}_{\alpha/\beta/\delta} - \boldsymbol{A}_{1/2/3} \cdot \left(\boldsymbol{D}_{\alpha/\beta/\delta}\right), \tag{13}$$

$$\boldsymbol{X}\left(t+1\right) = \frac{1}{3}\left(\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3\right), \tag{14}$$

where $\boldsymbol{X}$ are individual parameters, $\boldsymbol{X}_{\alpha/\beta/\delta}$ are respectively parameters of best wolves and $\boldsymbol{C}_{\alpha/\beta/\delta}$ and $\boldsymbol{A}_{\alpha/\beta/\delta}$ are calculated as follows:

$$\boldsymbol{A} = 2 \cdot a \cdot \boldsymbol{r}_1 - a, \tag{15}$$

$$\boldsymbol{C} = 2 \cdot \boldsymbol{r}_2, \tag{16}$$

where $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are random vectors in [0, 1] and component a linearly decreases from 2 to 0 over the course of algorithm iterations. Such a procedure allows for a smooth transition from exploration to exploitation and does not require setting any real value parameters of the actual algorithm [6].

It is also worth adding that the other elements of the fuzzy system (e.g. fuzzy sets) are not subject of optimization, which makes possible keeping the entire fuzzy system and fuzzy sets in a clear form. The idea of the method proposed in this paper is presented on Fig. 3. The proposed approach is new in the literature.

3.3 Data Set Preparation

The fuzzy systems used in this paper process the numeric parameters, however, the recommendation systems often include inputs with nominal values. Moreover, multiple nominal values can be assigned to single item attribute. To process such data, aggregation of nominal values was proposed in this paper. For each user and each attribute of rated by user items a list of unique nominal values is created. Then, the user preferences of each value are calculated as an average rate of the item that contains a particular value. Then, fuzzy system inputs for nominal attributes are calculated as an average preference of all values that occurs in a given attribute of an item (see Fig. 4).

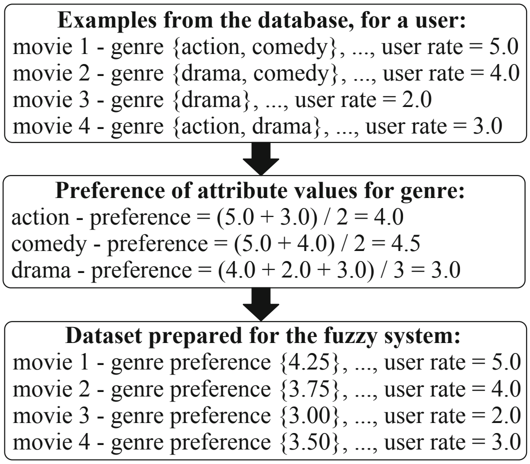


Fig. 4. Example of a dataset preparation for a user

It is worth to mention that the user's rate of an item may result from preferences of various attributes, and therefore the values of specific system inputs will not always be consistent with the rate of an item as is shown in Fig. 4. Moreover, the proposed aggregation may result in the loss of some information (the user may provide ratings based on attributes not included in the database and also it is not possible to accurately detect the preferences of specific combinations of values). Nevertheless, the proposed approach allows for the creation of fuzzy rules detecting dependencies between preferences of attributes and also provides very clear fuzzy rules (due to, among others, the low number of system inputs created).

3.4 Summary of the Proposed Method

The proposed method: (a) can be based on fuzzy rules generation methods with equally (uniform) spaced partitions, which allows obtaining clear and readable fuzzy sets, (b) it is based on transforming fuzzy systems into Zero-Order Takagi-Sugeno-Kang type, that are simple in interpretation, (c) it allows keeping trust of system prediction optimizing only singleton values in limited ranges of values calculated for each rule, (d) it uses recent and almost parameter-less optimization algorithm, which allows getting good results, and (e) a data set preparation method is used to create numeric inputs for the fuzzy system from nominal values.

4 Simulations

In the simulations the following system were tested: WM-T, NIT-T, NIT-S, WM-T+S, NIT-T+S, NIT-S+S. Moreover, a different numbers of fuzzy partitions

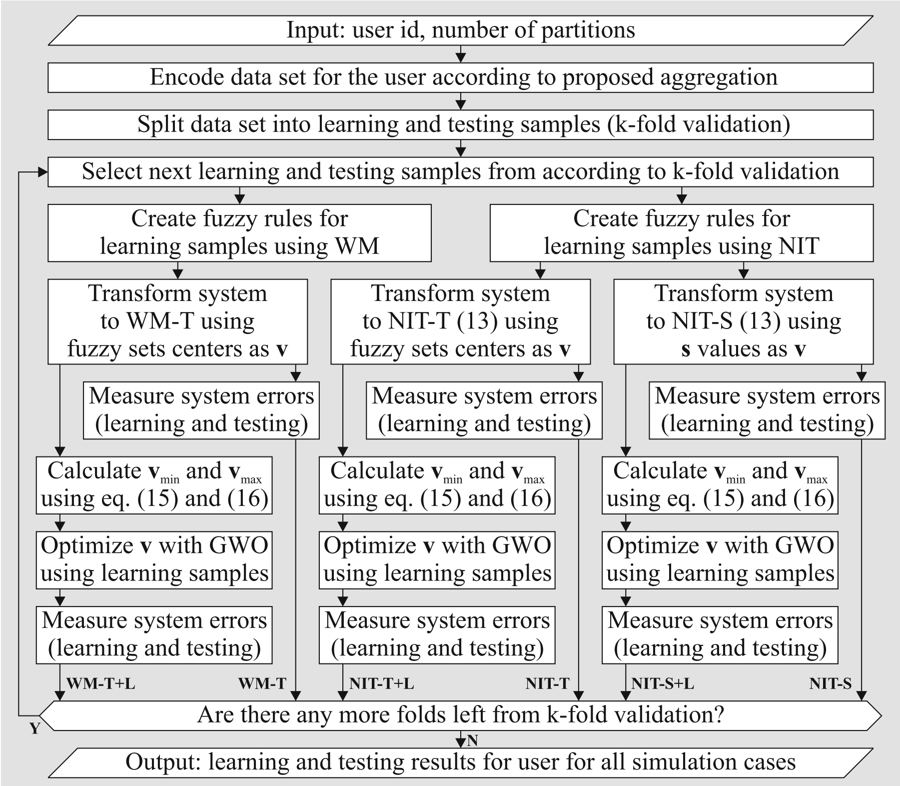


Fig. 5. Diagram showing the proposed methods and process of performed simulations for a single user.

were compared $m = \{3, 5\}$. A larger number of partitions would force to use more linguistic variables (e.g. very very low) and thus significantly reduce the transparency of fuzzy rules. In spite of this, only exemplary additional tests were made to show the possibilities of the proposed approach with larger number of partitions. The process of performed simulations for single user is presented in Fig. 5.

The following parameters was set for all systems: triangular norms = algebraic, fuzzy sets = Gaussian type. The following parameters of GWO were set: population size = 16, number of iterations = 100.

4.1 Data Set

For the simulations, a MovieLens 10m database is used [7], and three inputs are prepared: genre preference (multiple nominal values), year (numeric values), keywords preference (multiple nominal values). Moreover, a data sets were prepared for first 100 users that rated more than 30 movies from the database.

4.2 Results Verification

To verify the results for each simulation case 10-fold cross-validation was used. This process applied to use different data set samples not only in learning and testing phases but also using 90% of data samples (learning samples) for creating fuzzy rule base using WM and NIT methods. Moreover, a different error measures were used: *rmse* (this measure was used to optimize the system), *accuracy* (the predicted value was round to user rate and thus 10 different classes were obtained), *yesno* (the output value was set to class 1 if prediction was lower than average rate and to class 2 otherwise - such an approach is used e.g. in [17]). For both *accuracy* and *yesno* typical classification accuracy were measured.

4.3 Simulation Results

The simulation results in details are presented in Table 1. An overview of *rmse* depending on the amount of rated movies by users is presented in Table 2. The fuzzy rules in simulations were created basing on data set samples, thus the number of created fuzzy rules differ only for different users (for each user a different data sets are created - see Sect. 3.3). The dependencies between the number of fuzzy rules created for different groups of users are shown in average in Fig. 6 and in details in Fig. 7. The optimization process is presented in Fig. 8. Examples of the fuzzy system that allow to obtain best accuracy (NIT-S+S) are shown in Table 3. Exemplary results of using a higher number of fuzzy partitions are shown in Table 4.

Table 1. Simulation results in details, average stands for average results obtained for all users, st. dev stands for standard deviation, lrn stands for learning samples, tst stands for testing samples

m	**system**	*rmse*				*accuracy*				*yesno*			
		average		st. dev.		average		st. dev.		average		st. dev.	
		lrn	tst	lrn	tst	lrn	tst	lrn	tst	lrn	tst	lrn	tst
3	WM−T	0.368	0.424	0.057	0.182	55.38	50.89	7.03	10.76	94.78	93.12	3.31	8.49
	NIT−T	0.317	0.376	0.028	0.160	59.29	54.53	4.95	10.45	96.19	94.61	2.37	7.10
	NIT−S	0.350	0.438	0.018	0.167	53.92	45.74	4.66	10.21	96.54	93.23	1.39	8.25
	WM−T+S	0.275	0.354	0.022	0.149	64.27	56.62	5.61	10.12	98.94	96.87	0.63	4.53
	NIT−T+S	0.271	**0.351**	0.021	0.148	64.79	**56.96**	5.27	**10.02**	99.03	**96.97**	0.57	**4.48**
	NIT−S+S	**0.263**	0.357	**0.017**	**0.148**	**66.85**	56.52	**4.55**	10.31	**99.06**	96.25	**0.49**	5.42
5	WM−T	0.206	0.327	0.016	0.151	79.53	65.23	3.96	9.55	98.68	96.93	0.60	4.54
	NIT−T	0.202	0.323	0.014	0.153	80.55	66.22	3.55	9.56	98.91	97.11	0.47	4.45
	NIT−S	0.174	0.321	0.011	0.157	84.92	66.14	2.40	9.38	99.42	96.94	0.26	4.60
	WM−T+S	0.144	0.289	0.011	0.149	89.19	**71.30**	2.08	**9.14**	99.74	97.96	0.17	3.22
	NIT−T+S	0.142	0.288	0.010	**0.149**	89.46	71.28	2.14	9.21	99.77	**97.99**	0.17	**3.16**
	NIT−S+S	**0.132**	**0.288**	**0.010**	0.150	**90.68**	70.85	**1.73**	9.18	**99.79**	97.97	**0.10**	3.19

Table 2. *rmse* results in details with the division of users by the number of rated movies (o)

m	**system**	learning samples					testing samples				
		$o < 50$	$50 \leq o < 100$	$100 \leq o < 200$	$200 \leq o < 400$	$o \geq 400$	$o < 50$	$50 \leq o < 100$	$100 \leq o < 200$	$200 \leq o < 400$	$o \geq 400$
3	WM-T	0.352	0.350	0.377	0.418	0.406	0.463	0.409	0.415	0.435	0.419
	NIT-T	0.327	0.296	0.323	0.346	0.353	0.429	0.365	0.363	0.360	0.368
	NIT-S	0.289	0.328	0.377	0.425	0.480	0.442	0.435	0.427	0.452	0.504
	WM-T+S	0.266	0.253	0.294	0.303	0.330	0.402	0.347	0.344	0.323	0.352
	NIT-T+S	0.260	0.250	0.290	0.300	0.324	0.398	0.345	0.340	0.320	0.349
	NIT-S+S	0.230	0.240	0.289	0.310	0.345	0.394	0.354	0.344	0.337	0.370
5	WM-T	0.159	0.184	0.237	0.258	0.302	0.349	0.327	0.322	0.302	0.330
	NIT-T	0.157	0.178	0.233	0.258	0.297	0.347	0.323	0.315	0.303	0.323
	NIT-S	0.099	0.156	0.213	0.234	0.289	0.339	0.327	0.313	0.291	0.326
	WM-T+S	0.092	0.126	0.178	0.193	0.225	0.321	0.294	0.279	**0.251**	0.267
	NIT-T+S	0.092	0.124	0.175	0.188	0.219	0.322	**0.292**	0.277	0.251	**0.260**
	NIT-S+S	**0.070**	**0.114**	**0.168**	**0.186**	**0.222**	**0.317**	0.295	**0.277**	0.252	0.266

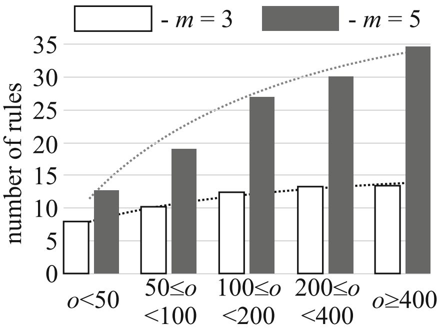


Fig. 6. A number of fuzzy rules created with the division of users by the number of rated movies (o)

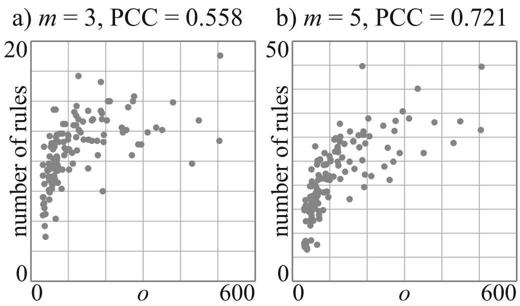


Fig. 7. Correlation between a number of rated movies (o) and a number of fuzzy rules created on prepared data sets, PCC stands for the Pearson Correlation Coefficient

4.4 Simulation Conclusions

The optimization of singleton parameters in specified ranges allow obtaining another increase in system accuracy (see WM-T+S, NIT-T+S, and NIT-S+S systems in Table 1).

The best *rmse*, *accuracy* and *yesno* were obtained for NIT-S+S system, where initial singleton positions resulted from the values s_k calculated by NIT method (see NIT-S+S system in Table 1).

The proposed solution allowed to achieve very high *yesno* recommendation accuracy (at level of 98% for testing data samples) and high classification accuracy of predicting exact user rate of the movie (72%) - see Table 1. In the latter, the increase in accuracy comparing to standard WM-T and NIT-T methods is higher than 5% (see Table 1).

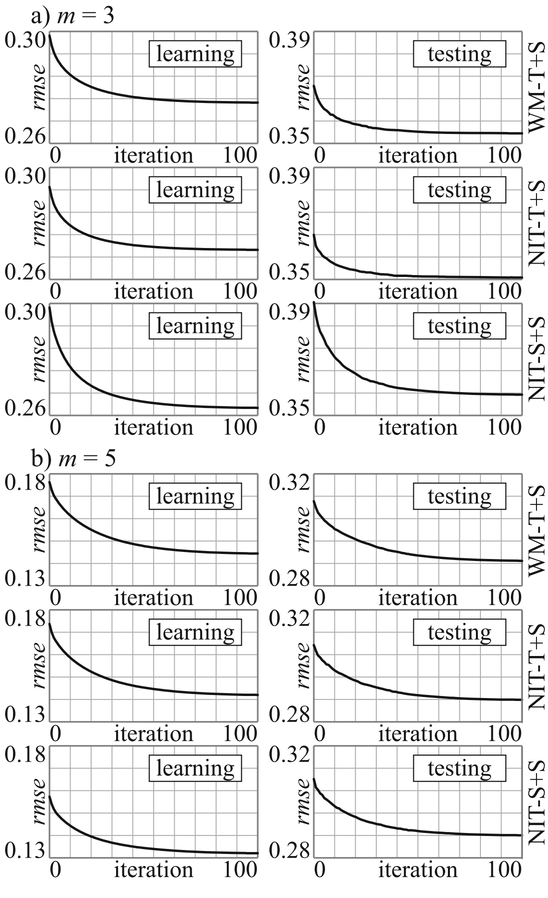
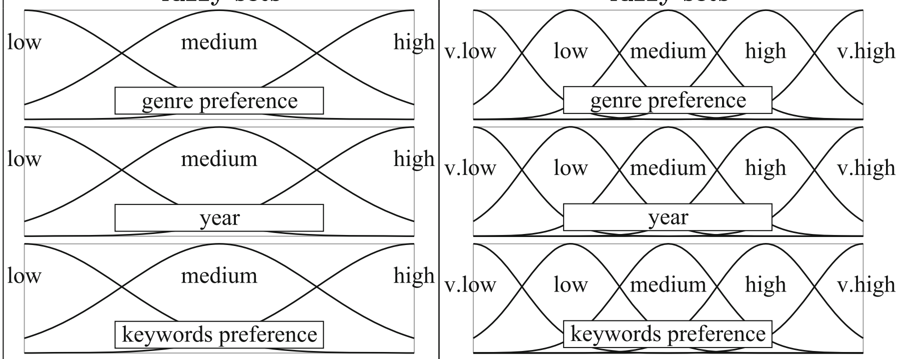


Fig. 8. Average *rmse* improvement during iterations of GWO

Table 3. Example of fuzzy system for NIT-S+S obtained for user 127 that rated 192 movies for $m = 3$ (left part of table) $m = 5$ (right part of table).

fuzzy sets



fuzzy rules ($m = 3$)

k	IF genre preference IS	IF year IS	IF keywords preference IS	THEN user rate =
1	low	medium	low	2.01
2	medium	high	medium	2.85
3	high	low	medium	3.85
4	high	medium	medium	3.21
5	high	high	low	2.00
6	high	high	medium	3.66
7	high	high	high	5.00

detailed results for selected user

$rmse = 0.29234$ (lrn), 0.30583 (tst)
$accuracy = 58.8\%$ (lrn), 56.4% (tst)
$yesno = 98.6\%$ (lrn), 97.1% (tst)

fuzzy rules ($m = 5$)

k	IF genre preference IS	IF year IS	IF keywords preference IS	THEN user rate =
1	v.low	medium	v.low	2.01
2	high	v.low	high	3.99
3	high	medium	medium	2.99
4	high	high	low	2.86
5	high	high	medium	3.02
6	high	high	high	4.15
7	high	v.high	low	2.85
8	high	v.high	medium	3.45
9	high	v.high	high	4.15
10	v.high	low	medium	3.07
11	v.high	medium	low	2.93
12	v.high	medium	medium	2.92
13	v.high	medium	high	5.00
14	v.high	high	medium	3.00
15	v.high	high	high	4.61
16	v.high	high	v.high	5.00
17	v.high	v.high	low	2.32
18	v.high	v.high	medium	3.03
19	v.high	v.high	high	4.65
20	v.high	v.high	v.high	5.00

detailed results for selected user

$rmse = 0.17194$ (lrn), 0.21830 (tst)
$accuracy = 87.9\%$ (lrn), 84.3% (tst)
$yesno = 100.0\%$ (lrn), 99.3% (tst)

The use of more partitions ($m = 5$) allowed to increase the accuracy of the system in every case (see Table 1), thus the number of created fuzzy rules increased (see Fig. 6).

The use of a higher number of partitions does not give a significant improvement of testing *rmse*, *accuracy* and *yesno*, especially if $m > 7$ (see Table 4). However, this results in more rules and the need to differentiate between more

Table 4. Additional comparison of results for NIT-S+S system with a higher number of fuzzy partitions.

m	*rmse*				*accuracy*				*yesno*				rules
	average		st. dev.		average		st. dev.		average		st. dev.		
	learn.	test.	learn.	test.	learn.	test.	learn.	test.	learn.	test.	learn.	test.	
3	0.263	0.357	0.017	0.148	66.85	56.52	4.55	10.31	99.06	96.25	0.49	5.42	10.87
5	0.132	0.288	0.010	0.150	90.68	70.85	1.73	9.18	99.79	97.97	0.10	3.19	21.78
7	0.081	0.273	0.009	0.168	95.54	74.05	0.91	8.73	99.88	97.43	0.03	4.03	29.14
9	0.051	0.270	0.006	0.178	97.98	75.72	0.43	8.11	99.92	97.18	0.02	4.45	35.20
11	0.039	0.273	0.005	0.188	98.62	75.68	0.34	8.26	99.97	97.13	0.01	4.41	38.79
13	0.029	0.272	0.005	0.191	98.96	76.36	0.27	7.80	99.98	97.23	0.01	4.34	41.86

linguistic labels of fuzzy sets, which significantly reduces the interpretability and readability of the system.

The *rmse* error calculated for learning samples increases simultaneously with the number of rated movies (see Table 2). This may be due to many factors: ratings based on attributes not included in the inputs, contradictions included in the user's ratings, use of proposed data set preparation method, etc.

The *rmse* calculated for unknown data samples (testing samples) is optimal in the case of 100–400 rated movies (in particular if $m = 5$ - see Table 2). This shows that the optimal number of rated movies for which there is no loss of information is contained in this range. Too many rated movies cause that the system has a too weak structure and would require to increase m or use of additional system inputs. Too few rated movies make it difficult to predict the correct recommendation for testing samples (see Table 2).

The number of fuzzy rules increases logarithmically along with the number of rated movies (see Fig. 6). The average number of fuzzy rules for $m = 3$ is close to 12, such a number may allow interpretation of the operation of the entire system. In the case of $m = 5$ average number of fuzzy rules is close to 28 and thus the interpretation of the operation of the entire system may be more difficult, which does not exclude the possibility of the interpretation of specific recommendations.

The correlation between the number of rated movies (o) and the number of created fuzzy rules according to the Pearson Correlation Coefficient is moderate in case of $m = 3$ and strong in case of $m = 5$ (see Fig. 7).

In simulation studies, the phenomenon of overfitting was not observed (see Fig. 8). Moreover, learning *rmse* usually decreased to the same extent as testing *rmse* (see Fig. 8).

It can be concluded that the exemplary fuzzy systems presented seem to be clear and interpretable at the same time ensuring high accuracy of operation (see Table 3).

Sample interpretations that can be drawn from the fuzzy rules presented in Table 3 are as follows: the user prefer older movies in some cases (see user rate for $k = 3$ and $k = 4$), the genre preference is less important for the user than keywords preference (see user rate for $k = 1$ and $k = 5$ vs $k = 6$), the keywords preference does not affect the result linearly (see user rate for $k = 5$, $k = 6$ and $k = 7$), etc.

Authors want to draw attention to the fact that interpreting results is much harder in the case of $m = 5$. It is worth noting, that the interpretations do not matter from the point of view of the explanation of specific recommendations and are given here as an example. In the case of specific recommendations, the user should only analyze these fuzzy rules that have influenced the result of the recommendation. Such solutions will be considered and analyzed in future work.

5 Conclusions

The proposed approach allows achieving high accuracy with a reasonable number of interpretable fuzzy rules. The use of ZO-TSK and optimization of singletons has allowed a significant improvement in results.

Conducted experiments proved that applying the Grey Wolf Optimizer to train the model gives better accuracy without losing interpretability of the system. The Zero-Order Takagi-Sugeno-Kang fuzzy system can be effectively used as a content-based recommender system that provides accurate results with intepretability, transparency, and explainability.

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

References

1. Bologna, G., Hayashi, Y.: Characterization of symbolic rules embedded in deep DIMLP networks: a challenge to transparency of deep learning. J. Artif. Intell. Soft Comput. Res. **7**(4), 265–286 (2017)
2. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adap. Inter. **12**(4), 331–370 (2002)
3. D'Aniello, G., Gaeta, M., Loia, F., Reformat, M., Toti, D.: An environment for collective perception based on fuzzy and semantic approaches. J. Artif. Intell. Soft Comput. Res. **8**(3), 191–210 (2018)
4. Dawar, D., Ludwig, S.A.: Effect of strategy adaptation on differential evolution in presence and absence of parameter adaptation: an investigation. J. Artif. Intell. Soft Comput. Res. **8**(3), 211–235 (2018)
5. Doran, D., Schulz, S., Besold, T.R.: What does explainable AI really mean? A new conceptualization of perspectives. CoRR, abs/1710.00794 (2017)
6. Faris, H., et al.: Grey wolf optimizer: a review of recent variants and applications. Neural Comput. Appl. **30**(2), 413–435 (2018)
7. Harper, F.M., Konstan, J.A.: The MovieLens datasets: history and context. ACM Trans. Interact. Intell. Syst. (TiiS) **5**(4), 19 (2016)

8. Jayabarathi, T., et al.: Economic dispatch using hybrid grey wolf optimizer. Energy **111**, 630–641 (2016)
9. Lipton, Z.C.: The mythos of model interpretability. Commun. ACM **61**, 36–43 (2018)
10. Liu, H., Gegov, A., Cocea, M.: Rule based networks: an efficient and interpretable representation of computational models. J. Artif. Intell. Soft Comput. Res. **7**(2), 111–123 (2017)
11. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
12. Mittal, N., Singh, U., Sohi, B.S.: Modified grey wolf optimizer for global engineering optimization. Appl. Comput. Intell. Soft Comput. **2016**, 8 (2016)
13. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_10
14. Prasad, M., Liu, Y.-T., Li, D.-L., Lin, C.-T., Shah, R.R., Kaiwartya, O.P.: A new mechanism for data visualization with Tsk-type preprocessed collaborative fuzzy rule based system. J. Artif. Intell. Soft Comput. Res. **7**(1), 33–46 (2017)
15. Riid, A., Preden, J.-S.: Design of fuzzy rule-based classifiers through granulation and consolidation. J. Artif. Intell. Soft Comput. Res. **7**(2), 137–147 (2017)
16. Rutkowski, L.: Computational Intelligence: Methods and Techniques. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-76288-1
17. Rutkowski, T., et al.: A content-based recommendation system using neuro-fuzzy approach. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE (2018)
18. Sadiqbatcha, S., Jafarzadeh, S., Ampatzidis, Y.: Particle swarm optimization for solving a class of type-1 and type-2 fuzzy nonlinear equations. J. Artif. Intell. Soft Comput. Res. **8**(2), 103–110 (2018)
19. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_9
20. Song, X., et al.: Grey Wolf Optimizer for parameter estimation in surface waves. Soil Dyn. Earthq. Eng. **75**, 147–157 (2015)