Metadata of the chapter that will be visualized in SpringerLink

Book Title	Artificial Intelligence and Soft Computing	
Series Title		
Chapter Title	Towards Interpretability of the Movie Recommender Based on a Neuro-Fuzzy Approach	
Copyright Year	2018	
Copyright HolderName	Springer International Publishing AG, part of Springer Nature	

Corresponding Author	Family Name	**Rutkowski**
	Particle	
	Given Name	**Tomasz**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Senfino Technologies
	Address	Czestochowa, Poland
	Division	
	Organization	Polish-Japanese Academy of Information Technology
	Address	Warsaw, Poland
	Email	tomasz.rutkowski@senfino.com
Author	Family Name	**Romanowski**
	Particle	
	Given Name	**Jakub**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Senfino Technologies
	Address	Czestochowa, Poland
	Email	
Author	Family Name	**Woldan**
	Particle	
	Given Name	**Piotr**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Senfino Technologies
	Address	Czestochowa, Poland
	Email	
Author	Family Name	**Staszewski**
	Particle	

	Given Name	**Paweł**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Senfino Technologies
	Address	Czestochowa, Poland
	Email	
Author	Family Name	**Nielek**
	Particle	
	Given Name	**Radosław**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Polish-Japanese Academy of Information Technology
	Address	Warsaw, Poland
	Email	

Abstract	In the paper, a neuro-fuzzy structure is implemented as a movie recommender. First, a novel method for transforming nominal values of attributes into a numerical form is proposed. This allows representing the nominal values, e.g. movie genres or actors, in a neuro-fuzzy system designed from scratch using the Mendel-Wang algorithm for rules generation. Several experiments illustrate performance of the neuro-fuzzy recommender.

Keywords (separated by '-')	Recommendation systems - Neuro-fuzzy systems - MovieLens

Towards Interpretability of the Movie Recommender Based on a Neuro-Fuzzy Approach

Tomasz Rutkowski[1,2](✉), Jakub Romanowski[1], Piotr Woldan[1], Paweł Staszewski[1], and Radosław Nielek[2]

[1] Senfino Technologies, Czestochowa, Poland
tomasz.rutkowski@senfino.com
[2] Polish-Japanese Academy of Information Technology, Warsaw, Poland

Abstract. In the paper, a neuro-fuzzy structure is implemented as a movie recommender. First, a novel method for transforming nominal values of attributes into a numerical form is proposed. This allows representing the nominal values, e.g. movie genres or actors, in a neuro-fuzzy system designed from scratch using the Mendel-Wang algorithm for rules generation. Several experiments illustrate performance of the neuro-fuzzy recommender. AQ1 AQ2

Keywords: Recommendation systems · Neuro-fuzzy systems MovieLens

1 Introduction

Recommender systems are very useful to suggest various services and products, e.g. movies, books, flights, financial investments or medical doctors. Such systems work based on information about recommended products (items) and/or information about users profiles. The most popular technique in recommendation systems is collaborative filtering (CF), see e.g. [1–5]. The recommender systems based on CF suggest the recommendation for a given user by checking the preferences of other users with respect to a degree of similarity, usually calculated using the Pearson or cosine similarity measures. Various soft computing techniques, see e.g. [6–10], can also be applied to design the CF recommenders. The main problem of the CF approach is a cold start, see e.g. [11,12], and sparsity of the rating matrix, see e.g. [13,14]. In contrast to the CF recommender systems, another approach, called content-based filtering (see e.g. [15–20]) ignores information about preferences of other users. The recommender systems of this type suggest the items characterized by similar features to those ones that the user preferred in the past. As it was mentioned in the literature, various methods have been proposed for designing recommender systems; however, the problem of interpretability of such system was very rarely studied.

L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10842, pp. 1–11, 2018.
https://doi.org/10.1007/978-3-319-91262-2_66

In this paper, a neuro-fuzzy structure is implemented as a movie recommender. First, a novel method for transforming nominal values into a numerical form is proposed. This allows representing nominal values, e.g. movie genres or actors, in a neuro-fuzzy system designed from scratch using the Mendel-Wang algorithm for rules generation. We also show several experiments illustrating the performance of the neuro-fuzzy recommender, and discuss the issue of interpretability.

2 Neuro-Fuzzy Recommender System

Recommendation systems require both numerical and nominal values to be considered. It allows achieving much higher efficiency in the recommendation by understanding the context and importance of components describing the situation concerning a person interested in the recommendation. Therefore, we develop a new method for converting nominal values to numerical ones.

For the purposes of our simulations, we have developed a method of converting nominal attributes to numerical form, taking into account the number of occurrences in samples assessed by the user. In the MovieLens database, users ranked their movies on a 1–5 rating scale. We divided all samples rated by one user into two groups: negative for ratings $\{1;\ 1.5;\ 2;\ 2.5;\ 3\}$ and positive for ratings $\{3.5;\ 4;\ 4.5;\ 5\}$. We determined a smaller range of ratings for positive samples because users are used to movie ranks, which are more positive for them. It is much more difficult to obtain negative movies ranks from users, so this causes a difference in the positive and negative values ranges. Rating range for negative samples is bigger because it allows balancing the number of samples to be learned for both classes. For nominal attributes, we need to determine which item position of the attribute value will be analyzed and what will be the significance of their position (eg. when movie genre is made up of several genres Horror-Drama-Thriller). We analyze the first three values, and their importance is specified in the Table 1.

Table 1. Initialized values of importance relative to position

Attribute	Values of importance on position		
	p_1	p_2	p_3
Genres	1,0	0,6	0,3
Actors	1,0	0,5	0,2
Directors	1,0	0,7	-
Producers	1,0	0,8	0,6
Spoken language	1,0	-	-

In the next step, occurrences of individual values of each attribute relative to the their position should be counted, separately for negative and positive samples as is shown in Table 2 for User 1 and User 2. More detailed sample with values of calculations based on the genre of the movie and on the ratings of one of the users are presented in the Table 3, and the calculation method is shown below:

$$x_{i,(n;p)} = \sum_{j=1}^{n} p_j * c_j \tag{1}$$

Formula (1) is the initial step of calculations, and it gives the possibility to obtain x_i which is unnormalized form of weight for attribute values. x_i is calculated separately for negative and positive samples, and index *"i"* represents next attribute values and *"j"* is index of importance position (in this example from 1 to 3). Value p_j is a value if importance from Table 1. c_j is the occurrence of the attribute value on specific position. The values for c_j are shown in Table 2.

The next step is to calculate normalized weights w_i for positive and negative samples by using Formula (2) where x_i is calculated by Formula (1), k_i represents all occurrences of attributes values. The k_i values are shown in Table 3 just like values of *COUNT* and *ALL*. Index *"i"* in Formula (2) is a representation of next attribute values. Table 3 is an exemplary representation of movie genres calculations for 3 most significant and 3 less significant attribute values.

$$w_{i,(n;p)} = \frac{x_{i,(n;p)}}{COUNT_i} * \frac{k_{i,(n;p)}}{ALL} \tag{2}$$

To obtain the final value of attribute value weight W_i, we propose Formula (3). Calculated values are obtained in previous steps, and exemplary values are shown in Table 3. The greater the weight, the better the positive recommendation, and the smaller the more negative for this particular feature.

$$W_i = \frac{1 - \frac{wn_i + wp_i}{MAX}}{2} \tag{3}$$

By applying the attributes analysis algorithm and calculating the composite attribute values, we can analyze the components of the several attribute's values and then reduce it to a single numerical value using the Formula (4). In this formula index *"k"* is an attribute number (from 1 to 5). Index *"z"* is an index of the selected value of the selected attribute *"k"*.

$$X_k = \frac{\sum_{j=1}^{n} p_{k,j} * W_{k,z}}{\sum_{j=1}^{n} p_{k,j}} \tag{4}$$

As we can convert all values to a numerical form, we can build rules in a neuro-fuzzy system. The simulations presented in this article were generated on the basis of fuzzy rules built by the Wang Mendel method [21] and then tuned by the backpropagation algorithm.

3 Data Set

The main source of data for our data set is the MovieLens 20M database. MovieLens is a publicly available database of movies and users who rated these movies.

As a result, it creates perfect conditions for testing recommendation systems. A significant number of movies, large number of users and ratings are the criteria required for testing. The MovieLens database fulfills these conditions. Movies in this database are described by genres, year, tags, 5-star rating and external services movies IDs.

However, we decided to enlarge the number of relevant information about the movies in order to understand the deeper needs of users and provide the possibility of interpreting of the recommendation. We took advantage of external services movies IDs contained in the MovieLens database and we use IMDB and TMDB external services to obtain more information about each of the movies.

Finally, our Data Set consists of 27278 movies, 20000263 ratings and following movies attributes: Genres, Actors, Directors, Producers, Spoken language and the values of the attributes are presented in Table 2. Attributes Genres and Spoken language are presented in Table 2 with a full list of their values. The other attributes are presented in the form of the three most and the three least significant values for the user due to their huge amount.

4 Testing Environment

The tests were performed based on our AI environment which is AI framework developed in C++ programming language using CUDA (Compute Unified Device Architecture) Technology provided by NVidia. Application of the CUDA allows for the high acceleration of calculations by using GPU compared to traditional calculations on CPU.

All of data about users, movies and ranks are stored in database. From the attributes and their values tables stored in our database, we select only those that we use for the experiment. However, for the purposes of other experiments, we store much more information about attributes.

The summary list of applied technologies looks as follows:

- MSSQL Server 2016
- CUDA Toolkit v9.0.176
- CUDNN v7
- BOOST v1.65.1 (vectors and logging features).

5 Experimental Results

5.1 Users Attributes Calculations

As an example of the weight calculation, we presented a list of attributes and their values for two exemplary users which are given in Table 2. Table 2 shows the number of appearances attribute values at three different positions (c_1, c_2, c_3) for both samples evaluated by the user as a negative and a positive. A detailed set of calculations of weights on the *Genres* of the movies is presented in Table 3 and the meaning of the column headers is as follows: k_n, k_p are values of sum

Table 2. Weights calculation and list of attributes and their values for two exemplary users

Attributes	User 1								User 2	
	Attributes values	Negative			Positive			Weight	Attributes values	Weight
		c_1	c_2	c_3	c_1	c_2	c_3			
Genres	Comedy	16	7	5	42	13	10	0,911		
	Drama	8	18	6	19	32	2	0,723	Drama	0,898
	Crime	0	6	5	8	14	6	0,658	Comedy	0,752
	Romance	0	1	8	3	9	12	0,607	Crime	0,614
	Animation	1	0	0	5	2	0	0,572	Romance	0,560
	Family	1	0	0	2	2	6	0,560	History	0,556
	Adventure	7	6	0	11	6	1	0,557	Adventure	0,554
	Documentary	1	0	0	5	0	0	0,554	War	0,551
	History	0	0	0	1	1	1	0,526	Thriller	0,548
	Sci-Fi	0	1	5	1	3	5	0,524	Mystery	0,538
	Mystery	0	2	1	1	2	3	0,521	Animation	0,537
	Fantasy	3	1	1	6	0	0	0,520	Sci-Fi	0,513
	Thriller	1	7	8	2	6	11	0,518	Fantasy	0,508
	War	0	0	0	0	1	2	0,516	Horror	0,500
	Western	0	0	0	0	0	1	0,504	Western	0,500
	Foreign	0	1	0	0	1	0	0,500	Music	0,493
	Music	1	2	1	1	0	0	0,476	Family	0,450
	Horror	2	0	1	0	0	0	0,469	Action	0,317
	Action	19	4	2	9	7	0	0,388		
Actors	Michael Douglas	0	0	0	5	0	0	1,000	S. Connery	0,983
	Harrison Ford	1	0	0	3	3	0	0,871	J. Nicholson	0,873
	Tom Hanks	1	0	0	4	0	1	0,833	M. Broderick	0,850
	...	...	...	...	...	...	...	...	...	...
	Will Smith	1	1	0	0	0	0	0,350	M. J. Fox	0,150
	Ben Stiller	2	0	0	0	0	0	0,300	E. Murphy	0,150
	John Cusack	4	1	0	0	0	0	0,050	B. Willis	0,127
Directors	Nick Park	0	0	–	5	0	–	1,000	J. Coen	1,000
	Joel Coen	0	0	–	3	0	–	0,800	B. Wilder	1,000
	Jay Roach	0	0	–	3	0	–	0,800	R. Zemeckis	1,000
	...	...	...	...	...	...	...	...	...	...
	Tom DiCillo	1	0	0	0	0	–	0,400	J. McTiernan	0,250
	John Dahl	1	0	0	0	0	–	0,400	R. Harlin	0,250
	Michale Bay	2	0	0	0	0	–	0,300	T. Scott	0,125
Producers	Paramount	2	0	0	13	1	0	0,990	Universal	0,919
	20th Cent. Fox	3	0	0	7	5	0	0,840	United Art.	0,888
	PolyGram	0	0	0	3	2	1	0,715	Columbia	0,743
	...	...	...	...	...	...	...	...	...	...
	J. Bruckheimer	3	0	0	0	0	0	0,376	RKO Radio	0,354
	Regency Enterp	4	0	0	0	0	0	0,334	Touchstone	0,303
	Touchstone	3	2	2	0	0	0	0,260	Walt Disney	0,257
Spoken language	en	57	–	–	110	–	–	0,866	en	0,737
	it	0	–	–	3	–	–	0,521	ja	0,508
	fr	0	–	–	1	–	–	0,507	zh	0,508
	cs	0	–	–	1	–	–	0,507	it	0,508

of all negative and positive occurrence of attribute values on each of positions (c_1, c_2, c_3) as shown in Table 2. *COUNT* column is the sum of k_n and k_p. The columns x_n and x_p are calculated based on the Formula (1). Values of sum_1, sum_2, ALL, MAX_1, MAX_2 and MAX are considered for full list of *Genres* attribute values, not only for six exemplary values.

Formula (2) is used to calculate partial weights w_i for negative and positives samples. In result, final weight W_i of attribute value can be calculated by using Formula (3).

Table 3. A detailed calculations of weights for the Genres attribute values

Attribute values	$\mathbf{k_n}$	$\mathbf{k_p}$	COUNT	$\mathbf{x_n}$	$\mathbf{x_p}$	$\mathbf{w_n}$	$\mathbf{w_p}$	Weight
Comedy	28	65	93	21,7	52,8	0,01597	0,09023	0,911
Drama	32	53	85	20,6	38,8	0,01896	0,05915	0,723
Crime	11	28	39	5,1	18,2	0,00352	0,03195	0,658
...								
Music	4	1	5	2,5	1,0	0,00489	0,00049	0,476
Horror	3	0	3	2,3	0,0	0,00557	0,00000	0,469
Action	25	16	41	22,0	13,2	0,03248	0,01247	0,388

$\mathbf{sum_1}$	$\mathbf{sum_2}$	ALL
134	275	409

$\mathbf{MAX_1}$	$\mathbf{MAX_2}$	MAX
0,03280	0,09023	0,09023

5.2 Measure of Effectiveness

The real measure of the effectiveness of recommendation systems is the user's feeling of the items recommended for him. In fact, the number of items recommended positively or negatively plays a secondary role. The most valuable thing is to recommend even one item, but one that adds the most value to the user of the recommendation system.

In order to be able to determine how much the recommendation is effective, we decide to perform experiments for two separate users, as shown in Table 4.

Here is a shortcut set in the headers of the table columns Tables 4 and 5 and their meaning is described below. *Ranks count* - is total movies ranks count created by user/users, *Train samples* - is part of *Ranks count* and it is used to train the system. *Test samples* - this is the remaining part *Ranks count* ant its amount is always 50 for each tested user. *UNR* - User Negative Ranks, *NR* - Negative Recommendations, *UPR* - User Positive Ranks, *PR*- Positive Recommendations, *Effectiveness* in meaning of ratio of User ranks to recommendations of the system. *UNR* and *UPR* are values of users ranks from our data set. *NR* and *PR* are values calculated by our recommender system. By describing the ratio *UNR* to *NR* and *UPR* to *PR* we show how many realistic movies ratings are equal to the recommendations of our system. It is worth mentioning here about the set of movies with the uncertainty of recommendation. We can not recommend them as positive or negative samples.

Table 4. Table of recommendation results for two separate users (User 1 and User 2)

	Ranks count	Train samples	Test samples	UNR/NR	UPR/PR	Effectiveness
User 1	226	176	50	3/5	15/17	18/22
User 2	251	201	50	4/8	26/34	30/42

Table 5. Table of recommendation results for groups of users (from 100 to 500)

Users count	Ranks count	Train samples	Test samples	UNR/NR	UPR/PR	Effectiveness
100	21731	16731	5000	532/774	2400/3119	2932/3893
200	45772	35772	10000	972/1381	4998/6449	5970/7830
300	71501	56501	15000	1558/2233	7484/9475	9042/11708
400	95263	75263	20000	2188/3155	9765/12315	11953/15470
500	117179	92179	25000	2954/4249	12128/15218	15082/19467

In the next step we made the same experiment as above but for a larger scale of users (from 100 to 500) as is shown in Table 5.

Tables 6 and 7 show the results of Leave-one-out cross-validation. First of them shows results for User 1 and User 2 and the second shows results for 100 users.

Table 6. Leave-one-out cross-validation for User 1 and User 2

	Rank count	Train samples	Test samples	Effectiveness
User 1	226	225	226	148/226
User 2	251	250	251	172/251

Table 7. Leave-one-out cross-validation for 100 users

Users count	Rank count	Train samples	Test samples	Effectiveness
100	16216	16116	10216	12801/16216

5.3 Interpretability of the Recommendation Result

One of the most important values of the recommendation system is to understand the reasons for the recommendation. Interpretability of the result of the recommendation is therefore key value.

For both users User 1 and User 2 we have prepared graphs of linguistic variables and rules between them. There are two forms of graphs - with all of rules and simplified. Simplified one is prepared to clarify the graph and to allow analyzing the interpretability of recommendation results.

Figures 1 and 2 present linguistic variables with full set of rules for User 1 and User 2. Similarly simplified rules for linguistic values for User 1 and User 2 are shown in Figs. 3 and 4.

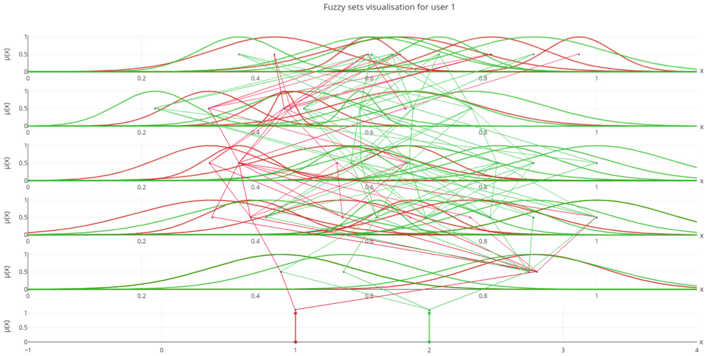

Fig. 1. Linguistic variables and rules for User 1

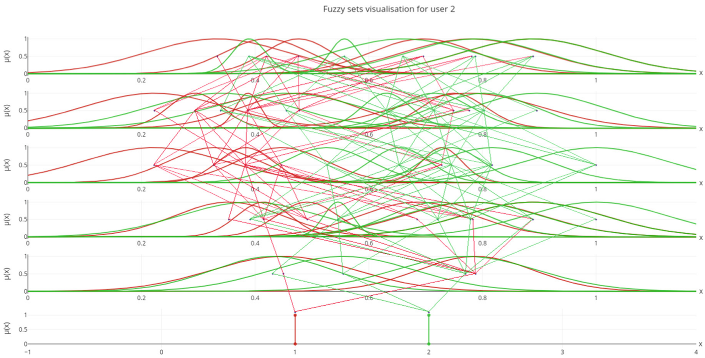

Fig. 2. Linguistic variables and rules for User 2

The order of linguistic variables in the graphs is the same as attributes order in the Table 2 (*Genres*, *Actors*, *Directors*, *Producers*, *Spoken languages*). The last axis in the graphs corresponds to the output from the system, with two values where value 1 indicating a negative recommendation and value 2 indicating a positive recommendation.

By applying our method of encoding nominal values, it is possible to interpret the recommendation by connecting rules from the systems with the value of the first linguistic variable.

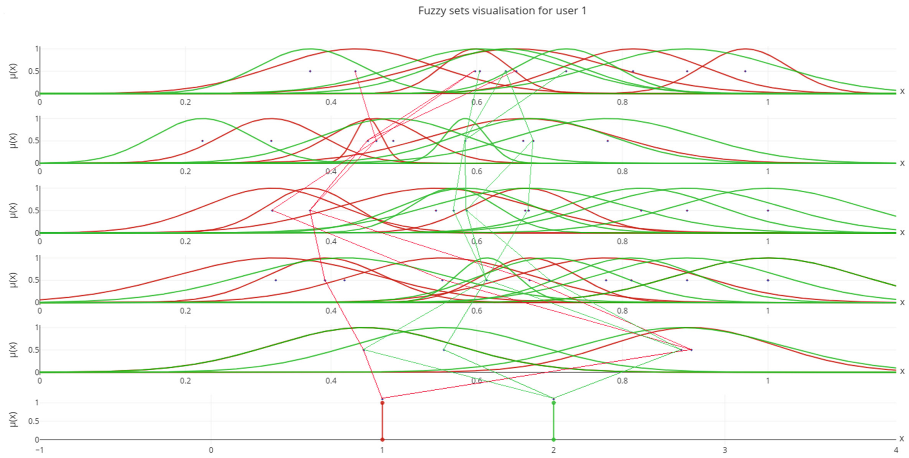

Fig. 3. Linguistic variables and simplified rules for User 1

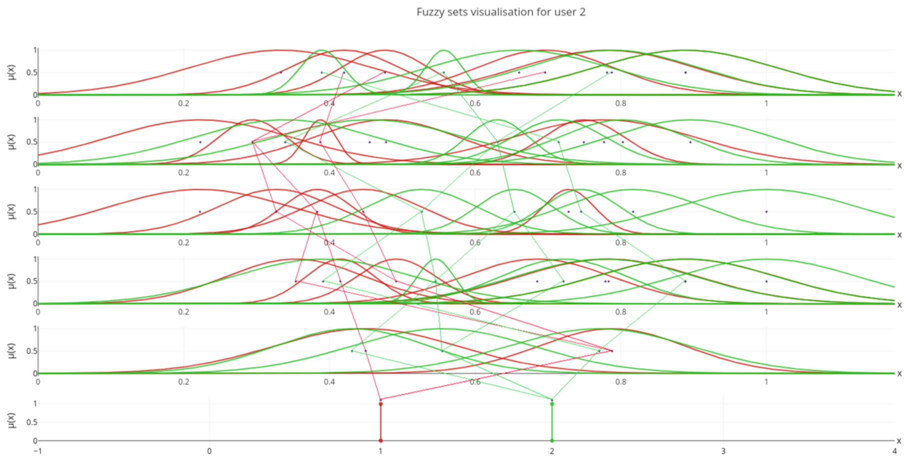

Fig. 4. Linguistic variables and simplified rules for User 2

Based on Fig. 5 and the assumption that the movie consists of the genre Comedy, its output value for User 1 equals 0,911 according to the Formula (4). Analyzing this value with regard to User 1, we see that this value falls almost perfectly into the positive set of the *Comedy* linguistic variable. In order to positively recommend the movie, the rules with the positive conclusion should be activated by input values of other linguistic variables. The rule must contain the set of *Comedy* genre indicated above. By analyzing fuzzy sets along with the rules in this way, we can easily obtain information about the user's preferences with respect to individual attributes and their values. If we distinguish all the values of our attributes with their combinations and extract linguistic variables, we can

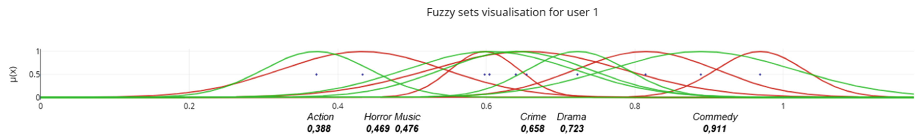

Fig. 5. Example of *Genres* linguistic variable

analyze them based on the fuzzy sets. Values of overlapping fuzzy sets indicate recommendations that are not certain. However, using fuzzy sets allow us to specify a degree of certainty. The rules are formulated by the use of appropriate fuzzy sets presented in Figs. 1, 2, 3, 4 and 5, which allows interpreting the final recommendation by matching the input values with the specific fuzzy sets of those rules.

6 Conclusions

In this paper, we present a way to interpret which attributes mostly influence the movie recommender. Interpretability is the biggest advantage of using a neuro-fuzzy approach. In the future work we plan to apply the rough set theory [22] for rule generation describing the movie recommender. That will allow us to explore different techniques to achieve the goal of building an interpretable and accurate recommender system.

References

1. Nie, F., Wang, H., Huang, H., Ding, C.: Joint schatten lp-norm robust matrix completion for missing value recovery. Knowl. Inf. Syst. **42**(3), 525–544 (2013). https://doi.org/10.1007/s10115-013-0713-z
2. Zhao, K., Pan, L.: A machine learning based trust evaluation framework for online social networks, pp. 69–74 (2015). https://doi.org/10.1109/TrustCom.2014.13
3. Anaissi, M., Goyal, M.: SVM-based association rules for knowledge discovery and classification (2015). https://doi.org/10.1109/APWCCSE.2015.7476236
4. Lu, J., Hoi, S., Wang, J., Zhao, P.: Second order online collaborative filtering. J. Mach. Learn. Res. **29**, 325–340 (2013)
5. Zhao, Q., Zhang, Y., Friedman, D., Tan, F.: E-commerce recommendation with personalized promotion, pp. 219–225 (2015). https://doi.org/10.1145/2792838.2800178
6. 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)
7. Beg, I., Rashid, T.: Modelling uncertainties in multi-criteria decision making using distance measure and topsis for hesitant fuzzy sets. J. Artif. Intell. Soft Comput. Res. **7**(2), 103–109 (2017)
8. 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)

9. 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)
10. Prasad, M., et al.: 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)
11. Wei, J., He, J., Chen, K., Zhou, Y., Tang, Z.: Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. **69**, 29–39 (2017)
12. Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K.: A literature review and classification of recommender systems research. Expert Syst. Appl. **39**(11), 10059–10072 (2012)
13. Alemeye, F., Getahun, F.: Cloud readiness assessment framework and recommendation system, November 2015. https://doi.org/10.1109/AFRCON.2015.7331995
14. Burke, R.: Hybrid recommender systems: survey and experiments. User Model User-Adap. Interact **12**(4), 331–370 (2002)
15. Baldominos, A., Albacete, E., Saez, Y., Isasi, P.: A scalable machine learning online service for big data real-time analysis (2015). https://doi.org/10.1109/CIBD.2014.7011537
16. Kao, C.-Y., Fahn, C.-S.: A multi-stage learning framework for intelligent system. Expert Syst. Appl. **40**(9), 3378–3388 (2013)
17. Tsuji, K., Yoshikane, F., Sato, S., Itsumura, H.: Book recommendation using machine learning methods based on library loan records and bibliographic information, pp. 76–79 (2014). https://doi.org/10.1109/IIAI-AAI.2014.26
18. Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A.: Recommendation systems: principles, methods and evaluation. Egypt. Inform. J. **16**, 261–273 (2015)
19. Portugal, I., Alencar, P., Cowan, D.: The use of machine learning algorithms in recommender systems: a systematic review. Expert Syst. Appl. **97**, 205–227 (2017)
20. Burke, R.: Hybrid web recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_12
21. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. IEEE Trans. Syst. Man Cybern. **22**(6), 1414–1427 (1992)
22. Pawlak, Z.: Rough set theory for intelligent industrial applications. In: Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials, IPMM 1999 (Cat. No.99EX296), vol. 1, pp. 37–44 (1999)

Author Proof

Author Queries

Chapter 66

Query Refs.	Details Required	Author's response
AQ1	This is to inform you that corresponding author has been identified as per the information available in the Copyright form.	
AQ2	Kindly note that the e-mail id is correctly identified for corresponding author.	