Performance Comparison of Featured Neural Network Trained with Backpropagation and Delta Rule Techniques for Movie Rating Prediction in Multi-criteria Recommender Systems

Mohammed Hassan
University of Aizu, Aizuwakamatsu, Fukushima, Japan
E-mail: d8171104@u-aizu.ac.jp

Mohamed Hamada
University of Aizu, Aizuwakamatsu, Fukushima, Japan
E-mail: hamada@u-aizu.ac.jp

Keywords: multi-criteria recommender systems, artificial neural network, prediction accuracy, backpropagation, delta rule

Received: November 22, 2016

Recommender systems are software tools that have been widely used to recommend valuable items to users. They have the capacity to support and enhance the quality of decisions people make when finding and selecting items online. Such systems work based on which techniques are used to estimate users' preferences on potentially new items that might be useful to them. Traditionally, the most common techniques used by many existing recommendation systems are collaborative filtering, content-based, knowledge-based and a hybrid-based which combines two or more techniques in different ways. The multi-criteria recommendation technique is a new technique used to recommend items to users based on ratings given to multiple attributes of items. This technique has been used and proven by researchers in industries and academic institutions to provide more accurate predictions than traditional techniques. However, what is still not yet clear is the role of some machine learning algorithms such as the artificial neural network to improve its prediction accuracy. This paper proposed using a feedforward neural network to model user preferences in multi-criteria recommender systems. The operational results of experiments for training and testing the network using two training algorithms and Yahoo!Movie dataset are also presented.

Povzetek: Opisana je primerjava več metod, tudi nevronske mreže, za napovedovanje uspešnosti filmov z večkriterijskim priporočilnim sistemom.

1 Introduction

Recommender systems are intelligent systems that play important roles in providing suggestions of valuable items to users. The types of suggestions given by the systems can be of different forms depending on the domain of recommendations. For example, in a movie recommendation problem such as Netflix[1], the systems can suggest the kinds of movies to watch. Similarly, music can be recommended to users in a music recommender systems like Pandora[2], or items to buy can be recommended in Amazon[3], or personalized online news recommender systems like Google-News[4] can recommend news for users to read [1, 2, 3, 4]. Recommender systems are classified based on the technique used during their design and implementation. Traditionally, collaborative filtering, content-based, knowledge-based, and a hybrid-based filtering are the commonly used techniques to design recommender systems. Therefore, knowing the recommendation techniques is at the heart of our understanding of recommender systems. Those techniques are sometimes called traditional techniques, and are increasingly becoming popular ways of building recommender systems [5].

However, despite their popularity and ability to provide considerable prediction and recommendation accuracies, they suffer from major drawbacks [6, 7, 8] because they work with just a single rating, whereas most of the time the acceptability of the item recommended may depend on several item's attributes [9]. Researchers have suggested that if ratings provided to those several characteristics of items would be considered during the prediction and recommendation process, it could help to enhance the quality of recommendations since complex opinions of users will be captured from various attributes of the item. Recent developments in this field have led to the existence of a new

This paper is based on Mohammed Hassan & Mohamed Hamada, Rating Prediction Operation of Multi-criteria Recommender Systems Based on Feedforward Network, published in the *Proceedings of the 2nd International Conference on Applications in Information Technology (ICAIT-2016).*

[1]https://www.netflix.com/
[2]www.pandora.com
[3]https://www.amazon.com/
[4]https://news.google.com/

recommendation technique known as the multi-criteria recommendation technique [6, 9] that exploits multiple criteria ratings from various items' characteristics to make recommendations. This technique has been used for a wide range of recommendation applications such as recommending products to customers [11, 10], hotel recommendations for travel and tourism [12], and so on. Nevertheless, having considered multi-criteria techniques as the answer to some of the limitations of traditional techniques, it is also logical to look at various ways of modeling the multiple ratings to enhance the prediction accuracies and recommendation qualities. However, few researchers have been able to advance on systematic research into improving the prediction accuracy [13]. In addition, no previous research has investigated the effect of using artificial neural networks to model users' preferences in order to improve the prediction operations of multi-criteria recommender systems [9]. Therefore, as an attempt to investigate the effectiveness of applying neural network techniques for improving prediction accuracies of multi-criteria recommender systems, this study seeks to examine the performance of backpropagation and delta rule algorithms to train the network using a multi-criteria rating dataset for recommending movies to users based on four attributes of the movies. This paper has been divided into five sections including this introduction section. The second part of the paper gives a brief literature review. The experimental methodologies are contained in the third section while the fourth section displays the results and discussion and the final section is concerned with the conclusion and presenting future research work.

2 Related background

2.1 Multi-criteria recommender systems

To be able to understand the concept of recommender systems, some mathematical notations U, I, δ, and ψ to represent the set of users, the set of items, a numerical rating, and a utility function are introduced respectively. The notation δ is the measure of the degree to which a user $\mu \in U$ will like $\iota \in I$, while the utility function ψ is a mapping from a $\mu \times \iota$ pair to a number δ, written as $\psi : \mu \times \iota \mapsto \delta$. The value of δ is a number within a specifically defined interval such as between 1 and 5, 1 and 13, or it can be represented using non-numerical values such as "like", "don't like", . . ., "strongly like", true or false, and so on [14]. Therefore, recommender systems try to predict the value of $\delta\ \forall \iota \in I$ that have not been seen by μ and recommend those with a high value of δ.

The methods of prediction and recommendation explained in the above paragraph are the mechanisms followed essentially by traditional recommendation techniques. Moreover, a similar approach is followed by the multi-criteria recommendation technique with the distinction that it uses multiple values of δ for each $\mu \times \iota$ pair. In the multi-criteria technique, the utility function ψ can be generally defined using the relations in equation 1.

$$\psi : \mu \times \iota \mapsto \delta_0 \times \delta_1 \times \delta_2 \times ... \times \delta_n \tag{1}$$

It is important to note however, there are n ratings in the above equation with the additional rating δ_0 called the overall rating which needs to be computed based on the other n values as in equation 2.

$$\delta_0 = f(\delta_1, \delta_2, \delta_3, ..., \delta_n) \tag{2}$$

The technique can work even without taking δ_0 into account so that there is no overall rating, only ratings of other attributes will be used to undertake the operation process. However, evidence observed from many researchers confirmed the greater efficiency of considering the overall rating rather than ignoring it [6]. The two common approaches used to model multi-criteria rating recommenders are heuristic-based approach that uses certain heuristic assumptions to estimate the rating of an individual item for a user, and a model-based approach that learns a model to predict the utility and recommends unknown items. This classification leads to grouping the multi-criteria rating algorithms into model- and heuristic-based algorithms. For the sake of this experiment, we only need to understand one model among model-based approaches known as the aggregation function model, but nonetheless, for a detailed explanation of the two categories of multi-criteria rating algorithms readers can refer to [8, 9].

The aggregation function approach starts by selecting and training a function or model (such as a neural network) to learn how to predict the overall rating from the criteria ratings. Secondarily, the multi-criteria problem will be decomposed into traditional recommendation problems so that missing ratings for each criterion can be treated as a single rating problem. Finally, the system uses the trained model and the single rating recommenders to predict the overall rating as in equation 2.

2.2 Artificial neural network model

An artificial neural network is one of the most powerful classes of machine learning models that can learn complicated functions from a data to solve many optimization problems. It aimed to mimic the functions of biological neurons that receive, integrate, and communicate incoming signals to other parts of the body [15]. Similarly, the artificial neural network contains sets of connected neurons arranged in a layered style (see Figure 1), where the input layer consists of neurons that receive input from the external environment and the output layer neuron receives the weighted sums of the products of input values and their corresponding weights from the previous layer and sends its computational result to the outside environment.

The features x_1, x_2, x_3, and x_4 in Figure 1 are inputs presented to the input layer, the parameters $\omega_0, \omega_1, \omega_2, \omega_3$, and ω_4 are the synaptic weights for links between the input and output nuerons. $\sum$ is the weighted sum of $\omega_i x_i$ for $i \in [0, 4]$, x_0 is a bias, and f is an activation function that

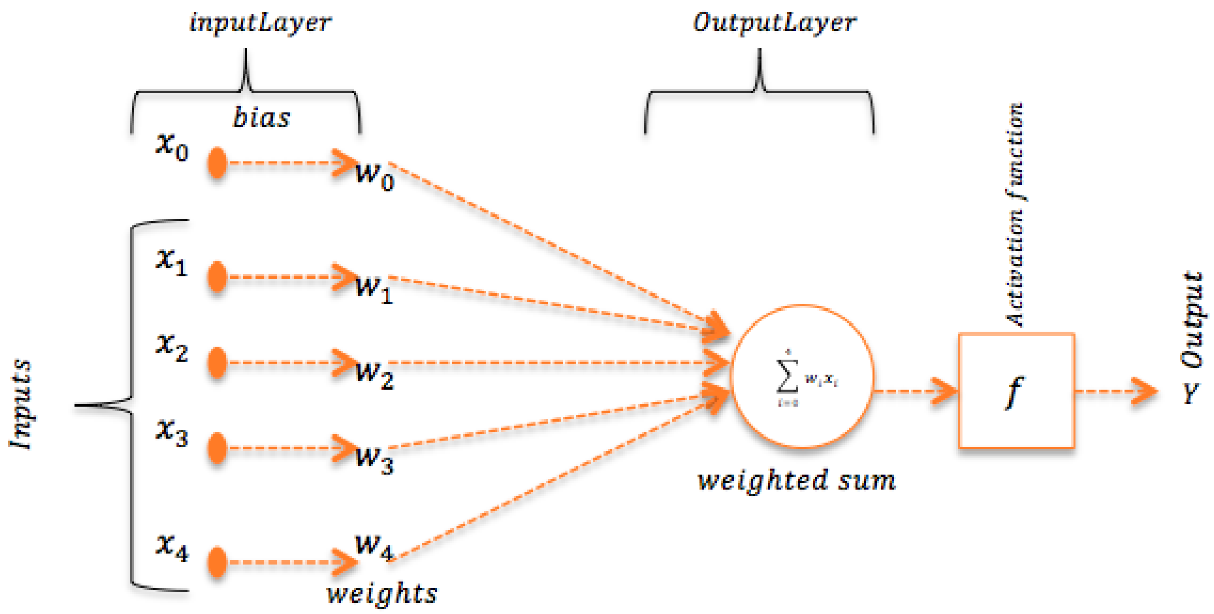

Figure 1: A Single Layer Neural Network

estimates the output Y written as $f(\sum_{i=0}^{4} \omega_i x_i)$. A feedforward network may contain more than two layers, where hidden layer(s) can be added between the input and output layers. The second experiment uses an extended version of the network presented in Figure 1 by adding one hidden layer between input and output layer. This is a brief explanation of how the neural network behaves; details of its learning process can be found in several machine learning books and articles [15, 16].

3 Experiments

The experiment was performed using Yahoo!Movies[5] datasets obtained from [10] for a multi-criteria movie recommendation system where movies are recommended to users based on four characteristics of movies, namely, action, story, direction, and visual effect of the movie which are represented as c_1, c_2, c_3, and c_4 respectively. In addition to those four criteria, an additional rating c_o, called overall rating criterion, was used to represent the final user's preference on a movie. The criteria values (ratings) in the dataset were initially presented using a 13-fold quantitative scale from A^+ to F representing the highest and the lowest preferences of the user respectively. In the same manner, we changed the rating representation to numerical form (13 to 1 instead of A^+ to F).

Table 1 consists of three parts: namely, *Original, Modified, and Normalized* datasets, where the first part displays the sample of the original dataset extracted, and the second part of the table is the same sample of the dataset modified into numerical ratings. Finally, for the network models to work faster and more efficiently, the numerically transformed dataset was normalized to real numbers between 0 and 1 through dividing each of the modified ratings by 13 (since 13 is the highest) as displayed in the last part of the same table. The dataset was well cleaned to avoid cases of uncompleted entries where ratings to some criteria will be missing, and also cases of users who rated few movies (less than five movies). Movies rated by a small number of users were removed completely from the dataset. This data cleaning process reduced the size of the dataset to a total of approximately 63,000 ratings sets. The dataset was divided into training and test data in a ratio of 75:25 for the two experiments. The target of the study was to use a feedforward network to learn how to estimate c_o from c_1, c_2, c_3, and c_4.

Two feedforward networks were developed using object oriented programming techniques in java [18] with learning capacities in delta rule and backpropagation. The Adaline network consists of an input and output layer as in Figure 1 with the input layer containing four neurons and a bias for passing the data to the output layer. The linear activation function f was used in the output neuron to process the weighted sum of the inputs x_i received from the input layer.

Furthermore, in addition to the two layers in the Adaline, a network containing an additional hidden layer with the same number of neurons as the input layers was used for backpropagation training with an additional activation function g (sigmoid function) that receives the weighted sum from the input layer and sends the result of its computation to the output neuron. For measuring the training and test error, mean square error in equation 3 for real output o_j (where $o_j = f(\sum_{i=0}^{5} x_i \omega_i)$) and the estimated output

[5] https://www.yahoo.com/movies/

	UserID	MovieID	Action c_1	Story c_2	Direction c_3	Visual c_4	Overall c_o
Original dataset	1	459	B	A^-	A	A^-	B^+
		554	A^-	A	A	A	A
		554	A^-	A^-	A	A^+	A^-
Modified dataset	1	459	9	11	12	11	10
		554	11	12	12	12	12
		554	11	11	12	13	11
Normalized dataset	1	459	0.692...	0.846...	0.923...	0.846...	0.769...
		554	0.846...	0.923...	0.923...	0.923...	0.923...
		554	0.846...	0.846...	0.923...	1.000	0.846...

Table 1: Sample of extracted and modified dataset

y_j, was used to compute the errors. Pearson correlation coefficient (PCC) presented in equation 4 was also used as a metric for measuring the relative relationship between the real and estimated output for the test data.

$$MSE = \frac{1}{2N}\sum_{j=1}^{N}(y_j - o_j)^2 \tag{3}$$

$$PCC = \frac{\sum(y_j - \overline{y})(o_j - \overline{o})}{\sqrt{\sum(y_j - \overline{y})^2}\sqrt{\sum(o_j - \overline{o})^2}} \tag{4}$$

4 Results and discussion

In each of the two algorithms, neurons' weights ω_i were initially generated at random and the network computes the outputs and the corresponding errors (as $\frac{1}{2}(y_j - o_j)^2$). Iteratively, the algorithms search for a set of weights ω_i $i = 0, 1, ..., 4$ that minimize the error. Since the two algorithms are based on gradient descent, the training begins at some points on the error function shown in equation 3 with defined ω_i, and tries to move to the optimal solution (global minimum) of the function. The rate of the movement is always determined by a parameter known as *learning rate* denoted by α which controls how much the $\omega_{i's}$ can be changed with respect to the observed training errors. Therefore, choosing the correct α is paramount since it can greatly influence the accuracy of the models. Deciding on the best value of α is not always obvious from the beginning of the experiment, as such, the study began by testing various values between 0.1 and 0.001 to find the one that could relatively produce the smallest error. The entire experiment was carried out using $\alpha = 0.007$, which produced the optimal error. The adaptive linear neuron (Adaline) network trained using delta rule shows a quick convergence within a few number of iterations (about 10 iterations) with a very good performance. On the other hand, the backpropagation algorithm prolongs the learning process where a large number of training cycles (epochs) have been used to monitor its performance and the result is presented in Figure 2. This figure shows the average MSE for the various numbers of training cycles. It shows that the convergence can only be attained at a very high number of iterations. However, for the purpose of comparison, the number of training cycles was set to 10,000 cycles (epoch = 10,000), the training error and correlations between the actual and estimated output of the test set for the two algorithms are shown in Table 2. Furthermore, to reaffirm the correlations between test results each of the two models and the actual values from the dataset, Figure 3 shows the curves, one for the actual values from the dataset, and the other two represent the corresponding predicted values by Adaptive linear neuron (Adaline)- and backpropagation (BPA)-based networks. The figure confirmed the accuracy of the Adaline network over the backpropagation-based network.

Table 2: **Performance Statistics**

Algorithm	Number of Iterations	Average Training MSE ($\times 10^{-3}$)	Percentage PCC
Adaline	10	5.34	94.4%
BPA	10,000	7.30	90.0%

5 Conclusion and future work

This study was carried out to investigate the relative performance of single layer and multilayer feedforward networks trained using delta rule and the backpropagation algorithm respectively.

The performance of each model was measured using MSE for the training and the percentage of the correct predictions were evaluated on the test data using Pearson correlation coefficient. From Figure 2 and Table 2, it can be seen that the backpropagation algorithm has a greater demand for longer training cycles to converge.

Moreover, the results indicate that the one layer network trained using the delta rule algorithm is more efficient than the two layer network which supports the tradi-

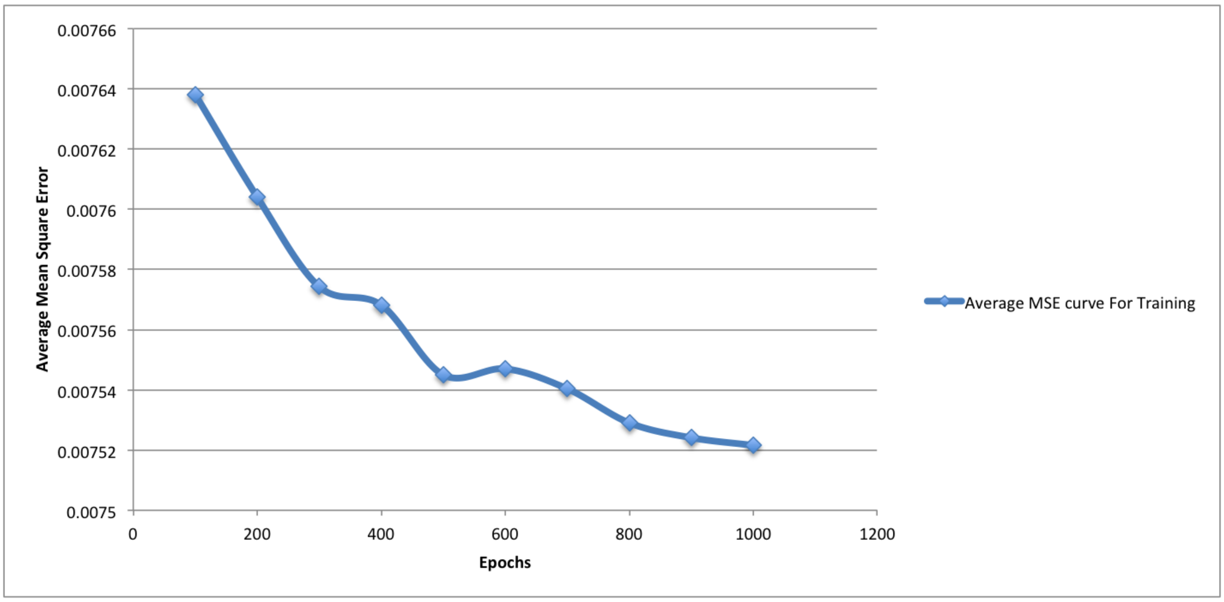

Figure 2: Average training MSE for Backpropagation

tional belief that a single layer network produces less error than a multilayered network [17]. Up to our last experiment with epochs of 10,000, backpropagation did not completely show final convergence, therefore, further investigation is recommended to estimate the approximate epochs required by the algorithm to converge and to know whether it will produce a better result than Adaline. The study confirmed the usefulness of training a neural network model with features of inputs obtained for predicting user preferences on items based on several characteristics of items in multi-criteria recommender systems. Future studies on the current topic are recommended to investigate the performance of more sophisticated neural network architectures and algorithms such as the restricted Boltzmann machine, deep neural networks, convolutional neural networks, and other similar neural networks. However, as the result of this study gives us a hint on the best network architecture and appropriate training algorithm to use, further work is required to extend this research by integrating the model with some popular collaborative filtering algorithms; such as the matrix factorization algorithm that can work on a single rating to predict individual criterion ratings to develop a complete multi-criteria recommender based on Adaline. Furthermore, as the scope of recommender systems covers many application domains like the domain of technology enhanced learning and e-commerce, investigating the effect of neural networks to improve their accuracies is a good direction for future research.

References

[1] Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender Systems Handbook* (pp. 1-34). Springer US.

[2] Mahmood, T., & Ricci, F. (2009, June). Improving recommender systems with adaptive conversational strategies, In *Proceedings of the 20th ACM conference on Hypertext and hypermedia* (pp. 73-82).

[3] Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11), 10059-10072.

[4] , Beam, Michael A (2014). Automating the News How Personalized News Recommender System Design Choices Impact News Reception. *Communication Research 41(8)* Sage Publications. pp. 1019–1041.

[5] Hassan, M., & Hamada, M. (2016). Recommending Learning Peers for Collaborative Learning Through Social Network Sites. *IEEE ISMS, Intelligent Systems, Modeling and Simulation.*

[6] Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems, 22(3)*, 48-55.

[7] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering, 17(6)*, 734-749.

[8] Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web, 10(4),* 415-441.

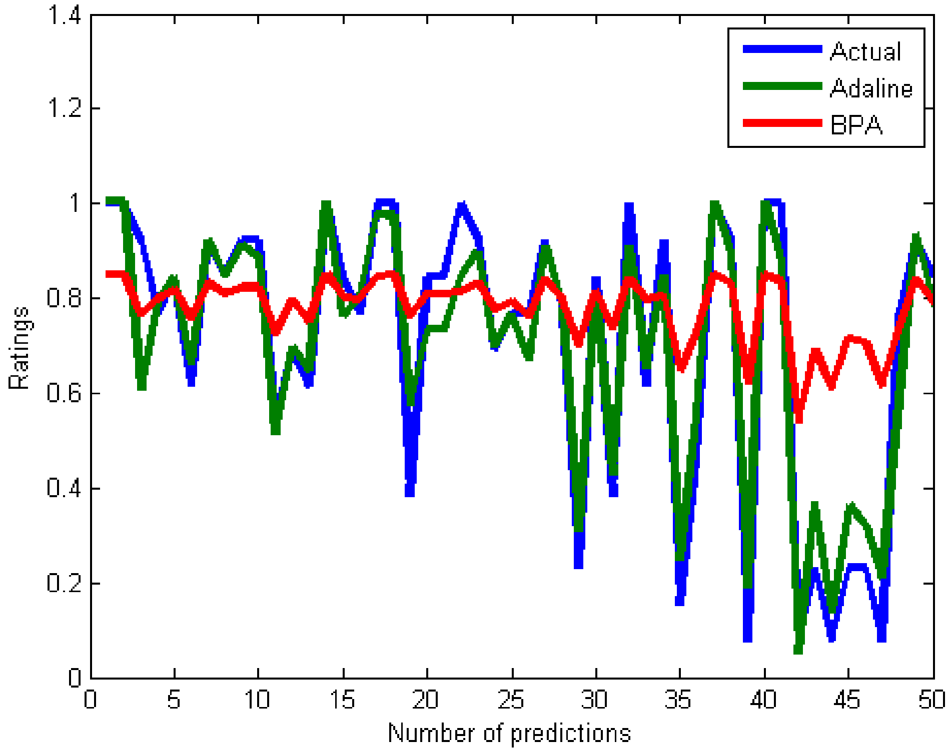

Figure 3: Curve of Actual and some testing results

[9] Adomavicius, G., Manouselis, N., & Kwon, Y. (2015). Multi-criteria recommender systems. *In Recommender systems handbook*, Springer US. (pp. 854–887).

[10] Lakiotaki, K., Matsatsinis, N. F., & Tsoukias, A. (2011). Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems, 26(2)*, 64-76.

[11] Palanivel, K., & Sivakumar, R. (2011). A study on collaborative recommender system using fuzzy-multicriteria approaches. *International Journal of Business Information Systems, 7(4)*, 419-439.

[12] Jannach, D., Gedikli, F., Karakaya, Z., & Juwig, O. (2012). *Recommending hotels based on multi-dimensional customer ratings.* na.

[13] Jannach, D., Karakaya, Z., & Gedikli, F. (2012, June). Accuracy improvements for multi-criteria recommender systems. *In Proceedings of the 13th ACM Conference on Electronic Commerce* (pp. 674-689). ACM.

[14] Ning, X., Desrosiers, C., & Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods.*In Recommender systems handbook* Springer US.(pp. 37-76).

[15] Graupe, D. (2013). *Principles of artificial neural networks (Vol. 7)*. World Scientific.

[16] Wasserman, P. D. (1993). *Advanced methods in neural computing.* John Wiley & Sons, Inc...

[17] Souza, A. M., & Soares, F. M. (2016). Neural network programming with Java. Packt Publishing Ltd.

[18] Kendal, S. (2009). *Object Oriented Programming using Java*. Bookboon.