JOURNAL of COMPUTER SCIENCES INSTITUTE

JCSI 13 (2019) 267-271
Wysłane: 2019-10-20
Przyjęte: 2019-10-28

Machine Learning as a method of adapting offers to the clients

Jacek Bielecki*, Oskar Ceglarski*, Maria Skublewska-Paszkowska
Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. Recommendation systems are class of information filter applications whose main goal is to provide personalized recommendations. The main goal of the research was to compare two ways of creating personalized recommendations. The recommendation system was built on the basis of a content-based cognitive filtering method and on the basis of a collaborative filtering method based on user ratings. The conclusions of the research show the advantages and disadvantages of both methods.

Keywords:recommender system; collaborative filtering; cognitive filtering; machine learning

*Corresponding author.

E-mail addresses: jacek.bielecki@pollub.edu.pl, oskar.ceglarski@pollub.edu.pl

Uczenie maszynowe jako metoda dostosowywania ofert do klientów

Jacek Bielecki*, Oskar Ceglarski*
Politechnika Lubelska, Katedra Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Systemy rekomendacji to aplikacje filtrujące dane, których głównym zadaniem jest dostarczanie spersonalizowanych rekomendacji produktów. Celem badań było dokonanie analizy i porównania dwóch metod uczenia maszynowego wykorzystywanych do generowania rekomendacji. System rekomendacji zbudowano na podstawie metody filtrowania kognitywnego opartej o treści oraz na podstawie metody filtrowania kolaboratywnego opartej o oceny użytkowników. Wnioski z przeprowadzonych badań pokazują wady i zalety obu metod.

Słowa kluczowe: System rekomendacji; filtrowanie kolaboratywne; filtrowanie kognitywne; uczenie maszynowe

*Autor do korespondencji.

Adresy e-mail: jacek.bielecki@pollub.edu.pl, oskar.ceglarski@pollub.edu.pl

1. Introduction

Recommendation systems (RS) are class of information filter applications whose main goal is to provide personalized recommendations of products, content and services to users. A recommendation system for an e-commerce site helps users to find products, such as movies, songs, books, gadgets, applications and restaurants that fit their personal preferences and needs [1]. Recommendation systems enhance e-commerce sales by converting browsers into buyers, exposing customers to new products, increasing cross-selling by suggesting additional products, building customer loyalty, increasing customers satisfaction based on their purchasing experience, and increasing the likelihood of repeat visits by satisfied customers. Each of these can be translated into increased sales and higher revenue [1]. In the age of e-commerce, it is important for companies to develop web-based marketing strategies such as product bundling to increase revenue. The e-commerce industry predominantly uses various machine learning models for product recommendations and analyzing a customer's behavioral patterns, which play a crucial role in exposing customers to new products based on their online behaviour [2]. Psychology studies show that if customers are shown products suited to their personality type or complementing their lifestyle, the chances of buying them grows considerably [2].

The most widely used filtering algorithms presented in the literature for the recommendation task are: collaborative filtering, demographic filtering, content-based filtering, and hybrid filtering [3]. The content-based method mostly takes into account the implicit rating by text mining process and makes a recommendation, whereas collaborative filtering considers only explicit ratings of users [4]. Furthermore, there are two types of collaborative filtering techniques frequently used in the recommendation system domain such as model-based and memory-based collaborative filtering. The model-based method develops a user model utilizing ratings of each user to evaluate the expected value of unrated items. On the other hand, memory-based method utilizes similarity measure computed from the explicit user rating to identify neighborhoods and perform prediction [1][4].

The content-based filtering makes recommendations based on user choices made in the past (e.g., in a web-based e-commerce RS, if the user has purchased comedy films in the past, the RS will likely recommend a newly released comedy that the user has not yet purchased on this website). The content-based filtering also generates recommendations using the content from objects intended for recommendation; therefore, specific content can be analyzed such as text, images, and sound [7]. The transformation of content described using a human-understandable language requires transformation into a machine-understandable language. This process is possible by using eg. Natural Language Toolkit [8]. These tools allow to save products in a multidimensional matrix. It is possible to determinethe similarity between them using mathematical functions [8]. Every recommendation method which based on users' profiles needs to create multidimensional matrices describing the user in a certain way. For the content based method, the attributes of the users' profiles are the movies that they rated [7]. The content-based approaches the focus on measuring the functional similarities between the content of services and user queries using keyword search or semantics-aware search. Keyword search methods usually have many limitations due to the insufficiencies in identifying semantically relevant keywords. Semantics-aware search methods can be further divided to two subgroups: logical ones based on ontologies and non-logical ones based on latent factor models (also called topic models) such as LDA (Latent Dirichlet Allocation) [9]. Logical semantics-aware methods require well-defined ontologies and semantic annotation of services and user queries, which makes them hard to apply; while the non-logical methods are generally not very effective due to the coarse-grained semantics captured by topic models [9].

Similarity measure in the recommender system is the statistical measure of how two users and items are related to each other. There are several traditional similarity metrics such as Cosine(COS), Pearson's Correlation (COR), Constrained Pearson's Correlation (CPC), Mean Squared Difference (MSD), Jaccard, JMSD etc. [10]. Cosine similarity measures the angle between two rated vectors where the smaller angle indicates greater similarity and higher angle show lesser similarity [6][11]. The cosine similarity could by computed by formula:

$$Sim(u,v)^{cos} = \frac{\sum_{i\in I(u,v)} R(u,i)\cdot R(v,i)}{\sqrt{\sum_{i\in I(u,v)} R(u,i)^2}\cdot\sqrt{\sum_{i\in I(u,v)} R(v,i)^2}} \quad (1)$$

where $R_{(u,i)}$ is the rating of the item i given by user u and $I_{(u,v)}$ is the number of co-rated items of users u and v [11, 12]. The range of cosine similarity is 0 to 1, where higher value signifies the closest similarity between users u and v.

The collaborative filtering (CF) approach is considered one of the most popular and effective techniques for building recommender systems [5]. The basic idea is to try to predict the user's opinion about different items and recommend the "best" items, using the user's previous preferences and the opinions of other similar users [1]. Collaborative Filtering allows users to provide ratings about a set of elements in such a way that when enough information is stored on the system, recommendations can be made to each user based on information provided by other users that are thought to have the most in common with them [1]. There are two types of collaborative filtering techniques frequently used in the recommendation system domain such as model-based and memory-based collaborative filtering. Model-based method develops a user model utilizing ratings of each user to evaluate the expected value of unrated items [5]. On the other hand, memory-based method utilizes similarity measure computed from the explicit user rating to identify neighbourhoods and perform prediction [5]. The traditional CF techniques are said to be memory-based because the original ratings database is used directly for generating the recommendations or making the predictions [6]. On the other hand, model-based approaches use ratings database to learn a predictive model which can be used to predict ratings of users for new items. Memory-based CF methods can be further divided into two groups, namely user-based and item-based algorithms [12]. The user-based algorithms look for users (also called neighbours) similar to the active user, and calculate a predicted rating as a weighted average of the neighbor's ratings on the active item. On the other hand, item-based algorithms look for similar items for an active user [12].

In 2005, Lemire and Maclachlan proposed Slope One family of algorithms to make the CF prediction faster than memory-based algorithms [13]. It has been shown that the Slope One is reasonably accurate despite its simplicity, efficiency, easiness to implement, updatability and scalability [12]. However, if there are no users or only a few users have rated the active item, the accuracy of the algorithm will decrease considerably [13][14]. Slope one algorithm adopts an easy but effective concept as a simple linear regression model to predict ratings. Its original idea was on the basis of what the authors call popularity differential between users and items. In general, the problem is to find functions of the form $f(x) = x + b$ where b is a constant and x is a variable representing rating values [12][13]. The Slope One algorithm is a typical item-based CF and mainly considers the users rating the active item and the other items rated by the active user [14]. It uses these ratings of the users to predict the rating of the active item. The Slope one could also be user-based CF and in this case it uses users ratings of the product to predict the ratings for other products [14]. In the basic Slope One, the constant b is defined as the average difference between each item and the item to be predicted; computing among the users that have rated both items [12]. This average deviation for two items i and j is calculated as:

$$dev_{j,i} = \frac{1}{|U_{i,j}|}\sum_{\in U_{i,j}}\left(r_{u,j} - r_{u,i}\right) \quad (2)$$

From every co-rated item i, a prediction for item j of user u can be obtained as $dev_{j,i} + r_{u,i}$, where $r_{u,i}$ represents the rating value given by user u to item i [12, 13]. A simple approach for combining these individual predictions is to compute the average over all co-rated items as:

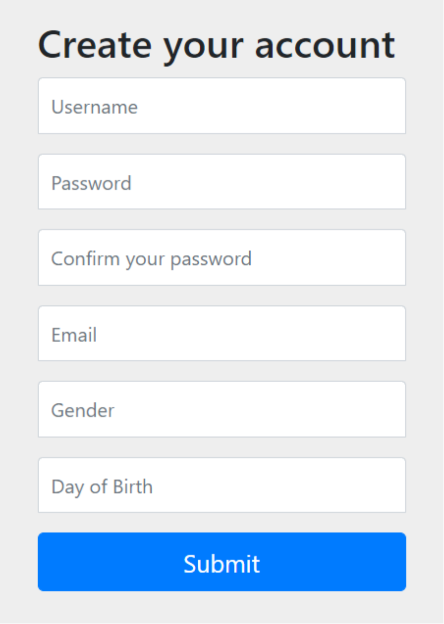
$$p_{u,j} = \frac{1}{|R_{u,j}|}\sum_{i\in R_{u,j}}\left(dev_{j,i} + r_{u,i}\right) \quad (3)$$

where $R_{u,j} = \{i\in Iu : |U_{i,j}| > 0\}$ is the set of relevant items [12][13].

However, current Slope-one based algorithms are all designed for static datasets, which are contradictory to real situations where dynamic datasets are mostly involved [15][16]. Note that in real applications, data increments can arrive at every millisecond, making a target dataset constantly change. Therefore, incremental recommenders which are able to address such data dynamics are greatly desired [16][17]. In this research, this problem was resolved by computing a prediction again when there was new ratings added to the dataset.

2. Research method

The research was made on the implemented movie store system, which allows user to watch movies and rate them. The movie store has been named "Intelligent Movie Store" and it bases on MovieLens latest datasets. The research was carried out on the group of 100 people of different age and different gender. Everyone, who participated in the research had to register their own account in movie store system. Account registration required entering all the necessary demographic user data that was used to create recommendations. Each participant of the research had to give his gender, age and country of residence. The registration form to the Intelligent Movie Store system was shown on Figure 1.



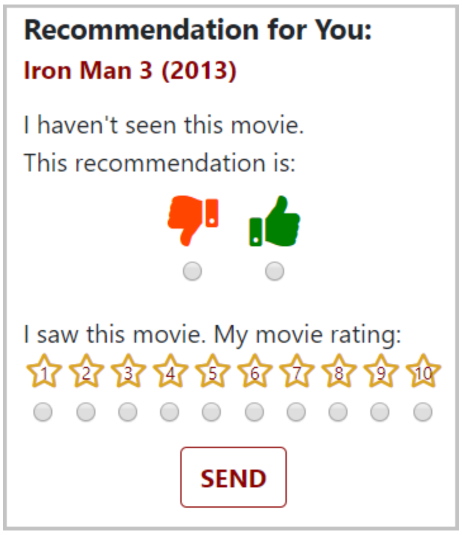
Create your account

Fig. 1. Registration form to Intelligent Movie Store system

After registration, each participant of the research has to indicate about Wight movies known to him/her movies which he/she likes and rate them. This part of the research was necessary to create recommendations based on collaborative filtering by users.

In the next step of the research, users have been split into two equal groups. Each one consisted of 50 people. Each group of people got the movie recommendations generated by different methods. The first one was getting the recommendations generated by content based method. First the NTLK tools have been used to save all movies in the multidimensional matrix. Then the cosine similarity method, which is described by formula (1), was used to compute the similarity between movies. This approach allowed to find movies which were similar to best rated movies and then recommend them. The second group of people had the recommendations generated by collaborative filtering method. In this case the Slope one algorithm was used and the predictions were computed using the (3) formula.

Every participant of the study got about 8 movie recommendations. Analysing the effectiveness of the recommendation system was based on user's responsiveness to the recommended products. Every recommendation could be rated by the user in positive or negative way. User could also ignore the recommendation. Every user's integration with the recommendation have their own numeric equivalent which means a specific level of user's responsiveness to this recommendation. For example, when user clicked on the recommendation to rate it positively - the responsibility status of that recommendation got status "2" in numeric equivalent. But when user rated low the recommendation, it got status "-2". In case when user got movie recommendation that is known to him, he could rate this movie on a scale of 0 to 10. When the user's rate was higher than 5, that recommendation got status "1" and when the rate was lower or equal to 5 the recommendation got status "-1". The example of movie recommendation with all possible response use cases is shown on Figure 2.



Recommendation for You:

Iron Man 3 (2013)

I haven't seen this movie.

This recommendation is:

I saw this movie. My movie rating:

Fig. 2. Example movie recommendation in Intelligen Movie Store system

Such a grading scale allows to determine if the recommendations were well matched to the users preferences or not.

3. Results

In the Table 1. the percentage distribution of movie ratings was shown. This ratings were collected in the first stage of research, before the users got any recommendations. There separated data is shown for both group of participants of research which have got recommendation generated by different methods in the next stage of research.

Table 1. Percentage distribution of movie ratings from the first stage of research work

	Method	
Rating	CB	CF
1	0.62%	0.25%
2	0.62%	1.24%
3	1.24%	3.72%
4	3.30%	4.71%
5	4.74%	2.98%
6	12.78%	10.42%
7	23.51%	22.83%
8	31.55%	26.55%
9	18.14%	22.08%
10	3.51%	5.21%

These data were grouped for both methods and then represented in the graph to show the normal distribution of movie ratings. The percentage distribution of each movie rating in the system is shown on Picture 3.

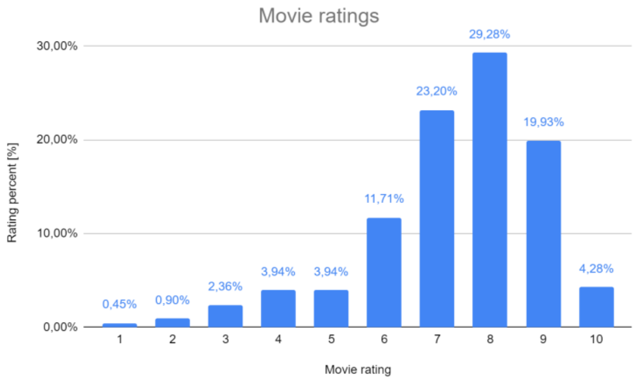


Fig. 3. The percentage distribution of each movie rating in the system

The content based method have got 43.42% of recommendations with status "2" and 30.48% of recommendations with status "1". On the other hand, the collaborative filtering method got 47.60% of recommendations with status "2" and 8% of recommendations with status "1". The percentage distribution of recommendations with different statuses is presented in Table 2.

Table 2. The percentage distribution of recommendations with different statuses

	Method	
Status	CB	CF
-2	25.00%	40.80%
-1	1.67%	2.80%
0	1.43%	0.80%
1	30.48%	8.00%
2	41.43%	47.60%

The percentage distribution of recommendations with different statuses have been also represented in graph which was shown on Figure 4.

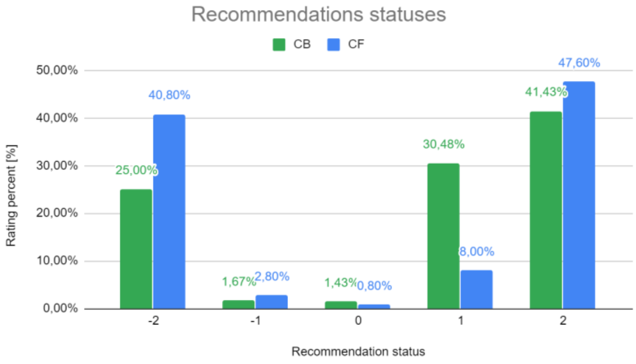


Fig. 4. The percentage distribution of recommendations with different statuses

The content based method got the effectiveness in level of 71.9% of all positive rated recommendations. A positively rated recommendation is a recommendation with the status 1 or 2. From the other side, the collaborative filtering method got the effectiveness only in level of 55.6% of positive rated recommendations. The ratio of positively rated recommendations to negatively rated ones according to both methods are shown on Figure 5.

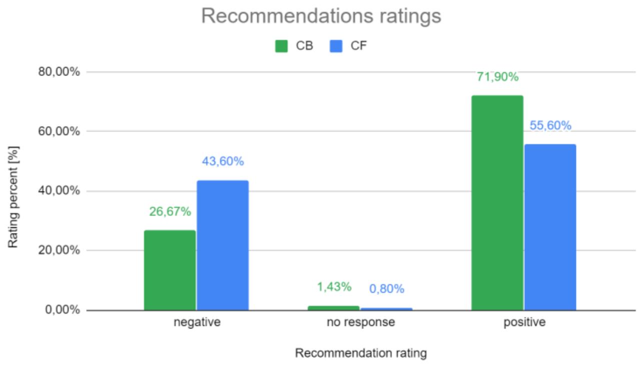


Fig. 5. The ratio of positively rated recommendations to negatively rated ones according to both methods

There were 1.43% of non rated recommendations for content based method and 0.80% of non rated recommendations for collaborative filtering method and this also have been shown on Picture 5.

4. Discussion of results and conclusions

The normal distribution of movie ratings (Figure 3) could be shifted to the right because of fact that users have selected and rated movies which were well known to them. Also the fact that in database of movies used In the research was based on the 9000 most popular movies according to MovieLens lab. It means that in the Intelligent Movie Store database were only well known, popular and interesting movies.

The content based method got the effectiveness in level of 71.9% of all positive rated recommendations and rom the other side, the collaborative filtering method got the effectiveness only in level of 55.6% of positive rated recommendations. There is some disparities in effectiveness of different recommending methods and this has been shown in Figure 5. Such a disparities in recommendation effectiveness results arise from the problem with a cold start, which is related to collaborative filtering method because of fact that it needs a lot of data about user's ratings of products to achieve better results [18][19][20]. A solution to this problem could be creating a recommendation system that is using different types of deep learning methods in case of different data access status. Eg. on the starting stage of movie store system work, the content based method could be used to avoid the cold start problem. In case when a lot of data with users movie ratings have been collected, the recommending method could be switched to collaborative filtering that should be more effective [18][20].

Bibliography

[1] M. Beladev, L. Rokach, B. Shapira, Recommender systems for product bundling, Knowledge-Based Systems, 1 Nov. 2016.

[2] A. Marwade, N. Kumar, S. Mundada, Augmenting E-Commerce Product Recommendations by Analyzing Customer Personality, Conference: 9th International Conference on Computational Intelligence and Communication Networks (CICN); 16-17 Sep. 2017.

[3] J. Bobadilla, Recommender systems survey, Knowledge-Based Systems, 46 (2013), 109-132.

[4] S. Jiang, Q. Xueming, S. Jialie, F. Yun, Author topic model-based collaborative filtering for personalized POI recommendations, IEEE transactions on multimedia 17:6 (2015), 907-918.

[5] R. Burke. Hybrid recommender systems: survey and experiments, UMUAI, 12 (4) (2002), 331-370.

[6] S. Bag, SK. Kumar, MK. Tiwari, An efficient recommendation generation using relevant Jaccard similarity, Information Sciences, May 2019.

[7] D. McIlwraith, M. Haralambos, B. Dmitry, Inteligentna sieć, Algorytmy przyszłości. Helion, Gliwice, 2017.

[8] Natural Language Toolkit, https://www.nltk.org/api/nltk.html.

[9] F. Xie , J. Wang , R. Xiong , N. Zhang, An Integrated Service Recommendation Approach for Service-Based System Development, Expert Systems With Applications, 2019.

[10] B. K. Patra, R. Launonen, V. Ollikainen, S. Nandi, A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowledge-Based Syst. 82 (2015) 163–177.

[11] H. J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Inf. Sci. (Ny). 178 (2008) 37–51.

[12] M. Saeed, E.G. Mansoori, A new slope one based recommendation algorithm using virtual predictive items, Journal of Intelligent Information Systems, June 2018, Volume 50, Issue 3, pp 527–547

[13] D. Lemire, A. Maclachlan, Slope One predictors for online rating-based collaborative filtering, SDM. SIAM, 2005 (Vol. 5 pp. 1–5)

[14] W. Yongqiang, Y. Liang, C. Bing, Learning to Recommend Based on Slope One Strategy, Web Technologies and Applications. Proceedings of the 14th Asia-Pacific Web Conference, APWeb 2012 pp: 537-4.

[15] QX. Wang, X. Luo, Y. Li, Incremental Slope-one recommenders, Neurocomputing, Volume 272, Journal of Computer Sciences Institute, 10 January 2018, pp 606-618

[16] X. Luo., Y.-N. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, Knowl. Based Syst., 27 (2012), pp. 271-280

[17] X. Luo., Y.-N. Xia, Q. Zhu, Y. Li, Boosting the K-nearest-neighborhood based incremental collaborative filtering, Knowl. Based Syst., 53 (2013), pp. 90-99

[18] Nguyen P., Wang J., Kalousis A.: Factorizing LambdaMART for cold start recommendations. Machine Learning, 21 July 2016

[19] T. Schreiner, A. Rese, D. Baie, Multichannel personalization: Identifying consumer preferences for product recommendations in advertisements across different media channels, Journal of Retailing and Consumer Services, May 2019

[20] A. Fiasconaro, M. Tumminello, V. Nicosia, V. Latora, R. N. Mantegna, Hybrid recommendation methods in complex networks. American Physical Society, 14 July 2015.