Analysis of Recommendation System Methods for Accuracy of Predicted Estimates

Nataliya Boyko, Petro Telishevskyi and Beata Kushka

Lviv Polytechnic National University, Profesorska Street 1, Lviv, 79013, Ukraine

Abstract

The Internet plays a major role in people's lives today. In the network, people communicate with friends, meet new people, develop themselves, buy goods and services, and spend time on entertainment, namely, watching movies, listening to music and so on. Because there is enough content on the Internet, it means that it is difficult for people to physically choose what they want at the moment. Therefore, web services use different levels of referral systems. Because, recommendation systems help us choose from a thousand not a lot of content that interests us, but what is not interesting to reject. Recommendation systems have been implemented for a long time, but have only recently been developed and applied. Namely, with the active development of the Internet. The most successful recommendation systems are systems based on collaborative filtering. Investigate the methods used in collaborative filtering, namely, User-based, Item-based and SVD. Conducting experimental studies on the data using methods for recommendation. A comparative characterization between the two methods after the experiments.

Keywords 1

Data Mining, Recommended System, Algorithms, SVD, method, Item-Based Method, NMF, RMSE

1. Introduction

One of the manifestations of information uncertainty is uncertainty caused by data gaps. The objective characteristics of certain processes can be changed or even distorted due to the loss of some data during their receipt, transmission or storage [3,8]. There is a need to recover such missed data and, importantly, to select the algorithms by which they will be recovered, because incorrect or insufficiently reliable recovery can cause more damage than the data gaps themselves [1, 5].

There are algorithms that allow you to process gaps with the necessary information, such as the Hot Deck method, the Barlet method, the Resampling algorithms, Zet, Zetbraid, EM estimation, regression modeling and value prediction. A feature of these algorithms is the filling of gaps with values that are selected by the algorithm [2, 4].

Recommendation systems are systems that try to solve the problem of information reloading on the Internet with the help of classification techniques and information retrieval. Using various techniques, they are created to search and recommend to users information that will be of interest to them.

These systems are widely used today in marketing, social networking and entertainment. Corporations use referral systems to increase traffic to their site as well as increase sales. For example, here are the statistics of well-known companies:

- In Netflix, 2/3 of the movies watched by users have been offered by the system.
- Google has improved (Click-through rate, CTR) by 38 percent.

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: nataliya.i.boyko@lpnu.ua (N. Boyko); petro.telishevskyi.kn.2017@lpnu.ua (P. Telishevskyi); beatakushka@yahoo.fr (B. Kushka)
ORCID: 0000-0002-6962-9363 (N. Boyko); 0000-0002-6187-740X (P. Telishevskyi); 0000-0002-4080-4607 (B. Kushka)

- Amazon sells 35% of all products with recommendations.

The most popular are systems that work on the model of explicit feedback (explicitfeedback), their main essence is that users leave reviews about a product or service and based on these reviews, recommendations are built in this work, we will explore a different approach, namely implicitfeedback. This model allows based on user behavior to predict the user's attitude to a particular product or service. As a result, we can automatically generate a content rating and use it for other users.

A recommended system is a system that recommends content to us among the vast amount of information from our needs. Users who use the resource, where there are recommendations, determine their interests using estimates [6, 10].

To create resources for watching videos, to store preferences, recommendation systems use user profiles, because these profiles store an estimate of the content of the resource. When a user evaluates, new recommendations for the user are calculated and compared with his profile [3, 12].

Ratings can be explicit or implicit. The evaluation that the user makes on the content that interests him is called explicit. And when the benefits are made on purchases, on the pages to which users went, it is an implicit rating.

Most recommendation systems use one of three approaches:

- Collaborative filtering;
- Content-based filtering;
- Hybrid filtration.

2. Methods

To date, there are many technologies and tools for analysis, classification. These include machine learning, BigData, Natural Language Processing (NLP), and referral systems. In many cases, such systems are created to predict and analyze user behavior.

Recommendation systems are designed to simplify and improve, speed up the user's search for the necessary content. They are very important, because they allow the user to interact only with the content that may interest him, and therefore increase the efficiency of the system.

Wang P. said that: "Recommendation systems are systems that aim to select specific objects that meet user requirements, where each of these objects is stored in computer memory and is characterized by a set of attributes."

Using specially collected information, recommendation systems can predict whether the user is interested in this content, rank. Or choose a specific set of N items that may be of interest to the user.

From these statements there are two problems which solve systems of recommendations. The first problem is the problem of predicting when it is necessary to predict whether this content will be interesting. As well as the problem (top-N) of choosing a set of data that might interest the user. This helps companies increase profits by reducing purchases of goods that will not be sold any time soon.

Modern recommendation systems provide high forecasting accuracy, but only if there is sufficient information about the user and his preferences. When a user does not provide enough information about himself, the system is unable to make the correct prediction

Lack of sufficient information can lead to the following problems.

The problem of similarity arises when it is difficult to determine the similarity between users because the number of features about the user is less than necessary for a quality recommendation. Researcher Yu also made an interesting conclusion about this problem, that in fact the content suffers from this problem no less than users. Because they typically use fewer features which is often not enough for popular algorithms such as feature-based, content-based algorithms.

A Cold Start Problem occurs when a newly registered user has not yet rated any content and the system has no information about it.

The problem of changing the taste. For example, today the buyer is looking for a product for himself and tomorrow he is looking for a gift for his mother, so the user may have incorrect ideas about his tastes.

The problem of new things happens when the system does not know about specific content because no one has evaluated it yet.

Unpredictable things depend on the user's preferences, especially in music, it is difficult for the system to evaluate such things, because everyone can have their own reaction to this content.

Basic algorithms in recommendation systems:

- Content based

This approach recommends that users use similar things to those they have liked in the past. Keyword search is often used to find similar things. In this case, the company Pandora music online service in the framework of MUSIC GENOME PROJECT to create for each of its songs a vector consisting of approximately 450 features. This allowed the use of more standard machine learning approaches that gave the output the likelihood that the user would like the music.

- Demographic based

Another approach is when there is a lot of information about users such as Facebook, Linkedin. Therefore, you can use this information to set the direction of recommendations regardless of previous user behavior.

- Coloborative filter

This method is based on user-content interaction. It only analyzes the rating and ignores information about the content or user. The key idea is that such users like similar things and if users have watched the same movies, there is a high probability that in the future they will also get the same recommendation. All you need is some kind of assessment that arises from the interaction of user and content.

There are two types of such data:

- Explicit - this is when the available rating matches the rating or preferences. Popular services offer users to evaluate things that interest them in different ways. For example Netflix, Youtube use the like / dislike system. At the same time, Amazon and Aliexpress trading platforms use a system of stars, the maximum number of which is 10.
- Implicit is the process by which a service collects information about a user's interaction with content with information about clicks, views, or purchases. It also takes into account the time: how much the user spent on the page and many other factors.

A striking example is online cinemas, where each film has a certain rating. It allows you to see in numerical terms the user's preferences for the film. The only problem is that some users do not leave any feedback, so the recommendation system may be inaccurate for this and implicit evaluation is used because it allows you to increase the amount of information about the user

Currently, many commercial pages on the Internet have their own recommendation pages. Ahead of others are sites such as Netflix, Amazon, Google, Linkedin. Many researchers also presented their versions of recommendation systems.

Today, there is a lot of research on referral systems, but they all face the problem of insufficient data. And implicit evaluation allows you to improve these results, because it does not require any additional action from the user.

There are several major problems when working with referral systems. Problems (1-2) were encountered at the beginning of the development of recommendation systems. Now they are successfully solved within the framework of modern algorithms.

1. The problem of similarity. A hybrid algorithm is used to solve this problem. Due to which we do not always need information about the user to give an accurate and adequate recommendation.
2. The problem of lack of data. It is solved by using implicit estimation, and also colloborative algorithm. This is usually sufficient for recommendations with high accuracy.

The main problem is the problem of cold start.

3. Cold start problem. Consider a problem with new content that has not yet received enough ratings to be recommended. This problem can be solved thanks to the content-based algorithm, hybrid algorithm. Due to certain categories that will be when mastering the content it is possible to show it along with similar content. However, not every content is labeled with enough classes or certain features to be used in recommendations. In this case, text mining uses several basic algorithms.

Here is an example based on the recommendations of websites.

1. Preprocessing stage

In this case, highlight the keywords, highlight the theme of the page, discard the stop words.

2. TF-IDF (TF — term frequency, IDF — inverse document frequency)

At this stage, the calculation is performed for each word *w* in each document *d*. $tf(w,d) = \frac{n_w d}{n_d}$,

where $n_w d$ - the total number of word entries in the document, n_d - the total number of words in the document. This method is used to discard redundant information and not store it in memory. Because in the relevance feedback algorithm, only 200 words with maximum weight will suffice.

3. Relewant feedback

This algorithm brings us to the problem. It is based on page ratings that the user likes but without overall page ratings. But it is usually used for new content, because it loses in efficiency to more traditional algorithms.

The first step is to find a TF-IDF for sites that the user likes (if the user has just started using the service, it is good practice to let him mark several sites (movies, books,…) that he visited and that he liked).

The second step is to find the similarity of the page to the user's preferences, which is calculated as a scalar product of the vectors of user weights and sites $k(u,d) = \sum_{w \in W_U} V_{ww} * tfdf(w,d)$, where W_U - words from the user profile. As a result, this algorithm allows you to explore the similarity of any page that has text to any user who has certain preferences.

The problem with the cold start for the user is that the system meets the user for the first time and there is no information about him in his memory. This problem can occur constantly. The first example is when a user has not registered and information about him is stored in cookies, but users always have the opportunity to delete their information. The system will then consider them as new users. The second problem arises when a user searches for goods for someone. For example, the user went to the site in search of a particular product and the relevant recommendations he needs only until the time of purchase. And then he will need another product. As a result, the user remains the same, but his tastes can be radically different.

The most common solution to this problem is to use geolocation. When a user has just registered or is using the service for the first time, we will show him information that is popular in his area of residence at a certain point in time. After a few likes or interactions with the content, the system will be able to show more accurate suggestions.

2.1. User-based method

A method that is based on the user liking products that have been selected by users similar to him (Formula 1) [11, 13].

$$\hat{r}_{ui} = \bar{r}_i + \frac{1}{\sum_{i' \in R(u)} |sim(i,i')|} \sum_{i' \in R(u)} sim(i,i')(r_{u,i'} - \bar{r}_{i'}). \quad (1)$$

The degree of similarity $sim(i,i')$ is calculated from the matrix of estimates *R*. The most commonly used similarity metric is the Pearson correlation and the cosine distance of the rows (columns) of the matrix (Formula 2-3).

$$sim(u,u') = \frac{\sum_{i \in R(u) \cap R(u')} (r_{u,i} - \bar{r}_u)(r_{u',i} - \bar{r}_{u'})}{\sqrt{\sum_{i \in R(u) \cap R(u')} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in R(u) \cap R(u')} (r_{u',i} - \bar{r}_{u'})^2}}. \quad (2)$$

$$sim(u,u') = \frac{\sum_{i \in R(u) \cap R(u')} (r_{u,i} r_{u',i})}{\sqrt{\sum_{i \in R(u) \cap R(u')} r_{u,i}^2 \sum_{i \in R(u) \cap R(u')} r_u}}. \quad (3)$$

2.2. Item-based method

A method that is based on the fact that the user will like products similar to those he has already chosen.

User A is characterized by objects that he has viewed or rated. For each of the selected / rated objects, m neighboring objects are defined, ie there are m most similar objects in terms of user views / ratings. When building a PC for movies, m takes values from 10 to 30. All neighboring objects are combined into a collection, which excludes objects viewed or evaluated by user A. And from the remains of the collection is built top n recommendations. Thus, in the item-based approach, all users who liked this or that object from the collection take part in creating recommendations (Formula 4) [14, 15].

$$\hat{r}_{ui} = \bar{r}_i + \frac{1}{\sum_{i' \in R(u)} |sim(i,i')|} \sum_{i' \in R(u)} sim(i,i')(r_{u,i'} - \bar{r}_{i'}). \quad (4)$$

2.3. SVD algorithm

Almost all collaborative filtering algorithms have such shortcomings as cold start, triviality of recommendation results, and so on. One of the fairly new algorithms that reduces the impact of typical collaborative filtering problems was the SVD algorithm, which was created to improve the results of conventional algorithms [16, 17].

SVD is a method of factorization of matrices, which is usually used to reduce the number of data set functions by reducing the size of space from *n* to *k*, where *k* <*n*. However, for the purposes of recommendation systems, we are only interested in matrix factorization, where parts retain the same dimension. Matrix factorization is performed on the matrix of user position ratings (Formula 5). Matrix factorization can be considered as a search for 2 matrices, the product of which is the original matrix (Formula 6) [12, 15].

$$\text{expected rating} = \hat{r}_{ui} = q_i^T p_u, \quad (5)$$

when q_i and p_i can be found so that the difference of the square errors between their product points and the known rating in the custom element matrix is minimal.

$$\text{minimum } (p,q) \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2. \quad (6)$$

2.4. NMF algorithm

NMF algorithm is a representation of the matrix *V* in the form of the product of the matrices *W* and *H*, in which all the elements of the three matrices are non-negative [9]. Let table *V* have size *m x n*. Denote by *r* the rank of the matrices *W* and *H*, as a rule *r << min (n, m)*. In contrast to the exact representation of the matrix in SVD and NMF, we obtain only an approximate equality (Formula 7) [3, 12].

$$V \approx WH. \quad (7)$$

The matrices *W* and *H* are chosen so as to minimize the loss function: $D(V,WH) \to \min$. In this case, *D* is given on the basis of Kulbak-Leibler divergence (Formula 8) [13, 17].

$$D(A,B) = \sum_{i,j} a_{ij} \log(\frac{a_{ij}}{b_{ij}}) - a_{ij} + b_{ij}. \quad (8)$$

3. Experiments

Experiments use the MovieLense100k open access data set. This data set includes detailed information about users, movies and information about users and their ratings on movies.

In this paper, we will use certain fields from this data set:

- User ID;
- Film ID;
- User rating for this movie.

This data set contains information about 1000 users, 1700 movies and 100000 user ratings for these movies. Table 1 will show the data structure.

Table 1

Data structure

user_id	item_id	rating
196	242	3
186	302	3
22	377	1
244	51	2
…	…	…

Next, divide the data into 2 parts:

- Test data (25%);
- Data for calculation (75%).

This data operation is required to predict the estimates of the elements based on the data to be calculated. When performing this action, then make a comparison with the test scores from the obtained estimates. This comparison will give us a better understanding of the efficiency of the algorithm used.

There are different evaluation indicators for evaluating the methods of the recommendation system. One of the most popular estimates is the root of the root mean square error (RMSE) (Formula 9). This is necessary for the accuracy of predicted estimates [6, 10].

$$RMSE = \sqrt{\frac{1}{N}\sum (r_i - \hat{r}_i)^2}, \quad (9)$$

where N – total number of assessments;

r_i – projected assessment;

$\hat{r}_i$ – the assessment that was made.

At the end of the experiments, the results of the image on the graph, where the x-axis will describe the test number, and the y-axis - the RMSE value for a method.

For experiments, use the methods mentioned in the previous section:

- User-based Collaborating filter(CF), Item-based CF;
- SVD, NMF.

The results are shown in Table 2 for these methods that worked with the ‘MovieLense 100k’ data set. The table records the values of five times the cross-calibration, and then shows the average RMSE value for each algorithm.

Table 2

The results of the methods

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average RMSE
RMSE User-based	1.0084	1.0052	1.0137	1.0194	1.0080	1.010936
RMSE Item-based	1.0431	1.0559	1.0393	1.0266	1.0412	1.041222
RMSE SVD	0.9373	0.9399	0.9369	0.9359	0.9333	0.936784
FRMSE NMF	0.9657	0.9664	0.9585	0.9630	0.9688	0.964507

For a comparative result, we show the results of all methods in Figure 1.

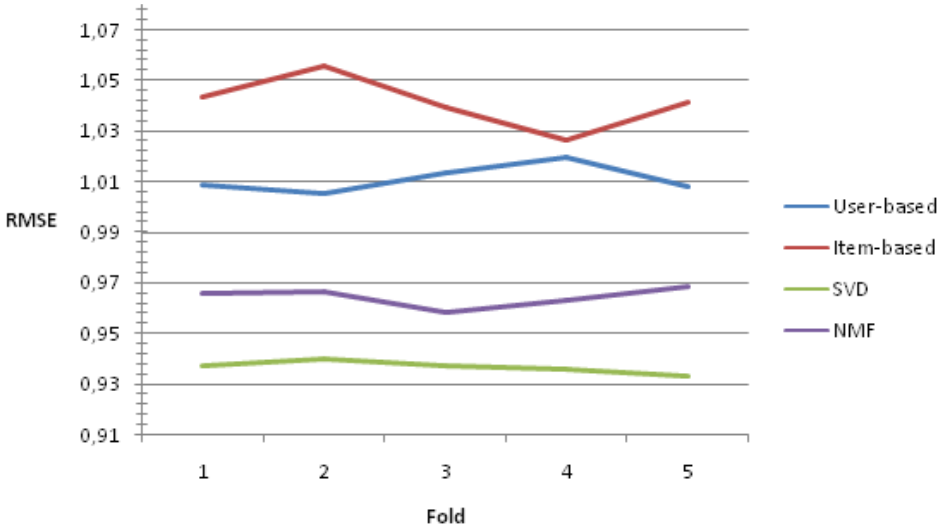

Figure 1: Results of all methods

As can be seen from the graph, in all tests the accuracy relative to RMSE is the best in the SVD algorithm. During all tests for this algorithm, the difference in accuracy is not large compared to the rest of the algorithms.

The average results show the NMF method, although the results of this algorithm are similar to the SVD algorithm.

The average results show the NMF method, although the results of this algorithm are similar to the SVD algorithm. This is determined by the fact that the presented algorithms use the decomposition of the matrix.

The worst results are User-base and Item-based, due to the fact that we use Pearson's correlation to find similar data. Sometimes the data does not correlate with the required data, which gives worse accuracy.

We will also conduct experiments for User-based and Item-based using different types of correlation. Namely, we use Pearson correlation, Cosine, Mean Squared Difference (MSD), Pearson baseline. All checks will be performed on our data from MovieLense 100K. Mean Abslolute Error was used to show which of the algorithms for the User-based and Item-based methods would be the best.

Table 3 and Figure 2 will show the results of experiments for User-based. The table records the values of five times the cross-calibration, and then shows the average value of the absolute error for each measure of similarity in the User-based method.

Table 3

Relative performance of different similarities for User-based

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average MAE
Pearson	0.8096	0.8002	0.7985	0.8023	0.8039	0.802902
MSD	0.7800	0.7736	0.7797	0.7692	0.7674	0.77396987
Cosine	0.8051	0.8050	0.8038	0.8068	0.8032	0.8047759
Pearson baseline	0.7919	0.7934	0.7917	0.7843	0.7971	0.7916556

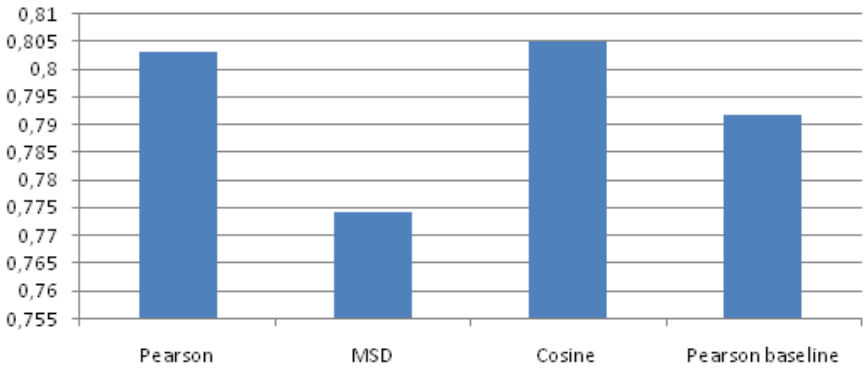

Figure 2: Influence of similarity calculation measure on User-based (CF)

Figure 2 shows that after using MSD, we get much better results. Therefore, this measure of similarity should be used, not correlation or cosine, because in such cases for the recommendation system using the User-based method with MSD the lowest values will be obtained, and they negatively affect the correctness of the recommendations. Also, the execution time in MSD is the lowest and this is shown in Table 4.

Table 4
User-based execution time for different degrees of similarity

Measures of similarity	Time, seconds
Pearson	27.8892565
MSD	21.47068977
Cosine	24.72791886
Pearson baseline	25.5328206

Table 5 and Figure 3 show the results of the experiments for Item-based.

Table 5
Relative performance of different similarities for Item-based

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average MAE
Pearson	0.8352	0.8367	0.8362	0.8316	0.8311	0.83416659
MSD	0.7701	0.7712	0.7717	0.7719	0.7644	0.769803984
Cosine	0.8110	0.8181	0.8137	0.8065	0.8100	0.8113568084
Pearson baseline	0.7837	0.7763	0.7841	0.7836	0.7777	0.78106792

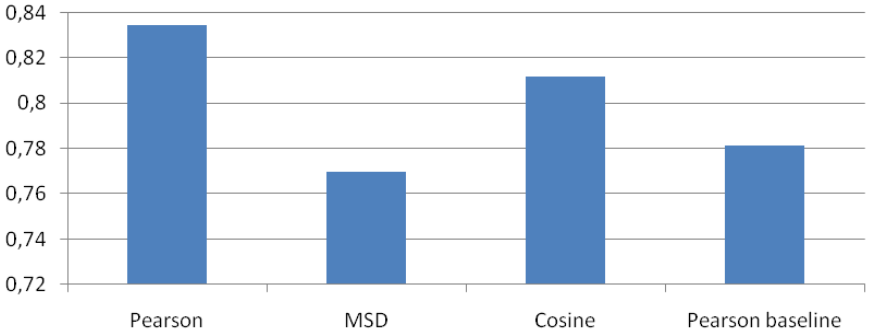

Figure 3: Influence of similarity calculation measure on Item-based (CF)

As with User-based in Item-based, a measure of MSD similarity should be used. MAE scores are the lowest. They have a positive effect on the system. Runtime is fastest in MSD and slowest in Pearson baseline. The final results are shown in Table 6.

Table 6
Item-based execution time for different similarities

Measures of similarity	Time, seconds
Pearson	33.631508
MSD	25.8418406
Cosine	32.7998433
Pearson baseline	34.0584611

3.1. SVD and NMF analysis

To understand how the SVD and NMF algorithm will behave on the proposed dataset MovieLense 100K, need training data and tests to be divided into 5 parts. Then it is necessary to determine with what accuracy for each user the Top 10 recommendations are defined. They are determined by the Formula 10.

$$\text{Precision@ k} = \frac{|\{\text{Recommende d items that are relevant}\}|}{|\{\text{Recommend ed items}\}|}. \quad (10)$$

The data shown in Table 7, where the numerical indicator is the average accuracy of the sum of the values of each user for the total number. Accuracy will be displayed as a percentage.

Table 7
Accuracy values Top 10 recommendations for users

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
SVD	60,13%	61,24%	62,33%	62,89%	63,69%
NMF	61,51%	61,75%	61,04%	61,24%	59,47%

Thus, from Table 7 results in all parameters equal to approximately 61%. Therefore, the top 10 recommendations will be true for users if you use SVD or NMF algorithms on a data set with MovieLense 100K.

4. Discussion of the results of experiments

Experiments show that SVD calculations will be better than User-based and Item-based. Because User-based and Item-based use correlation to find similar data, it doesn't always give us what we need.

The disadvantage of User-based and Item-based CFs do not fit into real scenarios and do not solve the known problem of cold start, ie when a new user or a new element enters the system.

It is said that cold start affects User and Item -based CF, ie increases RMSE for these methods.

And we can also assume that the algorithms described in this paper can give a better estimate if you increase the amount of data.

Let's list the disadvantages of correlation methods:

- Cold start problem;
- Poor forecasts for new / atypical users / facilities;
- Triviality of recommendations;
- Resource-intensive calculations. In order to make assumptions, we need to keep in mind all the estimates of all users.

And the second limitation may be the data set we used. Because the data provided by MovieLense was used for this study. And in order to see how our tonicity changes, it is desirable to use different data sets.

In order to improve the accuracy of the assessment should combine methods described methods of filtering content. This means the formation of a hybrid approach. Hybridity will consist of a

combination of two different approaches. This will avoid the disadvantages of collaborative filtering and content filtering. There are several approaches:

- Inclusion of element characteristics in collaborative filtering;
- Construction of a single model-based collaborative filtering and content filtering.

5. Conclusions

In an information-saturated world, referral systems play an important role in helping users interact only with information that is potentially of interest to them. Recommendation systems are a leading indicator of predicting user behavior in systems with a large amount of information. The systems provide the applications in which they are used with valuable customer offerings, increasing profits from business start-ups. Such a system helps the customer to provide the relevant goods, while generating revenue from the company. The system must also provide the newly added users with enough offers to meet the demand of the users. One way to increase forecasting accuracy is to encourage the user to evaluate the quality of the movies, thereby increasing the number of offers.

This study showed that the SVD algorithm provides better estimates than the described methods, which are based on correlation. It can also be said that it is advisable to use the User-based and Item-based CF methods independently, because the results of the other two algorithms are better. The described methods should be used in hybrid approaches. Further studies should take into account other parameters that affect the accuracy results.

6. References

[1] G. Tak´acs, I. Pil´aszy, B. N´emeth, D. Tikk, Matrix factorization and neighbor based algorithms for the netflix prize problem, in: Proceedings of the 2008 ACM conference on Recommender systems, ACM, 2008, pp. 267–274.

[2] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Sarwar Item-Based Collaborative Filtering Recommendation Algorithms, in: Proceedings 10th Int'l WWW Conf, 2001, pp. 285-295. https://dl.acm.org/doi/10.1145/1454008.1454049.

[3] I. Soboroff, C. Nicholas, Combining Content and Collaborationin Text Filtering, in: Proceedings of the Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering, 1999, pp. 86-91.

[4] M. T. Jones, Recommender systems. Introduction to approaches and algorithms., 2017. URL: https://www.ibm.com/developerworks/library/os-recommender1/

[5] C. Boehm, K. Kailing, H. Kriegel, P. Kroeger, Density connected clustering with local subspace preferences, in: Proceedings of the 4th IEEE Intern. conf. on data mining, IEEE Computer Society, Los Alamitos, 2004, pp. 27–34.

[6] N. Boyko, K. Boksho, Application of the Naive Bayesian Classifier in Work on Sentimental Analysis of Medical Data, in: Proceedings of The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19-21, 2020, pp. 230-239.

[7] D. Harel, Y. Koren, Clustering spatial data using random walks, in: Proceedings of the 7th ACM SIGKDD Intern. conf. on knowledge discovery and data mining, San Francisco, California, 200, pp. 281–286.

[8] D.J. Peuquet, Representations of space and time, Guilford Press, New York, NY, 2002.

[9] H.-Y. Kang, B.-J. Lim, K.-J. Li, P2P Spatial query processing by Delaunay triangulation, Lecture notes in computer science, vol. 3428, Springer/Heidelberg, 2005, pp. 136–150.

[10] M. Ankerst, M. Ester, H.-P. Kriegel, Towards an effective cooperation of the user and the computer for classification, in: Proceedings of the 6th ACM SIGKDD Intern. conf. on knowledge discovery and data mining, Boston, Massachusetts, USA, 2000, pp. 179–188.

[11] C. Zhang, Y. Murayama, Testing local spatial autocorrelation using, vol. 14, Intern. J. of Geogr. Inform. Science, 2000, pp. 681–692.

[12] N. Boyko, B. Mandych, Technologies of Object Recognition in Space for Visually Impaired People, in: Proceedings of The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19-21, 2020, pp. 338-347.
[13] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic sub-space clustering of high dimensional data, Data mining knowledge discovery, vol. 11(1), 2005, pp. 5–33.
[14] V. Estivill-Castro, I. Lee, Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram, in: Proceedings of the 9th Intern. Symp. on spatial data handling, Beijing, China, 2000, pp. 26–41.
[15] I. Turton, S. Openshaw, C. Brunsdon, Testing space time and more complex hyperspace geographical analysis tools, Innovations in GIS 7, Taylor & Francis, London, 2000, pp. 87–100.
[16] N. Boyko, N. Tkachuk, Processing of Medical Different Types of Data Using Hadoop and Java MapReduce, in: Proceedings of The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19-21, 2020, pp. 405-414.
[17] C. Aggarwal, P. Yu, Finding generalized projected clusters in high dimensional spaces, in: Proceedings of the Intern. conf. on management of data ACM SIGMOD, 2000, pp. 70–81.