DOI: 10.20535/kpi-sn.2020.3.209842
UDC 004.021

I.Yu. Kochubey*, O.S. Zhurakovska
Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine
*corresponding author: illa.kochubey1@gmail.com

MODIFIED ALGORITHM OF COLLABORATIVE FILTERING FOR FORMING USER RECOMMENDATIONS

Background. In our online life, we get a lot of information, and more and more people don`t want to rummage impassable jungle of information. Each of us wants to quickly find what is looking for. Many sites such as YouTube, Facebook and Twitter have already had recommender system and many people have used it. Recommender systems are becoming more and more popular.

Objective. The base algorithm of collaborative filtering which is used in recommender system is considered. We are trying to find bottleneck problems of base algorithm of collaborative filtering to improve it and take a gain in time.

Methods. We have analyzed the base algorithm of collaborative filtering and have found bottleneck problem. The main runtime of the algorithm is concentrated to calculate user similarity. We calculated the average rating for object in cluster with weighting factor. We use two criterions to compare the base algorithm with the modified algorithm. First criterion is the algorithm runtime. Second criterion is amount of elementary permutations we have to do to get recommendations which are provided by the base algorithm of collaborative filtering. The main factors which influence the algorithm runtime of collaborative filtering are: number of users, amount of objects and percentage of filling.

Results. The modified algorithm of collaborative filtering was compared with the base algorithm of collaborative filtering by two criteria. The difference between the results of both algorithms does not exceed 5%. The modified algorithm works faster than the base algorithm. Furthermore, with increasing the number of users or amount of objects the runtime difference will increase. The results of research are presented in graphs.

Conclusions. We have analyzed the base algorithm of collaborative filtering and methods to improve it**.** We can conclude on the feasibility of the modified algorithm of collaborative filtering from the research. The modified method gives a great gain in time. If systems start to use this modified algorithm, this can solve the problem with the runtime of the algorithm of collaborative filtering and allows giving recommendations faster than the system which uses the base algorithm.

Keywords: recommender system; collaborative filtering; modified algorithm of collaborative filtering; web directory.

Introduction

Since 2010s, popularity of recommender systems has been intensely growing [1]. Many sites such as YouTube, Facebook and Twitter have already had recommender systems [2]. Algorithms work fast and give good recommendations but over time the information in the database continues to grow and there comes a time when recommender systems can give recommendations just over a longer period of time [3]. This problem is growing up and the recommender system must be refreshed.

Most systems that suggest service offerings based on the user behavior use one of two basic approaches: collaborative filtering and content filtering [4]. In addition, recently, the principles of the above strategies have begun to be combined into a third hybrid filtering approach [5].

Content filtering offers elements based on the behavior of its users. That is, this approach uses retrospective view data.

Collaborative filtering makes suggestions based on the results of the analysis of the previous user behavior. This model is implemented based on the behavior of the user being considered, or more often, taking into account the behavior of other users of the cluster to which the user is considered. The second case is more effective [6].

In modern recommender systems, different methods are used to solve the problem of making personalized recommendations:

- item-based algorithm of collaborative filtering [7];
- neural collaborative filtering [8];
- the neighborhood approach [9];
- latent factor models [9];
- differential privacy protection [10].

In the article we have modified collaborative filtering to improve the runtime of the base algorithm.

Problem statement

In our article we find methods to improve the base algorithm of collaborative filtering. Searching and buying things online is becoming more and more popular in our life. The problem is that there are too many products and it is usually not easy for the user to choose something. Also, the web directory is not always trusted by the user. The user will have less suspicion if one sees that someone has bought a product or left a review. Therefore, the question arises as to the advisability of the recommendation sub-system. There are users who have left a product review or rated it on a ten-point scale. You need to analyze the information that users have displayed on the web directory and identify the object that is most comfortable for the user.

Substantive statement of the problem

The input data of our task are:

– B is a set of goods and services;

– b_i is an element i of set B;

– U is a set of users;

– u_j is an element j of the set U;

– i is an index of a recommended object (or, simply, object i).

The target function is

$$R = \{b_i \mid b \in B, i = \overline{1,c}\}$$

where R is a set of recommendations, and c is number of recommended objects;

Description of solution methods

In order to improve the algorithm of collaborative filtering, consider peculiarities of its routine. Then, we determine the narrow place where we can modify algorithm. There are many implementations of collaborative filtering but the main idea is that we can show in the next stages:

Stage 1. Read from the database or another place where we save our information the objects and their ratings that users put on the objects.

Stage 2. For each user from a cluster of users, to determine user similarity, for which we make recommendations, using the Pearson correlation coefficient. The similarity between sets of ratings of users u_1 and u_2 from a cluster is determined by formula

$$\mathrm{sim}(u_1,u_2) = \frac{\sum_{i=1}^{m}((u_1)_i - \overline{u_1})\cdot((u_2)_i - \overline{u_2})}{\sqrt{\sum_{i=1}^{m}((u_1)_i - \overline{u_1})\cdot\sum_{i=1}^{m}((u_2)_i - \overline{u_2})}} \qquad (1)$$

where u_1 is a set of user ratings for which we make recommendations; u_2 is a set of user ratings from cluster; m is a number of user ratings objects; $(u_1)_i$ is element i of set u_1; $(u_2)_i$ is element i of set u_2; $\overline{u_1}$ is mathematical expectation random variable u_1: $\overline{u_1} = \frac{1}{m}\sum_{i=1}^{m}(u_1)_i$; $\overline{u_2}$ is mathematical expectation random variable u_1: $\overline{u_2} = \frac{1}{m}\sum_{i=1}^{m}(u_2)_i$.

Stage 3. For each object of user u (for which we make recommendations) to calculate the measure which, after its normalization, shows how much the user may like the recommendation:

$$r_i = k \sum_{u' \in U} \mathrm{sim}(u,u')\cdot r_{u',\,i},\ (i = \overline{1,m}) \qquad (2)$$

where

$$k = \frac{1}{\sum_{u' \in U} |\,\mathrm{sim}(u,u')|} \qquad (3)$$

is a normalization factor; $\mathrm{sim}(u,u')$ is the similarity between two users calculated by formula (1); $r_{u',i}$ is a rate of object i exhibited by user u' from the cluster.

These are the main stages of the base algorithm of collaborative filtering. Of course, the measure of similarity between users can be determined differently but having evaluated the measures of similarity by the ELECTRE, a decision-making method, we can see that the Pearson correlation coefficient is the most accurate [11, 12].

Stage 4. For each measure r_i to make normalization by formula

$$rn_i = \frac{r_i}{r_{\max}} \qquad (4)$$

where $r_{\max} = \max_{r_i \in K}(r_i)$ (where K is a set of measure that shows how much the user may like the recommendation); rn_i is a normalized measure r_i.

Bottleneck problem

We have analyzed the base algorithm of collaborative filtering and have found a bottleneck problem in stage 2. The main running time of the algorithm is concentrated to calculate user similarity. So, our main goal was to move away from calculating user similarity. We calculated the average rating for an object in the cluster with a weight factor.

The modified algorithm of collaborative filtering

We show our modified algorithm of collaborative filtering using stages as in the base algorithm of collaborative filtering:

Stage 1. To read from the database or another place where we save our information the objects and their ratings that users put on the objects (the same as in stage 1 of the base algorithm).

Stage 2. We unite users in groups, where the number of users in the groups must be close or equal to $\sqrt{n}$, where n is a number of users in set U.

Stage 3. For each group, to determine the average rating of objects that users have rated by using formula

$$r_{av,i} = \frac{U_{f,i}}{U_a} \cdot A \qquad (5)$$

where U_a is a total number of users in group; $U_{f,i}$ is a number of users who rated the object;

$$A = \frac{\sum\limits_{u' \in U} r_{u',i}}{U_c}$$

where U_c is the total number of user ratings.

Stage 4. For each group, with the average rating of objects, to determine similarity with the user, for which we make recommendations using the Pearson correlation coefficient. The similarity between a user and a group is then calculated in the same way as the similarity between users. We can use formula (1) from stage 2 of the base algorithm.

Stage 5. For each object, to calculate the measure that shows how much the user may like the recommendation. We can use formula (2) from stage 3 of the base algorithm, but instead of similarity between users we take the similarity between the user and group. Then, to make normalization in the same way as in stage 4 of the base algorithm.

Numerical example of getting ratings by modified algorithm

Consider an example, in which we have 5 users and 5 objects. In Table 1, the rating of user i for object j is shown at the intersection of row i and column j.

Now, the users are united into two groups, as on stage 2 of the modified algorithm. Users 2 and 3 are in one group and users 4 and 5 are in another one. Then, we calculate the average rating of objects that users have rated by formula (5) from stage 3 of the modified algorithm. Average ratings of objects are shown in Table 2. For the 1st object in the 1st group average ratings is $\frac{1}{2}\left(\frac{5+0}{1}\right) = 2.5$.

***Table* 1.** The ratings of users

Users	Objects				
	1	2	3	4	5
1			5	2	
2	5	3			1
3		3		4	
4	2		2		2
5		4		1	

***Table* 2.** Average ratings of objects for groups

Groups	Objects				
	1	2	3	4	5
User, for which we make recommendations			5	2	
Group 1	2.5	3		2	0.5
Group 2	0.5	2	1	0.5	1

Each group can be replaced with a user with ratings from Table 2.

Then, for each user, with the average rating of objects, we calculate the similarity with the user, for which we make recommendations using the Pearson correlation coefficient by formula (1). The Pearson correlation coefficient for the user and group 1 is –0.6042. The Pearson correlation coefficient for the user and group 2 is –0.1925.

Then, for each object, we calculate the measure by formula (2) which shows how much the user may like the recommendation shown in Table 3. For object 1, the measure is

$$\frac{(2.5 \cdot |-0.6042| + 0.5 \cdot |-0.1925|)}{|-0.6042| + |-0.1925|} = \frac{1.60675}{0.7967} \approx 2.0158.$$

***Table* 3.** Value of measure (2) for objects

Objects	1	2	3	4	5
Measure that shows how much the user may like	2.0158	3	The object has already been evaluated by user 1	The object has already been evaluated by user 1	0.6208

Then, we make normalization by formula (3) as on stage 3 of the base algorithm, that is shown in Table 4.

***Table* 4.** Normalized measure (4) for objects

Objects	1	2	3	4	5
Normalized measure	0.6723	1	The object has already been evaluated by user 1	The object has already been evaluated by user 1	0.2069

Finally, the result of the modified algorithm is the ranking as follows: 2, 1, 5. So, the recommendation for user is object 2 (the highest priority), then is object 1 (medium priority), and the object 5 has the lowest priority.

Research on the relevance of the modified algorithm of collaborative filtering

We used two criteria to compare the base and modified algorithms:

1. Runtime of the algorithm.
2. The relative distance between the two algorithms (p), which shows the difference between the base algorithm and modified algorithm by formula

$$p = \frac{p_c}{p_{\max}} \cdot 100\% \qquad (6)$$

where p_c is an amount of elementary permutations we must to do with a ranking by the modified algorithm to get the ranking given by the base algorithm of collaborative filtering (or, in other words, it is a distance between two rankings); $p_{\max}$ is the maximum amount of elementary permutations between the two rankings.

We can see that the closer the value of (6) to zero, the smaller is the difference between two results of algorithms. For example, difference between two ranking (2, 3, 1) and (1, 2, 3) by formula (6) is $p = \frac{2}{3} \cdot 100\% \approx 67\%$.

The main factors which influence the runtime of algorithm are:

1. Number of users.
2. Amount of objects.
3. Percentage of filling. This factor shows how many rates have been put up by the user. For instance, our system has only 10 objects and a certain user has put up 5 rates. So, the percentage of filling will be 5/10 = 0.5.

If we want to see, how each of the factors influence our criteria, we have to choose one criterion and change it, while others are frozen as constants. Then we have to determine the boundaries of the factors.

Number of users

According to [11], we have to cluster users in group. The k-means method is one of the simplest among all clustering algorithms.

A peculiarity of the implemented algorithm consists in that we do not determine the value of the function among all users of the system, but only between users of the preformed cluster, which allows to significantly speed up the algorithm by reducing the input data for the algorithm. Another important factor is that filtering happens between such users. This means that users with the same preferences will be compared. Such actions make it possible to successfully combine it with the k-means algorithm [11]. As a result, we cannot use a large number of users in research. Ordinary cluster contains from 30 to 100 people inside. When we freeze this factor, the value of people is 50.

Amount of objects

New systems have a lot of objects, but to show efficiency of the modified algorithm collaborative filtering we can use a small amount of objects: it is from 200 to 1000 objects. When we freeze this factor, the value of objects is 400.

Percentage of filling

We can change the percentage of filling from zero to one, but in real life people never put up rates for all objects in system. Usually the number of scores compared to all objects will not be great, it is no more than 40%. So, in our research we will change the percentage of filling from 20% to 80%. When we freeze this factor, the value is 40%.

Results of research

The results of researching are shown in three points:

When we studied the unchanged number of users and number of objects, we took 50 users and 400 objects. The difference between the results of both algorithms calculated by formula (6) is shown in Fig. 1.

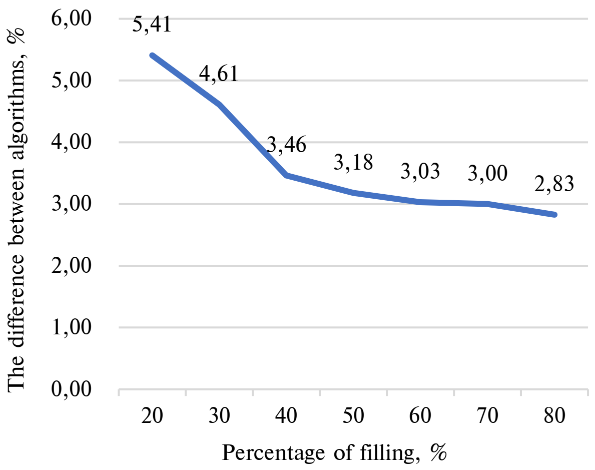

Fig. 1. The difference between algorithms when we changed the percentage of filling from 20% to 80%

Fig. 2 shows the runtime of the two algorithms versus the percentage of filling.

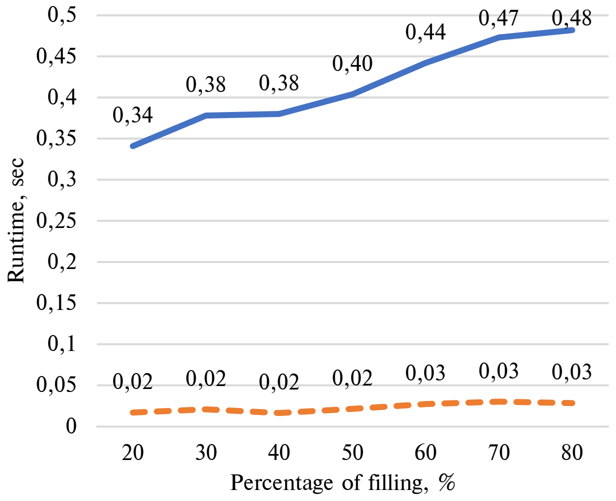

Fig. 2. Runtime of the two algorithms when we changed percentage of filling from 20% to 80% (the line shows the runtime of the base algorithm, and the dashed line shows the runtime of the modified algorithm of collaborative filtering)

So, we can see from the research that the difference between the results of both algorithms calculated by formula (6), when we changed the percentage of filling, is less than 5%. In Fig. 2, we can see that the runtime of the modified algorithm takes much less time with increasing the percentage of filling.

When we studied the unchanged number of users and percentage of filling, we took 50 users and 40 percentage of filling. The difference between the results of both algorithms calculated by formula (6) is shown in Fig. 3.

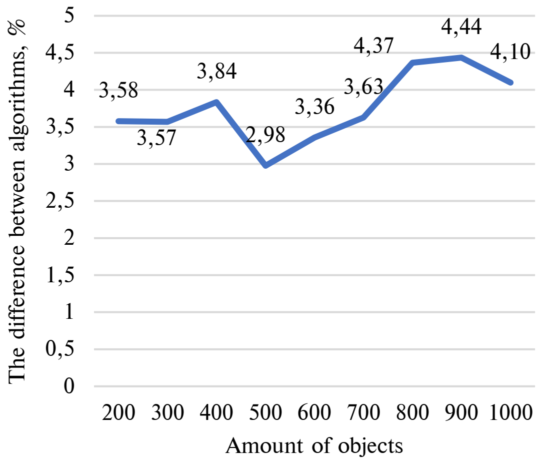

Fig. 3. The difference between algorithms when we changed the number of objects from 200 to 1000

Fig. 4 shows the results of the runtime of the two algorithms versus the number of objects.

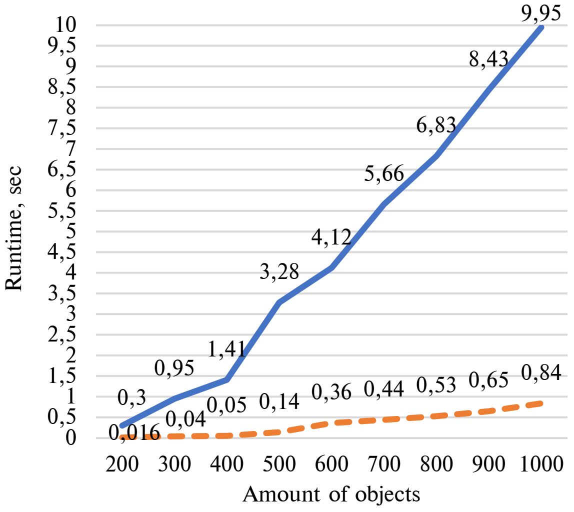

Fig. 4. Runtime of two algorithms when we changed number of objects from 200 to 1000. The line shows the runtime of base algorithm of collaborative filtering. Dashed line shows the runtime of modified algorithm of collaborative filtering

So, we can see from the research that the difference between the results of both algorithms calculated by formula (6), when we changed the number of objects, is less than 5%. In Fig. 4, we can see that the runtime of the modified algorithm takes much less time with increasing numbers of objects.

When we studied the unchanged number of objects and number of objects, we took 40 percentage of filling and 400 objects. The difference between the results of both algorithms calculated by formula (6) is shown in Fig. 5.

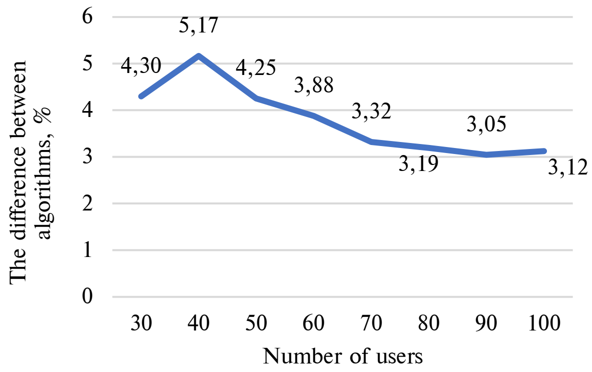

Fig. 5. The difference between algorithms when we changed the number of users in a cluster from 30 to 100

Fig. 6 shows the results of the runtime of the base and modified algorithms versus the number of users.

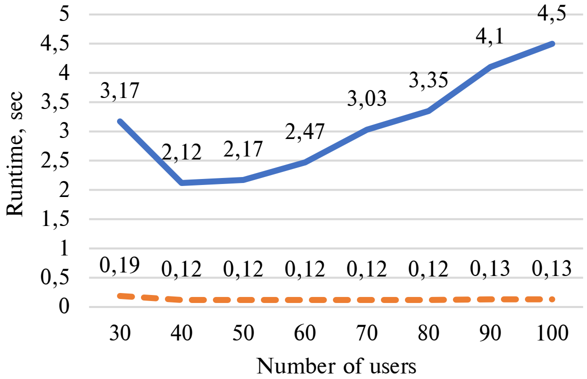

Fig. 6. Runtime of the two algorithms when we changed the number of users in a cluster from 30 to 100 (the line shows the runtime of the base algorithm, and the dashed line shows the runtime of modified algorithm)

So, we can see from the research that the difference between the results of both algorithms calculated by formula (6), when we changed the number of users in a cluster, is less than 5%. In Fig. 6, we can see that the runtime of the modified algorithm takes much less time with increasing numbers of users.

Research conclusion

In conclusion, of our researching, we can see that our modified algorithm of collaborative filtering works much faster than the base algorithm. The difference between the results of both algorithms does not exceed 5%.

Conclusions

We analyzed the work of the base algorithm of collaborative filtering. Then we found bottleneck place in algorithm. We formed stages of modifying the algorithm. The developed algorithmic software is applied to solve the problem of runtime. As a result of the research, we made sure that our modified algorithm works faster than the base algorithm of collaborative filtering. We used two criteria to compare the base and modified algorithms. It was shown that the difference between the results of both algorithms is insignificant, if the value of criterion (6) is close to zero and doesn't exceed 5%. We can see from the research that the results of both algorithms by this criterion are close. So, if systems start to use this modified algorithm, this can solve the problem with the runtime of the collaborative filtering algorithm and allow giving recommendations faster than the systems which use the base algorithm.

References

[1] P. Melville *et al.*, "Content-boosted collaborative filtering for improved recommendations", in *National Conference on Artificial Intelligence*, Edmonton, Canada, 2016, pp. 187–192.

[2] V. Srikar and R. Sudha, "Examining lists on twitter to uncover relationships between following, membership and subscription", in *Proc. 22nd Int. Conf. World Wide Web*, Rio de Janeiro, Brazil, 2013 pp. 673–676. doi: 10.1145/2487788.2488019

[3] P. Chatterjee *et al.*, *Advanced Multi-Criteria Decision Making for Addressing Complex Sustainability Issues.* Hershey, PA: IGI Global, 2019.

[4] H. Jafarkarimi *et al.*, "A naive recommendation model for large databases", *Int. J. Inform. Educ. Technol.*, vol. 2, no. 3, pp. 216–219, 2012.

[5] J.S. Breese *et al.*, "Empirical analysis of predictive algorithms for collaborative filtering", in *Proc. 14th Conf. Uncertainty in Artificial Intelligence*, San Francisco, CA, 1998, pp. 43–52.

[6] F. Ricci *et al.*, "Introduction to recommender systems handbook", in *Recommender Systems Handbook.* Boston: Springer, 2011, pp. 1–35. doi: 10.1007/978-0-387-85820-3_1

[7] B. Sarwar *et al.*, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, China, 2001, pp. 285–295.

[8] X. He *et al.*, "Neural collaborative filtering", in *Proc. 26th Int. Conf. World Wide Web*, Perth, Australia, pp. 173–182.

[9] Y. Koren and R. Bell, "Advances in collaborative filtering", in *Recommender Systems Handbook.* Boston: Springer, 2015, pp. 77–117.

[10] C. Yin *et al.*, "Improved collaborative filtering recommendation algorithm based on differential privacy protection", *J. Supercomput.*, vol. 76, pp. 5161–5174, 2020. doi: 10.1007/s11227-019-02751-7

[11] P. Lytvak and I. Kochubey, "Application of MAI to solve the problem of choosing an algorithm of the system of forming proposals to bookstore users", in *Aktuelle Themen im Kontext der Entwicklung der Modernen Wissenschaften*, Dresden, Germany, 2019, pp. 62–68.

[12] G.H. Tzeng and J.J. Huang, *Multiple Attribute Decision Making. Methods and applications.* Boka Raton: CRC Press, 2011.

І.Ю. Кочубей, О.С. Жураковська

АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ КОРИСТУВАЧАМ ВЕБ-КАТАЛОГУ

Проблематика. У нашому онлайн-житті ми отримуємо багато інформації, і все більше людей не хочуть аналізувати чи переглядати великі обсяги інформації. Кожен із нас хоче швидко знайти те, що шукає. На багатьох сайтах, таких як YouTube, Facebook і Twitter, вже є система рекомендацій, і багато користувачів віддають їй перевагу. Формування алгоритмічного забезпечення рекомендаційних систем на сьогодні є дуже актуальною проблемою.

Мета дослідження. Ми розглядаємо базовий алгоритм колаборативної фільтрації, який часто використовується в системах рекомендацій. Намагаємося знайти вузькі місця базового алгоритму колаборативної фільтрації для його вдосконалення, тобто покращення його швидкодії.

Методика реалізації. Ми проаналізували базовий алгоритм колаборативної фільтрації та виявили вузьке місце. Основний час роботи алгоритму зосереджений на обчисленні схожості користувачів. У модифікованому алгоритмі ми обчислюємо середню оцінку об᾿єкта в кластері з коефіцієнтом зважування. Для порівняння базового алгоритму з модифікованим алгоритмом ми використовуємо два критерії. Перший критерій – час роботи алгоритму. Другий критерій – це кількість елементарних перестановок, які ми повинні зробити, щоб отримати рекомендації, які дає базовий алгоритм колаборативної фільтрації. Основними факторами, які впливають на час роботи алгоритму колаборативної фільтрації, є: кількість користувачів, кількість об᾿єктів і відсоток заповнення.

Результати дослідження. З дослідження отримано результати відхилення у результатах між базовим та модифікованим алгоритмами, яке коливається між 3 та 5 %. Модифікований алгоритм працює швидше, ніж базовий, до того ж зі збільшенням кількості користувачів або кількості об᾿єктів різниця в часі роботи буде збільшуватися. Результати дослідження представлені у графіках.

Висновки. Ми проаналізували базовий алгоритм колаборативної фільтрації та методи його вдосконалення. Можна зробити висновок про доцільність використання модифікованого алгоритму фільтрації. Модифікований алгоритм дає великий виграш у часі. Якщо системи почнуть використовувати модифікований алгоритм, це зможе вирішити проблему з часом роботи алгоритму фільтрації і дасть змогу давати рекомендації швидше, ніж система, яка використовує базовий алгоритм.

Ключові слова: рекомендаційна система; колаборативна фільтрація; модифікований алгоритм колаборативної фільтрації; веб-каталог.

И.Ю. Кочубей, О.С. Жураковская

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ФОРМИРОВАНИЯ РЕКОМЕНДАЦИЙ ПОЛЬЗОВАТЕЛЯМ ВЕБ-КАТАЛОГА

Проблематика. В нашей онлайн-жизни мы получаем много информации, и все больше людей не хотят анализировать или просматривать большие объемы информации. Каждый из нас хочет быстро найти то, что ищет. На многих сайтах, таких как YouTube, Facebook и Twitter, уже есть система рекомендаций, и многие пользователи отдают ей предпочтение. Формирование алгоритмического обеспечения рекомендательных систем в настоящее время является очень актуальной проблемой.

Цель исследования. Мы рассматриваем базовый алгоритм коллаборативной фильтрации, который часто используется в системах рекомендаций. Пытаемся найти узкие места базового алгоритма коллаборативной фильтрации для его совершенствования, то есть увеличения его быстродействия.

Методика реализации. Мы проанализировали базовый алгоритм коллаборативной фильтрации и обнаружили узкое место. Основное время работы алгоритма сосредоточено на вычислении сходства пользователей. В модифицированном алгоритме мы вычисляем среднюю оценку объекта в кластере с коэффициентом взвешивания. Для сравнения базового алгоритма с модифицированным алгоритмом мы используем два критерия. Первый критерий – время работы алгоритма. Второй критерий – количество элементарных перестановок, которые мы должны сделать, чтобы получить рекомендации, которые дает базовый алгоритм коллаборативной фильтрации. Основными факторами, которые влияют на время работы алгоритма коллаборативной фильтрации, являются: количество пользователей, количество объектов и процент заполнения.

Результаты исследования. Получены результаты отклонения в результатах между базовым и модифицированным алгоритмами, которое колеблется между 3 и 5 %. Модифицированный алгоритм работает быстрее базового, к тому же с увеличением количества пользователей или количества объектов разница во времени работы будет увеличиваться. Результаты исследования представлены в графиках.

Выводы. Мы проанализировали базовый алгоритм коллаборативной фильтрации и методы его усовершенствования. Можно сделать вывод о целесообразности использования модифицированного алгоритма фильтрации. Модифицированный алгоритм дает большой выигрыш во времени. Если системы начнут использовать модифицированный алгоритм, это сможет решить проблему со временем работы алгоритма фильтрации и позволит давать рекомендации быстрее, чем система, которая использует базовый алгоритм.

Ключевые слова: рекомендательная система; коллаборативная фильтрация; модифицированный алгоритм коллаборативной фильтрации; веб-каталог.

Рекомендована Радою факультету
прикладної математики
КПІ ім. Ігоря Сікорського

Надійшла до редакції
28 листопада 2019 року

Прийнята до публікації
25 червня 2020 року