https://doi.org/10.31891/2307-5732-2026-365-73
УДК 519.816

ІВОХІН ЄВГЕН
Київський національний університет імені Тараса Шевченка
https://orcid.org/0000-0002-5826-7408
ivohin@univ.kiev.ua
ШЕЛЯКІН ГЛІБ
Київський національний університет імені Тараса Шевченка
https://orcid.org/0009-0002-7171-6535
shelyakingleb17@gmail.com

МОДИФІКАЦІЯ МЕТОДУ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ ТИПУ ITEM-BASED

У даній статті представлено вдосконалений підхід до формування рекомендацій на основі колаборативної фільтрації типу item-based, який інтегрує семантичні та часові чинники разом із методами кластерного аналізу. Основною метою запропонованих удосконалень є зниження обчислювального навантаження на рекомендаційний модуль та підвищення релевантності сформованих рекомендацій шляхом виключення контенту, що не має смислового навантаження, із одночасним збереженням контексту під час генерації рекомендацій для користувачів. У межах реалізації алгоритму використано методи колаборативної фільтрації в комбінації з алгоритмом кластеризації HDBSCAN, що забезпечило ефективну обробку великих обсягів даних. Застосування семантичних та часових характеристик об'єктів дало змогу істотно підвищити точність апроксимації оцінок користувачів і, відповідно, якість персоналізованих рекомендацій. Крім того, запропоновано метод оптимізації обробки даних з урахуванням динаміки змін інтересів користувачів у часовому розрізі. Розроблена система також передбачає можливість класифікації контенту за релевантними ознаками, що сприяє підвищенню точності фільтрації. Процедура попередньої обробки передбачає агрегацію даних із подальшим застосуванням кластеризації, що дозволяє зменшити обчислювальну складність формування рекомендацій. Обчислення подібності між об'єктами здійснюється з урахуванням як часових, так і семантичних характеристик. Розроблено програмне забезпечення для тестування запропонованого алгоритму на різних наборах даних. Експериментальна перевірка запропонованого підходу на різних датасетах продемонструвала його переваги над базовими методами за критеріями точності та продуктивності.

Ключові слова: *рекомендаційні системи, метод колаборативної фільтрації типу item-based, кластерний аналіз, метод семантичної подібності, часовий фактор.*

YEVHEN IVOKHIN, GLIB SHELYAKIN
Taras Shevchenko National University of Kyiv

MODIFICATION OF THE ITEM-BASED COLLABORATIVE FILTERING METHOD

This article presents an improved approach to generating recommendations based on item-based collaborative filtering, which integrates semantic and temporal factors with cluster analysis methods. The main goal of the proposed improvements is to reduce the computational load on the recommendation module and increase the relevance of the generated recommendations by excluding content that has no semantic meaning, while preserving the context when generating recommendations for users. The algorithm uses collaborative filtering methods in combination with the HDBSCAN clustering algorithm, which ensures efficient processing of large amounts of data. The use of semantic and temporal characteristics of objects made it possible to significantly improve the accuracy of user rating approximations and, accordingly, the quality of personalized recommendations. In addition, a method for optimizing data processing was proposed, taking into account the dynamics of changes in user interests over time. The developed system also provides for the possibility of classifying content according to relevant criteria, which contributes to increased filtering accuracy. The preprocessing procedure involves data aggregation followed by clustering, which reduces the computational complexity of generating recommendations. The similarity between objects is calculated taking into account both temporal and semantic characteristics. Software has been developed to test the proposed algorithm on different data sets. Experimental verification of the proposed approach on different datasets has demonstrated its advantages over basic methods in terms of accuracy and performance.

Keywords: *recommendation systems, item-based collaborative filtering method, cluster analysis, semantic similarity method, time factor.*

Стаття надійшла до редакції / Received 11.02.2026
Прийнята до друку / Accepted 11.03.2026
Опубліковано / Published 28.05.2026

Introduction

Collaborative filtering as a method of building recommendation systems emerged as a result of the rapid development of the Internet, when it became almost impossible to extract the necessary information from all the available information, and there was a need to search for mechanisms for its effective filtering. This search led to the development of algorithms that could measure the “similarity” between the preferences of a particular person and the preferences of others, and provide reasonable recommendations about what might interest a person in the future.

An example of such approaches is the method of using a collaborative filtering algorithm based on user information (user-based) and on comparisons of recommendation items (item-based). In our study, we decided to use the item-based method because it not only allows for a more "robust" assessment, but can also speed up data processing, since there are usually fewer products on the service than users, which in turn allows us to store less data.

However, given the growth in content and users, the computing power of servers may be insufficient for the effective operation of the above method. This study proposes to take into account the fact that user interests may change over time and that content can be divided in advance into subsets according to certain criteria (classified or clustered), which will allow the recommendation system to process data faster. As part of the study, an attempt was made to improve the item-based collaborative filtering method based on comparing objects using the time factor and semantic similarity and through the application of cluster analysis techniques.

Analysis of recent research and publications

The relevance of the issue under consideration is confirmed by contemporary Ukrainian scientists (Kulyagin I., Narozhny V. [1]; Kucheruk V., Glushko M. [2]; Leshchynskyi V., Leshchynska I. [3]) and foreign studies (Berkovsky S., Cantador I., Tikk D. [4]; Hinneburg A., Keim A.D. [5], Horasan F., Yurtttakal A., Gunduz S. [6], Lafia N., Capra L., Hailes S. [7], Marappan R. [8], Natarajan S., Vairavasundaram S., Natarajan S.[9], Qin Lui [10], Sander J. [11]) and others.

In particular, Kulagin A.I., Narozhny V.V., Tkachov V.M., Kuchuk G.A. [1] investigated methods for building recommendation systems to solve the problem of selecting the most relevant video when creating virtual art compositions. Kucheruk V.Yu., Glushko M.V. [2] proposed an improvement to the "Item to Item" algorithm of the collaborative filtering method for developing recommendation systems based on the cosine measure by assessing relevance. Leshchynskyi V.A., Leshchynska I.A. [3] improved the collaborative filtering method with implicit feedback based on ranking negative results in the input data matrix.

Birkovsky S., Santador I., and Tikk D. [4] conducted a thorough analysis of algorithms that can be used for content recommendations, considered the challenges that may be encountered during the development of recommendation systems, and proposed practical applications of optimization methods. In the work of Hinneburg A. and Keim A.D. [5] propose a new algorithm based on kernel density estimation for clustering in large multimedia databases, in which clusters can be identified by determining density attractors, and clusters of arbitrary shape can be easily described by a simple equation of the total density function. Neil Laffia, Licia Capra, and Stephen Hailes [7] proposed the use of a time function that gradually reduces the weight of older views when calculating the rating. This approach allows the system to focus on more recent interactions, which better reflects users' current interests and increases the relevance of recommendations. Qin Lui [10] proposed a new way of selecting users for the user-based method by dividing user features (attributes) into linear and hierarchical ones. In addition, he developed a new quantitative assessment formula, which he called User Attention, to calculate the similarity of users based on their preferences, as well as an index, which he called Project Popularity. Sander J. [11] generalized the well-known DBSCAN algorithm (called GDBSCAN), which allows point objects and spatially extended objects to be clustered according to their spatial and non-spatial attributes. In addition, he presented four applications using 2D points (astronomy), 3D points (biology), 5D points (earth sciences), and 2D polygons (geography), which demonstrate the applicability of GDBSCAN to real-world problems.

The goal of the study was to improve the item-based collaborative filtering method by applying the time factor, semantic similarity, and data clustering.

To achieve this goal, the following tasks were set:

1) To analyze the item-based collaborative filtering method.

2) To improve the item-based collaborative filtering method by applying time and semantic factors, as well as the HDBSCAN data clustering method.

3) To develop software to verify the effectiveness of the proposed modification.

4) To verify and analyze the effectiveness and constructiveness of the results obtained for data from different content areas.

Methods

The main research methods are based on the use of analytical techniques and conclusions regarding the impact of semantic, temporal, and group indicators on the improvement of collaborative filtering algorithms and HDBSCAN cluster analysis.

Problem statement

Let us consider the procedure for making a certain decision, taking into account the available set of recommendations. Suppose that the objects of the recommendations are certain entities, the characteristics of which can be divided into three groups:

1. Ratings. An assessment is understood to be a numerical value that is limited from below and above, for example, from 1 to 5, from 0 to 1, etc.

2. Textual information. Textual information is considered to be a finite list of characteristics of each object. Textual information includes, for example, descriptions, names, summaries, etc.

3. Time information. Time information refers to data about the time a user performs a certain action on an object. For example, the time when a user left their verbal (verbal) or non-verbal (numerical) feedback.

The task is to develop an algorithm that can be used to obtain approximations of user ratings using the above-mentioned characteristics without restrictions on the domain (problem) area, and which allows this to be done faster and with less error than the naive collaborative filtering method.

Results and discussion

First proposed by Goldberg in 1992 [12], the collaborative filtering method is based on user's responses to content. The main goal of the method is to calculate the rating of content that the user is not yet familiar with, using information about their previous reactions. The closer the rating is to the actual rating, with a reduced margin of error, the higher the quality of the final recommendation will be. As an example, let's consider any platform that provides recommendations based on users' preferences.

LinkedIn is a professional networking platform that allows users to create detailed profiles highlighting their work experience, education, skills, and achievements. LinkedIn serves as a hub for professionals to connect, share industry insights, and explore career opportunities.

LinkedIn's recommendation system personalizes content and connection suggestions to increase user engagement. This system uses the following combination of methods:

1. Collaborative filtering: analyzes user interactions, such as profile views, connections, and endorsements, to identify patterns and recommend relevant content or connections.
2. Content-based filtering: Checks user profile attributes, including job titles, skills, and industries, to suggest connections and content that match the user's professional experience.
3. Natural language processing (NLP): Processes text data from user profiles, posts, and articles to understand contextual connections and recommend relevant content.
4. Graph-based algorithms: Uses a network structure of connections to identify and suggest potential connections within the user's professional circle.

By integrating these methodologies, LinkedIn provides personalized content feeds, connection suggestions, and job recommendations, contributing to a more engaging and personalized users' experience.

Collaborative filtering is also used by such well-known platforms as Amazon, Temu, YouTube, Netflix, Spotify, Instagram, and others, as the advantages of this method are:

1. Independence from a specific domain area. The implementation of the method does not depend on the specifics of its use, which allows it to be applied in any field with almost no changes to the mathematical apparatus.
2. Transparency of the method. Differences from machine learning models make it possible to verify the correctness of the calculations and logic of the method at each iteration of the system, which greatly simplifies the development of recommendation system software.
3. Does not require a large number of input parameters. For the method to work correctly, a feedback matrix (user ratings) is sufficient, which distinguishes it from systems that require a significantly larger number of input parameters.

To obtain the most objective assessment of a recommendation system, it is necessary to have as many user ratings and methods for analyzing them as possible. Thus, the use of collaborative filtering allows for more accurate recommendations to users based on their own reactions to content.

It follows from the above that the main concepts used in the methodology are user (a person who must be registered on a certain service and has the ability to view, rate, purchase content on it, etc.); content (the content of the site with which the user interacts within the service and which can be evaluated by users based on a set of metadata, such as description, title, cost, tags, etc.); rating (determined by a numerical value from a set of natural numbers, which is limited by zero at the bottom and some positive integer at the top) and recommendation (a set that may consist of a group of meaningful suggestions (descriptions) that will be created by the recommendation system for the user).

Users' ratings for objects on a registered service are conveniently presented in the form of a matrix $R = \{R_{iu}\}$, $u = \overline{1,n}$, $i = \overline{1,m}$, n – number of users of the service and m – number of objects.

Let us consider an arbitrary user of the service, which is identified by a certain serial number u, $u = \overline{1,n}$. The task is to approximate the possible user rating u in terms of an object i, $i = \overline{1,m}$.

The approximation procedure can be described as follows:

1. For each object $\hat{\imath}$, we calculate how similar it is to object i.
2. We form a set of objects that are most similar to object i.
3. Calculate the rating of object i based on the ratings from the set of most similar objects.

Now let's describe each of the steps in this procedure in more detail. Since each object corresponds to a column of the matrix, we will calculate the similarity measure by columns, using, for example, the normalized sample correlation coefficient. The formula for estimating similarity $sim(i,\hat{\imath})$ of the objects i and $\hat{\imath}$ can be presented as:

$$sim(i,\hat{\imath}) = \frac{\frac{\sum_{u=1}^{n}(R_{\hat{\imath}u}-\overline{R}_u)(R_{iu}-\overline{R}_u)}{\sqrt{\sum_{u=1}^{n}(R_{\hat{\imath}u}-\overline{R}_u)^2}\sqrt{\sum_{u=1}^{n}(R_{iu}-\overline{R}_u)^2}}+1}{2} \quad (1)$$

where $\overline{R_u}$ – average user's rating u.

The resulting similarity value has the following properties:

- $sim(i,\hat{\imath}) \in [0,1]$;
- $sim(i,\hat{\imath}) = 0$, in the absence of user's ratings or any similarity between objects, or equal to 1 if the objects are as similar as possible in terms of ratings.

We then select a specified number of objects $s > 0$,that are most similar to the selected object i. To do this, we need to sort the obtained values in descending order and select s of the first objects. Sorting will ensure that the first s objects will have the highest similarity values, which are close to 1.

By renumbering these objects, we will calculate what possible rating a user *u* would give to the object ***i***. This can be done using the weighted average formula:

$$p_{iu} = \frac{\sum_{\hat{\imath}=1}^{s} R_{iu} sim(i,\hat{\imath})}{\sum_{\hat{\imath}=1}^{s} sim(i,\hat{\imath})}. \quad (2)$$

The essence of this assessment is that the greater the similarity between objects $\hat{\imath}$ and i , the greater the contribution of object $\hat{\imath}$ to the final assessment is.

A key improvement in the study is the introduction of an "aging" function, which increases the system's adaptability to changes in user preferences. Thanks to this, greater weight is given to recently viewed content, while the influence of outdated interactions (e.g., views made a week, month, or year ago) gradually decreases. This mechanism avoids generating irrelevant or outdated recommendations. The "aging" function is used in the form of $f(t) = e^{-t}$, $0 \leq t \leq \infty$, $f(t) \in [0,1]$ and ensures a smooth decrease in value over time, which corresponds to the natural change in the relevance of data. Thus, the further back in time an event is, the less influence it has on recommendations.

The algorithm for using the aging function is as follows:

1. Calculate the value of the "aging" function for the time information indicator for each object.
2. Sort the obtained values in descending order. This will ensure that objects that have been viewed recently will have a greater impact on the recommendation.
3. Chose first $v > 0$ objects objects that are currently trending with the user and, accordingly, have a greater influence on what they want to see.

The proposed modification will also contribute to improving the quality of recommendations, as the system will focus on user's interests, which change during content viewing, thereby expanding the thematic diversity of the proposed materials. Of particular importance is the identification of the trend toward novelty – the tendency of users to consume current content. In this context, taking time characteristics into account allows the system to provide recommendations focused on the latest materials that meet the current demands of the audience.

Semantic analysis plays a key role in the development of modern recommendation algorithms, as textual information contains multidimensional semantic characteristics that allow for deeper modeling of user preferences. Often, traditional metrics such as rating, duration, or viewing frequency cannot accurately reflect users' true attitudes toward an object. For example, a situation where a user does not watch a movie to the end but leaves a positive review may indicate that they liked the plot or the acting, even though they did not finish the movie itself. In another case, a user may give a product a low rating not because of its quality, but because of problems with delivery or service. Similarly, on social networks, the number of "likes" on posts does not always correspond to the actual popularity of the content, since "likes" can be artificial, provided by accounts that are not real users.

Text data provides a deeper understanding of user preferences, which significantly improves the relevance and personalization of recommendations and helps to generate recommendations that correspond to the user's sociocultural context. In order to take into account the semantic factor, let us introduce the following concepts: M – a set of textual information that reflects the essence of a certain object *i*. Each element of the set is a vector $(a_1, a_2, \dots, a_K)$, where a_j is some textual representation of the *j* attribute of the current object $j = \overline{1, K}$. As mentioned earlier, these can be descriptions, reviews, technical specifications, testimonials, and more. Then the measure of proximity between two objects i and $\hat{\imath}$ can be represented by the following formula:

$$sim_{meta}(i,\hat{\imath}) = \frac{\sum_{j=1}^{K} sim(i_{a_j}, \hat{\imath}_{a_j})}{K} \quad (3)$$

where $sim(i_{a_j}, \hat{\imath}_{a_j})$ – similarity coefficient of j characteristics of the objects i and $\hat{\imath}$ by using, for example, Siamese neural networks as evaluators.

Siamese neural networks are a class of neural networks used to compare two inputs and determine their similarity. These networks are called Siamese because they consist of two identical subnetworks, each of which processes one of the inputs and then combines their outputs.

Let's take a closer look at their structure and how they work. During training, the network accepts three values as input (w_1, w_2, s), where w_1, w_2– vector representations of a text,$s \in \{0,1\}$– expected similarity index between w_1 and w_2 (1 – similar, 0 – different). Each subnetwork receives one vector, and both vectors undergo the same processing (the number of layers in the networks, their type, and parameters are completely identical). The last, common layer receives two resulting features at the input and determines their similarity by measuring the Manhattan distance between the two features: $e^{(-|v_1 - v_2|)} \in [0,1]$, where v_1 and v_2 -– signs obtained at the output of subnets. The goal of training is to minimize the distance between similar objects and maximize it between dissimilar objects.

Siamese network architecture is a powerful tool widely used in tasks such as text similarity assessment, image comparison, etc. In the context of text similarity analysis, a Siamese network takes two text fragments as input, simultaneously encodes them using a shared embedding layer, and then compares the encoded vectors to evaluate their similarity or difference.

The main advantage of Siamese networks is their ability to learn from pairs of data with known similarity, making them ideal for tasks where similarity needs to be measured using labeled data. During training, the network minimizes the contrastive loss function, which encourages the model to distinguish between similar and dissimilar pairs of input data, ensuring high accuracy in detecting differences or similarities between objects.

Research shows that Siamese networks effectively solve various tasks related to similarity measurement, including searching for similar texts and images. In addition, they have been adapted to more complex architectures, such as triplet networks, which use three inputs instead of two. In this case, the network learns to evaluate the distance between a positive and negative example, allowing for even more accurate results in complex comparison tasks. Triplet networks add another level of complexity and accuracy, allowing them to work effectively with more complex and multidimensional data.

Let us formulate a data preprocessing (data aggregation) procedure for the collaborative filtering method based on object comparisons using the clustering method. The procedure consists of the following steps:

1. A similarity matrix of objects is constructed using the available data in the sample.
2. A clustering algorithm is used to divide the set of objects into corresponding clusters.
3. Data about the corresponding clusters is stored for future use.

Let's define a square symmetric matrix $S = \{S_{ij}\}$, $i, j = \overline{1, m}$, $S_{ij} = S_{ji}$, where m – a number of objects on the service, S_{ij} – similarity assessments i-th and j-th objects. This allows to store assessment results in a database and calculate assessments offline, taking into account that the number of objects is growing rather slowly. This also has a positive effect on the speed of data updates.

Another approach to improving the recommendation process is to use clustering methods to group information.

The HDBSCAN algorithm is a powerful tool for data clustering that successfully addresses issues related to determining the number and shape of clusters, as well as filtering out noise points. One of the key advantages of this method is the use of the density-based clustering principle, which has become popular due to its ability to process complex data structures and provide accurate results.

HDBSCAN has a built-in automatic parameter tuning system that allows it to independently determine the optimal values for parameters such as minimum density and cluster radius. This greatly simplifies the process of using the algorithm and reduces the need for manual tuning, making the method accessible even to users without in-depth knowledge of parameter tuning. In addition, HDBSCAN is quite fast and scalable, allowing it to work effectively with large amounts of data.

One of the features of HDBSCAN is its ability to detect clusters of any shape, including complex structures with intertwining or narrow gaps. In addition, this method is good at detecting noise points and can separate them from real clusters, providing even greater clustering accuracy. This approach allows you to obtain high-quality results in a variety of tasks, including creating personalized recommendations for users based on information similarity.

Let's consider how the algorithm works.

1. A "neighborhood matrix" is constructed—the distances between each pair of points are calculated and a neighborhood matrix is created. This allows denser areas to be separated from less dense areas.
2. A "minimum spanning tree" is constructed – a minimum spanning tree is calculated based on the neighborhood matrix. A minimum spanning tree is a graph that connects all points but has the smallest possible total weight.
3. A "segmentation tree" is constructed – a segmentation tree is calculated from the minimum spanning tree. The segmentation tree divides clusters into subclusters and increases the accuracy of clustering.
4. Based on the segmentation tree, a set of clusters that best describe the data is determined.

After using the HDBSCAN algorithm, it is necessary to save the obtained clusters, thereby performing preliminary data processing/aggregation, which can reduce the time required to generate recommendations and the complexity of the necessary calculations before the algorithm starts working. When using raw data, the recommendation algorithm must perform additional data processing and grouping during execution. This usually takes a long time and requires significant computing resources.

Based on the above similarity assessment methods, we can derive the final algorithm for calculating the object rating, which will look like this:

Let's take

$$U(i, \hat{\imath}) = \alpha * sim_{mark}(i, \hat{\imath}), \tag{4}$$

where $sim_{mark}(i, \hat{\imath}) \in [0,1]$ – the degree of similarity between objects assessed by users, and $\alpha \in [0,1]$ – weighted coefficient responsible for the magnitude of the assessment's impact $sim_{mark}(i, \hat{\imath})$ on the final assessment of the object.

Similarly, we define

$$V(i, \hat{\imath}) = \beta * sim_{meta}(i, \hat{\imath}), \tag{5}$$

where $sim_{meta}(i, \hat{\imath}) \in [0,1]$ – measure of similarity between objects based on comparison of their metadata, and $\beta \in [0,1]$ – weight coefficient responsible for the magnitude of the assessment's impact $sim_{meta}(i, \hat{\imath})$ in the final assessment of the object.

Then, the final similarity (proximity) score of objects i and $\hat{\imath}$, can be recorded using the above values:

$$sim_{hybrid}(i, \hat{\imath}) = (U(i, \hat{\imath}) + V(i, \hat{\imath}))/2. \tag{6}$$

It is evident that the proposed similarity assessment fully coincides with the properties of the collaborative filtering method based on object comparisons.

Now let's construct the final formula for calculating the user rating u for an object i taking into account aging and semantic factors:

$$p_{ui} = \frac{\sum_{\hat{i}=1}^{v} R_{u\hat{i}} sim_{hybrid}(i,\hat{i}) f(t_{\hat{i}})}{\sum_{\hat{i}=1}^{v} sim_{hybrid}(i,\hat{i}) f(t_{\hat{i}})}. \quad (7)$$

To verify the proposed improvement, experiments were conducted based on the developed software, which was created using the Python programming language and the free VS Code code writing environment.

The adequacy of the proposed method was verified using data sets from various problem areas, and the speed of recommendation calculations was measured.

The diagram (Fig. 1) shows the average speed of calculations for the corresponding approximations (the dotted line indicates the time required to execute the modified algorithm, taking into account the number of elements in the clusters, and the solid line indicates the time required to execute the naive algorithm. Since the naive method does not take into account our modifications, the execution time does not depend on the number of elements in the cluster). These data confirm that the modified algorithm demonstrates a faster mode of operation compared to the naive method.

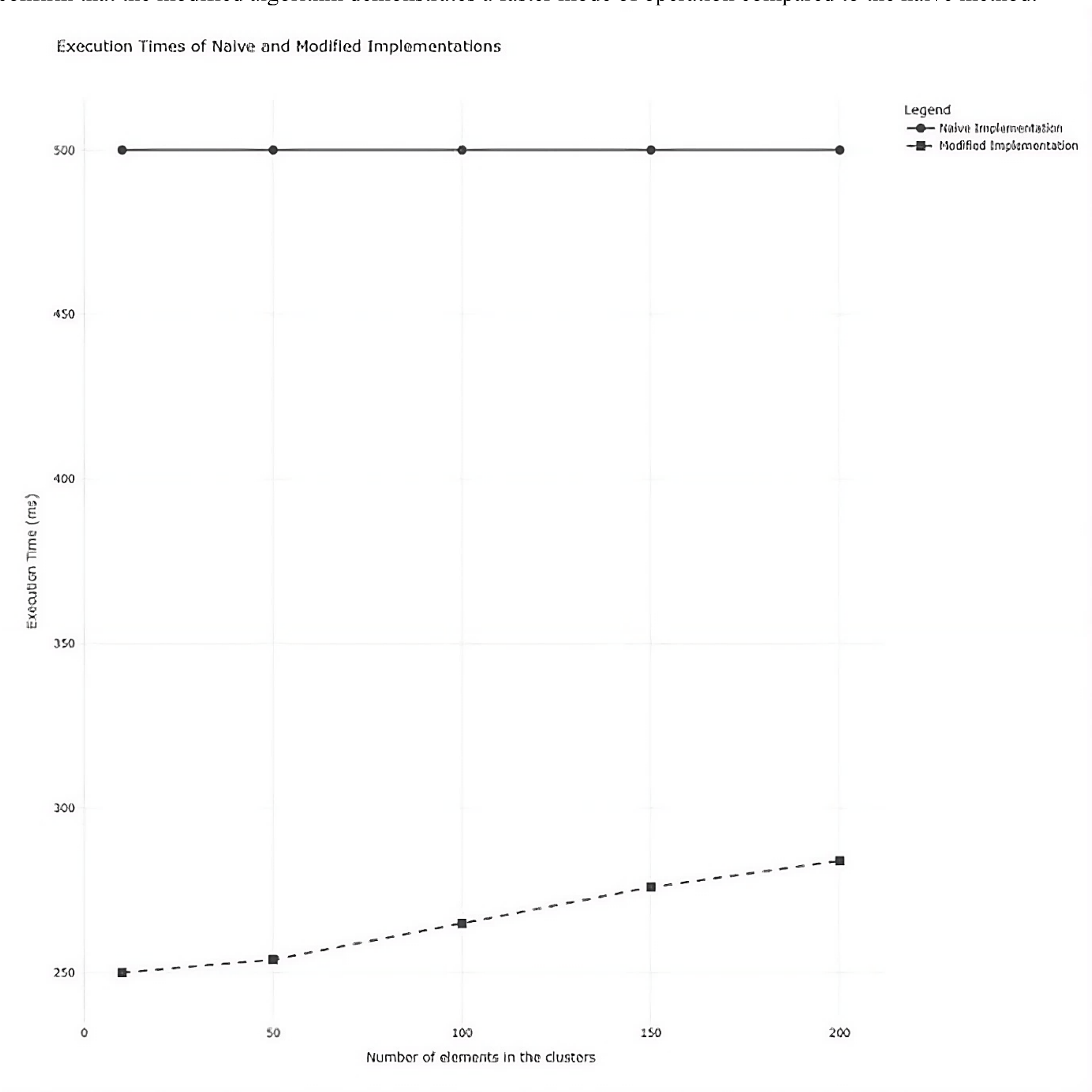

Fig. 1 Comparison of the speed of naive and modified methods

As an example of the practical application of the proposed method, a movie rating service was considered and possible errors in the results of recommendation formation were analyzed.

The results of statistical observation of the procedure for evaluating various recommendation objects by users were obtained. Figure 2 shows the number of objects involved in the recommendations on the x-axis. The y-axis shows the corresponding error value in the form of the root mean square deviation. The number of users involved in the experiment is 200. The values obtained for the modified method are marked with a dotted line, and for the naive method – with a continuous line. As can be seen from Fig. 2, the modified algorithm gives a smaller error than the naive method.

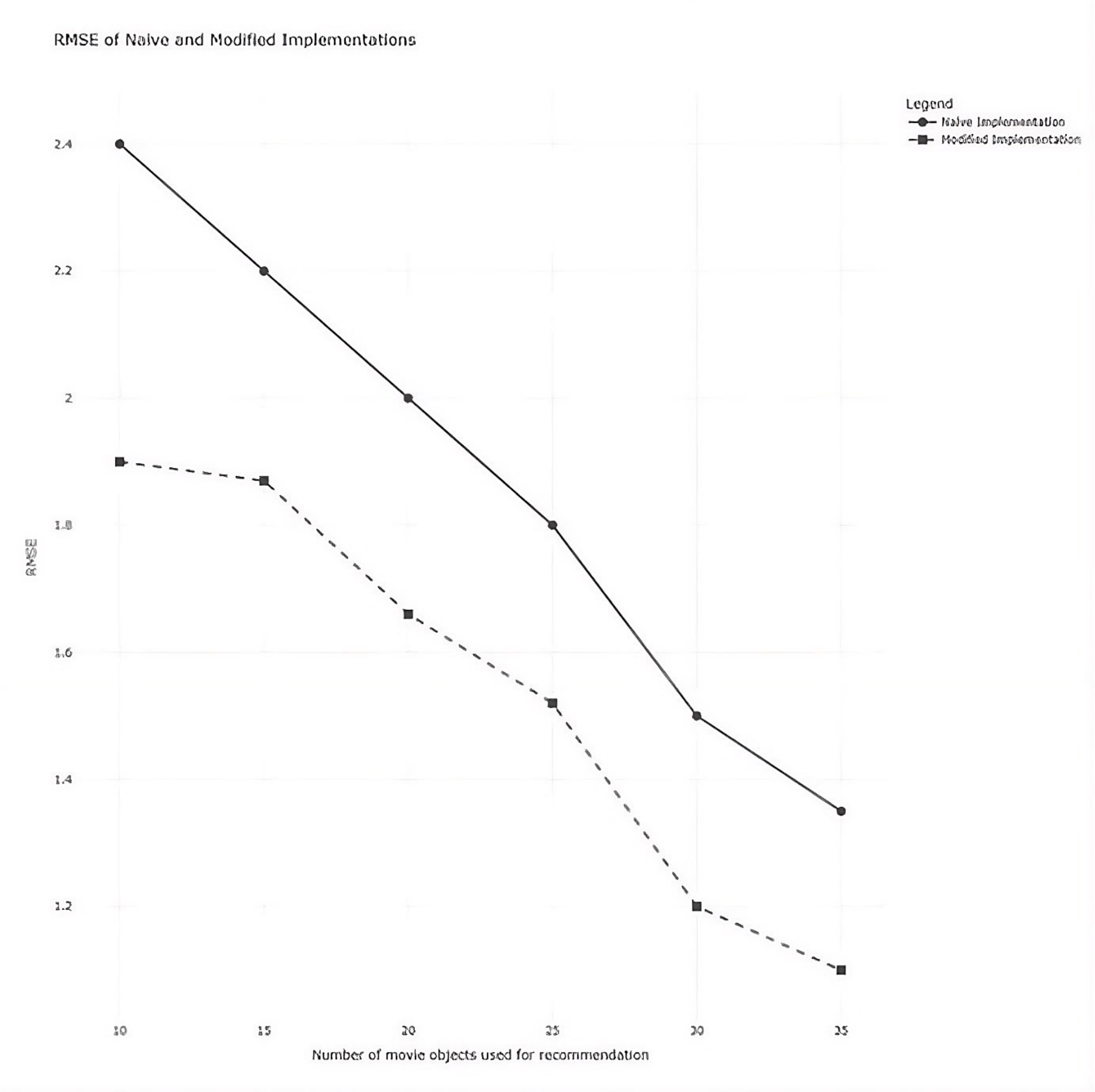

Fig. 2 Comparison of error values between the naive and modified methods

In another example (generating recommendations in the field of music tapes), statistical observation results were also obtained for the calculated error comparison results for the modified and naive methods. Here, the number of users involved in the experiment was 150. As can be seen in Fig. 3, the modified algorithm gives a smaller error than the naive method.

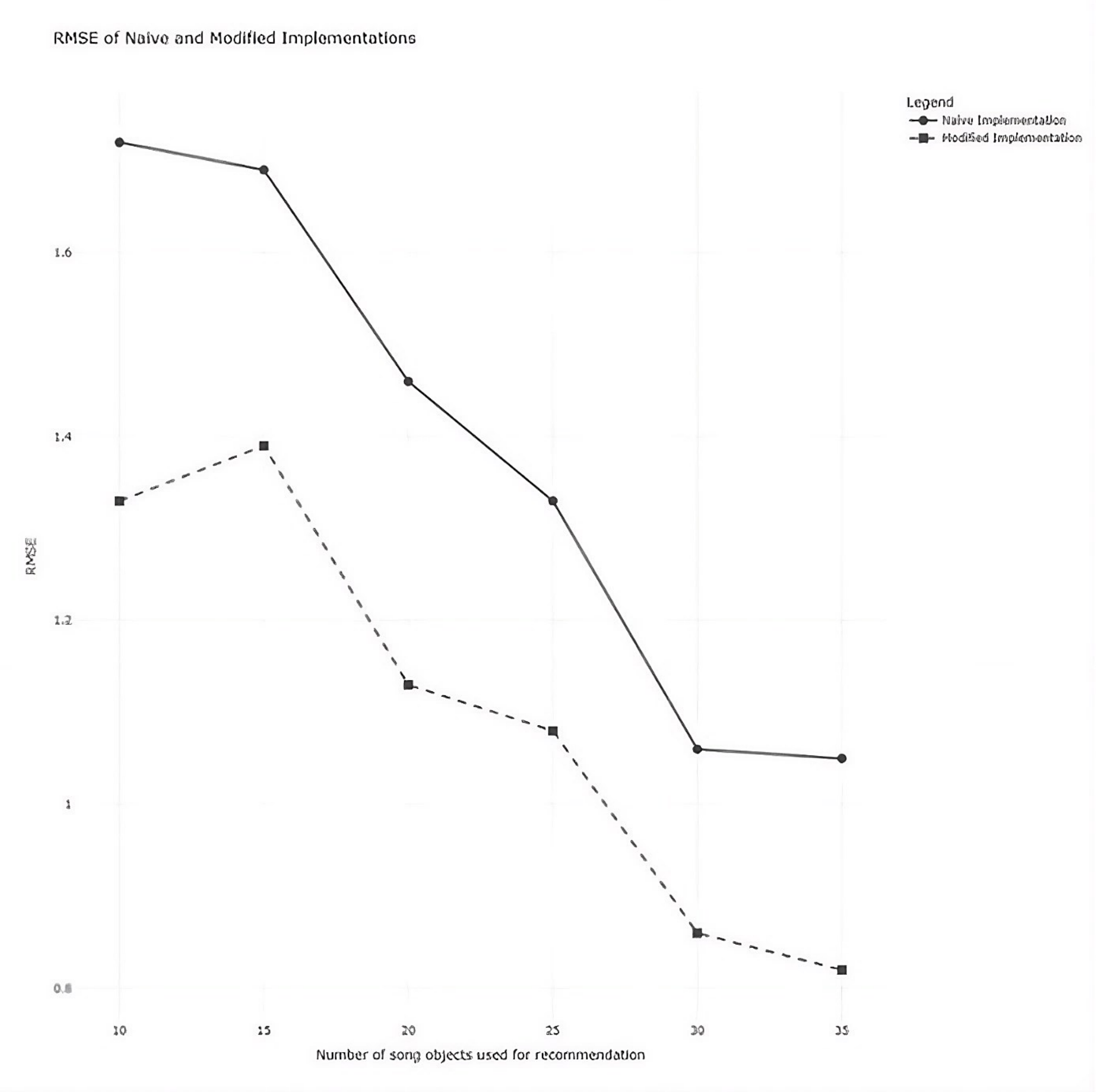

Fig. 3 Comparison of error values between the naive and modified methods

Conclusions

The described approach to modifying the item-based collaborative filtering method by introducing a time factor, semantic similarity, and data clustering is relevant for modern recommendation systems, as its goal is to improve algorithms that allow recommendation systems to more accurately predict user interests by applying new methods that take into account not only the similarity between objects, but also semantic analysis of content and changes in user interests over time.

The developed software made it possible to use the proposed modification in various problem (domain) areas. The results of the study show that applying modifications to collaborative filtering significantly improves the speed of calculations and the accuracy of recommendations. The modified algorithm has fewer errors, which improves the quality of service for users in various areas, such as movie or music track ratings.

Based on the experiments conducted, it can be concluded that the proposed methods work effectively on real data. An important feature is that the modification significantly reduces processing time and improves the accuracy of results, confirming the effectiveness of the proposed modification of collaborative filtering algorithms and its usefulness for modern recommendation systems that work with large amounts of data and changing user interests.

Література

1. Кулягін А. І., Нарожний В. В., Ткачов В. М., Кучук Г. А. Дослідження методів побудови рекомендаційних систем для розв’язання задачі вибору найбільш релевантного відео при створенні віртуальних арт-композицій // Системи управління, навігації та зв'язку. – 2022. – № 4. – С. 94–99. – DOI: https://doi.org/10.26906/SUNZ.2022.4.094

2. Кучерук В. Ю., Глушко М. В. Покращення алгоритму «Item to Item» методу колаборативної фільтрації для розробки рекомендаційних систем на основі косинусної міри шляхом оцінки релевантності // Scientific Journal «ScienceRise». – 2018. – № 1 (42). – С. 20–24. – DOI: 10.15587/2313-8416.2018.120886.

3. Лещинський В. А., Лещинська І. А. Удосконалення методу колаборативної фільтрації з неявним зворотним зв'язком на основі ранжування негативних результатів в матриці вхідних даних // Системи управління, навігації та зв'язку. – 2020. – № 3 (61). – С. 99–103. – DOI: https://doi.org/10.26906/SUNZ.2018.3.073.

4. Berkovsky S., Cantador I., Tikk D. Collaborative recommendations: algorithms, practical challenges and applications / S. Berkovsky, I. Cantador, D. Tikk. – Singapore: World Scientific Publishing, 2019. – DOI: https://doi.org/10.1142/11131.

5. Hinneburg A., Keim D. A. A general approach to clustering in large databases with noise / A. Hinneburg, D. A. Keim // Knowledge and Information Systems. – 2003. – Vol. 5, No. 4. – P. 384–415. – DOI: https://doi.org/10.1007/s10115-003-0086-9.

6. Horasan F., Yurttakal A., Gunduz S. A novel model based collaborative filtering recommender system via truncated ULV decomposition / F. Horasan, A. Yurttakal, S. Gunduz // Journal of King Saud University – Computer and Information Sciences. – 2023. – Vol. 35, No. 8. – DOI: https://doi.org/10.1016/j.jksuci.2023.101724.

7. Lathia N., Capra L., Hailes S. Temporal collaborative filtering with adaptive neighbourhoods / N. Lathia, L. Capra, S. Hailes // Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. – 2009. – P. 796–797. – DOI: https://doi.org/10.1145/1571941.1572133.

8. Marappan R. Recommender system for movielens datasets using an item-based collaborative filtering in Python / R. Marappan // International Journal of Mathematical, Engineering, Biological and Applied Computing. – 2022. – № 1 (1). – P. 42–43. – DOI: https://doi.org/10.31586/ijmebac.2022.340.

9. Natarajan S., Vairavasundaram S., Natarajan S., Gandomi A. H. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data / S. Natarajan, S. Vairavasundaram, S. Natarajan, A. H. Gandomi // Expert Systems with Applications. – 2020. – DOI: https://doi.org/10.1016/j.eswa.2020.113248.

10. Qin Liu A new collaborative filtering algorithm integrating time and multisimilarity / Q. Liu // Mathematical Problems in Engineering. – 2022. – DOI: https://doi.org/10.1155/2022/2340671.

11. Sander J. Density-based clustering in spatial databases / J. Sander // Data Mining and Knowledge Discovery. – 1998. – Vol. 2. – P. 169–194. – URL: https://link.springer.com/article/10.1023/A:1009745219419.

12. Goldberg D., Nichols D., Oki B. M., Terry D. Using collaborative filtering to weave an information tapestry / D. Goldberg, D. Nichols, B. M. Oki, D. Terry // Communications of the ACM. – 1992. – Vol. 35, No. 12. – P. 61–70. – DOI: https://doi.org/10.1145/138859.138867.

References

1. Kuliahin A. I., Narozhnyi V. V., Tkachov V. M., Kuchuk H. A. Doslidzhennia metodiv pobudovy rekomendatsiinykh system dlia rozviazannia zadachi vyboru naibilsh relevantnoho video pry stvorenni virtualnykh art-kompozytsii / A. I. Kuliahin, V. V. Narozhnyi, V. M. Tkachov, H. A. Kuchuk // Systemy upravlinnia, navihatsii ta zviazku. – 2022. – No. 4. – P. 94–99. – DOI: https://doi.org/10.26906/SUNZ.2022.4.094.

2. Kucheruk V. Iu., Hlushko M. V. Pokrashchennia alhorytmu «Item to Item» metodu kolaboratyvnoi filtratsii dlia rozrobky rekomendatsiinykh system na osnovi kosynusnoi miry shliakhom otsinky relevantnosti / V. Iu. Kucheruk, M. V. Hlushko // Scientific Journal «ScienceRise». – 2018. – No. 1 (42). – P. 20–24. – DOI: 10.15587/2313-8416.2018.120886.

3. Leshchynskyi V. A., Leshchynska I. A. Udoskonalennia metodu kolaboratyvnoi filtratsii z neiavnym zvorotnym zviazkom na osnovi ranzhuvannia nehatyvnykh rezultativ v matrytsi vkhidnykh danykh / V. A. Leshchynskyi, I. A. Leshchynska // Systemy upravlinnia, navihatsii ta zviazku. – 2020. – No. 3 (61). – P. 99–103. – DOI: https://doi.org/10.26906/SUNZ.2018.3.073.

4. Berkovsky S., Cantador I., Tikk D. Collaborative recommendations: algorithms, practical challenges and applications / S. Berkovsky, I. Cantador, D. Tikk. – Singapore: World Scientific Publishing, 2019. – DOI: https://doi.org/10.1142/11131.

5. Hinneburg A., Keim D. A. A general approach to clustering in large databases with noise / A. Hinneburg, D. A. Keim // Knowledge and Information Systems. – 2003. – Vol. 5, No. 4. – P. 384–415. – DOI: https://doi.org/10.1007/s10115-003-0086-9.

6. Horasan F., Yurttakal A., Gunduz S. A novel model based collaborative filtering recommender system via truncated ULV decomposition / F. Horasan, A. Yurttakal, S. Gunduz // Journal of King Saud University – Computer and Information Sciences. – 2023. – Vol. 35, No. 8. – DOI: https://doi.org/10.1016/j.jksuci.2023.101724.

7. Lathia N., Capra L., Hailes S. Temporal collaborative filtering with adaptive neighbourhoods / N. Lathia, L. Capra, S. Hailes // Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. – 2009. – P. 796–797. – DOI: https://doi.org/10.1145/1571941.1572133.

8. Marappan R. Recommender system for movielens datasets using an item-based collaborative filtering in Python / R. Marappan // International Journal of Mathematical, Engineering, Biological and Applied Computing. – 2022. – Vol. 1, No. 1. – P. 42–43. – DOI: https://doi.org/10.31586/ijmebac.2022.340.

9. Natarajan S., Vairavasundaram S., Natarajan S., Gandomi A. H. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data / S. Natarajan, S. Vairavasundaram, S. Natarajan, A. H. Gandomi // Expert Systems with Applications. – 2020. – DOI: https://doi.org/10.1016/j.eswa.2020.113248.

10. Qin Liu. A new collaborative filtering algorithm integrating time and multisimilarity / Q. Liu // Mathematical Problems in Engineering. – 2022. – DOI: https://doi.org/10.1155/2022/2340671.

11. Sander J. Density-based clustering in spatial databases / J. Sander // Data Mining and Knowledge Discovery. – 1998. – Vol. 2. – P. 169–194. – URL: https://link.springer.com/article/10.1023/A:1009745219419.

12. Goldberg D., Nichols D., Oki B. M., Terry D. Using collaborative filtering to weave an information tapestry / D. Goldberg, D. Nichols, B. M. Oki, D. Terry // Communications of the ACM. – 1992. – Vol. 35, No. 12. – P. 61–70. – DOI: https://doi.org/10.1145/138859.138867.