

**A. O. TURETSKYI, B. M. VORONA, M. A. VOVK, S. I. YERSHOVA**

### **DEVELOPMENT AND RESEARCH OF MODELS AND SOFTWARE FOR THE RECOMMENDER SYSTEM OF CONSUMER GOODS**

There have been proposed investigation of the problem of creating recommendations with technical description for building the Recommender System of consumer goods with help of modern algorithms, approaches, principles and contains the investigation of the most popular methods. It was defined, that the deployment of Recommender Systems is one of the rapidly developing areas for improving applied information technologies, tools for automatic generating offers service based on the investigation of the personal needs and profile of customers. It was investigated, that such systems have started to play a very important role in the fast growing Internet, as they help users to navigate in a large amount of information, because users are not able to analyze a large amount of information, because it is very difficult and takes a lot of time and effort, but due to such systems, namely Recommender Systems that are able to filter a large amount of information, and provide for users the information and recommendations their likes the problem can be solved and instead of providing the static information, when users search and, perhaps, buy products, Recommender Systems increase the degree of interactivity to expand the opportunities provided to the user. It was defined, that Recommendation systems form recommendations independently for each specific user based on past purchases and searches, and also on the basis of the behavior of other users with help of recommendation services, which collect different information about a person using several methods and at the same time all systems are shared. An overview of content-based, collaborative filtering and hybrid methods was performed. An overview of Alternating Least Squares and Singular Value Decomposition recommendation algorithms was performed. The design of the Recommender System of consumer goods software component was described. The main features of software implementation and programming tools for the system which is being developed were explained. The conclusions about the problems of Recommender Systems and the review of existing algorithms were made.

**Keywords:** recommender system, consumer goods, alternating least squares method, singular value decomposition method, models, software component, recommendations.

### **А. О. ТУРЕЦЬКИЙ, Б. М. ВОРОНА, М. А. ВОВК, С. І. ЄРШОВА РОЗРОБКА ТА ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА ПРОГРАМНИХ РІШЕНЬ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ВИБОРУ ТОВАРІВ МАСОВОГО ВЖИТКУ**

Запропоновано дослідження проблеми створення рекомендацій, з технічним описом для побудови рекомендаційної системи для вибору товарів масового вжитку за допомогою сучасних алгоритмів, підходів, принципів і містить дослідження найбільш популярних методів. Було визначено, що впровадження рекомендаційних систем є однією з областей, які швидко розвиваються для вдосконалення прикладних інформаційних технологій, інструментів для автоматичного генерування пропозицій, заснованих на дослідженні особистих потреб і профілю клієнтів. Було досліджено, що такі системи почали грати дуже важливу роль в швидко зростаючому Інтернеті, оскільки вони допомагають користувачам орієнтуватися у великій кількості інформації, користувачі не можуть аналізувати великий обсяг інформації, адже це дуже складно і також вимагає багато часу і зусиль, але завдяки рекомендаційним системам, які можуть фільтрувати великий обсяг інформації і надавати користувачам інформацію і рекомендації, які їм подобаються, проблема може бути вирішена і замість надання статичної інформації, коли користувачі шукають, і можливо, купують продукти, такі системи збільшують ступінь інтерактивності для розширення можливостей, що надаються користувачеві. Було визначено, що рекомендаційні системи формують рекомендації самостійно для кожного конкретного користувача на основі минулих покупок і пошуків, а також на основі поведінки інших користувачів за допомогою служб рекомендацій, які збирають різну інформацію про людину, що використовує кілька методів, і в той же час всі системи є загальними. Було проведено огляд методів фільтрації на основі контенту, спільної фільтрації і гібридних методів. Було виконано огляд алгоритмів альтернативних найменших квадратів і сингулярного розкладання. Описана конструкція рекомендаційної системи програмного забезпечення для вибору товарів масового вжитку. Зроблено пояснення деяких можливостей програмної реалізації і інструментів програмування для розробленої системи. Зроблено висновки про проблеми рекомендаційних систем і огляд існуючих алгоритмів.

**Ключові слова:** рекомендаційна система, товари масового вжитку, метод альтернативних найменших квадратів, метод сингулярного розкладу, моделі, програмний компонент, рекомендації.

### **А. О. ТУРЕЦЬКИЙ, Б. М. ВОРОНА, М. А. ВОВК, С. И. ЕРШОВА РАЗРАБОТКА И ИССЛЕДОВАНИЕ МОДЕЛЕЙ И ПРОГРАММНЫХ РЕШЕНИЙ ДЛЯ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ ВЫБОРА ТОВАРОВ МАССОВОГО ПОТРЕБЛЕНИЯ**

Предложено исследование проблемы создания рекомендаций, с техническим описанием для построения рекомендательной системы для выбора потребительских товаров с помощью современных алгоритмов, подходов, принципов и содержит исследование наиболее популярных методов. Было определено, что внедрение рекомендательных систем является одной из быстро развивающихся областей для совершенствования прикладных информационных технологий, инструментов для автоматического генерирования предложений, основанных на исследовании личных потребностей и профиля клиентов. Было исследовано, что такие системы начали играть очень важную роль в быстрорастущем Интернете, поскольку они помогают пользователям ориентироваться в большом количестве информации, пользователи не могут анализировать большой объем информации, это требует много времени и усилий, но благодаря рекомендательным системам, которые могут фильтровать большой объем информации и предоставлять пользователям информацию и рекомендации, которые им нравятся, проблема может быть решена и вместо предоставления статической информации, когда пользователи ищут и покупают продукты, такие системы увеличивают степень интерактивности для расширения возможностей, предоставляемых пользователю. Было определено, что рекомендательные системы формируют рекомендации самостоятельно для каждого конкретного пользователя на основе прошлых покупок и поисков, а также на основе поведения других пользователей с помощью служб рекомендаций, которые собирают различную информацию о человеке, использующем несколько методов, и в то же время все системы являются общими. Был проведен обзор методов фильтрации на основе контента, совместной фильтрации и гибридных методов. Был выполнен обзор алгоритмов альтернативных наименьших квадратов и сингулярного разложения. Описана конструкция рекомендательной системы программного обеспечения для выбора товаров массового потребления. Объяснены основные возможности программной реализации и инструменты программирования для разрабатываемой системы. Сделаны выводы о проблемах рекомендательных систем и обзор существующих алгоритмов.

**Ключевые слова:** рекомендательная система, товары массового потребления, метод альтернативных наименьших квадратов, метод сингулярного разложения, модели, программный компонент, рекомендации.

**Introduction.** Nowadays, we live in the age of information – the time of unrestricted access to the information resources, the time at which the amount of information published by various sites and other sources of information. The number of similar objects is so large and that it is difficult for the user to find the information they need among them through a regular review. Users are always care about their free time and want to use it with the benefit. Recommender Systems contribute to this process, as the system selects and provides itself a quantitative and qualitative assessment of the preferences of the user or a particular object.

Recommender Systems are used in the large number of applications. Firstly, Recommender Systems are used in Internet commerce to help users to choose the concrete products. Such services collect information about the benefits of users and try to offer them useful products. The best examples of companies using this approach are Amazon, LinkedIn, eBay, iTunes and others. Another important usage area user chooses books, music and movies. For example, Pandora, GoodReads, IMDb, Netflix and Hulu services use Recommender Systems for these purposes.

Now such systems are used extensively in e-commerce to provide recommendations. The objects of recommendations can be goods in an online store, a set of sections of the website, media content, other users of the web service. In modern conditions of big data, Recommender Systems are an indispensable mechanism for content retrieval. The saving time and the ease of use determine the relevance and necessity of such systems.

The problem of choosing consumer goods on the marketplace.

**The problem of choosing consumer goods on the marketplace.** The competition of products in the modern digital era is becoming more and more rigorous. Customers can easily access to the information about this product over the Internet. In addition, customers can share their opinions on products in the form of ratings or reviews through various web services, such as Amazon. Therefore, instead of relying on traditional TV ads or banners on the Internet, consumers can now view many competing products before they make their final purchasing decision.

Everyone faces the problem of finding and choosing (or choosing and searching, since these are different concepts) both in the world of the Internet and in simple things. Choosing a book, a movie for viewing at night, a household product, and even a modern gadget, without which it is difficult to imagine the life of modern society. And there are so many options, especially when you do not really know what you want, and even if you know, but you cannot try. In the modern world, there are many options and alternatives, from a variety of products from different suppliers in different marketplaces. But every vendor, recommends basically what he needs to sell, but not what the user could appreciate.

In the process of searching the user faces the problem of choice (and vice versa, after the choice of the problem of finding is arisen) a product that will satisfy its needs. Recommender Systems are used to solve this

problem. So, Recommender System (sometimes replacing "system" with a synonym such as platform or engine) is a subclass of information filtering systems that seeks to predict the "rating" or "preference" a user would give to an item.

**Recommender Systems.** Recommender Systems appeared on the Internet a long time ago, about 20 years ago. Recommender Systems – the class of information and search systems that allow to predict which objects will be interesting and useful to the users (provide recommendations to users), based on specific information about the user's profile. Nowadays, there are several (main) approaches of creating recommendations, which used in Recommender Systems:

- content-based filtering;
- collaborative filtering;
- hybrid approaches.

Content-based filtering constructs recommendations on the basis of a user's behavior and objects. For example, user profiles may include demographic information or answers on a specific set of questions, and object profiles may include genre names, actor names, artist names, and other attribute information, depending on the type of object.

Collaborative filtering constructs recommendations that's based on a model of prior user behavior. For example, such as purchase or estimation information. In this case, it does not matter what kind of objects were taken, but it can take into account implicit characteristics that would be difficult to include while creating a profile [1]. The model can be constructed solely from a single user's behavior or, more effectively, also from the behavior of other users who have similar features. When it takes other users' behavior into account, collaborative filtering uses group knowledge to form a recommendation based on like users. In essence, the recommendations are based on the automatic collaboration of multiple users and filtered by those who exhibit similar preferences or behaviors.

Hybrid approaches that combine collaborative and content-based filtering are also increasing the efficiency (and complexity) of Recommender Systems. In the process of work (hybrid approach), Recommender Systems collect data about users using a combination of explicit and implicit methods [2].

Examples of explicit data collection:

- to request the user to evaluate the object on a differentiated scale;
- to request a user to rank a group of objects from the best to the worst;
- representation of the two objects with the question of which of them is the best;
- suggestion to create a list of your favorite user objects.

Examples of implicit data collection:

- observing what the user is inspecting in online stores;
- tracking the online user behavior;
- tracking the contents of the user's computer.

Analysis of various sources of information showed that at the moment, shows that that the collaborative filtering methods are divided into three main groups.

The first group calculates the different similarity measures or similarities between the objects under consideration. Pearson correlation and cosine measure of similarity are the most widely used.

The second group of methods is the methods of intelligent data analysis, which involves the various methods of machine learning. The choice of method in this case depends on the domain area, the available statistical material and the power of the computer system. The last group of methods are a hybrid approaches that involves the use of methods, both from the first group and from the second, applying them in various combinations [3].

**Objectives.** Nowadays, the modern recommendation mechanisms are actively used on most popular social and business websites. They bring tremendous benefits to the owners of these sites and their users.

Most large-scale commercial and social websites recommend their users various offers, such as goods or services for further study of people, specifically their desires and needs. Intelligent reference systems (recommendation mechanisms) handle huge volumes of data to identify potential user benefits.

Consequently, web-based Recommender Systems became relevant to the development and active use in the trade of modern Internet technologies, including e-commerce platforms. Their feature, as noted, is to help the user to find the best alternatives in a multitude of web resources. Recommender Systems are primarily beneficial to users, because through such systems, they easily and quickly find the right, specific and useful information without spending much time for searching.

The purpose of this work is to develop a prototype of software for the Recommender System of consumer goods.

**Explanation of the choice of methods for the Recommender System.** Firstly, it is need to choose a recommendation algorithm that is needed to handle implicit user ratings. The data set consists entirely of the interaction between users and consumer goods. It does not contain information about the system users themselves or about products other than their names.

Therefore, we need an algorithm that could be trained without access to the attributes of the user and the product. This type of algorithms is called collaborative filtration algorithms. For example, the assumption that two users have similar tastes, because they are of the same age, or from one city, it is not collaborative filtering. And the decision that two people might like the same product, as they appreciated whether they already bought many identical or similar goods, is a good example.

A dataset for consumer goods can be enormous because it contains tens of millions of ratings. But in reality it is rather meager because it is sparse. Each user scans and estimates a fairly small percentage – from a few million. And some users can view and estimate only one product

An algorithm is needed to provide acceptable results for such users. Also, the algorithm must be able to scale, because it will have to build large models and, at the same time, quickly generate recommendations [4].

They are usually needed in the near future – within a second. We choose an algorithm from the class of so-called latent factors models, since we are developing a system based on implicit estimates of users. Such mechanisms attempt to explain the observed interactions between a large number of users and products due to the relatively small number of underlying reasons behind them.

The most popular in this class are Alternating Least Squares (ALS) and Singular Value Decomposition (SVD) algorithms. To begin, let's look at ALS, then go to SVD [5].

Two of the most popular recommendation algorithms: "SVD" i "ALS" factorize "user-product" matrix  $A$  on two: matrix "user-property"  $U$  and "property-goods"  $M$ .

SVD – "Singular Value Decomposition", method of decomposition of the matrix:

$$A = UDV^T, \quad (1)$$

where  $A$  – decomposition of the matrix;

$U, V$  – orthogonal matrices;

$C$  – orthogonal matrix.

Calculations by method SVD take more time and it's harder to parse them on a computer [6]. It also does not process very well the matrix with the missing values. When the dataset for the reference system is sparse, the missing values are equal to zero, even if the user could potentially give them the highest score.

In contrast to this mechanism, the algorithm ALS (Alternating Least Squares) works much faster, but with less precision [6]. This applies to any factorization method:

$$A \approx XY^T, \quad (2)$$

where  $A$  – factorization method;

$X, Y$  – matrices with low ranks.

One of the advantages of this algorithm is the fact that it has the ability to precisely customize what will be meant by the loss function. (The loss function is a function which minimized in the model fitting process and represents the chosen measure of disagreement between the observed data and the data "predicted" by a fitted function. For example, in most traditional methods for building generic linear models, the loss function (often called the smallest squares) is calculated as the sum of squares of deviations from the fitted line or plane) [7]. It is possible to ignore missed values. Since, it is necessary to handle a rather abundant number of records, it will be necessary to parallelize the calculation. Also very important is the speed of the algorithm. And for this, the ALS method is better.

Designing the Recommender System of consumer goods. A software product which is developed in this

project is a component of a complex large-scale system, data collection and processing, which helps to make profitable purchases of goods on various online resources and an independent software product, to provide high-quality recommendations to the user of the system.

The purpose of this project is to automate the process of processing and analyzing data to provide users with a system of up-to-date, high-quality, and non-trivial recommendations contained in the system database. Data in the system database can be entered by administrators of this system, whether gathered with help of software products from different sources of information, for example such as trading platforms.

For complete operation, the software is developed taking into account the ability to process a large number of requests from users of the product in the context of continuous updating of the data, as well as its reliability and security, openness and convenience for further improvements and parallelism.

This software product is developed as an independent software, but can interact with larger software systems by further developing REST services. The component consists of two parts: the server part and one web portal. The web portal will be used to provide recommendations to the user and to view information about them, and also used for information management. The server part performs the creating of recommendations and is responsible for the system as a whole.

The connection between the web portal and the component being developed on the server is performed through HTTP requests. During the initial request to the server, the component receives information from the database immediately about the existing recommendations, and if the data is successfully received – the information is returned to the client in response to the request. The creating of recommendations occurs with every update of the database to the actual data.

On the Fig. 1 component-deployment diagram is shown.

This software has two modes of operation: the initial

provision of recommendations to the user when the user has not published any reviews and recommendations based on collaborative filtering, provided that the user has published at least one review (fulfilled the grade) of the product.

The user receives topical recommendations, but if he is not satisfied with certain parameters or objects, it is possible to edit the request using filters. In this way, the software product can update its recommendations and provide more relevant information for the user.

To store the finite data in this software was used DBMS PostgreSQL, which based on SQL. Current version is 9.5, there are the following restrictions in PostgreSQL.

Current version is 9.5, there are the following restrictions in PostgreSQL:

- maximum database size – no restrictions;
- maximum table size – 32 TB;
- maximum recording size – 1,6 TB;
- maximum field size – 1 GB;
- maximum records in the table – no restrictions;
- maximum fields in the record – from 250 to 1600, depending on the types of fields;
- maximum of the indices in the table – there are no restrictions.

The strong points of PostgreSQL are:

- high-performance and reliable transaction and replication mechanisms;
- extensibility the system of embedded programming languages;
- inheritance;
- extensibility scalability [8].

**Features of software implementation, the system which is being developed.** At present, the Python programming language is very popular. The Python core syntax is minimal. At the same time, the standard library includes a large amount of useful features. Python supports several programming paradigms, including structural, object-oriented, functional, imperative and

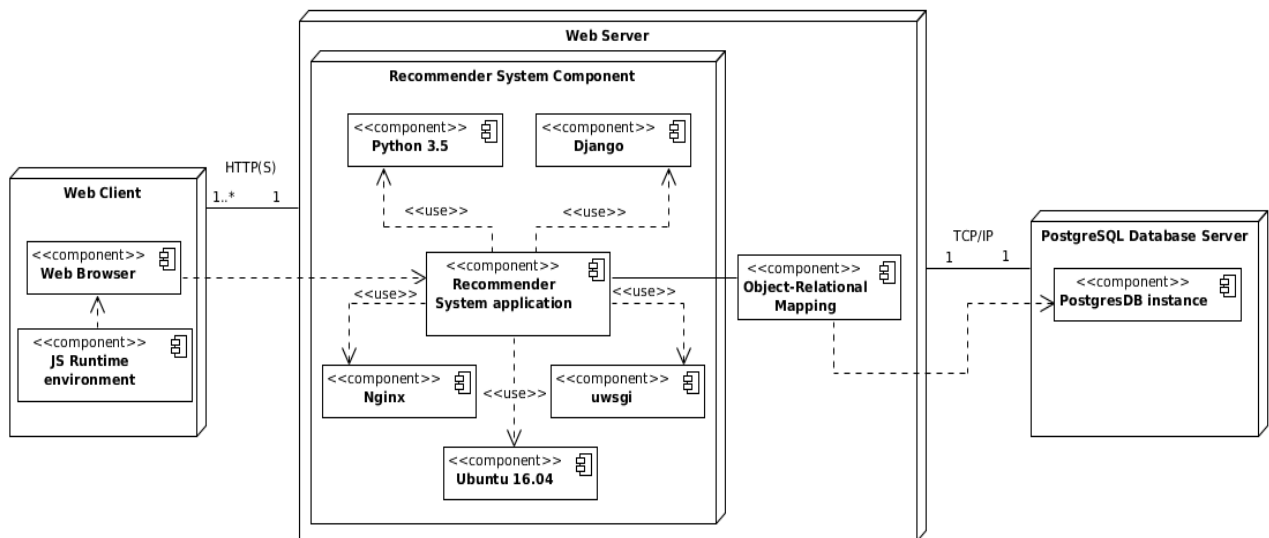


Fig. 1. Component-deployment diagram

aspect-oriented. The main architectural features are dynamic typing, automatic memory management, full introspection, exception handling mechanism, multithreaded computing support, and convenient high-level data structures. The code in Python is organized into functions and classes that can be merged into a module (which in turn can be grouped into packages). Python with packages NumPy, SciPy i MatPlotLib is actively used as a universal environment for scientific calculations as a substitute for common specialized commercial packages. Matlab, IDL and others.

Also, with help of Python, was developed a popular web framework that supports abstraction from low-level databases – Django. Django – this is a high-level web-based framework implemented on the basis of architecture MVC. MVC (Model-View-Controller) – one of the most fundamental architectures for applications, which separates the basic functionality of the application into a number of individual components. This achieves the main goal: one model for many applications. Django has a transparent design, provides for the rapid development of web-based applications, allows the development of dynamic web-sites [8].

In the area of distributed data analysis, the Hadoop framework uses demand, but there are alternatives that offer some important advantages over the typical Hadoop platform. Spark is a scalable data analysis platform that includes primitives for computing in RAM, and therefore has some performance benefits in relation to the Hadoop approach based on a clustered data storage scheme. Spark is implemented on Scala and supports this language Python and Java and provides a unique environment for data processing [9]. Spark is a clustered open source computing platform similar to the Hadoop, but with some useful features that make it an excellent tool for machine learning tasks. Namely, in addition to interactive queries, Spark supports distributed data sets in RAM, optimizing solutions for iterative tasks and reducing access time to data. Spark is implemented in Scala, but freely uses Python as an application development environment. Unlike Hadoop, Spark and Python form a tight integration where Python can easily manipulate distributed data sets as local collective objects [10].

Based on the benefits listed above, the Apache Spark framework and the Python language for creating the recommendations and the framework for implementing the web application for the development of the software system were used [11].

**Review of existing principles (approaches) and problems.** The content-based Recommender Systems try to find a similarity between the goods that people valued earlier, and only products that have a high degree of generality with consumer preferences will be recommended. Modern approaches of obtaining information require creating profiles of customers with tastes, preferences, needs, etc. Information for profiles can be obtained directly from the customer, for example, through questionnaires, or indirectly by analyzing the actions performed by the user.

The limitations of content techniques are related to the objects of recommendations. Therefore, for an

adequate work progress of the system a form is available – for automatic machine analysis, or manual assignment of all parameters.

Another problem is that two different objects, with the same set of properties, are indistinguishable. Although text documents are usually represented by the most representative keywords, content-based systems cannot distinguish a well-written article from poorly written, although they use the same words [12].

Also, the content-based Recommender Systems think too narrowly. The user receives recommendations only for those goods that are similar to goods that have already received his evaluation. To solve this problem, the random technique is used. In some cases, it is necessary to avoid the recommendations of items that are too similar to those already known, for example, another article on the same topic.

As in all Recommender Systems, there is a "cold start" problem. The user needs to evaluate a fairly large number of different products before the system can correctly understand his preferences and give him appropriate recommendations. Therefore, the system will not be able to give exact recommendations to a new customer who has made very few estimates.

An alternative to the content approach can be the collaborative filtering. This is a method of recommendation, in which only the reaction of users to objects is analyzed. The ultimate goal of the method is to as accurately as possible prediction of the estimation that the current user would put to the previously unvalued objects. The more estimates are collected, the more accurate the recommendations are. It turns out that users help each other in filtering objects. Therefore, this method is also called joint filtration.

Methods of collaborative filtering also face problems during operation. The first problem is that new products are regularly added to the Recommender Systems. Such systems in the development of recommendations are guided only by the preferences of users. Therefore, the Recommender System cannot recommend the product until it receives a sufficient number of estimates [12].

In any Recommender System, the number of estimates to be predicted is usually much larger than the number of estimates. It is important that the system is able to effectively predict estimates based on a small number of examples. It is also necessary to have a critical number of users. For example, in advisory systems dealing with movies, a large number of films can only be evaluated by a small number of users, and then these films will be recommended very rarely, even if the estimates of these few users were high. A small number of recommendations can be made to the owners of unusual tastes in comparison with the taste of the majority, for which there are no similar users in the system. Overcoming the problem of sparse estimates can, if searching for similar users use the information about the user, contained in his profile. Most of the recommendations are based on a limited understanding of users and goods, the analysis of which is mainly limited to the information contained in the profiles. Outside the analysis remain information contained in records of user transactions and other available

information. For example, traditional collaborative algorithms do not use information from user and product profiles at all, but they are limited only to information about the assessments made. The profiles themselves are still too primitive [13].

Most of Recommender Systems require a fairly active participation from the user. For example, before issuing a recommendation on a newsgroup, the system needs to obtain estimates of a large number of previously read articles. Since this way of obtaining information is not very user-friendly, methods of indirect evaluation of articles are created. For example, you can analyze the time that a user spent for reading an article, which would indirectly correspond to one or another evaluation. However, indirect estimates often suffer from inaccuracy. Thus, the problem of reducing the obsession of evaluations while maintaining the high quality of recommendations is quite acute for the developers. In particular, it is necessary to understand what the minimum number of evaluations is required from the user, so that this is sufficient for the development of accurate recommendations.

In addition, the current recommendation systems operate in the two-dimensional user-commodity space. This means that they issue recommendations based solely on the information about the user or product and bypass the party contextual information that may be of paramount importance in some applications (and in some special conditions). For example, in many cases, the utility of a product or service may depend on when the consumption occurs (time of year, day of week, time of day). Utility can also depend on who, in what company, under what conditions of the product is consumed.

In such cases, a simple product recommendation to the customer is not enough; when developing the recommendation, the system should refer to additional contextual information about the time and conditions of the intended consumption. In addition to the traditional methods of building a customer profile (such as reliance on keywords and questionnaire demographic information), new techniques have recently emerged, based on automatic word processing, network behavior analysis, etc. These methods allow you to take into account the interests and preferences of users and thereby expand the user profile [13].

In the literature, other approaches of collaborative filtration based on different models were proposed. Other models of collaborative filtering include Bayesian analysis, probabilistic relativistic model, linear regression model, maximum entropy model. Recently a large number of papers have been devoted to the search for more sophisticated probabilistic models of collaborative filtering.

As a results, such problems of Recommender Systems was defined:

- a huge amount of data;
- unfair user's estimations;
- "cold start" – no data about recently added users or objects to the system;
- rarefied ratings;

- difficulty in calculations while the process of working with large databases;
- recommendations of something fundamentally different.

**Conclusions.** In recent years, significant progress has been made in the development of Recommender Systems. Content-based, collaborative filtering and hybrid algorithms for developing recommendations were proposed. Some systems have found practical application in the commercial industry. Nevertheless, despite the progress, for a more efficient work in a large list of applications, the current generation of Recommender Systems requires further improvements. In this paper, various constraints faced by modern recommendatory methods were considered, and a review of necessary improvements was made that should make the work of the recommendation systems more effective. Such improvements include, among others, improved modeling of users and goods, the inclusion of contextual information in the recommender process, the possibility of multi-criteria assessments, the availability of more flexible and less intrusive recommendations.

In this work, the analysis of the subject area and the urgency of the problem of creating a recommendation system for consumer goods was carried out. The task was formulated and methods of creating recommendations were considered: Alternating Least Squares (ALS) and Singular Value Decomposition (SVD) algorithms.

The stage of designing was carried out and the information support of the creation of the Recommender System was developed.

The result of the work is the prototype software of the Recommender System of consumer goods, which can be used to demonstrate the main functions of the system.

#### References

1. George F. Luger *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 4th Edition. Pearson, 2002. 864 с.
2. Башмаков А. И., Башмаков И. А. *Интеллектуальные информационные технологии: учеб. пособие*. Москва: Изд-во МГТУ им. Н.Э. Баумана, 2005. 304 с.
3. Мельник К. В. Применение алгоритма коллаборативной фильтрации для обработки медицинских данных. *Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами*. Харків: НТУ «ХПІ», 2015. № 2. С. 193–198.
4. Brown T. A. *Confirmatory factor analysis for applied research*. The Guilford Press, 2006. 493 p.
5. Ройзнер М. *Як працюють рекомендаційні системи*. URL: <https://habrahabr.ru/company/yandex/blog/241455/> (дата звернення: 15.01.2018).
6. *Parallel Matrix Factorization for Recommender Systems*. URL: [http://www.cs.utexas.edu/~inderjit/public\\_papers/kais-pmf.pdf](http://www.cs.utexas.edu/~inderjit/public_papers/kais-pmf.pdf) (дата звернення: 18.01.2018).
7. *Функція витрат*. URL: <http://statistica.ru/glossary/general/funktsiya-poter>. (дата звернення: 07.02.2018).
8. Яковлев С. Ю. *Програмування на Python*. URL: [https://www.ibm.com/developerworks/ru/library/l-python\\_part\\_11/index.html/](https://www.ibm.com/developerworks/ru/library/l-python_part_11/index.html/) (дата звернення: 17.02.2018).
9. Johns T. Spark. *Альтернатива для швидкого аналізу даних*. URL: <https://www.ibm.com/developerworks/ru/library/os-spark/> (дата звернення: 21.02.2018).
10. Pentreath N. *Machine Learning with Spark*. Birmingham: Packt Publishing, 2015. 322 p.
11. Королева Д., Філіпов М. *Аналіз алгоритмів навчання рекомендаційних систем*. URL:

- <http://engjournal.ru/articles/816/816.pdf> (дата звернення: 23.01.2018).
12. *Рекомендаційні системи: Частина 1. Введення в підходи і алгоритми.* URL: <https://www.ibm.com/developerworks/ru/library/os-recommender1/index.html> (дата звернення: 23.01.2018).
  13. Гончаров М. *Data Mining: Рекомендаційні системи.* URL: [http://kek.ksu.ru/EOS/WM/50\\_132-670.pdf](http://kek.ksu.ru/EOS/WM/50_132-670.pdf) (дата звернення: 23.01.2018).
  6. *Parallel Matrix Factorization for Recommender Systems.* Available at: [http://www.cs.utexas.edu/~inderjit/public\\_papers/kais-pmf.pdf](http://www.cs.utexas.edu/~inderjit/public_papers/kais-pmf.pdf) (accessed: 18.01.2018).
  7. *Funktsiia vtrata.* Available at: <http://statistica.ru/glossary/general/funktsiya-poter> (accessed: 07.02.2018).
  8. Yakovlev C. *Prohramuvannia na Python Python* [Programming on Python]. Available at: [https://www.ibm.com/developerworks/ru/library/l-python\\_part\\_11/index.html/](https://www.ibm.com/developerworks/ru/library/l-python_part_11/index.html/) (accessed: 17.02.2018).
  9. Johns T. Spark. *Alternatyva dlia shvydkoho analizu danykh* [An alternative to fast data analysis]. Available at: <https://www.ibm.com/developerworks/ru/library/os-spark/> (accessed: 21.02.2018).
  10. Pentreath N. *Machine Learning with Spark.* Birmingham: Packt Publishing, 2015. 322 p.
  11. Koroleva D., Filipov M. *Analiz alhorytmiv navchannia rekomendatsiinykh system* [Analysis of the training algorithms for reference systems]. Available at: <http://engjournal.ru/articles/816/816.pdf> (accessed: 23.01.2018).
  12. *Rekomendatsiini systemy: Chastyna 1. Vvedennia v pidkhody i alhorytmy* [Advisory Systems: Part 1. Introduction to Approaches and Algorithms]. Available at: <https://www.ibm.com/developerworks/ru/library/os-recommender1/index.html> (accessed: 23.01.2018).
  13. Goncharov M. *Data Mining: Rekomedatsiini systemi* [Advisory Systems]. Available at: [http://kek.ksu.ru/EOS/WM/50\\_132-670.pdf](http://kek.ksu.ru/EOS/WM/50_132-670.pdf) (accessed: 23.01.2018).

#### References (transliterated)

Received 05.05.2018

#### Відомості про авторів / Сведения об авторах / About the Authors

**Турецький Андрій Олегович (Турецький Андрей Олегович, Turetskyi Andrii Olehovych)** – Національний технічний університет «Харківський політехнічний інститут», студент; м. Харків, Україна; ORCID: <https://orcid.org/0000-0003-3471-9657>; e-mail: [android.95t@gmail.com](mailto:android.95t@gmail.com)

**Ворона Борис Михайлович (Ворона Борис Михайлович, Vorona Borys Mykhailovych)** – Національний технічний університет «Харківський політехнічний інститут», студент; м. Харків, Україна; ORCID: <https://orcid.org/0000-0002-9649-0264>; e-mail: [borysvorona@gmail.com](mailto:borysvorona@gmail.com)

**Вовк Марина Анатоліївна (Вовк Марина Анатольевна, Vovk Maryna Anatoliivna)** – кандидат економічних наук, доцент кафедри програмної інженерії та інформаційних технологій управління, Національний технічний університет «Харківський політехнічний інститут»; м. Харків, Україна; ORCID: <https://orcid.org/0000-0003-4119-5441>; e-mail: [marihavovk@gmail.com](mailto:marihavovk@gmail.com)

**Єршова Світлана Іванівна (Єршова Светлана Ивановна, Yershova Svitlana Ivanivna)** – старший викладач кафедри програмної інженерії та інформаційних технологій управління, Національний технічний університет «Харківський політехнічний інститут»; м. Харків, Україна; ORCID: <https://orcid.org/0000-0003-3893-117X>; [svetlana.ershova.2016@gmail.com](mailto:svetlana.ershova.2016@gmail.com)