*Advances in Social Science, Education and Humanities Research, volume 447*
**Proceedings of the International Scientific Conference on Philosophy of Education, Law and Science in the Era of Globalization (PELSEG 2020)**

# Recommendation Assistant System for Social Networks and Search Services Based on Population Filtering Algorithm

Sergey Rodzin
Southern Federal University
Taganrog, Russia
srodzin@sfedu.ru

Victoria Bova
Southern Federal University
Taganrog, Russia

Yuri Kravchenko
Southern Federal University
Taganrog, Russia

Lada Rodzina
Southern Federal University
Taganrog, Russia

Olga Rodzina
Southern Federal University
Taganrog, Russia

Elmar Kuliev
Southern Federal University
Taganrog, Russia

***Abstract* — The authors present a hybrid model of a recommender system. The system includes the characteristics of collaborative and content filtering. Also, the article describes a population filtering algorithm and the architecture of a recommendation system based on it. The results of experimental studies on an array of benchmarks and an estimation of filtering efficiency based on a hybrid model and a population algorithm are presented. The results are compared with the traditional method of collaborative filtering using the Pearson correlation coefficient.**

***Keywords — recommendation system; collaborative and content filtering; population algorithm.***

## I. INTRODUCTION

On the Internet, recommendation systems in social networks, online stores, and search services are widespread. These systems help a person choose which new films to watch, what new music to listen to, what products to buy in the online store. In social networks, recommendations of friends, acquaintances, people with similar interests are common. The recommendation systems help the user choose among all the available objects exactly those that will be of interest to him based on the "votes" of the community.

The amount of information, goods in online stores, books, films, videos, games, as well as the number of other users on social networks is so large that it is unrealistic for the user to consider all available offers. Due to this, we can meet more and more often with systems that analyze the user profile and try to predict what will be most interesting for them at a given time. For example, under each item in the online store there you can see such an option "similar products", or "maybe you want to purchase ", or "usually, buy with this product". Referral systems are widely used on e-commerce websites to offer customers items (goods, films, services, etc.). They help users to choose from a large number of offers. One of the easiest ways is to recommend items that are the most popular. However, such an approach does not take into account the fact that users may have different tastes. Recommender systems should solve the problems of targeted promotion of goods and services taking into account specific user preferences based on the logging of data on customer actions. This data can have a significant amount, changes rapidly, and updates over time. The task of adapting to a specific user is complicated by the inherent uncertainty in the choice of a specific Internet resource and the particularities of the Internet.

Traditionally, recommendation systems are divided into the following types [1]:

- collaborative filtering based on ratings from other users. This type is characterized by high accuracy. However, without knowing anything about the user's preferences, these recommendation systems have a high entry threshold;
- content filtering based on data collected about each item or service. This type of system can make recommendations to new users, even without the ratings of other users, but the accuracy of their forecasts is not high, and the development time increases;
- expert filtering based on knowledge about the subject area, and not on information about the subject. This type of system is distinguished by the high complexity of developing and collecting data;

 

- hybrid systems, the basis of which is the algorithmic composition of individual filtering strategies (weighted, a combination of features from various sources, cascading, meta-learning, etc.).

Collaborative filtering (imhonet.ru, *last.fm*) is based on the principle of searching for similar users and ascertaining their preferences or based on searching for goods similar to a given one and ascertaining their ratings previously made by the user. One of the promising methods in the field of collaborative filtering is the SVD-method based on singular matrix decompositions.

Content filtering (Prismatic, Surfing bird) is based on knowledge of any properties of objects, regardless of their evaluation by other users. At the same time, the user does not need to "train" the system for his/her preferences.

Expert filtering (large online stores) is based on manually created associative rules, created and edited by qualified experts.

The quality of recommendation systems is judged by accuracy and completeness [2]. Accuracy is the percentage of items or services that are preferred by the user among all suggested recommendations. Completeness is the proportion of items or services recommended by the user, among all preferred for him.

We have made a hypothesis that a combination of collaborative and content filtering results potentially improves recommendations accuracy. A hybrid approach is useful if collaborative filtering begins when data sparseness is significant. In this case, the hybrid approach allows you to first weigh the results according to content filtering and then shift these weights towards collaborative filtering as data about a particular client [3].

## II. MATERIALS AND METHODS

### A. Problem Statement

Formally, the task of finding the recommended object can be represented as follows:

$$\forall u \in U, s'_u = \arg\max_{s \in S} h(u, s), \quad (1)$$

where $U$ is a set of users, $S$ is a set of objects that can be recommended for a user, $h$ is a function that determines how much some object $s$ is satisfied for some user $u$. Thus, it is necessary to select such an object $s' \in S$, at which satisfaction value for each user $u \in U$ is max.

The main issues of recommendation systems in solving the defined task are the following:

- sparse data. The recommendation system operates with a large amount of data, while users do not give ratings to most objects. As a result, the "object-user" matrix is very large and sparse, which makes it difficult to calculate recommendations and strengthens the problem of a "cold" start;
- "cold" start. New objects, new users present a big problem for recommendation systems due to the lack of data on users or objects recently appearing in the system;
- scalability. With the increase in the number of users in the system, the scalability problem appears. The collaborative filtering algorithm with complexity as *O(M*N)*, where *M* is the number of clients, *N* is a range of objects, where M = 107 and N = 106 are already too complicated for calculations. This requires greater scalability of the filtering algorithms since the recommender system must instantly respond to online requests from all customers, regardless of their purchase history and ratings;
- synonymy. Similar or identical objects have different names, and the recommender system is not able to detect these hidden connections and assigns these objects to different objects;
- fraud. Unscrupulous suppliers are trying to fraudulently raise the rating of their goods or services and lower the rating of their competitors;
- diversity. Algorithms based on sales and ratings create difficult conditions for the promotion of new and little-known objects since they are replaced by popular offers that have long been on the market;
- "black sheep". These are users whose opinions do not always coincide with the opinions of most others. Due to their unique taste, it is impossible to recommend something related to them.

Almost all modern Internet resources, including search engines, online stores, forums, and social networks focused on working with a large number of users, collect and log information about their activity. As well, they process it to personalize their content for each specific client. Information and data on user activity are characterized by a large volume and heterogeneity. The main goals of processing user activity data are clustering users and resources, building behavioral and socio-demographic profiles of users, segmenting the customer base, providing recommendations, the most interesting information and marketing offers.

### B. Hybrid Request Model

In recommendation systems [4, 5] we can see hybrid models based on collaborative and content filtering. These models can be classified as follows:

1. Building a unified model using the characteristics of collaborative and content filtering models.

2. Implementation of some characteristics of the content model in systems based on collaborative filtering.

3. Implementation of some characteristics of the collaborative filtering model in systems based on the content model.

4. Implementing collaborative and content filtering models separately and using combinations of recommendations received.

The basic principle of building a hybrid recommendation model is to use the filtering characteristics of content and collaborative filtering in one recommendation system [6].

The introduction of some characteristics of the content model into collaborative filtering systems implies that such hybrid recommender systems are not only built on a collaborative component but also include some content filtering data in the user profile. These data serve as the basis for calculating the similarity of user preferences instead of the overall evaluated objects [7, 8]. Not only objects with good reviews from other users are recommended for the user, but also those objects that the user may like based on his personal preferences.

The introduction of some characteristics of the collaborative filtering model into systems based on the content model implies a decrease in the dimension of profiles using content filtering.

Separate implementation of recommendation systems involves either a combination of ratings from two systems using a linear combination of ratings or the use of a system that should be better suited in a particular case.

The proposed hybrid filtration model is based on the application of the population algorithm as a machine learning method to solve the problem [9, 10]. A variety of input data in the hybrid model is supported by a population of custom "characteristics" encoded in a population algorithm.

Using a population of encoded solutions and special operators, the algorithm searches for a solution with the highest value of the objective function. The weighted values of the objects remain relevant, "noise" is eliminated. A method is proposed for solving the problems of sparse data and a cold start by encoding the "preference" additional component in solutions.

### *C. Population Algorithm for Collaborative Filtering*

A population algorithm explores the search space, synthesizes the solutions that are points of this space, requests an assessment of their quality or "fitness", which is then used to make "natural selection". In this way, they learn which areas of the search space contain the best solutions. Imagine the general scheme of a population filtering algorithm.

Let us suppose that in a finite state space $S$ we have some function $f(x)$, $x \in S$. We have to find $max\ \{f(x);\ x \in S\}$. As well, suppose that $x^*$ is a state with the maximum of the function $f_{max} = f(x^*)$.

The population algorithm for solving the problem includes the following steps.

Step 1. *Initialization* of a population (randomly or heuristically) $\xi_0 = (x_1, \ldots, x_{2n})$ from $2n$ individuals ($n$ is an integer). Assign $k = 0$. For each population $\xi_k$ defines

$$\xi_k = max\ \{f(x_i): x_i \in \xi_k\} \quad (2)$$

Step 2. *Generation* of a population $\xi_{k+1/2}$ using special operators.

Step 3. *Selection* and *reproduction* $2n$ individuals from population $\xi_{k+1/2}$ and $\xi_k$ and getting a new population $\xi_{k+S}$.

Step 4. If $f(\xi_{k+S}) = f_{max}$, then *stop*, otherwise $\xi_{k+1} = \xi_{k+S}$, $k = k + 1$ and move to step2.

Suppose $x^*$ is an optimum point. Define $d(x, x^*)$ as distance between points $x$ and $x^*$. If we have a set of optimums $S^*$, then $d(x, S^*) = min\{d(x, x^*): x^* \in S^*)\}$ is a distance between point $x$ and the set $S^*$. Such distance let us define as $d(x)$. Then $d(x^*) = 0$, $d(x) > 0$ for each $x \notin S^*$.

Considering that populations $X = min\{x_1, \ldots, x_{2n}\}$, suppose that

$$d(X) = min\{d(x): x \in X\} \quad (3)$$

This formula measures the distance between a population and the optimal solution.

Sequence $\{d(\xi_k);\ k = 0, 1, 2, \ldots\}$, generated by the population algorithm is a random sequence that is modeled by a homogeneous Markov chain.

Then the drift of a random sequence at time $k$ is defined as

$$\Delta(d(\xi_k)) = d(\xi_{k+1}) - d(\xi_k) \quad (4)$$

The algorithm shutdown time is estimated as

$$\tau = min\ \{k: d(\xi_k) = 0\} \quad (5)$$

The task is to study the relationship between time $\tau$ and the dimension of the problem $n$. At what drift values $\Delta(d(\xi_k))$ we can evaluate the expected value $E[\tau]$? The key issue here is the evaluation of the ratio $d$ and $\Delta$.

The population algorithm can solve the problem in a polynomial average time under the following drift conditions:

- if there is a polynomial $h_0(n) > 0$ ($n$ is dimension of the problem) such as $d(X) \leq h_0(n)$ for any defined population $X$, in other words, the distance from any population to the optimal solution is a polynomial function of the dimension of the problem;
- in any moment $k \geq 0$, if $d(\xi_k) > 0$, then the polynomial $h_1(n) > 0$ is exist, such as

$$E[d(\xi_k) - d(\xi_{k+1})|\ d(\xi_k) > 0] \geq 1/h_1(n) \quad (6)$$

in other words, random sequence drift $\{d(\xi_k);\ k = 0, 1, 2, \ldots\}$ is always positive towards the optimal solution and bounded by the inverse polynomial.

In other words, the estimate of the drift value is converted into an estimate of the running time of the algorithm, and the local property (drift in one step) is converted to a global property (the running time of the algorithm until the optimum is found)! It's easier to evaluate the drift - using the drift analysis, the conditions are defined, the fulfillment of which guarantees the solution of the filtering problem in polynomial time.

Note that this result was obtained under the assumption that the number of generations (which is equivalent to the number of calculations of the objective function) is the most important factor in estimating the computation time of the algorithm. This is probably true for most applications of population-based algorithms because the estimation of the

objective function there is the most time-consuming part of the algorithm, in contrast to the complexity of executing the operators of population-based algorithms, the complexity of which is estimated as $O(n) – O(n\ ln(n))$ [9].

How are solutions have encoded? Let us consider the structure of the coding solutions using the example of a recommendation system for watching movies. It has the form presented in Table 1.

### *D. Structure of Coding Solutions on the Example of a Recommendation System for Watching Films*

The structure of coding solutions on the example of a recommendation system for watching films is presented in Table 1.

TABLE I. CODING SOLUTIONS FOR A MOVIE RECOMMENDATION SYSTEM

| | | | | **9 parameters describing user profile** | | | **18 genres** | | | **9 additional movies' characteristics** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Rate** | **Year** | **Gender** | **Profession** | ***genre preferences*** | … | ***language*** | ***thriller*** | … | ***western*** | ***director*** | … | ***movie language*** |
| **w** | w | w | w | $w_5$ | … | $w_{13}$ | $w_{14}$ | … | $w_3$ | $w_{32}$ | … | $w_{40}$ |
| ***1*** | *2* | *3* | *4* | | | | | | *1* | | | |

The structure includes 40 elements: $w_1$ – rate, $w_2$ – launch year, $w_3$ – user's gender, $w_4$ – profession, $w_5 – w_{13}$ – nine genes that describe a user profile (from genre preferences to film language), $w_{14} – w_{31}$ – 18 movie genres, $w_{32} – w_{40}$ – 9 additional features. The more "weight" $w_i$, the more important this parameter is for evaluating user preferences.

This structure of decision coding in a population algorithm using the example of a recommendation system for watching films covers the entire range of input data.

### *E. Population Algorithm Architecture*

For the experiments, a prototype recommender system was built based on the population algorithm. The system implements the collection, analysis and use of ratings of some users to predict interest in films of other users. The prototype architecture of the recommender system is shown in Figure 1.

The recommendation system process is divided into three stages.

At the first stage, the user is registered, demographic data is collected, as well as information on the rating of films that he has already watched or bought.

In the second stage, the user profile is personalized. For those films that the user has already rated, a search is made from the database (DB) of the corresponding genres and characters of films that will be taken into account as preferences, along with ratings of other users. When recommending a film, the preferred types of elements are film genres, preferred directors, actresses and actors, producers, screenwriters, and language. This is especially effective when user ratings are not yet formed.

In the third stage, a population algorithm is used to search for the appropriate recommendation.

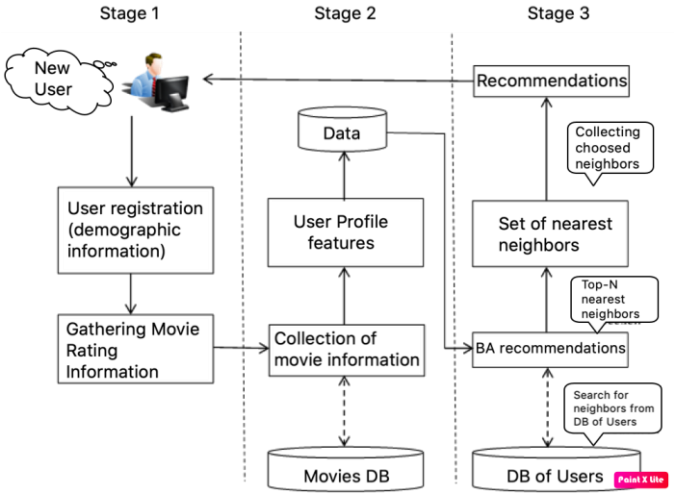


Fig. 1. Recommended system prototype architecture

## III. RESULTS

For the experiments, a data set *MovieLens* 1*M* [11] was selected. This set contains data on 1,000,000 ratings 6,040 users 3952 films. As contextual descriptions of objects dataset were taken the dataset *TagGenome*. Estimation of filtration efficiency based on a hybrid model and population algorithm was carried out by comparison with the traditional method of collaborative filtering using the Pearson correlation coefficient [12, 13]:

$$corr(user1, user2) = \frac{\sum_{i=0}^{n}(user1_i - \overline{user1})(user2_i - \overline{user2})}{\sqrt{\sum_{i=0}^{n}(user1_i - \overline{user1})^2}\sqrt{\sum_{i=0}^{n}(user2_i - \overline{user2})^2}}$$

where *user1*, *user2* are users and their rates; *n* is a number of object, $\overline{user1}$ and $\overline{user2}$ – is average user rating. Correlation coefficient *corr* takes values from –1 (absolute mismatch) to 1 (absolute match).

The experiments were repeated using various sets of weight parameters to show how they influence the outcome of the recommendations. In a sample of 10 users, the average value of the weight parameters was about 0.2649. Moreover, the weights describing the user profile ($w_5 – w_{13}$) turned out to be more significant than many characteristics of the film

To compare the population algorithm with the collaborative filtering algorithm using the Pearson correlation coefficient, a set of 5 decision coding structures with a different number of parameters was configured for 100 random users (8, 16, 24, 32, and 40). In Table 2 the composition of the parameters is given (√) for the Pearson algorithm and the population algorithm with a different number of parameters (PA(8) – PA(40)).

TABLE II. THE COMPOSITION OF THE FILTERING PARAMETERS FOR THE PEARSON ALGORITHM

| | Parameters | | | | |
|---|---|---|---|---|---|
| | ***User rating*** | ***Year, gender, profession*** | ***9 users profile's options*** | ***18 movie genres*** | ***9 additional movie features*** |
| **Pearson Algorithm** | √ | √ | | √ | √ |
| **PA (8)** | √ | | √ | | |
| **PA (16)** | √ | √ | √ | | |
| **PA (24)** | √ | | | √ | √ |
| **PA (32)** | √ | √ | | √ | √ |
| **PA (40)** | √ | √ | √ | √ | √ |

The prediction accuracy of the population algorithm for PA(8) – PA(40) turned out to be higher than the Pearson algorithm. In particular, the average prediction accuracy by the Pearson algorithm is about 70.1%, while the average accuracy of the population algorithm with different compositions of parameters ranged from 80% to 82%.

A comparison of the processing time of the population algorithm with a different composition of parameters showed that PA(8) exceeds PA(16), PA(24), PA(32) и PA(40). Its average operating time is about 20 seconds. The more parameters the algorithm includes, the longer the processing time.

## IV. CONCLUSION

Recommendation systems are a definite alternative to search algorithms because they allow you to detect objects that cannot be found last. Such systems try to predict which objects (movies, music, books, news, websites) will be of interest to the user, based on the content information about his/her profile and also using the preferences of other users to predict unknown preferences for this user.

The article proposes an approach based on a hybrid model that combines the results of collaborative and content filtering, which allows increasing the accuracy of recommendations to solve the problem of the sparseness of data and "cold" start. The model is based on the use of a population algorithm as a heuristic method for finding a solution. The drift analysis of the population algorithm showed that it solves the problem in polynomial time.

The experiments show that the developed population algorithm is superior in accuracy to the competing collaborative filtering algorithm based on the Pearson correlation coefficient.

The presented hybrid filtration model is static. In reality, the perception and popularity of certain products change over time, as do the tastes of users [14]. The recommendation system should take into account this temporal dynamic. The population approach, in our opinion, allows taking into account the temporal drift, which can improve the accuracy of recommendations.

## Acknowledgment

The reported study was funded by RFBR according to the research project № 18-429-22019 and project № 19-01-00412.

## References

[1] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques a survey of collaborative filtering techniques", Advances in Artificial Intelligence archive, pp. 1–19, 2009.

[2] G. Shani, A. Gunawardana, "Evaluating recommendation systems", Recommender Systems Handbook, pp. 257–297, 2011.

[3] R.Burke, "Hybrid Web recommender system", The Adaptive Web. pp. 377–408, 2007.

[4] P. Lops, M. Gemmis, G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends", Recommender Systems Handbook. pp. 73–105, 2011.

[5] G. Shani, A. Gunawardana, "Evaluating recommendation systems", Recommender Systems Handbook, pp. 257–297, 2011.

[6] S. Rodzin, O. Rodzina, "New computational models for big data and optimization", Proc. of the 9th IEEE Int. Conf. Application of Information and Communication Technologies (AICT'15), pp. 3–7, 2015.

[7] V. Bova, Yu. Kravchenko, S. Rodzin, E. Kuliev, "Hybrid method for prediction of users' information behavior in the Internet based on bioinspired search", Journal of Physics: Conference Series 1333 (ITBI'2019), pp. 1–7, 2019.

[8] Yu. Kravchenko, I. Kursitys, V. Bova, "The development of genetic algorithm for semantic similarity estimation in terms of knowledge management problems", Advances in Intelligent Systems and Computing. vol. 573, pp. 84–93, 2017.

[9] S. El-Khatib, S. Rodzin, Y. Skobtcov, "Investigation of optimal heuristical parameters for mixed aco-k-means segmentation algorithm for MRI images", Proc. of the Conf. on Information Technologies in Science, Management, Social Sphere and Medicine (ITSMSSM'2016), pp. 216–221, 2016.

[10] S. Rodzin, O. Rodzina, "Metaheuristics memes and biogeography for trans computational combinatorial optimization problems", Proc. of the 6th Int. Conference Cloud System and Big Data Engineering, pp. 1–5, 2016.

[11] F.M. Harper, J.A. Konstan, "The movieLens datasets: History and context", On Interactive Intelligent Systems (TiiS), vol. 5, no. 4, pp. 19, 2016.

[12] Y. Koren, R. Bell, C. Volinsky, "Matrix factorization techniques for recommender systems", IEEE Computer, vol. 42, no. 8, pp. 42–49, 2009.

[13] A. Hernando, J. Bobadilla, F. Ortega, "A non-negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model", Knowledge-Based Systems, vol. 97. pp. 188-202, 2016.

[14] P. Boucher-Ryan, D. Bridge, Collaborative recommending using formal concept analysis, Knowledge-Based System, no. 19(5), pp. 309–315, 2006.