DOI: http://doi.org/10.25728/cs.2024.4.4

AN ENTROPY-BASED COMPOSITE INDICATOR FOR EVALUATING THE EFFECTIVENESS OF RECOMMENDER SYSTEM ALGORITHMS

R. S. Kulshin* and A. A. Sidorov**

Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia

*✉ roman.s.kulshin@tusur.ru, **✉ anatolii.a.sidorov@tusur.ru

Abstract. The problem of forming a composite indicator for evaluating the effectiveness of recommender system algorithms is considered. A novel composite indicator is proposed by combining individual metrics using the entropy method. The testing base of this study consists of 12 algorithms (on the one hand) and 3 datasets (on the other). For each algorithm–dataset combination, we calculate partial criteria used in evaluating recommender systems. According to the results presented below, the composite indicator is an effective tool for evaluating the performance of recommender system algorithms. As is shown, the performance of the algorithms varies depending on the size and other basic characteristics of a particular dataset. This indicator can be used to develop more efficient algorithms and their ensembles as well as to optimize hyperparameters and improve the quality of recommendations.

Keywords: recommender systems, composite indicator, algorithms, metrics, datasets.

INTRODUCTION

In a hybrid world, when real social relations are closely intertwined with digital practices of communication and economic activities, recommender systems are becoming in-demand services between various business actors, both the suppliers and consumers of different types of goods and services in a market. For instance, under the constant growth of data volumes, the recipients of various benefits face the need for effective search and filtering of information [1], and their producers are looking for ways to improve the effectiveness of marketing activities to stimulate the consumption of goods and services. In this context, recommender systems provide personalized recommendations in various business segments: from online shopping to music and movie streaming.

In e-commerce, recommendations allow shoppers to find goods matching their interests and preferences. Recommender systems analyze data on previous purchases, goods views [2], and behavioral patterns to suggest the most relevant options. On music and movie streaming platforms, recommendations play a key role when meeting users' interests: personalized playlists and content recommendations are provided based on the analysis of preferences for genres, artists, and film directors and the consideration of ratings and reviews [3].

The technologies underlying recommender systems include machine learning as well as collaborative and content filtering algorithms. Many of them also involve the neural network approach [4] to analyze and predict user preferences more accurately. The demand for such software services and products results in the emergence of many algorithms with two parameters as their effectiveness evaluation metrics: time and memory, characterized by an asymptotic constraint. Big O is the most common method for evaluating algorithms [5]. The Big O notation describes the complexity of algorithms and allows comparing their effectiveness based on homogeneous data.

At the same time, many other metrics characterize the performance of recommendation algorithms. The list of evaluation criteria is quite extensive and includes both typical parameters of machine learning methods and specialized metrics of recommendation algorithms [6]. They measure the accuracy, completeness, diversity, reliability, and understandability of

recommendations, user satisfaction, catalog coverage, query processing speed, and other characteristics. Each metric has its advantages and limitations depending on the context of application and the goals of the recommender system.

The use of various metrics contributes to a better understanding of, first, the generalized aspects of recommendation algorithms that often have no common meaning for the scientific and professional community (performance, efficiency, effectiveness, quality) and, second, the suitability of algorithms for particular tasks and audiences.

In this situation, the following difficulties arise when comparing algorithms and selecting an appropriate one to solve a particular task:

• There are many metrics for evaluating the algorithms underlying recommender systems, each being suitable for different aspects of recommendations. This may produce ambiguous results when different metrics point to different algorithms as the best ones.

• Some metrics may be mutually related or dependent. An improvement in the value of one metric may worsen the value of another one. This complicates the choice of an appropriate algorithm.

• The values of metrics may vary depending on the context. For example, accuracy may differ in different fields of application.

Creating a unified evaluation metric for a recommendation algorithm is a necessary step to ensure the objectivity and comparability of recommendations as well as to improve their quality. This step serves to standardize the evaluation of algorithms' efficiency, select the most appropriate one for a particular task, and optimize resource utilization.

The main hypothesis of this paper is that all recommendation algorithms behave differently depending on the data context, the size of datasets, and other characteristics of the recommendation environment. To avoid the time-consuming empirical selection of an algorithm with a comparison of many metrics for a particular situation, we propose to create a supercriterion for evaluating recommendation algorithms.

1. METHODOLOGY

1.1. Composite Indicator as a Generalized Measure

As a single criterion, we introduce a composite indicator representing an aggregate combining several relevant metrics to measure a complex synthetic construct [7–10]. Nowadays, composite indicators are applied in various fields for evaluating:

– the productivity of companies, markets, and the economy as a whole [11–13],

– the patient's health status [14–16],

– environmental conditions [17–19],

– risks in various fields [20–22],

– the quality of education and student performance [23–25],

– et al.

They have advantages and drawbacks. For instance, the advantages of composite indicators include the following:

• Several individual indicators or variables are combined into a generalized measure, which is convenient for analysis and comparison [7, 9, 10, 24].

• Analysis is conducted based on the logical meaningful union of variables related to a particular field or topic [9, 10, 13].

• Composite indicators can be used in many fields and for different tasks, which makes them versatile tools for analysis and prediction [9, 13].

Speaking about the merits of composite indicators, we have to mention some of their disadvantages:

• The bottleneck of this methodology is the assignment of weights to individual indicators: there exists the possibility of manipulation when justifying the decision based on the calculation results of a given composite indicator [7, 24].

• The use of composite indicators may be limited by the availability of data required to justify the supercriterion structure in substantive terms [7, 13].

Despite these drawbacks, composite indicators can be effective because their applicability in many contexts makes them flexible tools adaptable to different tasks and conditions.

Application of the methodological approach based on calculating a composite indicator is quite disputable, which has led to the emergence of two camps in the scientific community: its opponents and supporters. Despite the diversity of views and controversial points (e.g., selecting the initial list of criteria, methods for determining weights and aggregation, and ways of normalizing indicators measured in different ranges and scales), the composite indicator is a rather convenient and well-interpretable tool for comparing alternatives and making decisions under multiple criteria. Due to its high degree of "customizability," this method can be adapted to different situations and needs.

With the diverse fields of application of this methodological approach and the possibility of working with heterogeneous data, we can use it in a new field, i.e., for comprehensively evaluating the effectiveness of algorithms. By using the composite indicator, we can consider various performance aspects of algorithms and summarize them into a single metric, which seems more convenient when analyzing performance

results described by a set of partial criteria. This will significantly improve the objectivity of the final evaluation.

Composite indicators, successfully used for analysis and evaluation in socio-economic fields [11–25], have not been previously applied to evaluate algorithms. The approach proposed below is novel due to adapting the methodology to the field of information technology. This paper mainly focuses on evaluating the effectiveness of recommendation algorithms, which opens new opportunities for analyzing them and improving their performance.

1.2. Metrics

To form a composite indicator, we select the main evaluation metrics for machine learning algorithms and personalized recommender systems.

• *Average Popularity* (*AP*) is the average popularity of recommended elements (items, objects) [26, 27]:

$$AP = \frac{1}{|U|}\sum_{u\in U}\frac{\sum_{i\in R(u)}\phi(i)}{|R(u)|},$$

where U denotes the set of all users with generated recommendations; $|U|$ is the number of such users; u is a particular user whose data are employed in calculations; $R(u)$ is the set of recommendations; $|R(u)|$ is the number of elements in this set; $\phi(i)$ is the number of recommendations of element i in training data.

High values of *AP* indicate that the system frequently recommends popular elements, which can attract users' attention. On the other hand, excessively high values of this metric may deteriorate the diversity of recommendations.

• *Grouped Area under the Curve* (*GAUC*) characterizes the quality of ranking for all users:

$$GAUC = \frac{1}{\sum_{u\in U}|R(u)|}\sum_{u\in U}|R(u)|\,AUC(u).$$

Here $AUC(u)$ is a quality evaluation metric for models in classification and ranking problems:

$$AUC(u) = \left(|R(u)|(n+1) - \frac{|R(u)|(|R(u)|+1)}{2} - \sum_{i=1}^{|R(u)|} rank_i\right) \Big/ \left(|R(u)|(n-|R(u)|)\right),$$

where $rank_i$ denotes the descending rank of element i of the set $R(u)$; n is the total number of user interactions with the element.

AUC reflects the area under the *Receiver Operating Characteristic* (*ROC*) curve, a graph showing the ratio of the share of correctly classified positive cases to the share of incorrectly classified positive cases when varying the decision threshold [28]. This metric quantitatively estimates the model's capability to discriminate between classes (e.g., positive and negative cases) under different thresholds [29].

GAUC demonstrates how well the model ranks recommendations for users (in other words, how effectively the model distinguishes between recommendations satisfying the user and those not). This metric is applied for tasks where the ranking of suggestions is important, such as goods or content recommendations.

• *Gini Index* (*GI*) characterizes the diversity of the distribution of generated recommendations:

$$GI = \frac{\sum_{i=1}^{|I|}(2i-|I|-1)P(i)}{|I|\sum_{i=1}^{|I|}P(i)},$$

where $P(i)$ denotes the popularity of each element after training the algorithm; $|I|$ is the rating of recommended elements in non-decreasing order.

GI serves to measure how uniformly the recommendations are distributed for each user. A more uniform distribution means that the recommendations are more diverse and meet the different interests of the user. If the distribution is close to ideal, *GI* will take a value near 0. If the recommendations are centered around a small number of elements, *GI* will be closer to the maximum value [30].

• *HitRate* (*HR*) is the ratio of the number of recommendations satisfying the user to the total number of recommendations [31]:

$$HR = \frac{1}{|U|}\sum_{u\in U}|\overline{R(u)}\cap R(u)|,$$

where $\overline{R(u)}$ denotes the set of recommendations satisfying the user.

HR indicates what share of recommendations helpful to the user; the higher its value is, the more successful the recommender system will be.

• *Precision* (positive predictive value) is the share of relevant elements among all recommended ones [32]:

$$Precision = \frac{1}{|U|}\sum_{u\in U}\frac{|\overline{R(u)}\cap R(u)|}{|\overline{R(u)}|}.$$

This indicator shows how accurately the system selects the entities that will be preferable to the user. The higher value *Precision* takes, the more accurate the recommendations will be.

• *Recall* is a measure to calculate the share of relevant elements from the entire set of generated recommendations [33]:

$$Recall = \frac{1}{|U|}\sum_{u\in U}\frac{|\overline{R(u)}\cap R(u)|}{|R(u)|}.$$

It indicates what share of relevant elements the system can consider in the recommendations.

At first glance, *Recall* and *Precision* are similar metrics; meanwhile, they have fundamental differences. *Recall* focuses on detecting as many relevant elements as possible and minimizing missed opportunities; *Precision*, on the accuracy of element selection and minimizing irrelevant recommendations.

A high *Recall* value means that the system covers the user's interests well but may include more noise in the recommendations. A high *Precision* value means that the system provides accurate recommendations but may miss some interesting elements.

• *Shannon Entropy* (*SE*) is the variety of recommendations generated for a user [34]:

$$SE = -\sum_{i=1}^{|I|} p(i)\log p(i),$$

where $p(i)$ denotes the probability of recommending element i.

If recommendations are diverse and cover different user interests, entropy will be high. Entropy estimation in recommender systems can help optimize the balance between personalization (providing recommendations meeting user's unique interests) and diversity (providing recommendations covering a wider range of interests).

• *Mean Average Precision* (*MAP*) is the overall quality of elements ranking:

$$MAP@K = \frac{1}{|U|}\sum_{u\in U}\left(\frac{1}{\min(|\overline{R(u)}|,K)}\times\sum_{j=1}^{|\overline{R(u)}|}|I|\overline{R(u)}_j\ Precision\right),$$

$$\overline{R(u)}_j \in R(u),$$

where K is the truncated number of generated recommendations; j is the index of the recommendation satisfying the user.

The truncated number of generated recommendations is a sample of the most recommended elements, specified by the system developer. It can take any value not exceeding the number of recommended elements. A value of 10 is most widespread in practice.

This indicator is a helpful metric for evaluating the quality of recommender system models, especially when it is important to consider the ranking of recommendations. Each user is treated separately to consider individual preferences and interests. Since *MAP* averages the *Precision* value of all users, this metric provides a generalized evaluation of the recommender system ranking.

• *Mean reciprocal rank* (*MRR*) is the ranking quality of the first element in the list of recommended ones [35]:

$$MRR = \frac{1}{|U|}\sum_{u\in U}\frac{1}{rank_u^*},$$

where $rank_u^*$ is the rank position of the first relevant element found by the algorithm for the user.

This indicator is widespread to evaluate the quality of search engines, recommender systems, and other tasks where it is important to consider the ranking of results. The higher the *MRR* value is, the better the result will be. The *MRR* value will equal 1 if the relevant element is always at the first position in the ranked list.

• *Normalized discounted cumulative gain* (*NDCG*) is a ranking quality indicator that considers the position–relevance relationship of elements in a ranked list [36]:

$$NDCG@K = \frac{1}{|U|}\sum_{u\in U}\left(\frac{1}{\sum_{i=1}^{\min(|R(u)|,K)}\frac{1}{\log_2(i+1)}}\times\sum_{i=1}^{K}\delta(i\in R(u))\frac{1}{\log_2(i+1)}\right),$$

where δ denotes the indicator function (if $i\in R(u)$, then δ = 1; otherwise δ = 0); i is the recommendation included in the truncated list.

NDCG is widespread to evaluate the quality of ranking in recommender systems, especially when the presence of relevant recommendations and their order in the list are both important.

In addition, metrics indicating memory and time consumption are taken for experiments. The consumed RAM is calculated using the *memory_profiler* package of the Python language. The training time of an algorithm is that required to tune hyperparameters or train embedding layers based on the *Recall* metric. Hyperparameters can be tuned based on a single metric. The *Recall* metric is chosen because of its importance for problems where missing data may have serious consequences.

The calculations presented below were performed on a PC with a 2.40GHz Intel(R) Xeon(R) Silver 4214R CPU and 132 Gb RAM. When reproducing the results, the time measured during an experiment may vary depending on the computing capabilities.

2. THE MATERIALS AND TOOLS OF THIS STUDY

2.1. Datasets

The following conditions are taken into consideration when forming the datasets. To make the source material diverse and the study adequate, we use three datasets representing different amounts of information about user preferences. The first two datasets, MovieLens 100k and MovieLens 1m, include 100 thousand and 1 million records, respectively. Both datasets contain valuable information about the user ratings of movies and demographic data. MovieLens is used for research in recommender systems and machine learning. Amazon Gift Card, part of Amazon Review Data, is chosen as the third dataset. Belonging to a different field than MovieLens, it has high data sparsity. Amazon Review Data (2018) is a set of user responses from the Amazon online store, including related information (product, user, ratings, and response text). This set contains about 35 million user responses over 18 years and is most widespread in machine learning. Amazon Review Data (2018) can be applied for various purposes, such as response tone analysis, natural language processing, and model training. The Amazon Gift Card dataset, which includes information about gift card responses, is taken for research.

Table 1 summarizes the characteristics of these datasets.

Table 1

The characteristics of datasets

Parameter	Datasets		
	MovieLens 100k	MovieLens 1m	Amazon Gift Card
Number of users	944	6041	128 878
Average number of user actions	106.04	165.59	1.1421
Number of elements	1683	3707	1549
Average number of actions with an element	59.45	26988	95.08
Number of intersections	100 000	1 000 209	147 194
Data sparsity, %	93.70	95.50	99.92

Thus, the selected sets satisfy the conditions described above. At the same time, MovieLens 1m is a dataset with a larger number of records compared to MovieLens 100k, which is important to analyze the impact of the number of dataset records on the algorithm evaluation.

When comparing Amazon Gift Card to MovieLens, one observes a significant difference in the average number of user actions: a value slightly exceeding 1 in the former case and over 100 in the latter. This difference creates significant challenges for recommendation algorithms. When dealing with Amazon Gift Card and its relatively small number of user actions, recommendation algorithms face difficulties in predicting and generalizing user preferences. The truncated amount of data may lead to underrepresentative samples, making forecasts and recommendations less accurate.

2.2. Algorithms

The following recommendation algorithms are evaluated within this study:

• Bayesian Personalized Ranking from Implicit Feedback (BPR) [37];

• Large-Scale Information Network Embedding (LINE) [38];

• Neural Collaborative Filtering (NeuCF) [39];

• Deep Matrix Factorization (DMF) [40];

• Spectral Collaborative Filtering (SpectralCF) [41];

• Simplifying and Powering Graph Convolution Network for Recommendation (LightGCN) [42];

• Variational Autoencoders for Collaborative Filtering (MultiVAE) [43];

• Collaborative Denoising Auto-Encoders (CDAE) [44];

• Ranking-Critical Training for Collaborative Filtering (RaCT) [45];

• Sparse Linear Method (SLIM) [46];

• Item-based collaborative filtering (ItemKNN) [47];

• Diffusion Recommender Model (DiffRec) [48].

This list covers different recommendation algorithms for maximum diversity. Note that currently, there are many more approaches than are considered in this paper. The ones above are selected to adequately represent various algorithms and their features in the context of recommendation generation. At the same time, other algorithms also deserve attention and may be the subject of further research in this area.

RecBole, an open-source library developed in the Python programming language and the PyTorch machine learning framework, is chosen to implement the algorithms. This platform offers a wide range of algorithms and approaches to building recommendations, as well as tools for developing, testing, and evaluating recommendation algorithms [49].

3. THE RESULTS OF THIS STUDY

3.1. Calculation of Algorithm Performance

Calculations were performed for all the three datasets. For clarity, we describe intermediate stages only for MovieLens 100k. When collecting metrics, the parameter K (the truncated number of generated recommendations) was set equal to 10.

All collected metrics, except for memory, preparation time, prediction time, and *Average Popularity*, are coefficients taking values between 0 and 1. Memory consumed is represented in megabytes; time metrics, in seconds; *Average Popularity*, in the number of user interactions with an element. The collected data are shown in Tables 2–4.

3.2. Model

We form the composite indicator model as follows: the 13 parameters are convolved into four subindicators of the second layer; *Preparation time*, *Prediction time* (the time to generate recommendations), and *Memory* (the memory size consumed) are aggregated into the Resources subindicator; *Recall* and *Precision* are reduced to Accuracy; the *GAUC*, *MMR*, *NDCG*, *HitRate*, and *MAP* metrics are convolved into the Ranking subindicator; *Average Popularity*, *Gini Index*, and *Shannon Entropy* are generalized into the Diversity subindicator. Figure 1 presents the structure of the composite indicator.

This model is based on the principle of logical union of parameters.

3.3. Calculations

To create the composite indicator, three main tasks have to be solved:

• normalize the partial criteria, as they have different dimensions and units of measurement;

• calculate the weights on the network layers;

• determine the principle of convolving the partial criteria into the composite indicator and its structural elements.

For the convenience of building the composite indicator, the values of the partial criteria should meet the following requirements:

– All partial criteria should be dimensionless.

– To compare different elements with each other, the values of the partial criteria should vary within the same range, e.g., from 0 to 1.

– All partial criteria should be unidirectional.

We apply the minimax value normalization of the source to satisfy these requirements:

$$e_{ij} = \frac{x_{ij} - \min\limits_{k=1,n}(x_{kj})}{\max\limits_{k=1,n}(x_{kj}) - \min\limits_{k=1,n}(x_{kj})},$$

where e_{ij} is the normalized value of the jth metric for the ith algorithm; x_{ij} is the factual value of the jth metric for the ith algorithm; n is the number of algorithms.

For some indicators, such as *Preparation time*, *Prediction time*, *Memory*, and *Average Popularity*, the normalization procedure is applied with value inversion in the same numerical range [0, 1], since increasing their values worsens the resulting evaluation:

$$e_{ij} = 1 - \frac{x_{ij} - \min\limits_{k=1,n}(x_{kj})}{\max\limits_{k=1,n}(x_{kj}) - \min\limits_{k=1,n}(x_{kj})}.$$

Table 5 presents the normalized values of the indicators.

To form the composite indicator, we involve an entropy method for finding the weights of partial criteria. This method is based on analyzing the estimates of the standard deviations of partial criteria over the entire set of elements under consideration [50].

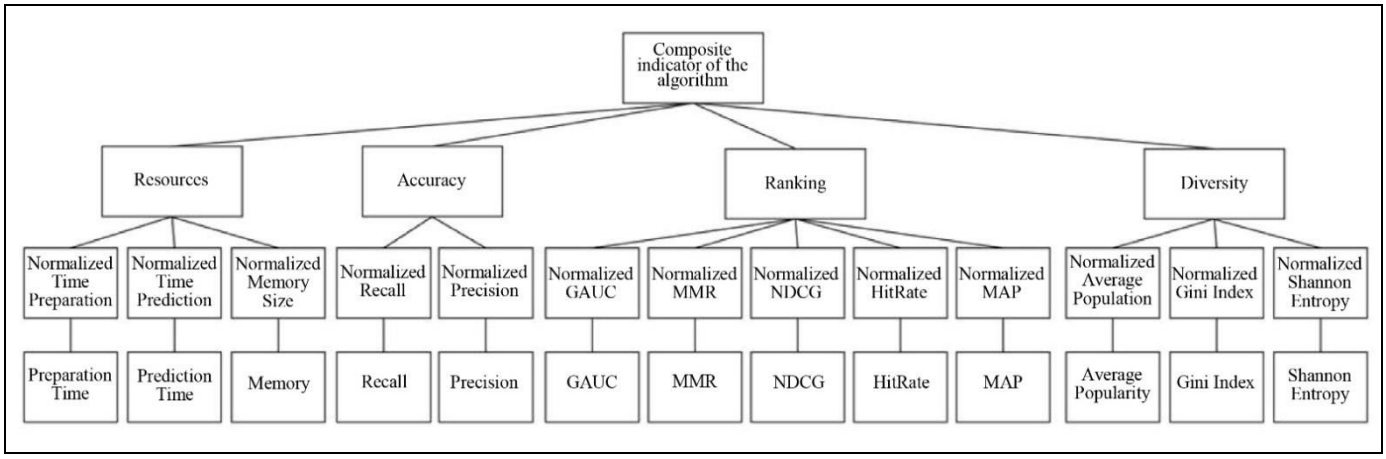

Fig. 1. The composite indicator.

Table 2

Performance of the algorithms for the MovieLens 100k dataset

Algorithm	Memory, Mb	Preparation time (T prep.), s	Prediction time (T pred.), s	Metrics									
				Recall	*Precision*	*GAUC*	*MMR*	*NDCG*	*HitRate*	*MAP*	*Average Popularity*	*Gini Index*	*Shannon Entropy*
BPR	290	20.275	0.192	0.239	0.191	0.918	0.482	0.286	0.772	0.174	241.928	0.925	0.0116
LINE	431.1	19.703	0.189	0.217	0.178	0.914	0.438	0.256	0.751	0.147	173.535	0.864	0.0104
NeuCF	454	40.751	0.923	0.238	0.189	0.919	0.459	0.277	0.766	0.165	226.602	0.911	0.0104
DMF	634.6	59.024	0.210	0.236	0.188	0.894	0.441	0.272	0.782	0.161	227.201	0.911	0.0116
SpectralCF	501.1	32.002	0.289	0.099	0.099	0.848	0.265	0.133	0.498	0.068	338.649	0.988	0.0497
LightGCN	448.3	66.221	1.191	0.226	0.177	0.909	0.453	0.268	0.750	0.161	257.067	0.945	0.0166
MultiVAE	629.6	51.722	0.532	0.237	0.182	0.911	0.463	0.276	0.766	0.165	239.964	0.923	0.0110
CDAE	516	7.853	0.220	0.098	0.096	0.698	0.279	0.135	0.504	0.071	320.434	0.986	0.0564
RaCT	404	112.653	0.662	0.263	0.201	0.918	0.486	0.298	0.808	0.178	223.918	0.892	0.0090
SLIM	416.7	0.455	0.310	0.275	0.217	0.911	0.523	0.324	0.812	0.202	246.862	0.931	0.0147
ItemKNN	521.4	0.741	0.323	0.247	0.193	0.907	0.462	0.283	0.785	0.170	217.709	0.914	0.0125
DiffRec	389.1	112.452	0.936	0.278	0.212	0.897	0.536	0.325	0.828	0.200	232.248	0.907	0.0111

Table 3

Performance of the algorithms for the MovieLens 1m dataset

Algorithm	Memory, Mb	Preparation time (T prep.), s	Prediction time (T pred.), s	Metrics									
				Recall	*Precision*	*GAUC*	*MMR*	*NDCG*	*HitRate*	*MAP*	*Average Popularity*	*Gini Index*	*Shannon Entropy*
BPR	790.5	210.782	2.695	0.163	0.200	0.927	0.445	0.256	0.742	0.152	1147.6	0.885	0.0035
LINE	931.2	190.290	2.680	0.147	0.175	0.920	0.405	0.225	0.707	0.129	721.73	0.791	0.0035
NeuCF	439.5	369.691	13.162	0.145	0.184	0.924	0.403	0.229	0.710	0.132	1136.8	0.896	0.0040
DMF	926.4	1666.525	2.468	0.152	0.188	0.888	0.424	0.239	0.724	0.138	1265.9	0.928	0.0054
SpectralCF	614.7	5327.285	2.691	0.142	0.187	0.922	0.411	0.233	0.701	0.137	1142	0.891	0.0037
LightGCN	601.2	5217.235	2.815	0.155	0.166	0.924	0.390	0.214	0.743	0.112	1136.1	0.889	0.0037
MultiVAE	1019.1	409.670	4.715	0.179	0.200	0.924	0.447	0.260	0.769	0.151	1137.6	0.876	0.0034
CDAE	1215.4	517.649	3.052	0.147	0.192	0.912	0.435	0.243	0.710	0.143	1511.7	0.954	0.0055
RaCT	1122.5	556.044	4.915	0.175	0.196	0.924	0.433	0.252	0.763	0.145	1128.3	0.875	0.0033
SLIM	916.7	6.819	5.379	0.193	0.225	0.909	0.503	0.296	0.791	0.182	1421.5	0.951	0.0068
ItemKNN	1739.1	9.702	4.583	0.163	0.199	0.917	0.447	0.256	0.742	0.152	1201.2	0.918	0.0045
DiffRec	797.2	500.809	7.739	0.200	0.232	0.917	0.511	0.303	0.806	0.186	1277.6	0.922	0.0053

Table 4

Performance of the algorithms for the Amazon Gift Card dataset

Algorithm	Memory, Mb	Preparation time (T prep.), s	Prediction time (T pred.), s	Metrics									
				Recall	*Precision*	*GAUC*	*MMR*	*NDCG*	*HitRate*	*MAP*	*Average Popularity*	*Gini Index*	*Shannon Entropy*
BPR	734.4	26.275	4.171	0.076	0.008	0.550	0.028	0.039	0.076	0.028	796.207	0.898	0.0115
LINE	1099.6	45.129	4.123	0.067	0.007	0.505	0.026	0.035	0.067	0.026	755.302	0.888	0.0091
NeuCF	1156.8	64.381	6.920	0.194	0.019	0.909	0.089	0.114	0.194	0.089	2974.811	0.993	0.1484
DMF	1268.3	319.145	4.464	0.219	0.022	0.862	0.093	0.122	0.219	0.093	2849.264	0.993	0.0993
SpectralCF	1356.6	179.778	3.587	0.213	0.021	0.908	0.092	0.121	0.213	0.092	2955.627	0.993	0.1501
LightGCN	1389.8	373.526	4.091	0.112	0.011	0.635	0.048	0.063	0.112	0.048	836.715	0.900	0.0087
MultiVAE	1515.4	152.994	5.120	0.315	0.032	0.915	0.125	0.170	0.315	0.125	1896.761	0.957	0.0061
CDAE	1505.8	48.741	4.803	0.212	0.021	0.786	0.093	0.121	0.212	0.093	2884.900	0.993	0.1631
RaCT	1514.7	191.619	5.590	0.319	0.032	0.913	0.125	0.170	0.319	0.125	1918.316	0.955	0.0061
SLIM	718.8	1.457	4.193	0.004	0.001	0.500	0.001	0.002	0.004	0.001	106.559	0.994	0.1921
ItemKNN	1146.1	3.545	3.960	0.172	0.017	0.741	0.073	0.096	0.172	0.073	316.197	0.875	0.0093
DiffRec	1408.8	231.541	11.727	0.231	0.023	0.758	0.101	0.132	0.232	0.101	2291.953	0.970	0.0096

Table 5

Normalized performance of the algorithms for the MovieLens 100k dataset

Algorithm	Memory, Mbs	Preparation time (T prep.), s	Prediction time (T pred.), s	Metrics									
				Recall	*Precision*	*GAUC*	*MMR*	*NDCG*	*HitRate*	*MAP*	*Average Popularity*	*Gini Index*	*Shannon Entropy*
BPR	1	0.8234	0.9965	0.7835	0.7926	0.9946	0.8007	0.8004	0.8296	0.7869	0.5858	0.4919	0.0549
LINE	0.5905	0.8285	1	0.6602	0.6786	0.9743	0.6364	0.6404	0.7653	0.5902	1	0	0.0295
NeuCF	0.5240	0.6409	0.2677	0.7790	0.7719	1	0.7172	0.7535	0.8102	0.7213	0.6786	0.3793	0.0295
DMF	0	0.4780	0.9785	0.7685	0.7669	0.8866	0.6501	0.7285	0.8584	0.6930	0.6750	0.3834	0.0549
SpectralCF	0.3874	0.7189	0.8995	0.0056	0.0248	0.6766	0	0	0	0	0	1	0.8587
LightGCN	0.5406	0.4138	0	0.7135	0.6711	0.9540	0.6932	0.7040	0.7620	0.6915	0.4941	0.6556	0.1603
MultiVAE	0.0145	0.5431	0.6576	0.7740	0.7141	0.9621	0.7309	0.7467	0.8102	0.7221	0.5977	0.4741	0.0422
CDAE	0.3442	0.9341	0.9684	0	0	0	0.0517	0.0135	0.0161	0.0179	0.1103	0.9854	1
RaCT	0.6692	0	0.5277	0.9167	0.8694	0.9955	0.8136	0.8629	0.9391	0.8204	0.6949	0.2285	0
SLIM	0.6323	1	0.8791	0.9845	1	0.9612	0.9524	0.9953	0.9518	1	0.5559	0.5397	0.1203
ItemKNN	0.3285	0.9975	0.8656	0.8290	0.8066	0.9435	0.7279	0.7858	0.8681	0.7563	0.7325	0.4003	0.0738
DiffRec	0.7124	0.0018	0.2549	1	0.9587	0.8975	1	1	1	0.9821	0.6444	0.3477	0.0443

For all evaluation criteria, the standard deviation is calculated on each dataset:

$$f_j = \sqrt{\frac{\sum_{i=1}^{N}(e_{ij} - \overline{e_j})^2}{N-1}},$$

where e_{ij} denotes the normalized value; $\overline{e_j}$ is the mean value of the metric for all algorithms; N is the number of algorithms.

Table 6 presents the standard deviations of different parameters.

Next, we determine the weights in each layer within each group of partial criteria and subindices (v_j denote the weights in the first layer). To calculate the weights, the standard deviation of a particular parameter is divided by the sum of the standard deviations in the group:

$$v_j = \frac{f_j}{\sum_{j=1}^{N_p} f_j},$$

where N_p denotes the number of metrics in the pth group.

Then, the value of each pth subindicator is found for all algorithms in the first layer via additive convolution:

$$a_{ip} = \sum_{j=1}^{N_p} e_{ij} v_j .$$

The next step is to calculate the standard deviations in the subindices:

$$s_p = \sqrt{\frac{\sum_{i=1}^{N}(a_{ip} - \overline{a_p})^2}{N-1}},$$

Table 6

Standard deviations for the MovieLens 100k dataset

Metric	Standard deviation
Memory	0.2198
T prep.	0.2730
T pred.	0.2914
Recall	0.2313
Precision	0.2196
GAUC	0.1718
MMR	0.2092
NDCG	0.2256
HitRate	0.2365
MAP	0.2229
Average Popularity	0.1827
Gini Index	0.2034
Shannon Entropy	0.2412

where $\overline{a}_p$ denotes the mean value of the pth subindicator.

Similar to the first layer, we calculate the weights in the second layer:

$$w_p = \frac{s_p}{\sum_{p=1}^{P} s_p},$$

where P is the number of subindices.

Table 7 shows the distribution of the metrics across groups, the sum of standard deviations in the group, and the weights of the first and second layers.

Table 7

Weight calculation for the MovieLens 100k dataset

Metric	Group	Sum of standard deviations in the group	Weight of the first layer	Weight of the second layer
Memory	1	0.784	0.280	0.274
T prep.	1		0.348	
T pred.	1		0.371	
Recall	2	0.451	0.512	0.303
Precision	2		0.487	
GAUC	3	1.066	0.161	0.286
MMR	3		0.196	
NDCG	3		0.211	
HitRate	3		0.221	
MAP	3		0.209	
Average Popularity	4	0.627	0.291	0.135
Gini Index	4		0.324	
Shannon Entropy	4		0.384	

The block diagrams in Figs. 2–4 demonstrate the composite indicator model with the weights of each parameter for the MovieLens 100k, MovieLens 1m, and Amazon Gift datasets, respectively.

Passing to the subindicators implies calculating the normalized values. For ease of understanding, Figs. 2–4 do not include this network layer, despite its factual presence, as in Fig. 1.

Table 8 shows the values calculated in the first layer. They characterize the evaluation for each group of the criteria (Resources, Accuracy, Ranking, and Diversity).

The last step is to perform the additive convolution of the second layer by analogy to the first layer:

$$b_i = \sum_{p=1}^{P} a_{ip} w_p ,$$

where P denotes the number of values in the first layer; the index i corresponds to the algorithm for which the calculations are carried out.

Table 9 presents the final value of the composite indicator for each dataset and the arithmetic mean of the evaluations of different datasets. The values in the cells characterize the integral evaluation of the algo-

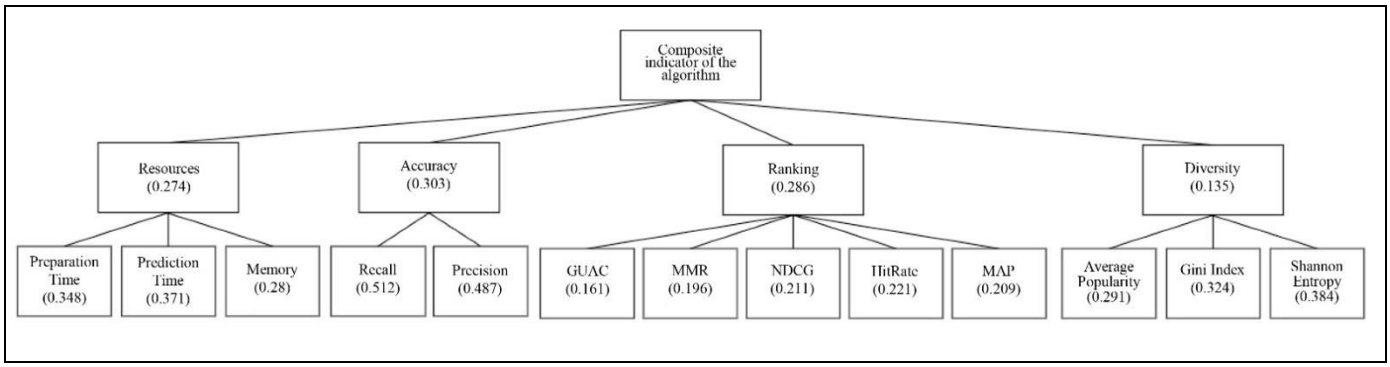

Fig. 2. The composite indicator weights for the MovieLens 100k dataset.

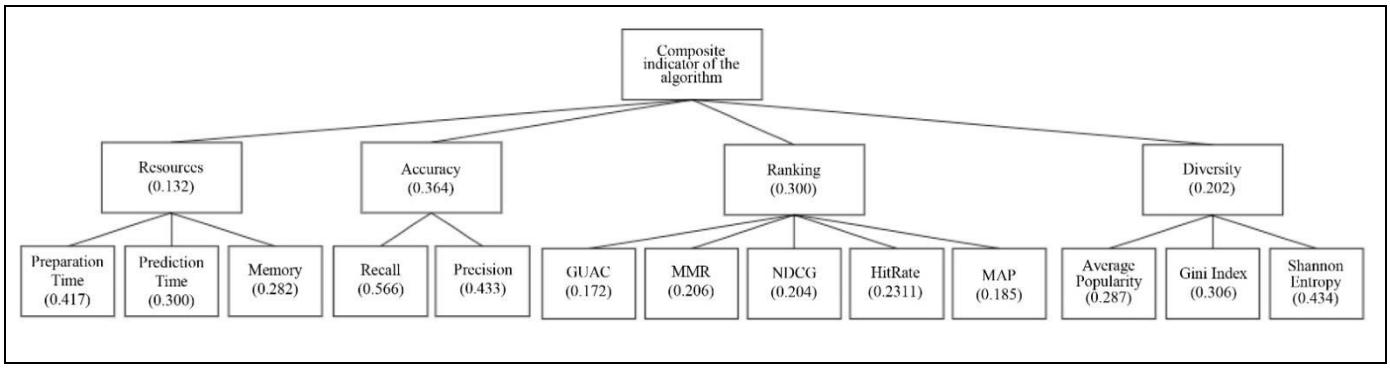

Fig. 3. The composite indicator weights for the MovieLens 1m dataset.

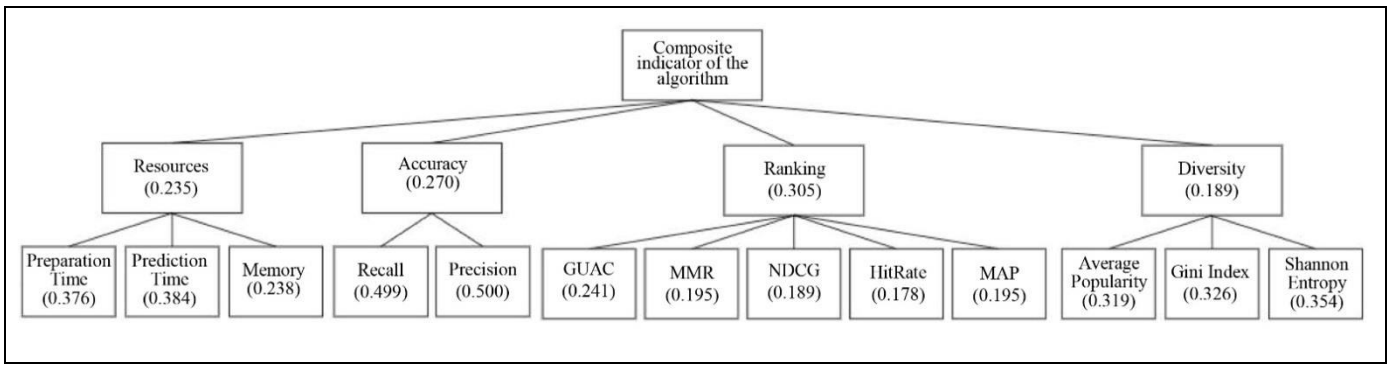

Fig. 4. The composite indicator weights for the Amazon Gift Card dataset.

Table 8

Calculation of subindicator values in the first layer for the MovieLens 100k dataset

Algorithm	Group / Subindicator			
	Resources	Accuracy	Ranking	Diversity
BPR	0.9372	0.7879	0.8354	0.3512
LINE	0.8255	0.6691	0.7106	0.3026
NeuCF	0.4695	0.7755	0.7920	0.3320
DMF	0.5300	0.7677	0.7600	0.3419
SpectralCF	0.6931	0.0149	0.1091	0.6544
LightGCN	0.2956	0.6928	0.7524	0.4181
MultiVAE	0.4374	0.7448	0.7873	0.3440
CDAE	0.7814	0	0.0203	0.7361
RaCT	0.3836	0.8936	0.8826	0.2765
SLIM	0.8520	0.9920	0.9727	0.3831
ItemKNN	0.7610	0.8181	0.8120	0.3715
DiffRec	0.2950	0.9799	0.9798	0.3174

Table 9

Final evaluation

Algorithm	MovieLens 100k	MovieLens 1m	Amazon Gift Card	Arithmetic mean of the evaluations
SLIM	0.8656	0.8390	0.4202	0.7083
DiffRec	0.7022	0.8649	0.5328	0.7000
MultiVAE	0.6184	0.5620	0.7356	0.6387
RaCT	0.6670	0.5058	0.7253	0.6327
ItemKNN	0.7402	0.4963	0.5591	0.5985
BPR	0.7834	0.5054	0.4051	0.5646
DMF	0.6426	0.3799	0.6043	0.5423
NeuCF	0.6362	0.3123	0.6525	0.5337
CDAE	0.3199	0.4090	0.6428	0.4572
LINE	0.6743	0.2874	0.3340	0.4319
SpectralCF	0.3145	0.2811	0.6506	0.4154
LightGCN	0.5637	0.2664	0.3265	0.3855

rithm by the set of criteria. The algorithms in this table are in descending order of the arithmetic mean of the evaluations.

According to Table 9, the same algorithms have different evaluations on the MovieLens 100k and 1m datasets, depending on the dataset size. For example, LINE demonstrates a high evaluation for the first dataset, but it drops by more than a factor of two for the second dataset. For SLIM, the evaluation remains almost the same when increasing the dataset size. Some algorithms (DiffRec, CDAE) show an increase in the evaluation. Thus, there is no obvious dependence of the algorithm's evaluation on the dataset size. This conclusion is also confirmed by the Pearson correlation coefficient between the algorithms on the MovieLens 100k and 1m datasets: the value is only 0.56, corresponding to a rather weak dependence.

The algorithms' evaluations on the MovieLens and Amazon Gift Card datasets do not correlate with each other. For example, BPR shows a high evaluation on the MovieLens dataset and a low evaluation on the Amazon Gift Card dataset.

A deeper analysis of the table with the algorithms' evaluations confirms the initial hypothesis: the same algorithm behaves differently and gives different performance depending on the dataset.

There is no clearly defined reason for this phenomenon, as each group of algorithms involves different principles. For example, clustering-based algorithms may inaccurately predict the element class due to

fuzzy cluster bounds if there are insufficient data. At the same time, under enough data and complete cluster bounds, the clustering results are accurate. Similar problems may arise in algorithms based on other principles.

Also, it is necessary to tune the hyperparameters of algorithms according to the data context considering more than one characteristic metric. A single metric can be applied to solve this problem.

4. DISCUSSION

The results of this study indicate the practical applicability of the proposed approach for selecting an optimal recommendation algorithm, especially in the context of particular datasets. Note that the method can be further improved by introducing user ranking to consider individual user preferences and prioritize the criteria most significant for them. For example, a user appreciating the accuracy of recommendations higher than the algorithm speed can assign appropriate weights, making the approach more flexible and personalized.

In addition, this approach can be applied to optimize the algorithms' hyperparameters. As a result, it becomes possible to systematically estimate the variations of hyperparameters and their impact on particular metrics and the total algorithm evaluation. The systematic estimation of the hyperparameters using this approach allows implementing an iterative model improvement process. Developers will be able to regularly analyze and optimize the parameters in accordance with changing requirements and data.

Hyperparameter optimization improves algorithm performance and accuracy. It also shows which parameters are most significant for achieving the best results: different tasks may require different optimal parameter settings.

An important line is the possible application of this method to form ensembles of algorithms for recommender systems. Evaluation maximization allows effectively combining different algorithms into an ensemble considering their contribution to the recommender system. Such an approach allows developing a more robust and effective recommender system that will cover the advantages and disadvantages of each algorithm to mitigate the shortcomings of particular algorithms and generate more accurate and relevant recommendations for users.

The approaches presented above have considered only the offline evaluations of recommendation algorithms, perhaps forming an incomplete picture of their effectiveness. In the future, a composite indicator may be developed to provide the online evaluation of such algorithms. It is also possible to combine online and offline metrics into a generalized measure. They can be based on such metrics as *Gross merchandise volume* (*GMV*), *Click-through rate* (*CTR*), and *Conversion rate* (*CVR*). These metrics will integrally evaluate the impact of recommendation algorithms on business indicators considering their offline and online aspects.

Other research areas may include analyzing the impact of dynamic data changes on the performance of recommendation algorithms. It is possible to study the performance of a method under changing user preferences, seasonal fluctuations, or temporal trends. Investigating the impact of changing trends in consumer behavior may be a key step toward developing more adaptive and stable recommender systems.

CONCLUSIONS

This paper has considered the problem of generating a composite indicator for evaluating the performance of recommender systems. For this purpose, different algorithms and datasets have been selected, including MovieLens and Amazon Gift Card. For each algorithm and dataset, different metrics have been calculated to represent different aspects of recommendation quality and reflect their accuracy, completeness, diversity, and other characteristics. The composite indicator has been formed by combining these metrics using the entropy-based weighting method. This approach considers the relative significance of each metric and provides balance in the comprehensive evaluation of the effectiveness of recommender systems.

The application of such an approach appears to be novel in recommender system evaluation. The entropy-based composite indicator is a unique tool for comparing different algorithms and datasets in terms of their overall performance. This approach covers different aspects of recommendation quality and can be applied in different recommender system scenarios.

According to the results presented, the composite indicator can be a useful tool for evaluating the effectiveness of recommender systems by combining various metrics into a single overall evaluation. Hence, it gains superiority over partial criteria in terms of informativeness and usability. This indicator provides a deeper and more comprehensive understanding of the effectiveness of recommender systems, representing a valuable tool for making well-grounded decisions in this area.

However, forming the composite indicator requires a large amount of calculations and data analysis. In addition, the choice of algorithms and datasets may significantly affect the evaluation results. Therefore, additional research and experiments with different al-

gorithms and datasets should be carried out to obtain more accurate results. These studies will provide a better understanding of the impact of different parameters on the final results and optimize the composite indicator formation process. With this approach, one will assess how well the composite indicator adapts to different recommender system scenarios and identify its limitations.

Additional research may also optimize the weights of combining different metrics into the composite indicator. This is crucial to consider the relative significance of each metric and balance the different aspects of recommendation quality.

Thus, further research on forming composite indicators for evaluating the effectiveness of recommender systems is necessary to improve their accuracy, generalizability, and applicability.

Acknowledgments. *This work was carried out within the state order of the Ministry of Science and Higher Education of the Russian Federation, project no. FEWM-2023-0013.*

REFERENCES

1. Fayyaz, Z., Ebrahimian, M., Nawara, D., et al., Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities, *Appl. Sci.*, 2020, vol. 10, no. 21, art. no. 7748. DOI: https://doi.org/10.3390/app10217748.
2. Amatriain, X. and Basilico, J., Recommender Systems in Industry: A Netflix Case Study, in *Recommender Systems Handbook*, Ricci, F., Rokach, and Shapira, B., Eds., Boston, MA: Springer, 2015, pp. 385–419. DOI: https://doi.org/10.1007/978-1-4899-7637-6_11.
3. Wibisono, C., Purwanti, E., and Effendy, F., A Systematic Literature Review of Movie Recommender Systems for Movie Streaming Service, *AIP Conf. Proc.*, 2023, vol. 2554, no. 1, art. no. 040005. DOI: https://doi.org/10.1063/5.0104316.
4. Kulshin, R., Sidorov, A., and Senchenko, P. Using Neural Networks with Reinforcement in the Tasks of Forming User Recommendations, *Journal of Physics: Conference Series*, 2022, vol. 2291, no. 1, art. no. 012005. DOI: 10.1088/1742-6596/2291/1/012005.
5. Sipser, M., *Introduction to the Theory of Computation. Course Technology*. Boston, MA: PWS Publishing, 2006.
6. Aixin, S., On Challenges of Evaluating Recommender Systems in an Offline Setting, *Proceedings of the* 17th ACM Conference on Recommender Systems, Singapore, 2023, pp. 1284–1285.
7. Jimenez-Fernandez, E. and Ruiz-Martos, M., Review of Some Statistical Methods for Constructing Composite Indicators, *Studies of Applied Economics*, 2020, vol. 38, no. 1, pp. 1–15. DOI: https://doi.org/10.25115/eea.v38i1.3002.
8. Nardo, M., Saisana, M., Saltelli, A., and Tarantola, S., Tools for Composite Indicators Building, *Report no. JRC31473*, Rome, Italy: European Commission, ISPRA, 2005.
9. Becker, W., Saisana, M., Paruolo, P., and Vandecasteele, I., Weights and Importance in Composite Indicators: Closing the Gap, *Ecological Indicators*, 2017, vol. 80, pp. 12–22. DOI: https://doi.org/10.1016/j.ecolind.2017.03.056.
10. Sidorov, A.A., Methodological Approach to the Integral Assessment of the State and Dynamics of Multidimensional Objects of Socio-economic Nature, *Control Sciences*, 2016, no. 3, pp. 32–40. (In Russian.)
11. Abberger, K., Graff, M., Müller, O., and Sturm, J., Composite Global Indicators from Survey Data: The Global Economic Barometers, *Rev. World Econ.*, 2022, vol. 158, pp. 917–945. DOI: https://doi.org/10.1007/s10290-021-00449-8.
12. Endrodi-Kovacs, V. and Tankovsky, O., A Composite Indicator for Economic Integration Maturity: The Case of Western Balkan Countries, *Eastern Journal of European Studies*, 2022, vol. 13, no 1, pp. 148–166. DOI: 10.47743/ejes-2022-0107.
13. Khadzhynova, O., Simanaviciene, Z., Mints, O., et al., Assessment of the EU Countries' Economic Security Based on the Composite Indicators, *WSEAS Transactions on Business and Economics*, 2022, vol. 19, pp. 690–700. DOI: 10.37394/23207.2022.19.61.
14. McDonnell, T., Cosgrove, G., Hogan, E., et al., Methods to Derive Composite Indicators Used for Quality and Safety Measurement and Monitoring in Healthcare: A Scoping Review Protocol, *BMJ Open.*, 2023, vol. 13, no. 7. DOI: 10.1136/bmjopen-2022-071382.
15. Kara, P., Valentin, J., Mainz, J., and Johnsen, S., Composite Measures of Quality of Health Care: Evidence Mapping of Methodology and Reporting, *PLoS One*, 2022, vol. 17, no. 5. DOI: 10.1371/journal.pone.0268320.
16. Asadi-Lari, M., Majdzadeh, R., Mansournia, M.A, et al., Construction and Validation of CAPSES Scale as a Composite Indicator of SES for Health Research: An Application to Modeling Social Determinants of Cardiovascular Diseases, *BMC Public Health.*, 2023, vol. 23, art. no. 293. DOI: https://doi.org/10.1186/s12889-023-15206-9.
17. Abenayake, C., Mikami, Y., Matsuda, Y., and Jayasinghe, A., Ecosystem Services-Based Composite Indicator for Assessing Community Resilience to Floods, *Environmental Development*, 2018, vol. 27, pp. 34–46. DOI: https://doi.org/10.1016/j.envdev.2018.08.002.
18. Gómez-Limón, J., Arriaza, M., and Guerrero-Baena, M., Building a Composite Indicator to Measure Environmental Sustainability Using Alternative Weighting Methods, *Sustainability*, 2020, vol. 12, no. 11, art. no. 4398. DOI: https://doi.org/10.3390/su12114398.
19. Alam, M., Dupras, J., and Messier, C., A Framework towards a Composite Indicator for Urban Ecosystem Services, *Ecological Indicators*, 2016, vol. 60, pp. 38–44. DOI: https://doi.org/10.1016/j.ecolind.2015.05.035.
20. Melo-Aguilar, C., Agulles, M., and Jordà, G., Introducing Uncertainties in Composite Indicators. The Case of the Impact Chain Risk Assessment Framework, *Front. Clim.*, 2022, vol. 4. DOI: https://doi.org/10.3389/fclim.2022.1019888.
21. Dolge, K. and Blumberga, D., Composite Risk Index for Designing Smart Climate and Energy Policies, *Environmental and Sustainability Indicators*, 2021, vol. 12, art. no. 100159. DOI: https://doi.org/10.1016/j.indic.2021.100159.
22. Do, H., Ly, T., and Do, T., Combining Semi-Quantitative Risk Assessment, Composite Indicator and Fuzzy Logic for Evaluation of Hazardous Chemical Accidents, *Sci. Rep.*, 2020, vol. 10, art. no. 18544. DOI: https://doi.org/10.1038/s41598-020-75583-8.
23. Avanesian, G., Mizunoya, S., and Delamonica, E., UNICEF Remote Learning Readiness Index: A Composite Indicator to Assess Resilience of Education Sector against Crises and

Emergencies, *Statistical Journal of the IAOS*, 2022, vol. 38, pp. 1–14. DOI: 10.3233/SJI-220051.
24. Segovia-Gonzalez, M. and Contreras, I., A Composite Indicator to Compare the Performance of Male and Female Students in Educational Systems, *Soc. Indic. Res.*, 2023, vol. 165, pp. 181–212. DOI: https://doi.org/10.1007/s11205-022-03009-1.
25. Hubelova, D., Odvarkova, V., and Chalupa, P., Selected Factors of Education Level in East African Countries: Comparative Method Using Composite Indicator, *Geographical Journal*, 2016, vol. 68, pp. 55–72.
26. Silveira, T., Zhang, M., Lin, X., et al., How Good Your Recommender System Is? A Survey on Evaluations in Recommendation, *Journal of Machine Learning and Cybernetics*, 2016, vol. 10, pp. 813–831. DOI: https://doi.org/10.1007/s13042-017-0762-9.
27. Hongzhi, Y., Cui, B., Li, J., et al., Challenging the Long Tail Recommendation, *Proceedings of the VLDB Endowment*, 2012, vol. 5, pp. 896–907. DOI: https://doi.org/10.14778/2311906.2311916.
28. Hanczar, B., Hua, J., Sima, C., et al., Small-Sample Precision of ROC-Related Estimates, *Bioinformatics*, 2010, vol. 26, pp. 822–830. DOI: https://doi.org/10.1093/bioinformatics/btq037.
29. Calders, T. and Jaroszewicz, S., Efficient AUC Optimization for Classification, in *Lecture Notes in Computer Science*, 2007, vol. 4702, pp. 42–53. DOI: https://doi.org/10.1007/978-3-540-74976-9_8.
30. Wenlong, S., Khenissi, S., Nasraoui, O., and Shafto, P., Debiasing the Human-Recommender System Feedback Loop in Collaborative Filtering, *Proceedings of the 2019 World Wide Web Conference*, San Francisco, 2019, pp. 645–651. DOI: https://doi.org/10.1145/3308560.3317303.
31. Zhang, Q., Cao, L., Zhu, C., et al., CoupledCF: Learning Explicit and Implicit User-item Couplings in Recommendation for Deep Collaborative Filtering, *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, 2018, pp. 3662–3668. DOI: https://doi.org/10.24963/ijcai.2018/509.
32. Bellogin, A., Castells, P., and Cantador, I., Precision-Oriented Evaluation of Recommender Systems: An Algorithmic Comparison, *Proceedings of the 5th ACM Conference on Recommender Systems*, Chicago, 2011, pp. 333–336. DOI: https://doi.org/10.1145/2043932.2043996.
33. Wang, Y., Application of Recall Methods in Recommendation Systems, *Proceedings of the 3rd International Conference on Signal Processing and Machine Learning*, Oxford, 2023, vol. 4, pp. 44–51. DOI 10.54254/2755-2721/4/20230344.
34. Zriaa, R., Sadiki, H., Ertel, M., et al., Qualitative Recommender System Using Entropy-Weighted Pedagogical Criteria for Effective Training in E-Learning Platforms, *Journal of Theoretical and Applied Information Technology*, 2023, vol. 101, no. 9, pp. 3517–3529.
35. Kumar, C. and Kumar, M., User Session Interaction-Based Recommendation System Using Various Machine Learning Techniques, *Multimed. Tools Appl.*, 2023, vol. 82, pp. 21279–21309.
36. Wang, Y., Wang, L, Li, Y., et al., A Theoretical Analysis of NDCG Ranking Measures, *Proceedings of the 26th Annual Conference on Learning Theory*, Princeton, 2013, vol. 30, pp. 25–54.
37. Steffen, R., Freudenthaler, C., Gantner, Z., and Schmidt, L., BPR: Bayesian Personalized Ranking from Implicit Feedback, *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, Montreal, 2009, pp. 452–461.
38. Jian, T., Qu, M., Wang, M., et al., LINE: Large-Scale Information Network Embedding, *Proceedings of the 24th International Conference on World Wide Web*, Florence, 2015, pp. 1067–1077. DOI: https://doi.org/10.1145/2736277.2741093.
39. He, X., Liao, L., Zhang, H., et al., Neural Collaborative Filtering, *Proceedings of the 26th International Conference on World Wide Web*, Perth, 2017, pp. 173–182. DOI: https://doi.org/10.1145/3038912.3052569.
40. Xue, J, Dai, X., Zhang, J., et al., Deep Matrix Factorization Models for Recommender Systems, *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Toronto, 2017, pp. 3203–3209. DOI: https://doi.org/10.24963/ijcai.2017/447.
41. Lei, Z., Lu, C., Jiang, F., et al., Spectral Collaborative Filtering, *Proceedings of the 12th ACM Conference on Recommender Systems*, Vancouver, 2018, pp. 311–319. DOI: https://doi.org/10.1145/3240323.3240343.
42. He, X., Deng, K., Wang, X., et al., LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation, *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Xian, 2020, pp. 639–648. DOI: https://doi.org/10.1145/3397271.3401063.
43. Dawen, L., Krishnan, R., Hoffman, M., and Jebara, T., Variational Autoencoders for Collaborative Filtering, *Proceedings of the 2018 World Wide Web Conference*, Lyon, 2018, pp. 689–698. DOI: https://doi.org/10.1145/3178876.3186150.
44. Yao, W., DuBois, C., Zheng, A., and Ester, M., Collaborative Denoising Auto-Encoders for Top-N Recommender Systems, *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, San Francisco, 2016, pp. 153–162. DOI: https://doi.org/10.1145/2835776.2835837.
45. Lobel, S., Li, C., Gao, J., and Carin, L., RaCT: Toward Amortized Ranking-Critical Training for Collaborative Filtering, *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, 2020.
46. Ning, X. and Karypis, G., SLIM: Sparse Linear Methods for Top-N Recommender Systems, *Proceedings of the IEEE 11th International Conference on Data Mining*, Vancouver, 2011, pp. 497–506. DOI: 10.1109/ICDM.2011.134.
47. Mukund, D. and Karypis, G., Item-Based Top-N Recommendation Algorithms, *ACM Transactions on Information Systems*, 2004, vol. 22, pp. 143–177. DOI: https://doi.org/10.1145/963770.963776.
48. Wang, W., Xu, Y., Feng, F., et al., Diffusion Recommender Model, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Taipei, 2023, pp. 832–841. DOI: https://doi.org/10.1145/3539618.3591663.
49. Zhao, W., Mu, S., Hou, Y., et al., RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms, *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, Queensland, 2021, pp. 4653–4664. DOI: https://doi.org/10.1145/3459637.3482016.
50. Raev, A.G., On One Way to Determine Weighting Coefficients of Particular Criteria in Development of an Integral Additive Criterion, *Avtomat. Telemekh.*, 1984, no. 5, pp. 162–165. (In Russian.)

This paper was recommended for publication by RAS Academician D.A. Novikov, a member of the Editorial Board.

Received March 5, 2024, and revised May 14, 2024. Accepted July 19, 2024.

Author information

Kulshin, Roman Sergeevich. Postgraduate, Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia
✉ roman.s.kulshin@tusur.ru
ORCID iD: https://orcid.org/0000-0002-6891-1869

Sidorov, Anatolii Anatol'evich. Cand. Sci. (Econ.), Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia
✉ anatolii.a.sidorov@tusur.ru
ORCID iD: https://orcid.org/0000-0002-9236-3639

Cite this paper

Kulshin, R.S. and Sidorov, A.A., An Entropy-Based Composite Indicator for Evaluating the Effectiveness of Recommender System Algorithms. *Control Sciences* **4**, 37–51 (2024). http://doi.org/10.25728/cs.2024.4.4

Original Russian Text © Kulshin, R.S., Sidorov, A.A., 2024, published in *Problemy Upravleniya*, 2024, no. 4, pp. 44–60.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com