ISSN 0147-6882, Scientific and Technical Information Processing, 2025, Vol. 52, No. 5, pp. 1–12. © Allerton Press, Inc., 2025.
Russian Text © The Author(s), 2024, published in Iskusstvennyi Intellekt i Prinyatie Reshenii, 2024, No. 1, pp. 103–117.

Reducing the Search Space for Optimal Clustering Parameters Using a Small Amount of Labeled Data

V. I. Yuferev[a], * and N. A. Razin[a], **

[a] *Central Bank of Russia, Moscow, Russia*

**e-mail: YuferevVI@cbr.ru*

***e-mail: RazinNA@cbr.ru*

Abstract—This article presents a method for reducing the search space of clustering parameters. This is achieved by selecting the most appropriate data transformation methods and dissimilarity measures at the stage preceding the actual execution of clustering. To compare the selected methods, it is proposed to use the silhouette coefficient, which considers class labels from a small labeled dataset as cluster labels. The results of experimental validation of the proposed approach for clustering news texts are presented.

Keywords: clustering, parameter search, search space reduction, dissimilarity measures, machine learning

DOI: 10.3103/S0147688225700340

INTRODUCTION

Publications devoted to the search for parameters of the machine learning process mainly focus on searching for parameters of individual stages of the process, predominantly model hyperparameters [22, 26, 28]. By machine learning process parameters, we mean parameters of the stages included in this process. At the same time, the parameters that influence the training of the machine learning model are called model hyperparameters to distinguish them from model variables optimized by the machine learning algorithm itself, called model parameters (not included in machine learning process parameters). Examples of possible stages and their parameters for a specific case of the machine learning process, text clustering, are provided in Table 1.

Approaches to searching the parameters of all stages of the machine learning process in aggregate (such approaches are often denoted by the term AutoML), unlike individual stages, have received less attention [21]. In particular, publications exist on the possibility of applying a meta-learning approach [4], the essence of which is training a special model based on multiple tasks, capable of making predictions about which data transformation methods, machine learning algorithms and their parameters are optimal for a new task fed to the model input. However, this approach is complex to implement and requires preliminary training of the meta-model. The random search method is simple to implement [2], the essence of which is selecting parameter values from certain random distributions in the parameter space of the original dimensionality and subsequent evaluation of the results of the machine learning process for each combination of

Table 1. Example of proposed division of clustering process parameters

Data Transformation		Clustering		
1. Text preprocessing parameters	2. Text vectorization algorithms	3. Clustering algorithm	4. Clustering algorithm hyperparameters	
Stop word set, lemmatize/no, convert case/no, others	TF-IDF on n-grams of words/characters, n-gram length, ... Embedding models, others	K-Means, DBScan, ohers	4.1 Dissimilarity measure	4.2 Number of clusters, initialization method, others

parameter values obtained in this way. There are also studies on the application of Bayesian optimization to simultaneous search for parameters of the machine learning process, e.g., [21]. The essence of the approach consists in modeling the quality function from the parameters of the machine learning process using a Gaussian process. This method is considered one of the most effective for finding optimal parameters. Nevertheless, as with random search, obtaining estimates when building a quality function model requires fixing the values of all parameters in the original high-dimensional space, i.e., conducting experiments in sufficient volume. In general, it can be said that other approaches can be used to optimize the search for parameters of the machine learning process (e.g., evolutionary [23], particle swarm [14], and the Nelder–Mead simplex method [16]), considering the optimized process as a "black box" provided there is an optimized function and taking into account that this function is not necessarily convex and differentiable, and the optimized parameters can simultaneously be discrete, categorical, continuous, and dependent [7, 28]. At the same time, as in the cases of Bayesian optimization and random search, a set of values of all parameters can be evaluated after complete execution of the entire process. Since the execution of the entire machine learning process can be computationally intensive, this entails high computational complexity of parameter search.

One approach to reducing the computational complexity of the parameter search process for machine learning is multi-fidelity optimization [7]. The approach involves executing the machine learning process when searching for parameters only on part of the data, or only using fewer algorithm iterations than required. However, as a result, for each run of the machine learning process, not the exact value of the final quality is obtained, but some approximation of it.

A common approach in practice (but not independently designated in scientific publications due to its obviousness) for reducing the parameter space of the machine learning process is sequential search for the best parameters of individual stages. Sequential search makes it possible to reduce the computational complexity of experiments to find the best parameter values by performing only the initial stages of the machine learning process (for parameters of all stages except the last) and due to the smaller dimensionality of parameter spaces considered at individual stages. However, the relationship between choosing the best parameters for individual stages and the quality of the final result of the entire process is often not obvious, resulting in parameter values that are best for individual stages not necessarily being the best in terms of final quality.

Thus, the question of how to divide the search for parameters of the entire machine learning process into separate stages for parameter subsets using criteria related to the final quality assessment of the entire machine learning process is relevant. We propose a solution to the formulated problem for a specific case of the machine learning process, i.e., the clustering process using labeling for part of the data.

1. STANDARD CLUSTERING

Clustering is a type of machine learning, in which an unlabeled set of objects is divided into groups [11]. In dependence on the goals with which clustering is performed, various approaches to assessing its quality are used. If the priority is to obtain an assessment of the structural properties of the resulting division regardless of labeling, then internal quality assessment methods are applied. Also, in practice, situations arise when the analyst has an idea about which of the clustering results is more preferable. This representation is fixed in the form of labeling of part of the clustered data. Thus, we can speak of the following task solved by the analyst using clustering: to obtain a data partition that is closest to the labeling. In order to evaluate which of the partitions of the original data is closer to the available labeling, external methods for assessing clustering quality are used [10, 25].

In the standard clustering process, the following main stages can be distinguished [8, 15, 27]: data acquisition, data transformation, clustering, and clustering quality assessment. The stages of data transformation and clustering allow for variability determined by their parameters. Obviously, the obtained clustering quality assessment characterizes the choice of values of all parameters of the specified stages: the method and parameters of data transformation, the selected clustering algorithm and its hyperparameters. In connection with this, there arises the need to search for the best values in the space formed by all the specified parameters of the clustering process. By the best values, we mean here those that give the maximum clustering quality assessment using labeled data with the help of given external quality metrics. In general, in order to find the best (in terms of external quality metrics) parameters of the clustering process, all stages need to be executed repeatedly [15], scheme in Fig. 1.

With a large number of parameters, a complete enumeration of all values is computationally expensive, as the total number of possible variants is the product of the number of all possible values for each parameter. This is especially relevant in cases where the clustered dataset is large, and the clustering stage, accordingly, requires significant computational resources. The approach we propose involves

—Dividing the parameter space of the clustering process into two: parameters affecting the method of data representation together with the dissimilarity measure (Table 1, columns 1, 2, and 4.1) and parame-

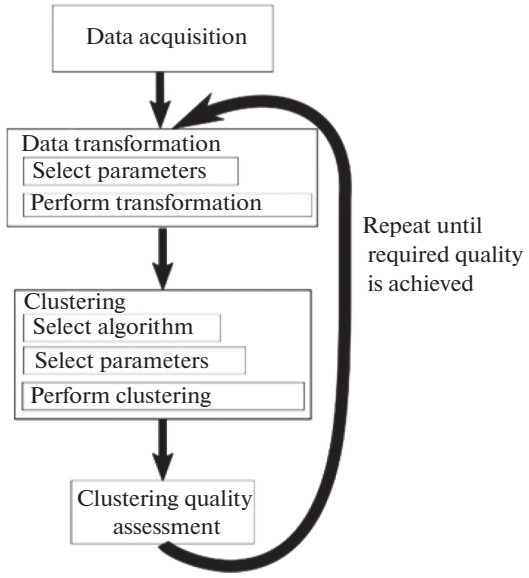

Fig. 1. Iterative scheme for searching for the best parameters of the clustering process.

ters of the clustering stage directly: algorithm and its hyperparameters (Table 1, columns 3 and 4.2).

—Sequential search for the best parameters in each space based on a criterion related to the final quality of the solution to the task facing the analyst (obtaining a partition of the original data that is closest to the available labeling).

2. PROPOSED APPROACH

In existing publications, clustering is considered as a process of studying a set of points and grouping points into clusters according to some distance measure. The clustering algorithm and its hyperparameters determine only the way in which data points are divided into clusters, and the basis for its application is the dataset itself and the distance measure defined on this dataset [17]. In the context of clustering, often along with the term *distance measure* the term *dissimilarity measure* is used as a synonym, we will also understand by these two terms one concept, i.e., a numerical assessment of similarity or dissimilarity between objects in the feature space, used in the context of clustering. For convenience, we will denote the dataset and the distance measure defined on it by the term *dissimilarity space* (DS). Note that the use of the concept of distance measure (and dissimilarity measure), as opposed to a distance metric, implies that it does not necessarily satisfy the axioms for distance metrics.

We propose to search for the best parameters of the clustering process through a separate search for optimal parameters of the data transformation stage together with the most suitable dissimilarity measure used at the clustering stage using a small amount of labeled data. Let us call this stage the assessment of DS. As a result, the clustering process takes the form shown in the diagram in Fig. 2.

Existing publications also consider issues of searching for optimal data transformation methods [1,

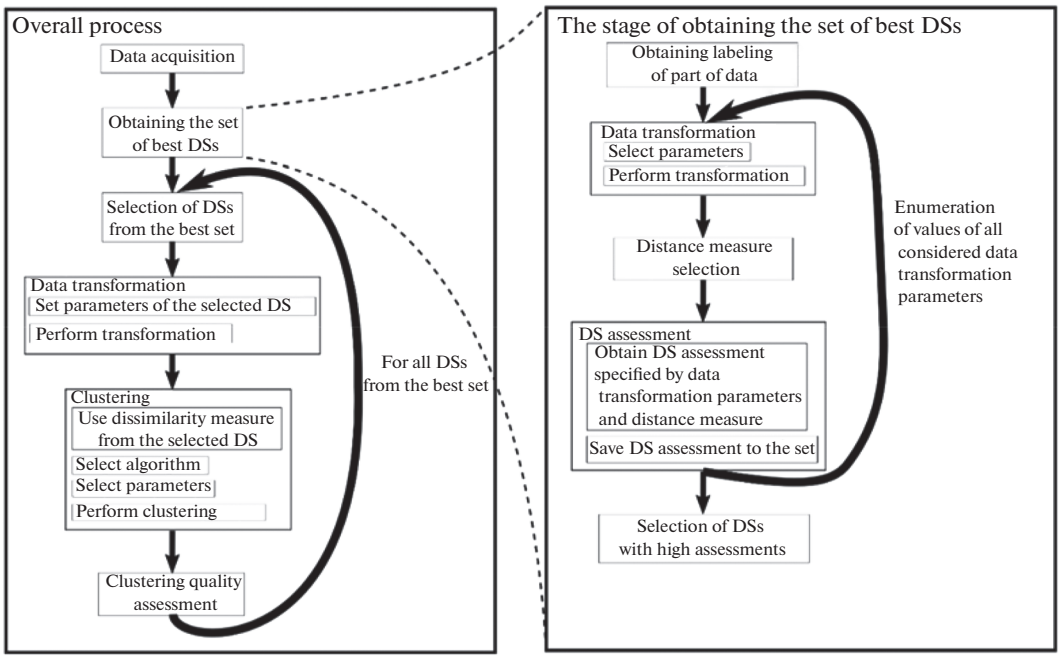

Fig. 2. Proposed scheme for searching for the best parameters of the clustering process.

5, 6] and dissimilarity measures [3, 9] for performing clustering. However, parameter value assessment is either performed after clustering, or does not use labeled data, i.e., does not directly relate to the task facing the analyst. Also note that a scheme similar to the proposed one is given in [1]. In the specified work, the purpose of introducing an additional stage is to assess the "clusterability" of the original data, i.e., to assess whether there is a structure in the original data that could be extracted using clustering without taking into account how this structure relates to the available labeling. Unlike [1], we propose to introduce an additional stage for a different purpose, namely, to assess the correspondence of the data transformation method and the selected distance measure to the available labeling.

The idea of searching for the best feature selection method before performing clustering based on analysis of the entropy of pairwise distances between vector representations of clustered objects has also been described [5]. A description of the Filter-model approach for feature selection for clustering, the essence of which is feature selection before clustering and independently of it based on maximizing assessments: feature dependences, entropy of pairwise distances, and Laplacian assessment, was given [6]. In both of these works, the study of properties of data representation also does not address the question of how the representation of these objects relates to labeling.

External clustering quality assessment methods show to what extent cluster labels coincide with class labels from labeled data [12]. Obviously, this degree of proximity depends on the DS in which clustering is performed, i.e., on how well the DS corresponds to the task solved by the analyst (formulated in the Introduction). Let us formulate informally the concept of DS correspondence to the task as follows:

—DS corresponds to the task if examples of one class are located closer to each other than to examples of other classes.

—DS does not correspond to the task if most examples are located closer to examples of other classes rather than to examples of the same class.

—Intermediate correspondence is expressed in intermediate location of class objects between the two described extreme cases.

To be able to compare DSs, it is necessary to specify a numerical expression for assessing DS correspondence to the task. For internal clustering quality assessment, the silhouette coefficient is used, defined in a manner similar to the formulation presented above with the only differences being that instead of class labels, cluster labels are used, and instead of the term "other classes" the term "nearest other cluster" is applied [19, p. 55].

We will use for assessing DS correspondence to the solved task the method of calculating the silhouette coefficient: $silhouette_i = \frac{b-a}{\max(b,a)}$, where silhouette$_i$ is the silhouette coefficient for example i, a is the average difference between i and all other examples of the cluster to which i belongs, and b is the average difference between i and all examples of the cluster whose examples have the smallest average difference with i among all clusters. At the same time, we will use as object labels the classes to which these objects belong. Since the silhouette coefficient for classes is calculated separately for each object, it is proposed to calculate the value of the DS correspondence assessment based on the entire labeled dataset by simple averaging of the values obtained for individual objects (in accordance with the recommendation in [19]).

The stage of assessment of DS correspondence to the task is proposed to be used for searching for the best DS (having maximum assessments) by enumerating values of data transformation stage parameters as well as dissimilarity measures compatible with these parameter values that are supposed to be used at the clustering stage. After determining the list of best DS, it is proposed to conduct experiments to determine optimal DS of the clustering algorithm and its hyperparameters in these found DSs.

The idea of using the silhouette coefficient for labeled data has been previously proposed. In particular, the silhouette coefficient was considered as a measure of class separability in the context of identifying biases in the dataset [20]. Unlike our proposed approach, in [20] the coefficient value was calculated on the complete set of labeled data. In addition, due to the fact that the silhouette coefficient is applied to the complete dataset, as well as the quadratic computational complexity of the algorithm, in [20] a modified silhouette coefficient calculation algorithm was applied, which according to the authors' assumption gives less accurate results than the original. The algorithm modification consists in excluding intra-cluster distances from consideration and using only inter-cluster ones. In the target application conditions considered in this work, due to the initially small volume of labeled data, such optimization is not required.

3. EXPERIMENTAL

Experimental verification of the applicability of the proposed approach was performed using news clustering as an example.

The experiments were intended to:

- Prepare a set of DSs.
- Perform clustering in each of the selected DSs and evaluate clustering results using external clustering quality assessment methods.
- For each DS, calculate its assessment based on the silhouette coefficient value (hereinafter, SC) using a small amount of labeled data.

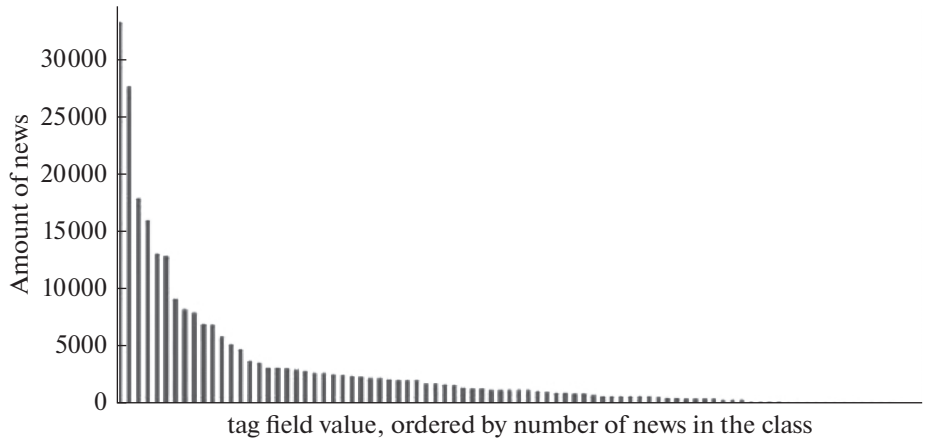

Fig. 3. Distribution of news by tag field values.

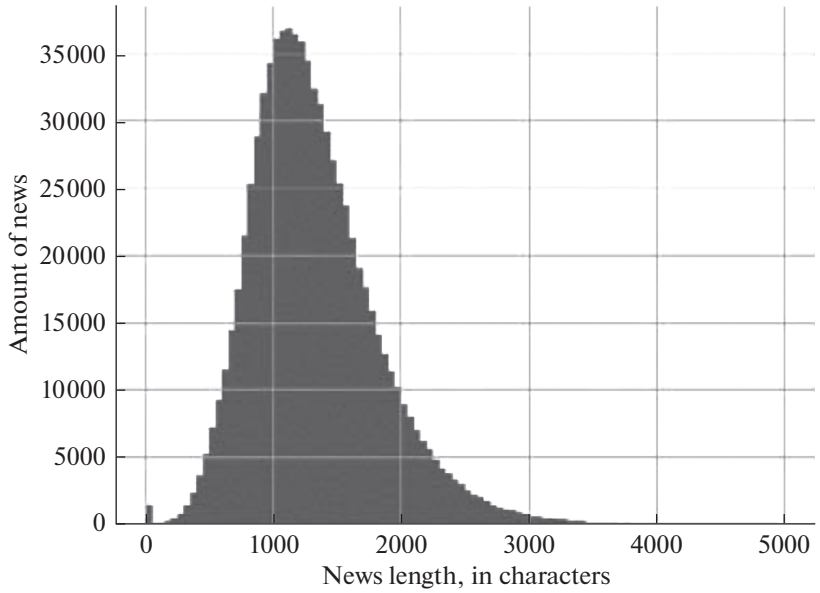

Fig. 4. Distribution of news by their length in characters.

• Compare DSs assessments obtained using SC and using external clustering quality assessments.

The study was performed using the text dataset lenta-ru-news (https://github.com/yutkin/Lenta.Ru-News-Dataset) from the lenta.ru news portal. The dataset contains 800 975 news items. The dataset contains the following fields: url, which is the link by which the news was obtained, title, which is the news headline, text, which is the news text, topic, which is enlarged news category, tags, which is the detailed news category, and date, which is the news date.

For experiments, the news category from the tag field was used, due to the fact that it contains more classes (94 classes) than the enlarged category topic (23 classes). For each news item in the tags field, one value is assigned characterizing the news topic. In total, the dataset in the tag field contains 94 unique values. The value All in the tag field is assigned to 453 762 news items. The average number of texts in the class of the tag field, excluding All values, is 3694. The distribution of topics by values of the tag field (except for the All value) is presented in Fig. 3. The average length of news texts is 1333 characters. The distribution of text lengths has the form presented in Fig. 4.

3.1. Dataset Preparation

Data preparation includes the following stages:

1. All texts were transformed by excluding all characters that are not numerical digits or letters of the Russian or Latin alphabet.

2. Examples with the value All in the tag field were excluded from the original dataset, because visual analysis reveals differences in topics in different texts marked with the same tag All, while when considering examples marked with other values in the tag field, correspondence between the text topic and the value contained in the tag field is apparent. This manipulation is justified, because in real cases when there is a small labeled dataset, or when external clustering quality assessment is performed, it is reasonable to expect that examples correspond to target class labels rather than general ones.

3. Texts that are too long (more than 4000 characters) and too short (less than 50 characters) were excluded from consideration.

4. Text lemmatization was performed using Yandex MyStem 3.1.

5. Complete duplicates of lemmatized texts were removed.

As a result, the original dataset was reduced to 258780 news items. The resulting set of texts will be denoted as ULS for subsequent references.

3.2. Dissimilarity Space Selection

At the first stage, several arbitrary DSs were selected defined by the method of transforming news texts into vector form as well as by the dissimilarity measure defined on the resulting vectors.

Text transformation methods into vector form are defined as sequentially applied text processing stages (brief designations are given in parentheses):

• For the first processing stage, either lemmatization is performed or not performed, in a total of two stage implementation variants (lemm and not_lemm, respectively).

• Then vectorization is performed using either the TF-IDF approach or bag-of-words, a total of two stage implementation variants (tf_idf and countvec, respectively).

• Next, either L2-normalization of vectors obtained at the previous stage is performed or not performed, a total of two stage implementation variants (L2 and no_norm).

Thus, there are $2 \times 2 \times 2 = 8$ possible variants for transforming texts into vector form. For convenience of subsequent mention, we will denote approaches by indicating brief stage designations separated by hyphens, e.g.: "lemm-tf_idf-l2."

The following four were selected as possible dissimilarity measures (brief designations for subsequent references are given in parentheses): Euclidean (euclidean), cityblock (cityblock), radial based function with the parameter gamma = 0.1 (rbf), and cosine distance (cosine).

Thus, by combining possible methods of transforming texts into vector form and possible dissimilarity measures, we have $8 \times 4 = 32$ possible DSs. For subsequent references, DSs are denoted using a combination of the brief designation of the approach for transforming texts into vector form and the brief designation of the dissimilarity measure separated by hyphen, e.g.: "lemm-tf_idf-l2-euclidean."

3.3. Clustering and Its Quality Assessment

The Kmedoids algorithm [13] was selected as the clustering algorithm, because it makes it possible to apply the selected dissimilarity measures. The algorithm divides the dataset into a specified number of clusters, selecting real objects (medoids) from the dataset as cluster centers, unlike the K-means algorithm, where cluster centers are computed. The algorithm was applied with the following fixed parameters: alternating optimization, kmedoids++ initialization, and maximum number of iterations of 300.

Adjusted mutual information (AMI) [24] was selected as the external clustering quality metric. AMI is a normalized measure of mutual information that, unlike mutual information, accounts for random cluster coincidence. AMI is more suitable in cases where there is class imbalance in the data (shown in [18]). From Fig. 3, it can be seen that the available dataset indeed has significant class imbalance.

Clustering quality assessment was performed in each DS for different values of the "number of clusters" parameter of the clustering algorithm in the range from 2 to 950 (49 values total). For clustering quality assessment, text sets were formed, each of 10000 examples by sampling without replacement from the original ULS dataset. A total of 17 such datasets were generated to conduct experiments on all values of the number of clusters parameter on each generated dataset.

Figure 5 shows the obtained dependence of the average AMI value on the number of clusters parameter for clustering in different DSs. Dissimilarity spaces in the legend are ordered by decreasing value of the maximum average value. From the figure, it can be seen that over the entire observed set of "number of clusters" parameter values, the graphs of the leading 15 DSs are clearly distinguishable (marked with a black curly bracket on the right).

Based on the goals of determining DSs most corresponding to the task, based on visual analysis of the graphs of AMI dependence on the "number of clusters" parameter, we will define the DS assessment as the maximum average AMI value for the corresponding DS.

3.4. Assessment of Dissimilarity Space Correspondence to the Task

DS assessment based on the proposed approach was performed for datasets of different sizes (10, 20,

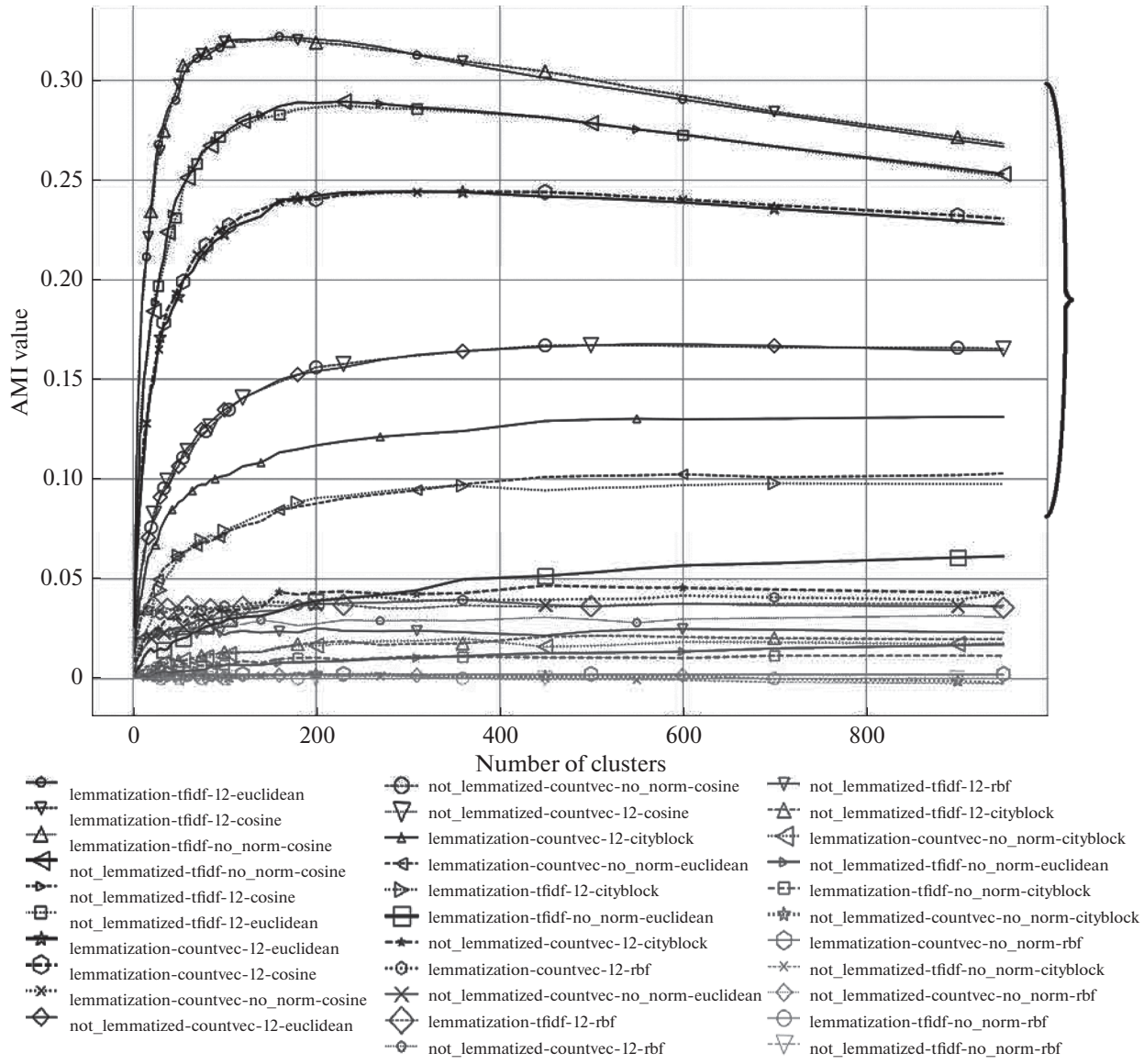

Fig. 5. Dependence of AMI on the "number of clusters" parameter for clustering in different DSs.

30, 40, 50, 80, 120, 160, 200, 250, 300, and 500 examples). Datasets of each specified size were generated 20 times. Let us denote each such dataset as T_i^s, where s is the size of the generated sample, and i is the iteration number at which the dataset of size s was generated.

T_i^s generation was performed by sampling from the original ULS dataset without replacement. Additionally, when generating the dataset, the requirement of having more than one text for each tag field value from T_i^s was ensured. Generation for a given size s was performed as follows:

(1) T_i^s = select s random examples from ULS without replacement.

(2) GT_i^s = group T_i^s by the tag field.

(3) NGT_i^s = obtain the number of texts from all GT_i^s groups whose size > 1 text.

(4) if $NGT_i^s < s$, add one random example without replacement to T_i^s from ULS, go to step 2.

(5) if $NGT_i^s > s$, return one random example from T_i^s to ULS, go to step 2.

(6) return from T_i^s to ULS all texts included in GT_i^s in groups of size 1.

(7) T_i^s is the result.

Assessment was performed by calculating SC for each DS on each of the generated T_i^s datasets using

values from the tag field as cluster labels. Since the purpose of calculating SC is to determine the best DS, i.e., to build a ranking, not the absolute value of SC is important but the value of DS assessments relative to each other. This means that the original SC value can be subjected to transformations if this leads to building a better ranking (criterion provided below). The following types of transformations were considered:

• Scaling to the interval [0, 1]: $\mathrm{SilC}^{01}(x) = (x - \mathrm{min\_val})/(\mathrm{max\_val} - \mathrm{min\_val})$, where x is the absolute SC value in one of the DS for which the scaled SC value is calculated, $\mathrm{SilC}^{01}(x)$ is the scaled SC value, min_val is the minimum SC value (obtained in one of the DSs) on dataset T_i^s, and max_val is the maximum SC value obtained in one of the DSs on dataset T_i^s.

• Softmax transformation: $\mathrm{SilC}^{\mathrm{softmax}}(x) = \frac{e^x}{\sum_j^N e^{x_j}}$, where N is the number of DSs, x_j is the SC value for the jth DS.

As a criterion for the validity of choosing a transformation (and, accordingly, ranking), we will use the correlation of the obtained values of transformed DS assessments with DS assessment based on AMI values obtained during clustering. The $\mathrm{SilC}^{\mathrm{softmax}}(x)$ assessment was selected as the most suitable, due to the fact that additional experiments showed that it gives the highest average correlation value for all dataset sizes. From the fact that the softmax transformation gives higher correlation with the ranking based on AMI assessment, it can be assumed that DSs with higher assessments correlate more strongly with each other than DSs with lower ones.

Figure 6 presents the dependence of $\mathrm{SilC}^{\mathrm{softmax}}(x)$ on dataset size for the entire set of DSs. Here, interpretation of the absolute assessment value is not assumed; only the mutual arrangement of DS matters. DSs in the legend are ordered by decreasing assessment at the maximum number of clusters (i.e., by the position of the rightmost points of the graph), DS graph designations coincide with designations in Fig. 5.

From Fig. 5, it can be seen that DS not_lemmatized-tfidf-L2-cityblock (graph marked with a black triangle on the right) divides all DSs into two parts. Let us call all DSs whose graphs are above this line good (graphs marked with a curly bracket with a solid line), and all DSs located below this line bad (graphs marked with a curly bracket with a dashed line).

When comparing good DSs with the order of DSs based on clustering quality (in the legend of Fig. 5), it can be seen that the list of good DSs does not contain DS lemmatization-countvec-no_norm-euclidean, which is in the 14th position in the AMI-based ranking, and also contains DS not_lemmatized-countvec-L2-cityblock, which is in the 16th position in the AMI-based ranking; otherwise, the list of good DSs contains the leading 15 DSs from the AMI-based ranking.

Let us assess the degree of correlation of assessments of good and bad DSs separately with assessments of these DSs based on AMI. The results of assessment based on Pearson correlation coefficient are presented in Figs. 7 and 8, respectively. On the horizontal axis, the graphs show the size of the dataset, on which $\mathrm{SilC}^{\mathrm{softmax}}(x)$ calculation was performed.

From Figs. 7 and 8, it can be seen that the assumption that DSs with high assessment both in the AMI-based ranking and in the SC-based ranking correlate more strongly with each other than DSs with lower assessment proved correct. Since the purpose of the proposed approach is to determine the most suitable DS for clustering, in assessing its applicability it is possible to focus on the degree of correlation of DSs from the good list. From the graph in Fig. 7, it can be seen that the average correlation coefficient value for datasets with a size of 40 examples and more exceeds 0.8, which indicates strong correlation when using datasets of the specified size for calculating $\mathrm{SilC}^{\mathrm{softmax}}(x)$, while the average p-value does not exceed 10-3, which indicates high statistical significance of the obtained result.

3.5. Assessment of Efficiency of Applying the Proposed Clustering Process Compared to the Iterative Process

The proposed approach is oriented toward solving a relatively narrow class of problems (clustering with a small amount of labeled data) and uses its features. In this context, comparison with the approaches mentioned in this article, oriented toward optimizing the search for parameter values in the general case, is inconsistent. In connection with this, let us assess how much the number of required experiments for clustering execution, the most computationally intensive stage of the overall process, can be reduced without considering the application of optimization approaches.

Suppose the researcher has C possible clustering algorithms, for each algorithm $c_i \in C$ it is required to investigate a set of parameters P_i, for each parameter $p_{ij} \in P_i$ it is required to enumerate v_{ij} different values. Then the total number of variants for enumerating clustering parameter values is $N = \sum_{i=1}^{C} \prod_{j=1}^{P_i} v_{ij}$. Let it also be required to consider A data transformation methods, for each method $a_i \in A$ it is required to consider d_i compatible dissimilarity measures, then the number of resulting DSs is $M = \sum_{i=1}^{|A|} d_i$. The total number of experiments required to check the values of all parameters of the iterative clustering process is $T = NM$.

Let m be the selected number of best DSs determined using the ranking based on SC, then the total number of clustering experiments that will need to be conducted in case of applying the proposed approach

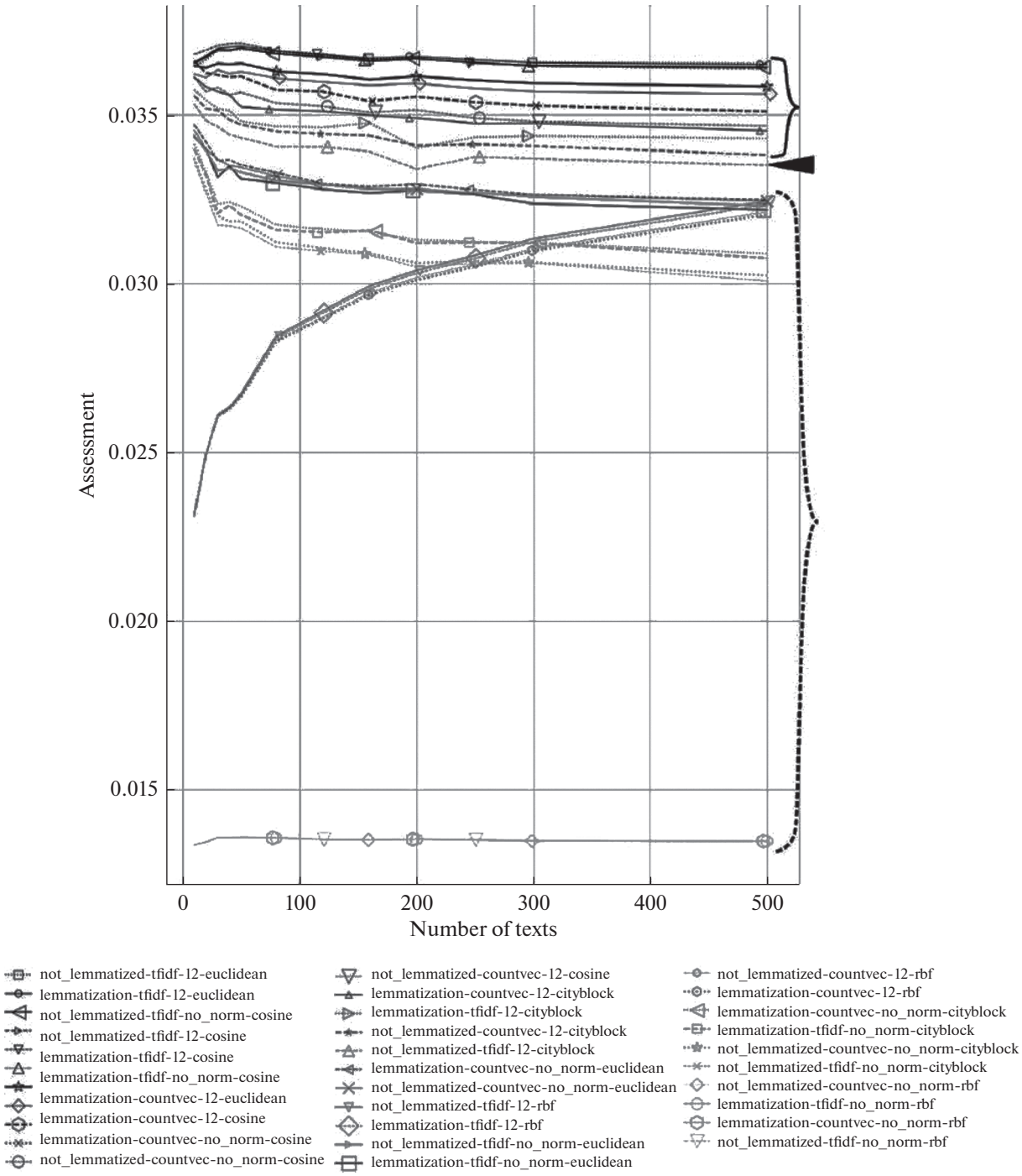

Fig. 6. Dependence of $\mathrm{SilC}^{\mathrm{softmax}}(x)$ on data portion size.

is $t = Nm$, and $w = m/M$ shows what proportion of the original set of clustering experiments needs to be conducted. Let $M = 32$ (as presented in this work), $C = 3$, all $P_i = 1$, all $v_{ij} = 10$, $m = 4$.

Then the number of variants for enumerating clustering parameter values (N) is 30, the total number of clustering experiments for the iterative process (T) is 960, and the number of clustering experiments for the proposed process (t) is 120. Hence, the proportion of clustering experiments that needs to be conducted when implementing the proposed process, from the number of required experiments in the original process (w) is 0.125, which corresponds to a reduction by a factor of $1/0.125 = 8$.

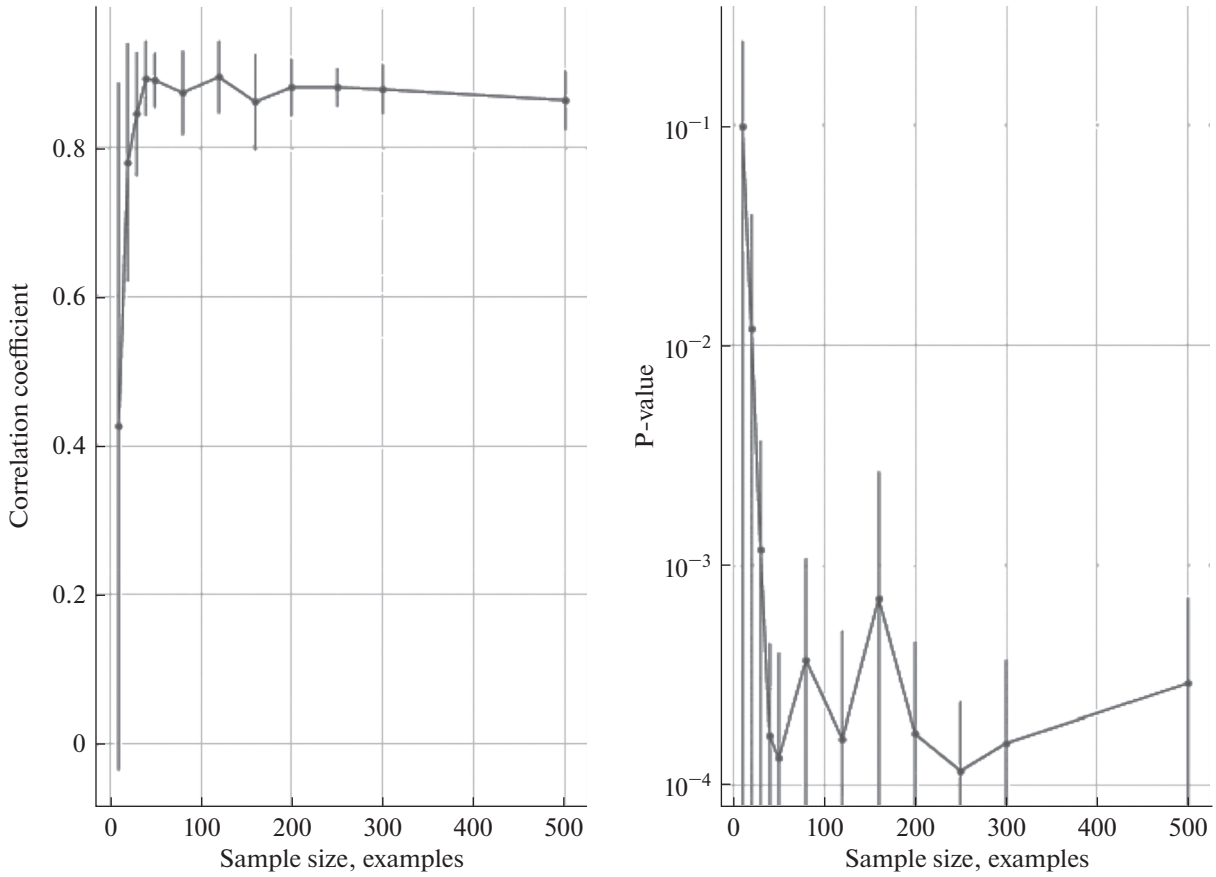

Fig. 7. Dependence of the mean value and standard deviation of Pearson correlation coefficient of the $SilC^{softmax}(x)$ assessment and AMI assessment for good DSs on the dataset size used for $SilC^{softmax}(x)$ calculation.

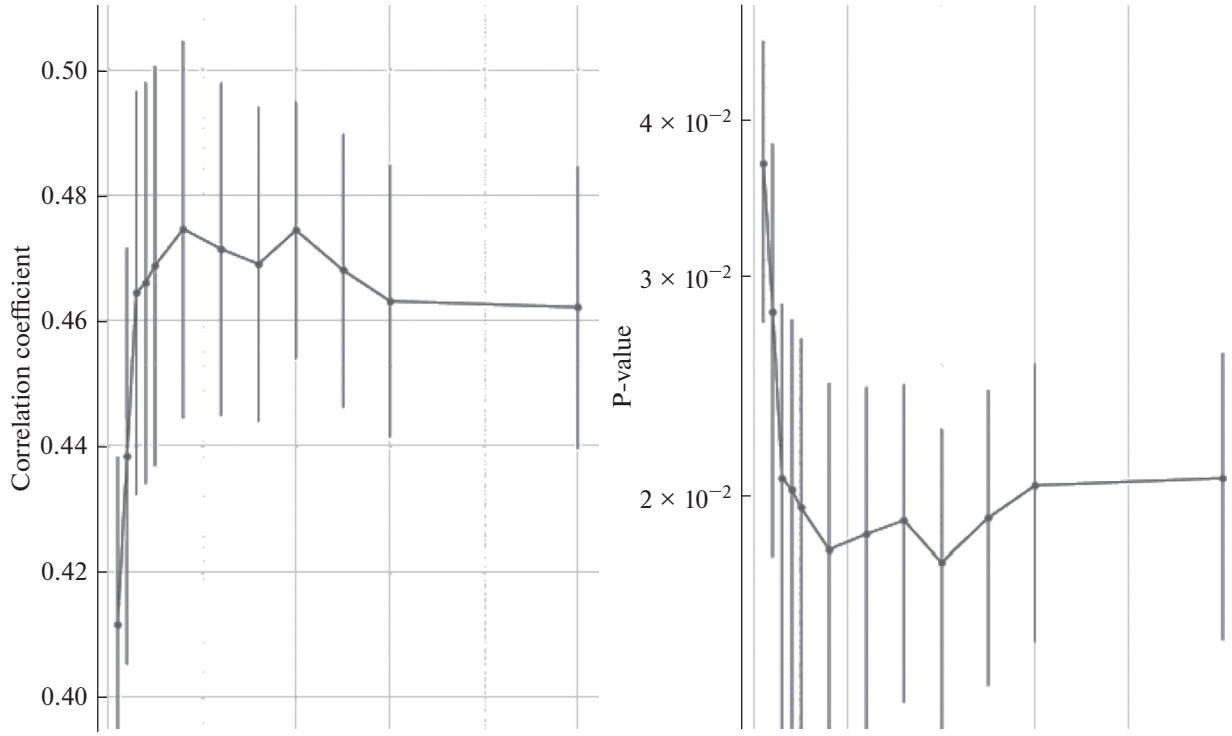

Fig. 8. Dependence of the mean value and standard deviation of Pearson correlation coefficient of the $SilC^{softmax}(x)$ assessment and AMI assessment for bad DSs on the dataset size used for $SilC^{softmax}(x)$ calculation.

CONCLUSIONS

In this work, it was proposed to reduce the search space for optimal parameters of the clustering process through its division into two stages: (1) search for the best dissimilarity spaces (DSs), defined by data-transformation parameters together with the dissimilarity measure defined on the transformed data, and (2) search for optimal algorithms and clustering parameters for the best DSs found in the previous stage. At the same time, comparison of DSs by degree of correspondence to the solved task at stage 1 is proposed to be performed based on the silhouette coefficient using a small amount of labeled data.

Using the lenta-ru-news text dataset, the applicability of the proposed DS assessment method at stage 1 was verified. Verification was performed by comparing DS assessment based on SC with results of external clustering quality assessment on 10000 examples with the variable "number of clusters" parameter. For assessments obtained on datasets with a size of 40 examples or more, a high degree of correlation with external clustering quality assessment was established. This makes it possible to conclude that the proposed assessment method is applicable in the proposed clustering process on the considered dataset using the data preprocessing methods, clustering algorithm, and their parameters considered in the work. On this basis, it is possible to assume that the proposed method can also be applied to other datasets using other data preprocessing methods, clustering algorithms, and their parameters.

An assessment of the expected efficiency improvement in the number of required experiments when applying the proposed clustering process compared to the iterative one was provided.

As further research directions, it is advisable to consider the possibility of assessing the effectiveness of combining the proposed approach with existing methods for optimizing the search for machine learning process parameters, the influence on the correlation between assessment based on the silhouette coefficient and external clustering quality assessment: the degree of data diversity, the proportion of labeled classes from the set, types of clustered data (such as textual, tabular, visual, etc.), and the type of clustering algorithm used; it is also advisable to investigate other measures, in addition to the SC, for assessing DS correspondence to the solved task.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

This article reflects the personal position of the authors. The content and results of this study should not be considered, including being cited in any publications, as the official position of the Bank of Russia or an indication of official policy or regulatory decisions. Any errors in this material are exclusively those of the authors.

REFERENCES

1. Ackerman, M., Adolfsson, A., and Brownstein, N., An effective and efficient approach for clusterability evaluation, *arXiv Preprint*, 2016. https://doi.org/10.48550/arXiv.1602.06687
2. Bergstra, J. and Bengio, Y., Random search for hyper-parameter optimization, *J. Mach. Learn. Res.*, 2012, vol. 13, no. 2, pp. 281–305.
3. Bora, M.D.J. and Gupta, D.A.K., Effect of different distance measures on the performance of *K*-means algorithm: An experimental study in Matlab, *Internatinonal Journal of Computer Science and Information Techonolgies*, 2014, vol. 5, no. 2, pp. 2501–2506.
4. Brazdil, P., Giraud-Carrier, C., Soares, C., and Vilalta, R., *Metalearning: Applications to Data Mining*, Cognitive Technologies, Berlin: Springer, 2008. https://doi.org/10.1007/978-3-540-73263-1
5. Dash, M., Choi, K., Scheuermann, P., and Liu, H., Feature selection for clustering—A filter solution, *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, Maebashi City, Japan, 2002, IEEE, 2002, pp. 115–122. https://doi.org/10.1109/icdm.2002.1183893
6. *Data Clustering: Algorithms and Applications*, Aggarwal, C.C. and Reddy, C.K., Eds., New York: Chapman and Hall/CRC, 2014. https://doi.org/10.1201/9781315373515
7. Feurer, M. and Hutter, F., Hyperparameter optimization, *Automated Machine Learning*, Hutter, F., Kotthoff, L., and Vanschoren, J., Eds., The Springer Series on Challenges in Machine Learning, Cham: Springer, 2019, pp. 3–33. https://doi.org/10.1007/978-3-030-05318-5_1
8. Hernández-Reyes, E., García-Hernández, R.A., Carrasco-Ochoa, J.A., and Martínez-Trinidad, J.F., Document clustering based on maximal frequent sequences, *Advances in Natural Language Processing*, Lecture Notes in Computer Science, vol. 4139, Berlin: Springer, 2006, pp. 257–267. https://doi.org/10.1007/11816508_27
9. Holder, Ch., Middlehurst, M., and Bagnall, A., A review and evaluation of elastic distance functions for time series clustering, *Knowl. Inf. Syst.*, 2023, vol. 66, no. 2, pp. 765–809. https://doi.org/10.1007/s10115-023-01952-0
10. Hui, X. and Li, Z., *Clustering validation measures, Data Clustering: Algorithms and Applications*, Boca Raton, FL: CRC Press, 2014, pp. 571–606.
11. Jain, A.K., Murty, M.N., and Flynn, P.J., Data clustering, *ACM Comput. Surv.*, 1999, vol. 31, no. 3, pp. 264–323. https://doi.org/10.1145/331499.331504

12. Kassambara, A., *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*, STHDA, 2017.
13. Kaufman, L. and Rousseeuw, P., *Clustering by means of medoids, Data Analysis Based on the L1-Norm and Related Methods*, Dodge, Y., Ed., North-Holland, 1987, pp. 405–416.
14. Li, Y., Zhang, Y., and Wei, X., Hyper-parameter estimation method with particle swarm optimization, *arXiv Preprint*, 2020. https://doi.org/10.48550/arXiv.2011.11944
15. Mahdavi, K., Enhanced clustering analysis pipeline for performance analysis of parallel applications, *PhD Dissertation*, Barcelona: Universitat Politècnica de Catalunya, Departament d'Arquitectura de Computadors, 2022. https://doi.org/10.5821/dissertation-2117-375586
16. Nelder, J.A. and Mead, R., A simplex method for function minimization, *Comput. J.*, 1965, vol. 7, no. 4, pp. 308–313. https://doi.org/10.1093/comjnl/7.4.308
17. Nguyen, Q.H. and Smith, V.J.R., Internal quality measures for clustering in metric spaces, *International Journal of Business Intelligence and Data Mining*, 2008, vol. 3, no. 1, pp. 4–29. https://doi.org/10.1504/ijbidm.2008.017973
18. Romano, S., Vinh, N.X., Bailey, J., and Verspoor, K., Adjusting for chance clustering comparison measures, *J. Mach. Learn. Res.*, 2016, vol. 17, no. 1, pp. 4635–4666.
19. Rousseeuw, P.J., Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *J. Comput. Appl. Math.*, 1987, vol. 20, pp. 53–65. https://doi.org/10.1016/0377-0427(87)90125-7
20. Schneider, M., Grinsell, J., Russell, T., Hickman, R., and Thomson, R., Identifying indicators of bias in data analysis using proportionality and separability metrics, *Proceedings of SBP-BRiMS Conference*, Washington, DC, 2019. http://sbp-brims.org/2019/proceedings/papers/working_papers/Schneider.pdf. Cited January, 2024.
21. Thornton, C., Hutter, F., Hoos, H.H., and Leyton-Brown, K., Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms, *Proceedings of the 19th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining*, Chicago, 2013, Ghani, R., Senator, T., Bradley, P., and He, J., Eds., New York: Association for Computing Machinery, 2013, pp. 847–855. https://doi.org/10.1145/2487575.2487629
22. Tong, Y. and Hong, Z., Hyper-parameter optimization: A review of algorithms and applications, *arXiv Preprint*, 2020. https://doi.org/10.48550/arXiv.2003.05689
23. Vincent, A.M. and Jidesh, P., An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms, *Sci. Rep.*, 2023, vol. 13, no. 1, p. 4737. https://doi.org/10.1038/s41598-023-32027-3
24. Vinh, N.X., Epps, J., and Bailey, J., Information theoretic measures for clusterings comparison, *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, 2009, New York: Association for Computing Machinery, 2009, pp. 1073–1080. https://doi.org/10.1145/1553374.1553511
25. Vysala, A. and Gomes, J., Evaluating and validating cluster results, *Computer Science Information Technology*, Wyld, D.C. et al., Eds., AIRCC Publishing Corporation, 2020, vol. 10, no. 9, pp. 37–45. https://doi.org/10.5121/csit.2020.100904
26. Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., and Deng, S., Hyperparameter optimization for machine learning models based on Bayesian optimization, *J. Electron. Sci. Technol.*, 2019, vol. 17, no. 1, pp. 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120
27. Xu, R. and Wunschii, D., Survey of clustering algorithms, *IEEE Trans. Neural Networks*, 2005, vol. 16, no. 3, pp. 645–678. https://doi.org/10.1109/tnn.2005.845141
28. Yang, L. and Shami, A., On hyperparameter optimization of machine learning algorithms: Theory and practice, *Neurocomputing*, 2020, vol. 415, pp. 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Translated by O. Pismenov