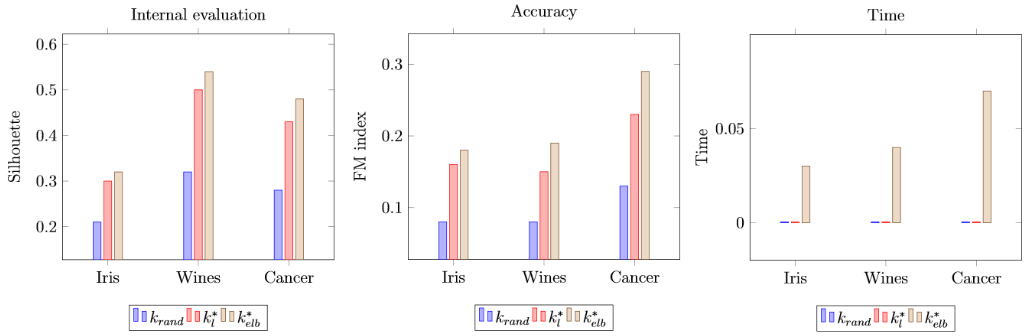
Graphical Abstract

OPIC: the Method of Machine Learning Automation Based on Objective Prior Information Criteria

Ildar Baimuratov, Dmitry Mouromtsev



Highlights

OPIC: the Method of Machine Learning Automation Based on Objective Prior Information Criteria

Ildar Baimuratov, Dmitry Mouromtsev

- Hyperparameter optimization leads to learning complexity gain
- Hyperparameter selection can be considered as decision-making under uncertainty
- Information measures can be considered as utility functions in hyperparameter selection
- Objective prior distributions can be used for uncertainty modeling
- OPIC allow selecting hyperparameters in constant time

OPIC: the Method of Machine Learning Automation Based on Objective Prior Information Criteria

Ildar Baimuratov*, Dmitry Mouromtsev

ITMO UNiversity, Kronverksky Pr. 49, bldg. A, St. Petersburg, 197101, Russia

Abstract

In this research, we consider the problem of computational complexity gain during automated hyperparameter optimization of machine learning algorithms. We analyze present hyperparameter optimization heuristics and conclude that all of them inevitably lead to complexity gain. Instead, we propose to consider the hyperparameter selection problem as decision-making under uncertainty and to use decision criteria for choosing hyperparameters value, combined with information estimation as utility function and objective prior distributions for uncertainty modeling. As a result, we developed two objective prior information criteria (OPIC), that allow selecting hyperparameters in constant time. Experimental research showed that the results of learning based on the proposed automation method are much better than ones, based on random hyperparameters.

Keywords: automated machine learning, hyperparameter optimisation, decision-making under uncertainty, Kullback–Leibler divergence, objective priors

1. Introduction

In the last decades, machine learning techniques proved to be useful in many practical tasks. However, the success of using machine learning substantially depends on the effective solution of multiple related tasks. One of them is hyperparameter selection.

*Corresponding author

Email addresses: baimuratov.i@gmail.com (Ildar Baimuratov), d.muromtsev@gmail.com (Dmitry Mouromtsev)

The problem is that hyperparameter selection in most cases is performed manually, based on experts' experience or intuition. Manual hyperparameter selection has several evident drawbacks. First, new machine learning techniques emerge continuously, while existing ones get more and more complicated. For manual hyperparameter selection, an expert needs a considerable theoretical and practical background. Second, manual hyperparameter selection is usually performed as tuning, which takes a lot of iterations and human time. Finally, manual hyperparameter selection may lead to suboptimal results. It may be caused by limitations of human computational abilities or inefficiency of a chosen heuristic. All this makes it difficult to apply machine learning techniques. Even experts need a long time for developing a satisfying solution, while efficient use of machine learning by non-experts appears to be impossible at all.

In recent years, the research field aimed at automation of machine learning emerged. Informally automated machine learning is learning, where a user only inputs data into a system and the system designs the process of learning most efficiently by itself. Systems of automated machine learning allow avoiding expert assessment in several tasks, particularly, in hyperparameter selection. As a result, automation methods allow using state-of-the-art machine learning techniques by users, that don't have enough time or other resources to master them in detail. Systems of automated machine learning already proved to be able to outperform expert-based solutions [1].

The present approach to machine learning automation supposes hyperparameter selection to be an optimization task. In this paper, we consider the problem of complexity gain, caused by automated hyperparameter optimization techniques. We conclude that under this approach complexity gain is inevitable. The goal of this paper is to develop the method, which allows improving the quality of learning by selecting hyperparameter values without complexity gain. We propose to consider hyperparameter selection as decision-making under uncertainty and to apply decision-making criteria combined with information estimation as utility function and objective prior distributions as uncertainty models.

There are numerous formulations of the hyperparameter optimization task: hyperparameter optimization [2], full Model Selection [3] or combined algorithm selection and hyperparameter [4] [5]. In all cases such optimization consists of defining an objective function, assessing results of learning, and considering a learning algorithm as a black box, matching hyperparameters to the objective function. Optimization is implemented as searching for the combination of hyperparameters that yields the result with the best value of the objective function.

The limitation of optimization algorithms performance was formulated as the No Free Lunch theorem [6]. There are researches considering the NFL theorem itself. The one most related to the present research is Everitt et al. [7], which will be discussed in detail later. Its authors apply algorithmic information framework to show that a free lunch is possible, but they don't provide any computable model. Instead of that, we propose a combinatorial approach, which proved to be useful for adjustment of mutual information and rand index in clustering comparing [8].

The outline of the paper is as follows. In the next section, we present a detailed analysis of the hyperparameter optimization task and complexity gain problem. In Section 3 we review several popular heuristics for hyperparameter optimization and evaluate complexity gain they imply. In Section 4 we present our decision theory inspired approach and review relevant criteria for decision-making under uncertainty. In Section 5 we present preliminary definitions resulting in the formal model of learning. In Section 6 we define the information measure intended to be the objective function. In Sections 7 we present the developed hyperparameter selection criteria. Sections 8 provides the results of empirical research. Finally, there are discussion and conclusion sections.

Earlier we published some results in Baimuratov et al. [9]. We presented there the information estimation measure, described in Section 6, and the objective prior distribution from Section 7.2. Combining them, we got the expected information measure, described in Section 7.2 as well, which we used for clustering evaluation. In the present research, we introduce another objective prior and add decision-making criteria to the information measure and objective priors to get the hyperparameter selection method.

2. The Problem of Complexity Gain in Hyperparameter Optimization

Formulation of the hyperparameter optimization task is following. A search space is a combination of an algorithm space $A = \{a^1, ..., a^k\}$ and a corresponding hyperparameters space $\Lambda = \{\Lambda^1, ..., \Lambda^k\}$. Selection of an algorithm $a^j \in A$ and its hyperparameters $\lambda^j \in \Lambda^j$ is based on a loss function $L(a^j_{\lambda_j}, D_{train}, D_{test})$, defined on a train data D_{train} and test data D_{test} with k-fold cross-validation. Denoting the optimal algorithm as a^* and its optimal hyperparamters as λ^*, we have

$$a^*_{\lambda*} \in argmin_{a^j \in A, \lambda \in \Lambda^j} \frac{1}{k} \sum_{i=1}^{k} L(a^j_\lambda, D^i_{train}, D^i_{test}). \tag{1}$$

Consider computational complexity gain during this process. The search is performed on the following sets:

- algorithms A;
- hyperparameters Λ;
- cross-validation partitions K.

In the same time each search iteration includes:

- implementing an algorithm a with particular hyperparameters λ;
- evaluating a result with the loss function L.

Therefore, the time complexity $T(a^*_{\lambda^*})$ of finding the optimal configuration $a^*_{\lambda^*}$ of the algorithm a^* and its optimal hyperparamters λ^* can be described as

$$T(a^*_{\lambda^*}) = k|A_\Lambda|T(a_\lambda)T(L), \tag{2}$$

where k is the number of cross-validation partitions, $|A_\Lambda| = |A||\Lambda|$, $T(a_\lambda)$ is average time complexity of learning implementation and $T(L)$ is average time complexity of calculating loss function.

Consider an example. Suppose there is a d-dimensional data set with n objects and we want to cluster them with a centroid-based clustering algorithm. Centroid-based clustering algorithms are algorithms, such as k-means, k-medoids, or k-medians, therefore, let $|A| = 3$. The key hyperparameter of centroid-based algorithms is number of clusters k [10], which ranges from 1 to n, therefore, let $|\Lambda| = n$. In addition, usually 10-fold cross-validation is applied for robustness, i.e. $k = 10$. Suppose, we use the Calinski–Harabasz index for evaluating results of clustering. It depends on the number of clusters k and the number of objects n, therefore, $T(L) = O(kn)$. Finally, the complexity of the algorithms itself is $T(A) = O(nkdi)$, where i is the number of iterations, required for convergence, which in the worst case equals to $i = 2^{\Omega(\sqrt{n})}$ [11]. Summing up, the complexity of the finding the optimal learning model $a^*_{\lambda^*}$ in the described task is

$$T(a^*_{\lambda^*}) = O(2^{\Omega(\sqrt{n})} k^2 n^3 d). \tag{3}$$

Therefore, hyperparameter optimization may lead to superpolinomial complexity gain.

3. Existing Optimization Heuristics

In practice, the search space is limited only externally: by a number of algorithms and hyperparameters, available in a particular system, or by a user's input. Hyperparameter optimization with large data sets or with complex learning models leads to enormously high computational complexity. It can take from several hours to several days.

There are several heuristics for speeding-up hyperparameter optimization. They reduce the complexity of optimization but sacrifice the quality of results. We consider several optimization heuristics and analyze the complexity gain they imply.

Grid search and *random search* [12]. In these two methods, complexity reduction is achieved by searching not through the whole hyperparameter space, but through its subset, the size of which is specified by a user. In the case of grid search, this subset is specified according to some distribution, in case of random search it is formed randomly. Applying the formal approach from the previous section, the complexity gain of these methods can be described as follows:

$$T(a^*_{\lambda^*}) = k|A'_{\Lambda'}|T(a_\lambda)T(L), \tag{4}$$

where $A'_{\Lambda'} \subseteq A_\Lambda$, therefore,

$$k|A'_{\Lambda'}|T(a_\lambda)T(L) \leq k|A_\Lambda|T(a_\lambda)T(L). \tag{5}$$

Ensemble method [13]. In ensemble learning, we still consider several algorithms and hyperparameters configurations, i.e. the subset $A'_{\Lambda'}$, but instead of iteratively searching the best one, they are considered in parallel. An ensemble ensures robustness itself, hence it does not require cross-validation, but it requires some voting function V, which depends on the number of target functions under consideration, to combine them into the final output

$$T(a^*_{\lambda^*}) = |A'_{\Lambda'}|T(a_\lambda)T(L)T(V), \tag{6}$$

where $T(V)$ denotes average time, required for computing the voting function V.

Bayesian optimization [14] also includes restriction of hyperparameter space, but additional complexity reduction is achieved by prioritizing hyperparameter values with a less expensive acquisition function Ac, based on some prior probability model. After evaluating the selected configuration the probability model is updated. The complexity gain is the following:

$$T(a^*_{\lambda^*}) = |A_\Lambda|T(Ac) + k|A'_{\Lambda'}|T(a_\lambda)T(L)T(U), \tag{7}$$

where U is the probability model update function.

Meta-learning [5]. Instead of evaluating the utility of available configurations, in meta-learning, it is meta-data of the data set or of the task being evaluated. Usually, this evaluation considered as a distance metric D, measuring the distance between the present meta-data and prior meta-data M, obtained in previous experiments. Each meta-data instance is associated with some configuration, and the priority of configurations is determined as the closeness of the corresponding prior meta-data to meta-data at hand. This evaluation does not require the performing of learning itself, therefore, it does not imply calculating losses and cross-validation. The complexity gain is described as following:

$$T(a^*_{\lambda^*}) = |M|T(D), \tag{8}$$

where $T(D)$ is time for calculating distance metric D.

There are other optimization heuristics, such as gradient descent [15] or evolutionary algorithms [16]. Nevertheless, each of these optimization heuristics leads to computational complexity gain. We generalize this result in the following way. According to the “No free lunch” theorem, there is no optimization method, that allows achieving average complexity lower, than of random search. Only prior information about which method is better for a particular task can help to reduce the complexity. However, you have to pay an additional computational costs to obtain such prior information in turn. Therefore, the described approach to hyperparameter selection inevitably leads to computational complexity gain to some degree.

4. Decision-making Criteria

We propose to consider hyperparameter selection as decision-making under uncertainty and, consequently, to use decision-making criteria. There are several criteria for optimal decision in the decision theory. Consider the following [17]:

- The principle of maximum expected utility, or the *Bayes' criterion.* If the probability distribution P of the states of nature Ω is known, then the optimal decision b^*, according to the Bayes' principle, is defined as the decision a from the set of decisions A, that has maximum expected utility $u(a, \omega)$, where $\omega \in Omega$:

$$b^* = arg \max_i \sum_j P_j u(a_i, \omega_j). \tag{9}$$

- The principle of insufficient reason, or the *Laplace's principle.* If the probability distribution P of the states of nature Ω is unknown, one can assume, that they are equiprobable, then the optimal decision l^*, according to the Laplace's principle, is defined as following:

$$l^* = arg \max_i \sum_j \frac{1}{n} u(a_i, \omega_j), \tag{10}$$

  where $n = |\Omega|$, i.e. the number of states of nature.

We don't consider more complex decision criteria, as they require a user's input for non-deterministic values, like the pessimism coefficient in Hurwicz's criterion.

Let there is some hyperparameter h, a learning model lm and an objective function f, then the hyperparameter selection task can be represented as decision-making under uncertainty task, where values of hyperparameter h are considered as decisions, learning result $lm(h)$ – as an outcome and values of the objective function $f(lm(h))$ – as a utility. Then it becomes possible to apply decision-making criteria for defining the optimal hyperparameter value h^*. To do this, it is necessary to define a particular objective function and a learning models probability distribution to model uncertainty.

5. The Formal Learning Model

In order to define learning models probability distributions, we consider a general, or formal learning model. Usually, a learning model includes

- a set of input data X, or domain;

- a set of possible outputs Y, or co-domain;
- a target function $f : X \to Y$.

In this work, we consider partitions as an example of a base for defining learning models distribution. This partition model is applicable in such tasks as classification or clustering, where partitions correspond to the distribution of labels.

Let there is a target function $f : X \to Y$. We consider partition as the function *part*, that associates each $x \in X$ with a subset X_i, such that

$$X_i = \{x \in X : f(x) = y_i\}. \tag{11}$$

Therefore, each learning model can be associated with some partition. Let there are k such subsets, then the list $X_1, ..., X_k$ is a k-part partition $part^k(X)$ of the set X. Summing up, the function $part^k$ associates the set X with a k-part partition $part^k(X) = X_1 \cup ... \cup X_k$. The set of all partitions of X we denote as $Part(X)$, the set of all k-part partitions of X – as $Part^k(X)$.

Now we can use objective prior distributions, based on combinatorial properties of partitions, as probability distributions of learning models. In particular, we consider probability distributions, based on the relation between partitions of the *set* X and the partitions of the *number* $n = |X|$. Each k-part partition $part^k(X)$ of the set X can be associated with a k-part partition $part^k(n) = n_1 + ... + n_k$ of the number n, where $|X_i| = n_i$. We denote the set of all partitions of the number n as $Part(n)$ and the set of all k-part partitions of the number n – as $Part^k(n)$.

6. The Information Measure

In the present research we propose to use information estimation as the objective function for evaluating partitions. As starting point for defining a partition information measure we consider self-information

$$I(x_i) = -\log P(x_i) \tag{12}$$

and information entropy [18]

$$H(X) = -\sum_i P(x_i) \log P(x_i), \tag{13}$$

where X is an arbitrary random variable, $x_i \in X$ and $P(x_i)$ is probability of x_i. Further, we propose to apply normalization of self information to be able to compare partitions of sets of different sizes

$$I(x_i) = -\log_{|X|} P(x_i). \tag{14}$$

Consider the distribution produced by partition $part^k(X)$, i.e. the distribution of objects into the subsets $X_1, ..., X_k$, as the random variable. Then the probability distribution $P(X_i)$ is based on the relation $\frac{|X_i|}{|X|}$, therefore, the normalized self-information $I(X_i)$ of the subset X_i can be represented as

$$I(X_i) = -\log_{|X|} \frac{|X_i|}{|X|}. \tag{15}$$

But it follows from logarithm property that

$$I(X_i) = -(\log_{|X|} |X_i| - \log_{|X|} |X|) \tag{16}$$

and

$$I(X_i) = 1 - \log_{|X|} |X_i|. \tag{17}$$

Therefore, the normalized information entropy $H(part^k(X))$ of partition $part^k(X)$ is defined as

$$H(part^k(X)) = 1 - \sum_i \frac{|X_i|}{|X|} \log |X_i|. \tag{18}$$

Further, partition informativeness is intuitively evaluated as higher, as more the distribution, produced by partition $part^k(X)$, differs from the original distribution $P(x)$, where the original distribution is the uniform distribution $P(x) = \frac{1}{|X|}$. Therefore, we propose to use Kullback–Leibler divergence [19] $D_{KL}(P(part^k(X))||P(x))$ for measuring information gain produced by partition:

$$D_{KL}(P(part^k(X))||P(x)) = \sum_i P(X_i) \log \frac{P(X_i)}{P(X_i)}. \tag{19}$$

According to logarithm properties

$$D_{KL}(P(part^k(X))||P(x)) = \sum_i P(X_i)(\log P(X_i) - \log P(X_i)). \tag{20}$$

As $P(X_i) = \frac{|X_i|}{|X|}$ and $P(x) = \frac{1}{|X|}$, we have

$$D_{KL}(P(part^k(X))||P(x)) = \sum_i \frac{|X_i|}{|X|}(\log \frac{|X_i|}{|X|} - \log \frac{1}{|X|}). \tag{21}$$

After normalization we have

$$D_{KL}(P(part^k(X))||P(x)) = \sum_i \frac{|X_i|}{|X|}(\log_{|X|} \frac{|X_i|}{|X|} - \log_{|X|} \frac{1}{X}). \tag{22}$$

And finally, reducing

$$D_{KL}(P(part^k(X))||P(x)) = \sum_i \frac{|X_i|}{|X|}(\log_{|X|} .|X_i|) \tag{23}$$

Summing up, we propose to use the value $D_{KL}(P(part^k(X))||P(x))$ for estimating information gain of the partition $part^k(X)$. For shortness, we will denote it as $D(part^k(X))$.

7. The Prior Information Criteria

In this section we propose two prior information criteria, the first is based on the Laplace's decision-making criteria, the second – on the Bayes' one.

7.1. The Laplace Prior Information Criterion

If the probability distribution of learning models is unknown, the optimal hyperparameter value can be defined, based on the Laplace's criterion. To do this we define the uniform distribution of partitions of the number n:

$$P(part^k(n)) = \frac{1}{|Part(n)|}. \tag{24}$$

Substituting the distribution $P(part^k(n))$ and the information measure $D(part^k(X))$ as utility function to the Laplace's criterion, we get Laplace prior information criterion (LPIC) for defining the optimal number k_l^* of subsets in partition:

$$k_l^* = arg \max_i \sum_j \frac{1}{|Part(n)|} D(part_j^{k_i}(X)). \tag{25}$$

We are going to show that the LPIC is robust, i.e. it has the global maximum k_l^* such that $k_l^* \neq \min(k)$ and $k_l^* \neq \max(k)$. To do this consider the expected partition information $ED(part^k(X)$

$$ED(part^k(X)) = \sum_i \frac{1}{|Part(n)|} D(part_j^k(X)). \tag{26}$$

Let $n = 75$ and $k = 1, ..., 75$, then the plot of $ED(part^k(X))$ is presented at the Picture 1. As we see, the maximal informative partition has $k = 10$.

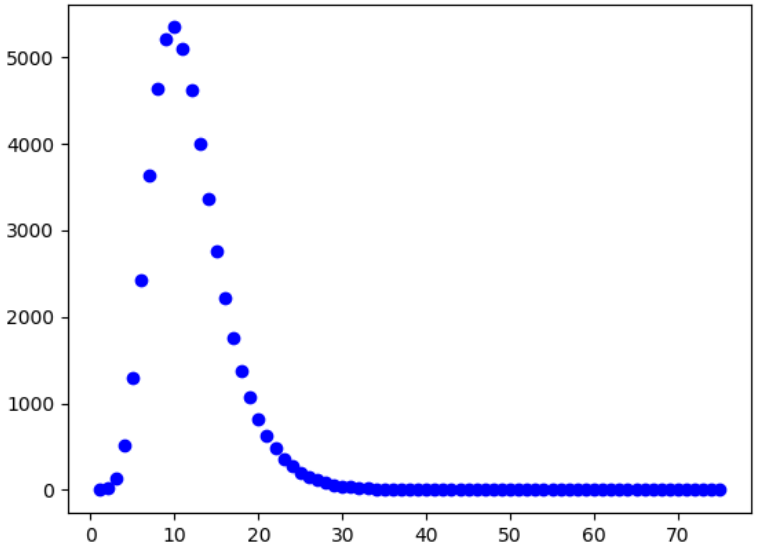


Figure 1: $ED(part^k(X))$ for $n = 75$ and $k = 1, ..., 75$

7.2. The Bayes Prior Information Criterion

However, the distribution of partitions $Part^k(n)$ of the number n relatively to partitions $Part^k(X)$ of the set X, such that $n = |X|$, is not uniform due to possibility of different combinations of elements in the subsets $X_1, ..., X_k$ and permutations of the subsets itself. Thereby, we suggest to consider application of the Bayes' criterion, but it requires to define an objective prior distribution of partitions of the number n.

The set of all possible partitions $Part(X)$ of the set X is represented by its exponential object X^X, therefore, the number of such partitions is described as $|Part(X)| = |X|^{|X|}$. Further, we denote the number of partitions from the set $Part^k(X)$, that fall under some particular number partition $part^k(n)$ as $|part^k(n)|$. This number is described by the following formula:

$$|part^k(n)| = \frac{n!}{n_1!...n_k!} \frac{n!}{k!(n-k)!} \frac{k!}{k_1!...k_m!}, \tag{27}$$

where k_j is the number of subsets with some equal number of elements n_i. Therefore, we define objective prior probability of the number partition $part^k(n)$ as relation of the number of related set partitions $|part^k(n)|$ to the total number of partitions $|Part(X)|$ of the set X

$$P(part^k(n)) = \frac{|part^k(n)|}{|Part(X)|}. \tag{28}$$

Substituting this distribution and the information measure to the Bayes' criterion, we get Bayes prior information criterion (BPIC) for defining the optimal number of subsets k_b^*:

$$k_b^* = arg \max_i \sum_j \frac{|part^{k_i}(n)|}{|Part(X)|} D(part_j^{k_i}(X)). \tag{29}$$

We are going to show that BPIC is robust, i.e. it has the global maximum k_b^* such that $k_b^* \neq \min(k)$ and $k_b^* \neq \max(k)$. To do this, consider expected partition information $ED(part^k(X))$

$$ED(part^k(X)) = \sum_i \frac{|part_i^k(n)|}{|Part(X)|} D(part_i^k(X)). \tag{30}$$

Let $n = 75$ and $k = 1, ..., 75$. The plot of $ED(part^k(X))$ is presented on the Picture 2. As we see, $k_b^* = 47$.

Further, we are going to show that the model of the objective prior distribution is correct. To do this, we will use Monte Carlo method: we will simulate s random partitions to get posterior distribution and calculate expected information $ED(part^k(X))$ for each k. Thereby we will find the posterior optimal number of subsets k_p^*. The model of the objective prior distribution will be correct, if the posterior k_p^* is equal to the prior k_b^*. Let $n = 75$ and

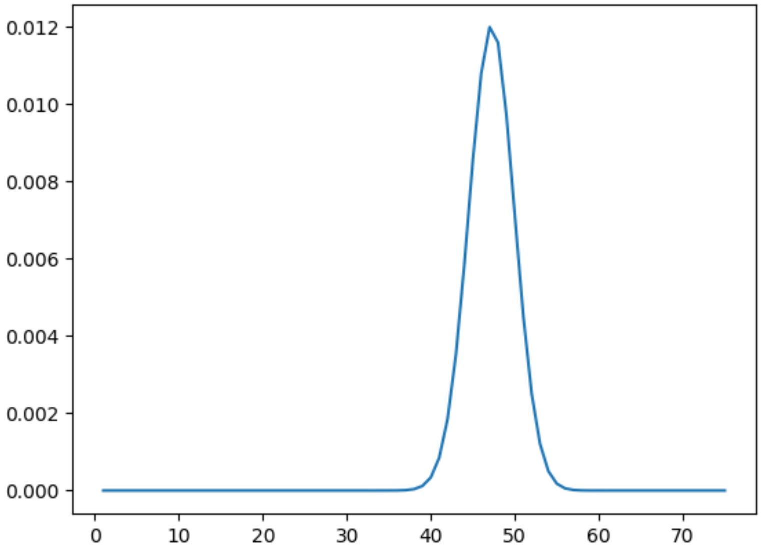


Figure 2: $ED(part^k(X))$ for $n = 75$ and $k = 1, ..., 75$

$s = 1, ..., 10000$, the result of the experiment is presented at the Picture 3. As we see, k_p^* becomes similar to k_b^*, starting from near $s = 1000$. Therefore, the model of the objective prior distribution is correct.

As LPIC and BPIC are both prior and depend only on metadata, particularly, on the size n of the data set, the values k_l^* and k_b^* can be determined before learning itself and only once for each n. Therefore, this hyperparameter selection task is reduced to matching n with the predefined value k_l^* or k_b^*. This task has constant complexity $O(1)$.

8. Experiments

We performed experimental research of the proposed machine learning automation method. In particular, we researched automation of determining the optimal number of clusters in the clustering task. To research the dependency of the prior information criteria values on different meta-data, we used synthetic data sets. To research the effectiveness of the proposed automation method we used though toy, but real data sets.

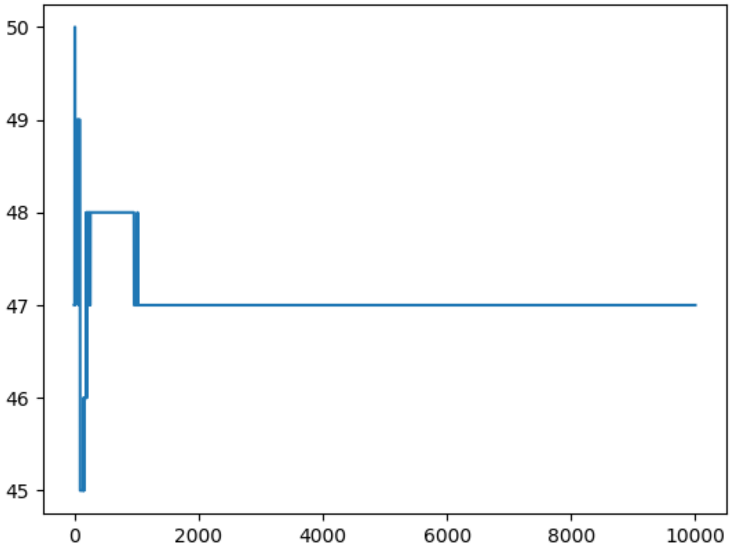


Figure 3: k_p^* for $n = 75$ and $s = 1, ..., 10000$

8.1. Experiments on Synthetic Data

In research on synthesized data sets, we considered the following task: to compare under different parameters dependency of values of a prior criterion with one of the values, obtained by posterior optimization based on an internal clustering metric.

We used K-means [20], namely sklearn.cluster.KMeans on Python, as clustering algorithm, elbow method [21], pyclustering.cluster.elbow, as the internal posterior criterion and the method make_blobs from the sklearn.datasets.samples_generator module as a data set generator. The algorithm for obtaining the prior optimal number of clusters k^* is described by the Procedure 1. We used LPIC for initial determining k^*. The Python realization is available at GitHub [22].

The dependency of the optimal number of clusters inf and of one according to the elbow method *elbow* on the true number of clusters *true* k, set as generation parameter, is presented at the Picture 4. The dependency of inf and *elbow* on the number of dimensions d of a data set is presented on the Picture 5. The dependency inf and *elbow* on the size n of a data set is

Procedure 1 Obtaining k^*

```
Input: n
Read $k^*$ for $n$
if $k^*$ is unknown then
    $k^* = 0$, $ed = 0$
    for $k = 1...n$ do
        $t = ED(k, n)$
        if $t > ed$ then
            $k^* = k$, $ed = t$
        end if
    end for
    Store $k^*$ for $n$
end if
Output: $k^*$
```

presented at the Picture 6.

It is seen on the plots that the optimal, according to LPIC, number of clusters does not depend on the true number of clusters, nor the number of dimensions, but depends on the size of a data set and correlates to the results of posterior optimization at some degree.

Further, we compared the results of automated learning based on LPIC with ones, based on internal posterior optimization, under different true numbers of clusters. To do this we generated several samples with $n = 50$, $d = 4$ and random true number of clusters. For each sample, we determined the optimal number of clusters, according to LPIC, and one, according to the elbow method. Then we performed K-means clustering with the corresponding parameters. Comparing results by another internal metric, Calinski-Harabasz index [23], is presented at the Picture 7, evaluation of accuracy concerning the true number of clusters – at the Picture 8.

As we see on the plots, the results of learning based on LPIC on average are slightly worse, than of posterior optimization, according to both the internal metric and accuracy, however, using LPIC we have enormous complexity reduction at the same time.

8.2. Experiments on Real Data

We did experiments of the proposed automation method on random samples from real, not synthesized data sets. We chose 'Iris', 'Wines' and 'Breast

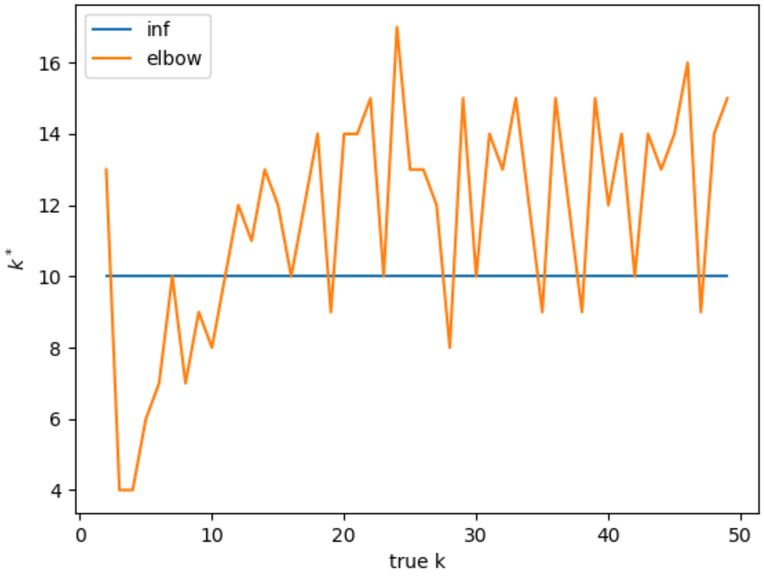


Figure 4: Dependency of *inf* and *elbow* on *true k*

cancer' [24] data sets. For each data set we made $s = 100$ random samples of size $n = 100$.

For each sample, we determined the optimal number of clusters k_l^* based on LPIC. We compared the results of clustering, based on the obtained values of k_l^*, with the results, based on a random number of clusters k_{rand}, and ones, obtained by the elbow method. We compared them by the internal metric – Silhouette index [25], namely "sklearn.metrics.silhouette_score", by accuracy – Fowlkes–Mallows index [26], "sklearn.metrics.fowlkes_mallows_score" and by time, spent on obtaining the optimal hyperparameter value.

The results for 'Iris' data set are presented at the Table 8.2, for 'Wine' data set – at the Table 8.2, for 'Breast cancer' data set – at the Table 8.2. As we see, the proposed automation method allows to improve the quality of results comparing to learning, based on random hyperparameter values, but it doesn't imply complexity gain. According to the results of experimental research, we improved the quality of results on 0,09-0,18 points by the Silhouette metric and on 0,07-0,1 points by the FM index, depending on the data set.

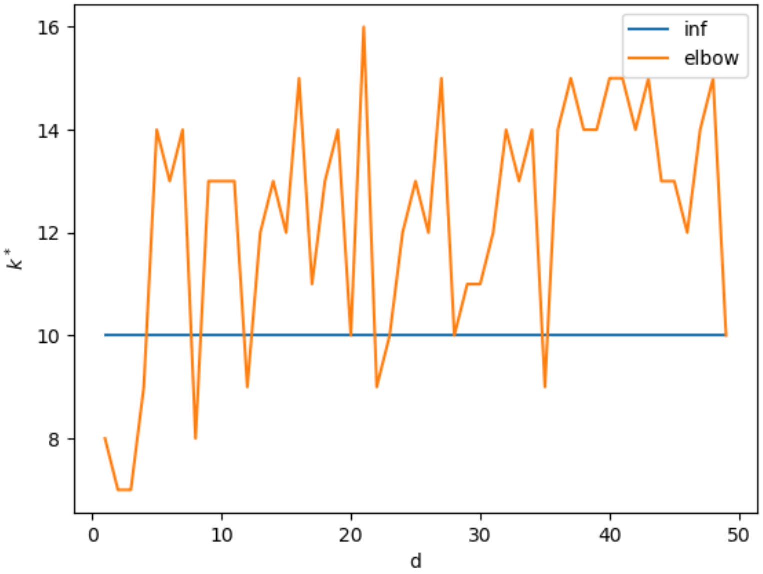


Figure 5: Dependency of inf and $elbow$ on d

	Silhouette	FM index	Time
k_{rand}	0,21	0,08	0,00
k_l^*	0,3	0,16	0,00
k_{elb}^*	0,32	0,18	0,03

Table 1: 'Iris' data set

	Silhouette	FM index	Time
k_{rand}	0,32	0,08	0,00
k_l^*	0,50	0,15	0,00
k_{elb}^*	0,54	0,19	0,04

Table 2: 'Wines' data set

	Silhouette	FM index	Time
k_{rand}	0,28	0,13	0,00
k_l^*	0,43	0,23	0,00
k_{elb}^*	0,48	0,29	0,07

Table 3: 'Breast cancer' data set

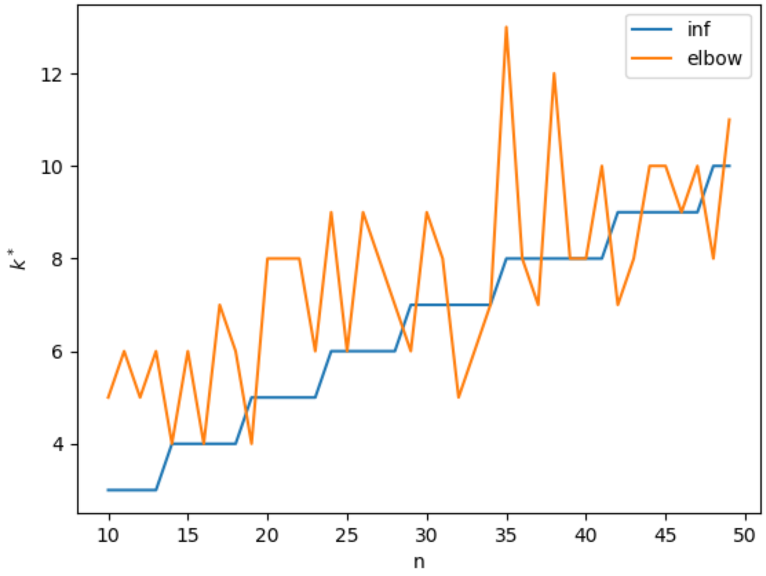
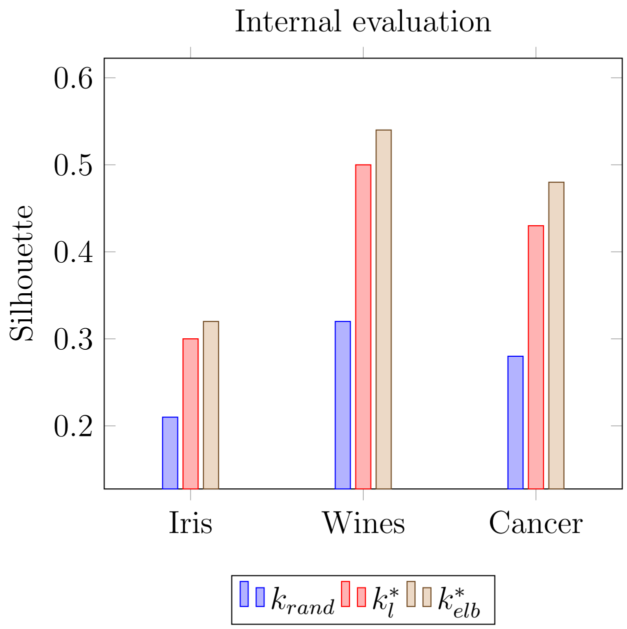


Figure 6: Dependency of inf and $elbow$ on n



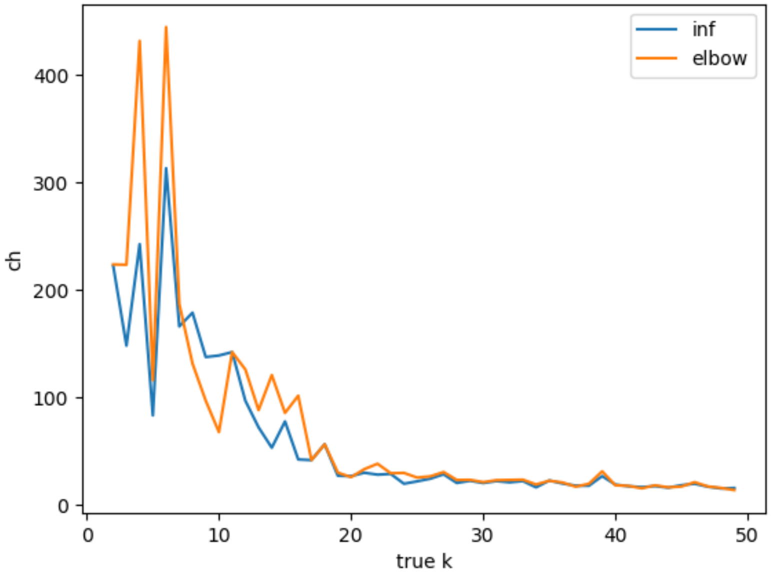
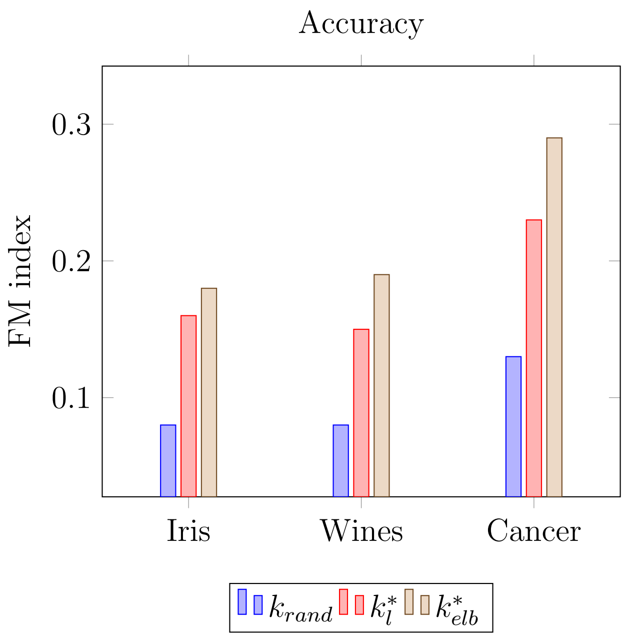


Figure 7: Comparing with Calinski-Harabasz index



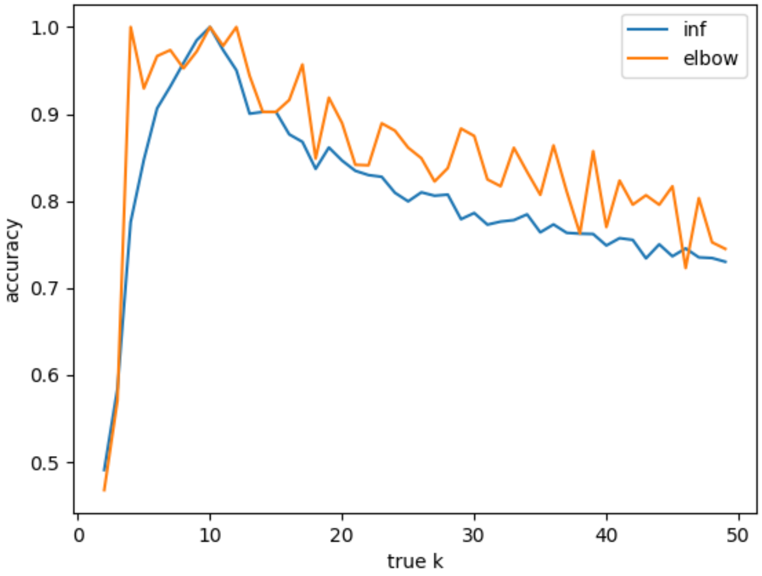
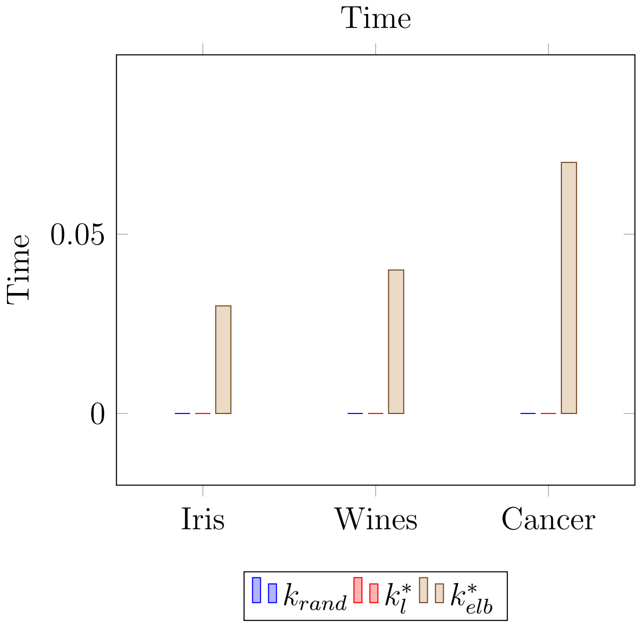


Figure 8: Comparing with accuracy



9. Discussion

The present research is closely related to the work Everitt et al. [7], where the authors apply Kolmogorov complexity to define a target function complexity and the universal distribution from algorithmic information theory [27] as distribution of target functions. Further, they use the notion of block uniformity from [28] to prove that there is a free lunch under the universal distribution, as it is not block uniform. In that case, the expected optimization time grows linearly regardless of an optimization strategy. However, this result is only theoretical as the authors don't provide any computable model for the universal distribution and don't present any empirical research. It is due to the fact that Kolmogorov complexity is incomputable by itself.

In contrast, instead of algorithmic concepts of Kolmogorov complexity and universal distribution we propose to use combinatorially inspired concepts of partition information and objective distributions, both are fully computable. Translating to the language of the paper, partition function $part(n)$ is indeed histogram $h : Y \to \mathrm{N}$, such that $|Y| = n$, the value $|part(n)|$ corresponds to the base class B_h of the histogram h. According to the definition of block uniformity, the objective distribution, used for LPIC, is block uniform, as every partition is equiprobable in it, while the objective distribution, used for BPIC, is not, therefore, there is a free lunch.

10. Conclusion

In this work, we presented the method of machine learning automation, called OPIC. This method is based on decision-making criteria, information estimation, and objective prior distributions. As a special case, we formulated the two prior information criteria for determining the optimal number of subsets in partitions, that are applicable in clustering automation. These criteria depend only on the size of a data set and allow determining the optimal number of subsets before learning. As a result, it becomes possible to improve the quality of results without computational complexity gain. The experimental research showed that OPIC improves the results of learning, comparing to ones, based on random hyperparameter values.

Further research will be focused, first, on applying the proposed automation method to other kinds of machine learning tasks, such as dimensionality reduction, that require other combinatorial models, and second, on developing new prior information criteria, using other decision-making principles.

11. Acknowledgement

This work was supported by the Government of Russian Federation (Grant 08-08)

References

[1] F. Hutter, L. Kotthoff, J. Vanschoren (Eds.), Automatic machine learning: methods, systems, challenges, Challenges in Machine Learning, Springer, Germany, 2019.

[2] B. Komer, J. Bergstra, C. Eliasmith, Hyperopt-sklearn: Automatic hyperparameter configuration for scikit-learn, in: Proceedings of the 13th Python in Science Conference (SCIPY 2014), 2014, pp. 32–37.

[3] H. J. Escalante, M. Montes, E. Sucar, Ensemble particle swarm model selection, in: Proceedings of the 2010th International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1–8.

[4] C. Thornton, F. Hutter, H. Hoos, K. Leyton-Brown, Auto-weka: Combined selection and hyperparameter optimization of classification algorithms, KDD (2012).

[5] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, F. Hutter, Efficient and robust automated machine learning, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, MIT Press, Cambridge, MA, USA, 2015, p. 2755–2763.

[6] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1997) 67–82.

[7] T. Everitt, T. Lattimore, M. Hutter, Free lunch for optimisation under the universal distribution, 2014 IEEE Congress on Evolutionary Computation (CEC) (2014).

[8] A. J. Gates, Y.-Y. Ahn, The impact of random models on clustering similarity, 2017. `arXiv:1701.06508`.

[9] I. Baimuratov, Y. Shichkina, E. Stankova, N. Zhukova, N. Than, A bayesian information criterion for unsupervised learning based on an objective prior, in: Computational Science and Its Applications – ICCSA 2019, Springer International Publishing, Cham, 2019, pp. 707–716.

[10] G. Hamerly, C. Elkan, Learning the k in k-means, Advances in Neural Information Processing Systems 17 (2004).

[11] D. Arthur, S. Vassilvitskii, How slow is the k-means method?, Proceedings of the Annual Symposium on Computational Geometry 2006 (2006) 144–153.

[12] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[13] T. G. Dietterich, Ensemble methods in machine learning, in: Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00, Springer-Verlag, Berlin, Heidelberg, 2000, p. 1–15.

[14] F. Hutter, H. H. Hoos, K. Leyton-Brown, Sequential model-based optimization for general algorithm configuration, in: Proceedings of the 5th International Conference on Learning and Intelligent Optimization, LION'05, Springer-Verlag, Berlin, Heidelberg, 2011, p. 507–523.

[15] L. Franceschi, M. Donini, P. Frasconi, M. Pontil, Forward and reverse gradient-based hyperparameter optimization, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1165–1173.

[16] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11, Curran Associates Inc., Red Hook, NY, USA, 2011, p. 2546–2554.

[17] T. Denœux, Decision-making with belief functions: A review, International Journal of Approximate Reasoning 109 (2019) 87–110.

[18] C. E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (1948) 379–423.

[19] S. Kullback, R. A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1951) 79–86. URL: http://www.jstor.org/stable/2236703.

[20] D. Arthur, S. Vassilvitskii, K-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, Society for Industrial and Applied Mathematics, USA, 2007, p. 1027–1035.

[21] R. L. Thorndike, Who belongs in the family, Psychometrika (1953) 267–276.

[22] I. Baimuratov, Prophet, 2020. URL: https://github.com/ldrbmrtv/Prophet.

[23] T. Caliński, H. JA, A dendrite method for cluster analysis, Communications in Statistics - Theory and Methods 3 (1974) 1–27.

[24] D. Dua, C. Graff, UCI machine learning repository, 2017. URL: http://archive.ics.uci.edu/ml.

[25] P. Rousseeuw, Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. comput. appl. math. 20, 53-65, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[26] E. B. Fowlkes, C. L. Mallows, A method for comparing two hierarchical clusterings, Journal of the American Statistical Association 78 (1983) 553–569.

[27] M. Li, P. M. Vitnyi, An Introduction to Kolmogorov Complexity and Its Applications, 3 ed., Springer Publishing Company, Incorporated, 2008.

[28] C. Igel, M. Toussaint, A no-free-lunch theorem for non-uniform distributions of target functions, J. Math. Model. Algorithms 3 (2004) 313–322.