ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

Sample Importance Guided Progressive Sampling-Based Bayesian Optimization for Automatic Machine Learning

Sufen Chen[1,a], Xueqiang Zeng[2,b,*]

[1]School of Information Engineering, Nanchang Institute of Technology, Nanchang, Jiangxi Province 330099, P.R. China
[2]School of Computer & Information Engineering, Jiangxi Normal University, Nanchang, Jiangxi Province 330022, P.R. China
[a]csf@nit.edu.cn, [b]xqzeng@jxnu.edu.cn
**Corresponding author*

Abstract: *In the era of big data, machine learning-based data analysis has been integrated into almost all walks of modern life. Before applying machine learning, a machine learning algorithm with its proper hyper-parameters have to be decided, where rich machine learning knowledge and lots of practical manual iterations are required. In order to popularize machine learning and allow non-professionals to use machine learning to solve problems, automatic machine learning model selection is particularly important. Among various existing automatic machine learning model selection methods, Progressive Sampling-based Bayesian Optimization (PSBO) is one of the most efficient and effective ones. However, PSBO adopted the progressive sampling with the traditional random sampling strategy, which does not consider the importance of individual samples. Based on the idea that more important and effective samples will make the model training results better, the paper proposed a Sample Importance Guided Progressive Sampling-based Bayesian Optimization (SIG-PSBO) for automatic machine learning. SIG-PSBO defines the sample importance by the difficulty to distinguish categories in a PCA feature space. Then samples with higher sample importance are more likely to be sampled for the subsequent model training. Extensive experiment results showed that the SIG-PSBO method can significantly shorten the search time and reduce the classification error rates compared to the original PSBO method.*

Keywords: *Automatic machine learning, Sample importance sampling, Principal component analysis, Progressive sampling*

1. Introduction

With a growing fountain of data in the modern information age, machine learning is increasingly beneficial to intelligent data analysis and has become the mainstream technology of intelligent data analysis. At the same time, machine learning is considered to be a dark art in some sense that can be acquired by only a few people [1]. In particular, algorithms and hyper-parameter selection require a large number of manual iterations to realize, which not only takes time and effort but also requires a large number of professionals. Combined with the growing needs of people in real life, this limits the development of machine learning applications in many fields. Therefore, in order to lower the bar for ma-chine learning and make it available to non-professionals to solve problems, automatic machine learning model selection is particularly important to let non-professionals select proper machine learning algorithms and the corresponding hyper-parameter values in the individual application.

Nowadays, Automated Machine Learning (Auto-ML) has emerged as a new promising sub-field of machine learning. Auto-ML, or a combination of automation and machine learning, is a new line of research that could allow computers to perform more complex tasks independently, freeing human hands. The target of Auto-ML is to realize the automation of machine learning step by step based on the principles of optimization and ma-chine learning itself [1]. It has been paid more and more attention not only in machine learning, but also in computer vision, data mining and natural language processing. Au-to-ML promises to make it easier for users to access machine learning for new domains and to reduce the tedious manual work of applying machine learning to existing domains. Auto-ML was originally introduced to automate model hyper-parameter searches. Nowadays, Auto-ML has been developed to include the automation of other tasks in the ma-chine learning workflows, such as model interpretability,

ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

feature engineering [2], data cleaning [3], and model deployment, as part of its goals. The ultimate goal of Auto-ML tools is to make machine learning easier to implement by providing ready-made solutions for users with lower technical backgrounds.

Auto-ML [4] has been successfully applied to many important problems, many automatic machine learning tools [5] were created, such as Auto-WEKA [6], Auto-Sklearn [7], Google's Cloud [8], NNI and Feature Labs [9], and so on. Among the existing Auto-ML tools, Auto-WEKA is one of the most advanced automatic machine learning methods and the first use of Bayesian optimization method. Auto-WEKA only needs to press the button, can automatically instantiated highly parameters of machine learning framework, and make full use of all the learning algorithm of the WEKA [10]. Through Auto-WEKA, non-experts can easily build high-quality classifiers that are suitable for the specific ap-plication scenario. Based on the currently popular Scikit-Learn machine learning library [11], Auto-Sklearn can automatically seek out a more appropriate classification model from some existing machine learning tools by searching the appropriate model and optimizing its corresponding super parameters. NNI (Neural Network Intelligence) is an open source automatic machine learning tool of Microsoft. NNI has a very unique value, compared with other current automatic machine learning tools or services. Unlike most existing automatic machine learning services and tools, NNI requires the user to provide the training code and specify the search scope of the hyper-parameter. Google's Cloud Auto-ML uses neural architecture search to free customers from the difficult and time-consuming architectural design process. Feature-Labs is a commercial product that implements automated feature engineering designed to build new and modified feature sets to enhance the performance of machine learning tools. All these powerful tools pro-vide great convenience for non-experts to use machine learning tools in their own re-searches and applications.

However, in the face of large hyper-parametric space, traditional optimization methods can no longer support the efficient selection of automatic machine learning models. Most existing Auto-ML tools are time-consuming, and the fundamental obstacle is that it takes a long time to test the machine learning algorithm and the combination of super-per-parameter values on the entire data set. In order to speed-up the optimization process, we putted forward a novel Progressive Sampling based Bayesian Optimization (PSBO) for automatic machine learning in the previous work [12]. Based on the framework of Bayesian optimization, PSBO uses progressive sampling and several optimization methods for automatic selection of machine learning models. The key idea of PSBO is to use the cheap information computed on small sample size to guide the search process to focus on promising areas when training sample expanded. Compared to the existing methods, PSBO significantly improves the computational efficiency without sacrifice the search effectiveness. Although the success of PSBO, one drawback of PSBO is the random sampling is used in the progressive sampling without considering the importance of samples. For complex hyper-parameter combination search space, it is difficult to guarantee the representativeness of samples drawn by random sampling method.

By consideration of the importance of samples, we proposed a Sample Importance Guided Progressive Sampling-based Bayesian Optimization method (SIG-PSBO) for automatic machine learning. In the process of progressive sampling, PCA is first used to reduce sample dimensionality. In the PCA reduced feature space, the sample importance and the corresponding sampling area are measured by the difficulty to distinguish categories. In the subsequent sampling, samples with higher sample importance are more sampled for model training. In other words, the sampling of SIG-PSBO is more focused on the distinguishing hyper-parametric search space. We show that our approach can significantly shorten search time and reduce classification error rates compared to the original PSBO method.

Our contributions are summarized as follows:

1) An effective Sample Importance Guided Progressive Sampling-based Bayesian Optimization method (SIG-PSBO) based on PCA dimension reduction is proposed, whose major sampling strategy is to focus on the distinguishing hyper-parametric search space guided by the sample importance.

2) Extensive experiments have been carried out on several datasets and the results show that the proposed SIG-PSBO model yields more satisfactory results than state of the art methods.

2. Related work

The main work of this paper is to improve the traditional sampling methods such as random sampling by using the importance progressive sampling method based on PCA to improve the efficiency of

ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

automatic machine learning model selection. Therefore, this chapter will briefly review the following two points:

2.1. Progressive Sampling for Automatic Machine Learning Model Selection

Sampling refers to the selection of data from all data for analysis in order to find useful information in a larger data set. During model training, in order to enable the model to better learn the characteristics of the data, we often perform data sampling to achieve better results. Progressive sampling [13] starts with a small sample, and as long as the accuracy of the model is sufficiently improved, small samples are gradually increased until a sufficient volume of samples is obtained. When determining a hyper-parameter combination for a machine learning algorithm, faced with a very large hyper-parameter space, it would be more appropriate to use Bayesian optimization based on progressive sampling, as shown in Figure 1, the validation sample remains unchanged in each round. The training sample is expanded in each round, and its size is doubled in each round, which will try to avoid searching in the hyper-parameter space region containing low-quality combinations.

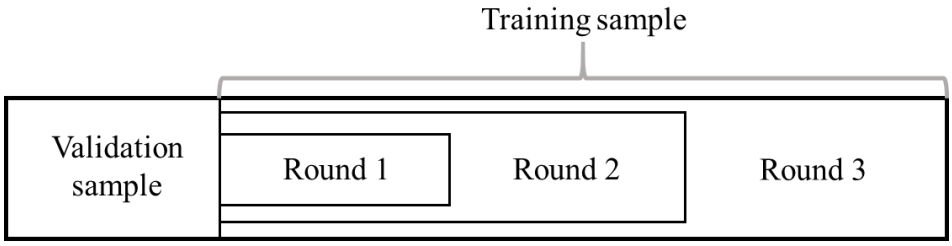

Figure 1: Example of progressive sampling used in automatic machine learning model selection method

2.2. Dimension Reduction Using PCA

Principal component analysis (PCA) is a common unsupervised learning method [14], which uses orthogonal transformation to convert a series of linearly related variables into a set of linearly unrelated new variables, which also become principal components, thus using the new variable shows the characteristics of the data in a smaller dimension. The principal component is a linear combination of the original variables, and its number is not more than the original variables. After the combination, we have obtained a batch of new observational data. The meaning of these data is different from the original data, but it contains most of the characteristics of the previous data, and has a lower dimension, which is convenient for further analysis.

In general, the easiest way to obtain a low-dimensional subspace [15] is to apply a linear transformation to the original high-dimensional space, given a sample set $X = \{x_1, x_2, \dots, x_m\} \in R^{d\times m}$in a d dimensional space, and then obtain a sample set Z in an n dimensional space, where $n \ll d$ The specific process is as follows: Firstly, sample feature demeanization (data centeralization) is carried out, that is, for each feature of the sample, the value of the current feature is subtracted from the mean value μ_j of the modified feature in the sample set, for the j-th feature of the i-th sample:

$$x_i^{(j)} = x_i^{(j)} - \mu_j \tag{1}$$

Where$\mu_j = \frac{1}{m}\sum_{k=1}^{m} x_k^{(j)}$, and the mean vector of all features is $\mu = (\mu_1, \mu_2, \dots, \mu_d), \mu \in R^d$.And then we calculate the covariance matrix for the sample:

$$\Sigma = XX^T \tag{2}$$

Then, the covariance matrix $\Sigma = XX^T$ is decomposed by eigenvalue decomposition, and the eigenvectors $\{u_1, u_2, \dots, u_l\}$ corresponding to the largest first l eigenvalues are taken to form the transformation matrix U, Finally, generate the dimensionless sample set:

$$\mathrm{Z} = \{\mathrm{z}_1, \mathrm{z}_2, \dots, \mathrm{z}_\mathrm{m}\}, \mathrm{z}_\mathrm{i} = \mathrm{U}^\mathrm{T}\mathrm{x}_\mathrm{i}, \mathrm{i} = 1,2, \dots, \mathrm{m} \tag{3}$$

The approximation of the original sample set:

$$\widehat{\mathrm{D}} = \left\{\mathrm{U}_{\mathrm{Z}_\mathrm{i}} + \mu\right\}, \mathrm{i} = 1,2, \dots, \mathrm{m} \tag{4}$$

Results output $\mathrm{Z} \in R^{l\times m}$, $\widehat{D} \in R^{d\times m}$. $\mathrm{Z} \in R^{l\times m}$ is the projection of the sample in the new space,

ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

which is the low-dimensional approximation after the reduction of the high dimensional X.

3. Method

In this section, we will introduce our proposed SIG-PSBO method in detail. Firstly, we will introduce the PCA based progressive importance sampling method in detail in Section 3.1, and then introduce other main technologies in Section 3.2. Finally, our complete SIG-PSBO method is described in Section 3.3.

3.1. PCA-Based Important Progressive Sampling

The importance sampling method based on PCA [16] is used in each progressive sampling. The key idea of using this method is to define a new basis for the hyper-parametric space *D*space$_i$ of each progressive sampling (i is the total space of the i-th progressive sampling), which can take more hyper-parametric combinations in the difficult area of space *D*space$_i$ than in the easy area with greater probability to generate a new sample set (considering the whole new basis). If other samples around a sample belong to the same category as the sample, this area is an easy area, On the contrary, if a sample is different from other surrounding samples, it is regarded as a difficult area, samples in difficult areas are important samples. The importance sampling procedure based on principal component analysis is a local method, which is applied to the region of interest in *D*space$_i$. A simple example is shown in figure 2, after PCA dimension reduction area $[-\lambda_1, \lambda_1]$ to the region of interest for us, for difficult areas, namely the diagram shows the two types of samples, has mixed in two classes of samples on the border, with a blend of two kinds of samples and cannot accurately identify the category, so many times in the area sampling, less time sampling in other areas.

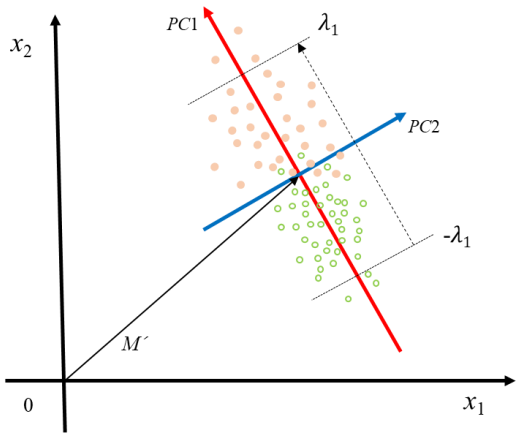

Figure 2: Example of PCA for dimensionality reduction

Firstly, given the initial set S of at least d conflict free hyper-parametric combinations (d is the dimension of spatial *D*space$_i$) and the region RS in *D*space$_i$ requiring importance sampling, PCA was used for dimension reduction calculation to obtain the sample volume V_S (V_S = PCA (S)) is calculated by principal component analysis. Then m_{R_S} combinations are taken from R_S and m_{V_S} combinations are taken from V_S. the hyper-parametric combinations obtained in V_S should also be in regional R_S. Finally, return to the new set S magnified to $m_{R_S} + m_{V_S}$ combinations. The reason to keep sampling in R_S is to obtain a conflict free configuration that is not in V_S, allowing the V_S to be recalculated to better cover the region of interest. Region V_S is more suitable for the difficult region of *D*space$_i$ than R_S. therefore, sampling in V_S can increase the probability of finding conflict free hyper-parameter combinations in the low gap difficult region of *D*space$_i$.

3.2. Other Technical Details

To better optimize our method and make it suitable for data sets of any size, we also use other technologies to improve the performance of the method.

We consider using the Hamming distance [17] between each hyper-parameter value combination. The default number of hyper-parametric value combinations we select for testing is $n_c = 10$, and the distance threshold defaults to $t_d = 2$. The Hamming distance between two hyper-parameter combinations is defined as the number of different hyper-parameter values in the combination. Machine

ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

learning algorithms usually have fewer hyper-parameters, and the number of hyper-parameters determines the maximum possible Hamming distance between two combinations. Therefore, if the Hamming distance between the two combinations is > 2, they can be considered far enough from each other. If the number of hyper-parameter combinations with error rate $< 100\%$ used in the last round of test $\leq n_c$, select all these combinations for test, otherwise select n_c of these combinations for test.

To improve the robustness of the generalization error rate estimated for the combination of machine learning algorithm and hyper-parametric values selected for testing. We use a technique similar to multiple cross validation [18] to use multiple folds on a small data set to increase the robustness of the obtained estimates. We use k ($k = 3$) times progressive sampling. In any contraction that is not the last round, we perform k folds. In the i-th ($1 \leq i \leq k$) collapse, the i-th part of m data instances forms the verification sample. The average of the error rates of all k estimated models is used as the estimated error rate of the wheel combination. In order to prevent over fitting, in the last round, we also added a step to perform k-fold cross validation to evaluate the most suitable machine learning algorithm and hyper-parametric value combination currently searched. In the implementation, we set $k = 10$ for a small dataset and $k = 3$ for a large dataset [19]. For each combination, cross validation will produce k error rate estimates. If there are extreme values in these estimation error rates, their average values will be greatly affected. Therefore, when comparing the two combinations, the paired method is used instead of the traditional average method to obtain more stable results. We compare the error rate estimates obtained by the two combinations in each fold cross validation, win the combination with more times, and finally become the best machine learning algorithm and hyper-parametric value combination.

3.3. The Complete Algorithm Processes

The automatic machine learning model selection method using PCA-based importance progressive sampling conducts in five rounds. The complete process is shown in Algorithm 1.

Algorithm 1: SIG-PSBO

1: Form the initial training and validation samples,

// the first round

2: For each machine learning algorithm, test its default value and 20 hyper-parameter value combinations,

3: Identify and delete some unsuitable algorithms,

// the middle rounds

4: For round two to four {

5: Increase the training samples,

6: For each remaining algorithm {

7: For the combinations of hyper-parameter value used in the previous round, the Hamming distance between the combinations is considered to update their error rate estimates,

8: Conduct Bayesian optimization, use the PCA-based importance progressive sampling method to select 10 combinations for testing and modify the regression model,

9: }

10: Identify and delete some unsuitable algorithms,

11: }

// the last round

12: Use the cross-validation method to select an algorithm and a final combination of hyper-parameter values, use this combination to build a model on the entire data set, and return this model,

In the first round, we test each applicable machine learning algorithm. For each algorithm and technique, test hyper-parametric values and the default combination (if any) of a predetermined number of hyper-parametric values (the default value is 20). First, eliminate the error rate $\geq \mu$ algorithm, where $\mu = 0.5$ is the error rate threshold. After that, if the number of remaining algorithms exceeds 40%, the combination with high error rate will continue to be eliminated until the number of remaining algorithms

ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

reaches 40%. The purpose of selecting $\mu = 0.5$ and 40% is to strike a balance between deleting enough algorithms in the first round to improve search efficiency and not deleting high-quality algorithms too early.

In each round that followed, not the last, we apply $\mu \times 0.8$ to reduce the error rate difference threshold of each round, expand the training samples, increase the feature selection time L_f and model training time L_t, test and adjust the promising combination on the extended training set to further narrow the search space. For each remaining machine learning algorithm in the previous round, we will complete it in three steps. The first step is to consider the Hamming distance between the hyper-parametric value combinations, select the hyper-parametric value combination used in the previous round of test, and obtain its error rate estimation in this round. The second step obtains the error rate estimation of the combination used in the previous round but not selected in the first round. The last step is to build a regression model using all combinations of the algorithms tested so far. Then C Bayesian Optimization cycles are performed. In the second round, we set $C = 3$ and reduce C by 1 in each subsequent round. In each cycle of Bayesian optimization, choosing 10 new combinations to test and used to modify the regression model. In order to better cover the new area of the hyper-parametric space and focus on the extraction of hyper-parametric combinations, we use the importance progressive sampling method based on PCA to extract samples. Finally, the first-round method is used to identify and remove invalid algorithm combinations, but at this time, the target percentage of the retention algorithm is increased from 40% to 70%.

In the last round, we use the cross-validation method to select the final combination, and use the final combination to build a model on the whole data set as the final model returned by our automatic selection method.

4. Results

We have conducted extensive experiments on various datasets and compared the results with those of PSBO [12], which shows the effectiveness of SIG-PSBO.

4.1. Experiment Description

In the experiment, we compare the automatic machine learning model selection using PCA based importance progressive sampling method with PSBO automatic selection method. Our purpose is to prove the effectiveness and feasibility of using PCA based importance progressive sampling method for automatic model selection. Considering all 39 classification algorithms in the standard Weka package [20], using 10 well-known benchmark datasets. Divide each data set into a training set and a test set. The training data is used in the search process, and the training data is used when evaluating the error rate of the automatic selection method returning to the final model. The other set-tings in the experiment are basically the same as those in Auto-WEKA [9].

4.2. Overall Results of the Search Process

For each data set, we run the PSBO and SIG-PSBO method for 5 times. Take the mean value and standard deviation of the results of 5 times, as shown in Table 1, and display as mean value $\pm$ standard deviation. Three indicators are given in the table, which are the time spent on the search process, the number of tested machine learning algorithms and combinations of different hyper-parametric values, and the error rate of the final model on the test data.

From the experimental results of Table 1, the search process of SIG-PSBO method takes much less time than that of PSBO. The reduced search time is 0.2h on averaged, which greatly improves the search efficiency.

On account of the importance progressive sampling method based on PCA, both the PSBO and SIG-PSBO methods can test different combinations of machine learning algorithms and hyper-parametric values faster and more than the Auto-WEKA method, especially on large datasets. Compare to the PSBO method, SIG-PSBO test less than that required to fully explore the search space without sacrificing the search performance.

On almost all data sets, our sampling Bayesian optimization method based on sample importance guidance achieves lower error rate than the PSBO method. There are only two exceptions. On the Wine Quality datasets, SIG-PSBO is 0.88% worsen than PSBO. On the Waveform dataset, SIG-PSBO is 0.91%

ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

worsen than PSBO.

Table 1: Overall results of the search process.

Data set	Time spent (hours)		# of distinct combinations tested		Error rate on the test data (%)	
	PSBO	SIG-PSBO	PSBO	SIG-PSBO	PSBO	SIG-PSBO
Yeast	1.7±0.4	1.5±0.3	1,602±23	1,400±29	38.88±1.20	36.20±1.10
German Credit	1.0±0.3	0.8±0.2	1,602±29	1,420±25	27.00±1.11	25.00±1.05
Wine Quality	4.2±0.8	4.0±0.8	1,633±33	1,480±35	34.12±0.70	35.00±0.70
Waveform	3.5±0.6	3.6±0.4	1,596±21	1,600±33	14.29±0.12	15.20±0.11
Semeion	4.7±0.7	5.0±0.8	1,594±10	1,480±15	5.03±0.21	4.88±0.12
Secom	1.1±0.1	1.0±0.1	1,699±47	1,500±30	7.83±0.09	6.20±0.09
Madelon	2.6±0.3	2.2±0.3	1,763±49	1,660±40	18.31±1.47	16.02±1.20
Convex	5.5±0.4	5.1±0.4	1,766±73	1,635±55	23.37±0.87	22.60±0.87
Dexter	4.4±0.6	4.3±0.5	1,569±28	1,556±28	5.11±1.20	4.90±0.90
Dorothea	7.3±0.9	6.6±0.7	1,625±25	1,680±25	5.74±0.48	5.50±0.52
Averaged	3.6±0.5	3.4±0.5	1,645±34	1,541±32	17.97±0.75	17.15±0.67

5. Conclusions

For the task of automatic machine learning model selection, this paper designed an effective Sample Importance Guided Progressive Sampling-based Bayesian Optimization method (SIG-PSBO) based on PCA dimension reduction. Beyond the classical progressive sampling, SIG-PSBO uses the sample importance defined on the PCA reduced feature space to guide the sampling process. As a result, the sampling of SIG-PSBO is more focused on the distinguishing hyper-parametric search space. Extensive experiments on several benchmark datasets show that compared with the current existing automatic selection methods, this method greatly shortens the search time and improves the accuracy of the search results. In the future, we will continue to study other theories and technologies to further improve search efficiency without reducing the quality of search results.

Acknowledgements

This research was funded by Natural Science Foundation of China, grant numbers 61866017.

References

[1] Su, M., Liang, B., Ma, S., Xiang, C., Zhang, C., Wang, J. (2021) Automatic Machine Learning Method for Hyper-Parameter Search. Journal of Physics: Conference Series 2021, 1802, 032082 (8pp), doi:10.1088/1742-6596/1802/3/032082.

[2] Kanjilal, R., Uysal, I. (2021) The Future of Human Activity Recognition: Deep Learning or Feature Engineering? Neural Process. Lett. 53, 561–579, doi: 10.1007/s11063-020-10400-x.

[3] Gemp, I., Theocharous, G., Ghavamzadeh, M. (2017) Automated Data Cleansing through Meta-Learning. In Proceedings of the Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., pp. 4760–4761.

[4] He, X., Zhao, K., Chu, X. AutoML: (2021) A Survey of the State-of-the-Art. Knowl. Based Syst. 212, 106622, doi:10.1016/j.knosys.2020.106622.

[5] Bezrukavnikov, O., Linder, R. (2021) A Neophyte With AutoML: Evaluating the Promises of Automatic Machine Learning Tools. CoRR 2021, abs/2101.05840.

[6] Kotthoff, L., Thornton, C., Hoos, H.H., Hutter, F., Leyton-Brown, K. (2019) Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA. In Automated Machine Learning - Methods, Systems, Challenges, pp. 81–95.

[7] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J.T., Blum, M., Hutter, F. (2019) Auto-sklearn: Efficient and Robust Automated Machine Learning. In Automated Machine Learning - Methods, Systems, Challenges, pp. 113–134.

ISSN 2616-5775 Vol. 5, Issue 1: 32-39, DOI: 10.25236/AJCIS.2022.050107

[8] Zoph, B., Le, Q.V. Neural Architecture Search with Reinforcement Learning. (2017) In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.

[9] Kanter, J.M., Veeramachaneni, K. (2015) Deep Feature Synthesis: Towards Automating Data Science Endeavors. In Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, October 19-21, 2015, pp. 1–10.

[10] Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K. (2013) Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. In Proceedings of the The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013, pp. 847–855.

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2012) Scikit-Learn: Machine Learning in Python. CoRR 2012, abs/1201.0490.

[12] Zeng, X., Luo, G. (2017) Progressive Sampling-Based Bayesian Optimization for Efficient and Automatic Machine Learning Model Selection. Health Inf. Sci. Syst. 5, 2, doi: 10.1007/s13755-017-0023-z.

[13] Ros, F., Guillaume, S. (2021) A Progressive Sampling Framework for Clustering. Neurocomputing 2021, 450, 48–60, doi:10.1016/j.neucom.

[14] Zhang, T., Yang, B. Big Data Dimension Reduction Using PCA. (2016) In Proceedings of the 2016 IEEE International Conference on Smart Cloud, SmartCloud 2016, New York, NY, USA, November 18-20, 2016, pp. 152–157.

[15] Wang, S.-H., Huang, S.-Y., Chen, T.-L. (2020) On Asymptotic Normality of Cross Data Matrix-Based PCA in High Dimension Low Sample Size. J. Multivar. Anal. 175, doi:10.1016/j.jmva.2019.104556.

[16] Rosell, J., Suárez, R., Pérez, (2013) A. Path Planning for Grasping Operations Using an Adaptive PCA-Based Sampling Method. Auton. Robots 35, 27–36, doi: 10.1007/s10514-013-9332-5.

[17] Grabowski, S., Kowalski, T.M. (2021) Algorithms for All-Pairs Hamming Distance Based Similarity. Softw. Pract. Exp. 51, 1580–1590, doi:10.1002/spe.2978.

[18] Rao, R.B., Fung, G. (2008) On the Dangers of Cross-Validation. An Experimental Evaluation. In Proceedings of the Proceedings of the SIAM International Conference on Data Mining, SDM 2008, April 24-26, 2008, Atlanta, Georgia, USA, pp. 588–596.

[19] Marcot, B.G., Hanea, A.M. (2021) What Is an Optimal Value of k in K-Fold Cross-Validation in Discrete Bayesian Network Analysis? Comput. Stat. 36, 2009–2031, doi: 10.1007/s00180-020-00999-9.

[20] Witten, I.H., Frank, E., Hall, M.A. (2011) Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition., Morgan Kaufmann, Elsevier, p. 629, ISBN 9780123748560.