Technoeconomics. 2023, Vol. 2, No. 4 (7). Pp. 4–15.
Техноэкономика. 2023, Том 2, № 4 (7). С. 4–15.

Scientific article
UDC 330.47
DOI: https://doi.org/10.57809/2023.2.4.7.1

STUDY OF THE CONCEPT OF BAYESIAN OPTIMIZATION AND PRACTICAL USE OF ITS ALGORITHMS IN THE PYTHON PROGRAMMING LANGUAGE

Anton Chernyagin ✉, **Sergey Svetunkov**

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ a.s.chernyagin@gmail.com

Abstract. The aim of the study is to explore the principles of Bayesian optimization and its potential for solving complex problems, including economic ones. This article presents the main aspects of Bayesian optimization such as selection of a priori distribution, estimation of posterior distribution and selection of optimal model parameters. An example of applying Bayesian optimization to find hyperparameters using the Python programming language is presented. Bayesian optimization algorithms and their application to improve machine learning models were studied. The use of Bayesian optimization algorithm for finding hyperparameters can be useful in the future for optimizing various machine learning models such as neural networks, SVM and others.

Keywords: Bayesian optimization, hyperparameters, Python, machine learning

Citation: Chernyagin A., Svetunkov S. Study of the concept of Bayesian optimization and practical use of its algorithms in the Python programming language. Technoeconomics. 2023. 2. 4 (7). 4–15. DOI: https://doi.org/10.57809/2023.2.4.7.1

Научная статья
УДК 330.47
DOI: https://doi.org/10.57809/2023.2.4.7.1

ИССЛЕДОВАНИЕ ПОНЯТИЯ “БАЙЕСОВСКАЯ ОПТИМИЗАЦИЯ” И ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ ЕЕ АЛГОРИТМОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Антон Чернягин ✉, Сергей Светуньков

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия

✉ a.s.chernyagin@gmail.com

Аннотация. Цель исследования заключается в изучении принципов байесовского подхода в различных областях, а также подробное изучение байесовской оптимизации и ее возможностей для решения сложных задач, в том числе, экономических. В данной работе представлены основные аспекты байесовской оптимизации, такие как выбор априорного распределения, оценка апостериорного распределения и выбор оптимальных параметров модели. Приведен пример применения байесовской оптимизации для нахождения гиперпараметров с помощью языка программирования Python. Были изучены алгоритмы байесовской оптимизации и их применение для улучшения моделей машинного обучения. Использование алгоритма байесовской оптимизации для нахождения гиперпараметров может быть полезным в будущем для оптимизации различных моделей машинного обучения, таких как нейронные сети, SVM и другие.

Ключевые слова: Байесовские методы, Байесовская оптимизация, гиперпараметры, Python, машинное обучение

Для цитирования: Чернягин А., Светуньков С. Исследование понятия «Байесовская оптимизация» и практическое использование ее алгоритмов на языке программирования Python // Техноэкономика. 2023. Т. 2, № 4 (7). С. 4–15. DOI: https://doi.org/10.57809/2023.2.4.7.1

Introduction

Bayesian optimization is a current and important area in the field of machine learning and statistics. It is based on the application of Bayes formula to determine the optimal model parameters given a priori knowledge and observed data. This scientific article studies the principles of Bayesian optimization, its application in various machine learning problems and the potential for improving existing methods.

The Bayesian approach allows us to account for uncertainty and a priori knowledge about model parameters, which makes it particularly useful for analyzing small data and solving complex problems. At the same time, Bayesian optimization has a theoretical underpinning and allows us to give the model parameters a meaningful interpretation. The study of Bayesian optimization is necessary and meaningful because this method can solve complex function optimization problems, considering the noise in the data and the cost of estimating the function. The study of Bayesian optimization can lead to the development of new methods and algorithms that can be applied to solve practical problems in various fields. In addition, Bayesian optimization can be used to select reasonable information that determines the whole modeling process in econometrics. Thus, the study of Bayesian optimization has great significance for various fields of science and practice.

Bayesian methods of decision making in economics consider the application of Bayesian optimization for decision making in the activities of individual economic entities. Any organization operates in the economy under conditions of uncertainty, which requires an increase in the accuracy of estimates when making economic decisions, since the financial results of the organization depend on it. Application of Bayesian optimization allows to increase the probability of making rational economic decisions.

Materials and Methods

Bayesian methods are statistical methods based on Bayes' theorem, which allows updating probabilistic estimates of events based on new data. These methods have been widely used in various fields including economics, medicine, genetics, speech recognition, space exploration, insurance, and others. They can be useful for parameter estimation, data prediction, model comparison, decision making under uncertainty, and many other tasks.

Bayesian theory and methods are named after Thomas Bayes (1702-1761), an English mathematician and clergyman who was the first to propose the use of Bayes' theorem to adjust beliefs based on updated data. His work An Essay towards solving a Problem in the Doctrine of Chances was published in 1763, two years after the author's death. However, methods using Bayes' theorem became widespread only towards the end of the 20th century, when computationally intensive calculations became possible with the development of information technology.

The principle of Bayesian methods is to use a priori knowledge of the model parameters to obtain posterior distributions of the parameters after taking into account new data. This allows uncertainty and prior experience to be taken into account when making decisions. Bayesian methods also allow models to be updated based on new data, making them flexible and adaptive.

Bayesian methods have found applications in medical diagnosis, image modeling, genetics, speech recognition, economics, space exploration, insurance, and other fields. They are used to estimate parameters, predict data, compare models, make decisions under uncertainty, and establish causal relationships.

Bayesian optimization in economics can be applied in different contexts to solve a variety of problems:

– Portfolio optimization:

Bayesian optimization can be used to find the optimal allocation of assets in a portfolio in order to maximize returns or minimize risk. The model can take into account various factors such as expected returns, volatility, and correlations between different assets (Laumanns, 2002).

– Pricing:

In business, Bayesian optimization can help in determining the optimal price of a product or service. The model can take into account data on market trends, consumer preferences, and competitive factors.

– Marketing campaigns:

When planning marketing campaigns, Bayesian optimization can be used to determine the optimal budget, timing, and distribution channels for advertising. This can help maximize the expected effect of an advertising campaign.

– Inventory management:

For businesses that sell goods, Bayesian optimization can be useful in inventory management. The model can help optimize inventory levels with respect to demand, delivery time and storage costs.

– Business Process Optimization:

Bayesian optimization can be used to optimize business processes such as manufacturing, logistics or human resource management. The model can suggest optimal parameters to improve efficiency and reduce costs.

– Financial planning:

In financial planning, Bayesian optimization can help determine the optimal budget allocation between different projects or business lines.

– Risk Analysis:

A Bayesian optimization model can be used to analyze risks and select optimal risk management strategies under uncertainty.

However, it is important to note that successful application of Bayesian optimization in economics requires a good understanding of the context of the problem, proper choice of model parameters and careful interpretation of the results. Bayesian methods are a powerful tool to incorporate uncertainty and prior experience into decision making, making them an important tool in various fields of knowledge, including economics.

Bayesian optimization is a method that combines probabilistic models with optimization techniques to efficiently find optimal hyperparameters. Hyperparameters are parameters that are used to control the learning process, as opposed to model parameters that are tuned during training. Bayesian optimization allows us to select the next point to be estimated using the model of the model performance evaluation function. Bayesian optimization can be used to optimize hyperparameters in a variety of domains including machine learning, deep learning, natural language processing, and others. It can be particularly useful in tasks where model performance function estimation is expensive, such as in tasks with large amounts of data or complex models (Fujimoto, 2023).

The advantages of Bayesian optimization include efficiency and the ability to work with a black box, that is, a function that has no analytical expression. It can also be used to optimize multiple hyperparameters simultaneously. Disadvantages include the need to select an appropriate model of the model performance evaluation function and computational complexity (Downey, 2018).

Examples of problems in which Bayesian optimization can be useful include optimization of hyperparameters of neural networks, selection of optimal parameters for machine learning algorithms, optimization of parameters in optimization problems, and others. Bayesian optimization is a powerful tool for optimizing hyperparameters in various domains. It can efficiently find the optimal values of hyperparameters using probabilistic models and optimization methods. However, the selection of a suitable model of the model performance evaluation function and computational complexity can be problems that need to be considered when using this method (Smirnova, 2022).

Bayesian optimization uses Gaussian processes to model the unknown function to be optimized. A Gaussian process is a probability distribution over functions that is updated based on new data. Bayesian optimization adaptively selects the next point to evaluate the function, which reduces the number of evaluation operations. Bayes formula is the basis of Bayesian statistics and is used to calculate the posterior probability based on a priori probability and new data. The Bayes formula is as follows:

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)} \tag{1}$$

where P(A|B) is the posterior probability, P(B|A) is the likelihood (probability of occurrence of event B given event A), P(A) is the a priori probability (probability of occurrence of event A), and P(B) is the full probability (probability of occurrence of event B).

In Bayesian optimization, Bayes formula is used to calculate the posterior probability of the distribution of functions based on observed data. The posterior distribution of functions is the basis for selecting the next point for function evaluation. The choice of the next point is based on which point maximizes the expected improvement of the function. The expected improvement of the function is calculated based on the posterior distribution of functions and the a priori distribution of hyperparameters (Cuesta Ramirez, 2022).

A Gaussian process is a stochastic process such that every finite collection of random variables has a multivariate normal distribution, that is, every finite linear combination of these random variables has a normal distribution.

Gaussian processes can be used to model an unknown function and can be used in Bayesian optimization to build a model of the model performance evaluation function of the model. The basic properties of Gaussian processes can be defined through the covariance function. Some of these properties include stationarity, isotropy, smoothness and periodicity of the process. If the process is stationary, then the covariance function depends only on the difference between two points (Galuzzi, 2020).

Advantages of Gaussian processes include the ability to work with a black box, i.e., a function that has no analytical expression, and the ability to be used in Bayesian optimization. Disadvantages include the need to select an appropriate model of the function to evaluate model performance and computational complexity. Examples of tasks in which Gaussian processes may be useful include network traffic modeling, statistical modeling, parameter optimization in optimization problems, and others (Pico-Valencia, 2021).

Gaussian processes are a powerful tool for modeling unknown function and optimization in various domains. Gaussian process and Bayesian optimization are closely related. Bayesian optimization uses Gaussian processes to model the unknown function to be optimized. A Gaussian process is used to model the unknown function to be optimized and is a probabilistic model that describes the distribution of function values at different points. Bayesian optimization uses Gaussian processes to estimate the unknown function and select the next point to be estimated. The Gaussian process allows for uncertainty in the data and adaptively selects the next point to estimate the function, thus reducing the number of estimation operations.

The Bayesian optimization algorithm consists of two main parts:

1. Probabilistic function model: Bayesian optimization starts with an a priori distribution over the function to be optimized, which reflects the uncertainty about the function under study. With each new observation of the function, the a priori distribution is updated and a posterior distribution over the possible functions is obtained.

2. Selecting the next point to estimate: based on the posterior distribution, the next point to estimate the function is selected.

To use Bayesian optimization, the following steps should be followed:

1. Define the function to be optimized;
2. Select an a priori distribution for the function;
3. Evaluate the function at the initial points;
4. Update the a priori distribution with new observations and obtain the posterior distribution;
5. Select the next point to estimate based on the posterior distribution;
6. Repeat steps 4-5 until a stopping criterion (e.g., a given number of iterations or convergence) is reached.

Thus the Bayesian optimization algorithm solves the following problems (Feliot, 2017):

1. Optimization of complex functions: Bayesian optimization allows to find the maximum of functions with unknown structure, for example, when selecting hyperparameters for machine learning models;

2. Accounting for the cost of function estimation: in some cases, function estimation can be expensive (e.g., training a neural network). Bayesian optimization adaptively selects the next

point to be estimated given the information from previous iterations, thus reducing the number of estimation operations;

3. Noise control: the function may return different values for the same set of parameters due to noise in the data. Bayesian optimization accounts for this noise and allows finding optimal parameters given this uncertainty.

4. Balance between exploration and exploitation: the Bayesian optimization algorithm takes into account both the already known values of the function and the uncertainty in the unexplored regions of the parameter space, which allows more efficient exploration of the parameter space and finding optimal values (Garrido-Merchбn, 2020).

Bayesian optimization can be relevant for the following economic problems:

1. Decision Making by Individual Economic Entities: Bayesian Methods of Decision Making in Economics examines the application of Bayesian methods of decision making to the activities of individual economic actors

2. Monetary policy analysis: Bayesian vector autoregression model can be used to estimate the impact of various factors on the economy such as monetary policy, external shocks and other variables, which allows us to obtain robust estimates for models with a large number of variables on samples of limited size (Sheikh, 2022).

3. Estimating the impact of factors on the economy: a Bayesian approach can be used to estimate the interdependence of household income inequality and economic growth rates.

Thus, Bayesian optimization can be useful for decision making by individual economic actors, analyzing monetary policy and assessing the impact of various factors on the economy (Pozhidaeva, 2023).

What is more, Bayesian methods can help in decision-making when there is uncertainty in the data or when it is necessary to take into account previous knowledge and experience. For example, Bayesian methods can be used to predict future trends in the market, to determine the optimal price of a product or service, and to assess risk and make decisions in investment activities.

Results and Discussion

Bayesian optimization in the Python programming language in the Microsoft Visual Studio Code environment.

One type of problem often faced by scientists in both academia and industry is the optimization of black-box functions that are expensive to evaluate.

Black box functionality is a term used to refer to a system whose internal structure and mechanism of operation are very complex, unknown or unimportant within the scope of a given task. In the context of software testing, black box means that the tester does not necessarily know the internal structure of the program or system, but tests its functionality based on input and output data. This testing method is used to verify that the software performs all stated functions and customer requirements in full according to the documentation (Bishop, 2006).

In cybernetics, systems engineering, and physics, a black box is a system that is viewed as having some "input" for inputting information and an "output" for displaying the results of operation, with the processes occurring during the operation of the system unknown to the observer. The black box approach developed in the exact sciences in the 1920s and 1940s and was borrowed by other sciences, including behavioral psychology.

Thus, a black box is a system whose internal design and operating mechanism are complex, unknown, or unimportant to the task at hand, and is used in a variety of fields including software testing, aviation, and automatic control theory.

This simplifies the programming and data processing process because the programmer can use a black-box function without having to know all the details of its implementation. Instead, he can focus on what input data he needs to provide and what results he expects to get (Sharma, 2021).

However, using a black box function can have disadvantages. For example, if the function does not work correctly, it may be difficult for the programmer to determine the cause of the error or to correct it. Also, if the function is not well documented, it may be difficult to understand exactly what input data and what output data it expects.

The concept of a black box expensive to evaluate means that it costs a lot of money or resources to perform a function or operation and its inner workings cannot be understood. A good example of a black box function expensive to evaluate is the hyperparameter optimization of a deep neural network. Each iteration of training can take up to several days, and it is impossible to analyze in advance the values of hyperparameters that will lead to the best performance of the model (Subasi, 2020).

It is possible to perform a cross-grid search of all possible hyperparameter values, but with so many training iterations to be repeated, this would result in a huge computational cost. A more efficient method is needed to find the best set of hyperparameters using the least number of iterations. Bayesian optimization can be used for this task (Gaudrie, 2020).

Bayesian optimization for the black box function in this case has 2 components (Pandita, 2020):

1. The black box function to be optimized is: f(x). We need to find a value of 'x' that globally optimizes f(x). This is a probabilistic model of the function, it is also sometimes called objective function, objective function or loss function. In the general case, we only have knowledge of the inputs and outputs of f(x) (Morice-Atkinson, 2018).

2. An acquisition function: a(x), which is used to generate new values of x to be evaluated with f(x). a(x) internally relies on a Gaussian process model N(X, y) to generate new values of x.

The optimization process itself is as follows:

1. Definition of black box function f(x), acquisition function a(x) and search space of parameter 'x'.

2. Generating several initial values of 'x' randomly and measuring the corresponding results of f(x).

3. Setting up a Gaussian process model N(X, y) on X = x and y = f(x).

4. The acquisition function a(x) then uses N(X, y) to generate new values of 'x' as follows. The model N(X, y) predicts changes in f(x) as a function of 'x'. The value of 'x' that results in the largest predicted value in N(X, y) is then offered as the next sample 'x' to estimate f(x).

5. Steps 3 and 4 should be repeated until the value of 'x' that leads to the global optimum of f(x) is obtained. At the same time, all historical values of 'x' and f(x) should be used to train the Gaussian process model N(X, y) in the next iteration - as the number of data points increases, N(X, y) becomes better at predicting the optimum of f(x).

This experiment uses the bayes_opt library to find the hyperparameter C of the SVC model trained on sklearn breast cancer data.

Support Vector Machine (SVM) is a powerful machine learning algorithm used for classification and regression tasks. Within the classification task, SVM is based on the idea of finding an optimal separating hyperplane in the feature space that maximizes the gap between classes of data.

Support Vector Classifier (SVC) model is one of the variations of SVM for classification task. It works by finding an optimal hyperplane that separates the training data into two classes. The optimal hyperplane is chosen to maximize the distance (gap) between the closest points of each class, which are called support vectors (Lyu, 2018).

The hyperplane in SVC is defined by a set of weights (weights) and bias (bias), and training the model consists of tuning these parameters based on the training sample. However, it is important to note that in the case of nonlinear data, SVC can use so-called kernel functions, which allow the model to build nonlinear separating hyperplanes in higher dimensional space. The SVC model can also be used to solve the one-vs-all multiclass classification problem. It shows good performance for medium to large training sample sizes, although careful tuning of hyperparameters may be required to achieve optimal results. The overall performance of the SVC method and SVM in general makes it a popular choice for classification tasks in various fields such as computer vision, bioinformatics, financial analytics, and others (Popovic, 2019).

The hyperparameter C in the Support Vector Classification (SVC) model is a regularization parameter that controls the balance between maximizing the width of the separating band and minimizing classification errors. The value of C determines how much we want our model to be adapted to the noise in the data. If the value of C is very small, then the model will be more flexible and will have a larger error on the training data but a smaller error on the test data. If the value of C is very large, the model will be less flexible and will have smaller error on the training data but larger error on the test data. The value of C should be chosen optimally for the particular classification task (Nguyen, 2018).

The components of the optimizer are:

1. The black box function f(x) is the ROC AUC score that we want to maximize to get the best performing model.

2. The acquisition function a(x) is used as the upper confidence bound ("ucb") function, which is of the form: a = mean + kappa * std. Both mean and std are outputs from a Gaussian process model N(X, y). kappa is a hyperparameter of the optimizer that balances exploration and exploitation when searching for x.

The out-of-the-box Python code to perform the above optimization steps is as follows.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.datasets import load_breast_cancer
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score
from bayes_opt import BayesianOptimization, UtilityFunction
import warnings
warnings.filterwarnings("ignore")

# Prepare the data (dataset download, identify X and Y).
cancer = load_breast_cancer()
X = cancer.data
y = cancer.target
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = y, random_state = 42)
# Normalizing data.
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Define the black box function to optimize.
def black_box_function(C):    # C - SVC hyperparameter for optimization.
    model = SVC(C = C)
    model.fit(X_train_scaled, y_train)
    y_score = model.decision_function(X_test_scaled)
    f = roc_auc_score(y_test, y_score)
    return f
# Set range of C to optimize for.
pbounds = {"C": [0.1, 10]}
# Create a BayesianOptimization optimizer for black box .
optimizer = BayesianOptimization(f = black_box_function, pbounds = pbounds, verbose = 2, random_state = 42)
optimizer.maximize(init_points = 5, n_iter = 10)
print("Best result: {}; f(x) = {}.".format(optimizer.max["params"], optimizer.max["target"]))
```

Fig. 1. Bayesian optimization of the hyperparameter

Running the above Python code produces the following output.

```
|   iter    |  target   |     C     |
-------------------------------------
| 1         | 0.9975    | 3.808     |
| 2         | 0.9979    | 9.512     |
| 3         | 0.9979    | 7.347     |
| 4         | 0.9975    | 6.027     |
| 5         | 0.9966    | 1.645     |
| 6         | 0.9981    | 8.433     |
| 7         | 0.9981    | 8.041     |
| 8         | 0.9981    | 8.237     |
| 9         | 0.9981    | 8.868     |
| 10        | 0.9981    | 8.69      |
| 11        | 0.9981    | 7.901     |
| 12        | 0.9914    | 0.1       |
| 13        | 0.9975    | 2.727     |
| 14        | 0.9975    | 4.913     |
| 15        | 0.9981    | 10.0      |
=====================================
Best result: {'C': 8.432539826625984}; f(x) = 0.9981132075471698.
```

Fig. 2. Result of running the Python code

From the above acquired results, the optimizer was able to determine that using a hyperparameter value of C = 8.432 results in the best model performance.

Conclusion

Thus, how Bayesian optimization works was studied and how it was used to find hyperparameters of a machine learning model. For small datasets or simple models, the speedup in finding hyperparameters may be negligible compared to grid search. However, for very large datasets or deep neural networks, it may not be economically feasible to test every sample on a grid, and the use of Bayesian optimization will improve the efficiency of the hyperparameter search process.

Using Bayesian optimization code to find hyperparameters of black-box functions, we can apply the knowledge gained in the following directions:

1. Machine learning: optimizing the hyperparameters of machine learning models such as neural networks, decision trees and support vector method to improve their performance and accuracy.

2. Financial research: applying Bayesian optimization to tune the parameters of econometric models used in financial analysis and forecasting.

3. Industrial optimization: using Bayesian optimization to determine the optimal parameters of processes and systems in various industries.

4. Tuning classification algorithms: determining the significance of hyperparameters and tuning classifiers with an extensive set of hyperparameters for better validation of results.

5. Comparison and analysis of results: evaluating statistical results using criteria such as the Mann-Whitney criterion to compare the performance of classical and extended Bayesian optimization.

Bayesian optimization of regression model hyperparameters used in programming can be useful for economic purposes in several aspects:

1. Predicting future trends in the market: Bayesian optimization can help in predicting future trends in the market, which can be useful for production, investment, and strategic planning decisions.

2. Determining the optimal price of a product or service: By optimizing the hyperparameters of a regression model, more accurate forecasts of demand and prices can be obtained, which can help in making pricing decisions.

3. Risk assessment and decision making in investment activities: Bayesian optimization can help in risk assessment and decision making in investment activities by allowing uncertainty and previous experience to be taken into account in decision making.

Thus, Bayesian optimization of regression model hyperparameters applied in programming can be useful for decision making under uncertainty in economic problems such as forecasting, pricing, and investment activities.

REFERENCES

Bishop C.M. 2006. Pattern Recognition and Machine Learning. Springer, 738.

Cuesta Ramirez J. 2022. A comparison of mixed-variables Bayesian optimization approaches. Advanced Modeling and Simulation in Engineering Sciences, 9 (1), 1-29. doi:10.1186/s40323-022-00218-8

Downey A.B. 2018. Bayesian models, 182.

Feliot P.A. 2017. Bayesian approach to constrained single- and multi-objective optimization. Journal of Global Optimization, 67 (1), 97-133. doi:10.1007/s10898-016-0427-3.

Fujimoto Yu. 2023. Controller tuning with Bayesian optimization and its acceleration: Concept and experimental validation. Asian Journal of Control, 25 (3), 2408-2414. doi:10.1002/asjc.2847

Galuzzi B. 2020. Hyperparameter optimization for recommender systems through Bayesian optimization. Computational Management Science. doi:10.1007/s10287-020-00376-3

Garrido-Merchan E.C. 2020. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. Neurocomputing, 380, 20-35. doi:10.1016/j.neucom.2019.11.004

Gaudrie D. 2020. Targeting solutions in Bayesian multi-objective optimization: sequential and batch versions. Annals of Mathematics and Artificial Intelligence, 88 (1), 187-212. doi:10.1007/s10472-019-09644-8

Laumanns M. 2002. Bayesian Optimization Algorithms for Multi-objective Optimization. Lecture Notes in Computer Science, 2439, 298-307.

Lyu W., P. Xue. 2018. An efficient Bayesian optimization approach for automated optimization of analog circuits. IEEE Transactions on Circuits and Systems. Part 1: Regular Papers, 65 (6), 1954-1967. doi:10.1109/TCSI.2017.2768826

Morice-Atkinson X. 2018. Learning from the machine: Interpreting machine learning algorithms for point- and extended-source classification. Monthly Notices of the Royal Astronomical Society, 481 (3), 4194-4205. doi:10.1093/mnras/sty2575

Nguyen T.D. 2018. Stable Bayesian optimization. International Journal of Data Science and Analytics, 6 (4), 327-339. doi:10.1007/s41060-018-0119-9

Pandita P. 2021. Computationally efficient bayesian optimization for multi-objective industrial applications. AIAA Scitech 2021 Forum, 1, 1-13. doi:10.2514/6.2021-1482

Pico-Valencia P. 2021. Bringing Machine Learning Predictive Models Based on Machine Learning Closer to Non-technical Users. Advances in Intelligent Systems and Computing, 1273, 3-15. doi:10.1007/978-3-030-59194-6_1

Popovic M. 2019. Formal verification of local and distributed python software transactional memories. Revue Roumaine des Sciences Techniques. Series Electrotechnique et Energetique, 64 (4), 423-428.

Pozhidaeva. N.A. 2023. Application of Bayesian process definition strategy for priority optimization (on the example of the banking sector). Economics: yesterday, today, tomorrow, 13 (3), 530-535. doi:10.34670/AR.2023.83.23.085

Sharma P. 2021. Role of machine learning and deep learning in securing 5G-driven industrial IoT applications. Ad Hoc Networks, 123, 102685. doi:10.1016/j.adhoc.2021.102685

Sheikh H.M. 2022. Bayesian optimization for mixed-variable, multi-objective problems. Structural and Multidisciplinary Optimization, 65 (11), 1-14. doi:10.1007/s00158-022-03382-y

Smirnova V.S., Shalamov V.V., Efimova V.A., Filchenkov A.A. 2020. Hyperparameter optimization based on combining a priori and a posteriori knowledge of the classification task. Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics, 20 (6), 828-834.

Subasi A. 2020. Practical Machine Learning for Data Analysis Using Python. Practical Machine Learning for Data Analysis Using Python, 1-520. doi: 10.1016/B978-0-12-821379-7.00008-4

СПИСОК ИСТОЧНИКОВ

Bishop C.M. 2006. Pattern Recognition and Machine Learning. Springer, 738.

Cuesta Ramirez J. 2022. A comparison of mixed-variables Bayesian optimization approaches. Advanced Modeling and Simulation in Engineering Sciences, 9 (1), 1-29. doi:10.1186/s40323-022-00218-8

Downey A.B. 2018. Bayesian models, 182.

Feliot P.A. 2017. Bayesian approach to constrained single- and multi-objective optimization. Journal of Global Optimization, 67 (1), 97-133. doi:10.1007/s10898-016-0427-3.

Fujimoto Yu. 2023. Controller tuning with Bayesian optimization and its acceleration: Concept and experimental validation. Asian Journal of Control, 25 (3), 2408-2414. doi:10.1002/asjc.2847

Galuzzi B. 2020. Hyperparameter optimization for recommender systems through Bayesian optimization. Computational Management Science. doi:10.1007/s10287-020-00376-3

Garrido-Merchan E.C. 2020. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. Neurocomputing, 380, 20-35. doi:10.1016/j.neucom.2019.11.004

Gaudrie D. 2020. Targeting solutions in Bayesian multi-objective optimization: sequential and batch versions. Annals of Mathematics and Artificial Intelligence, 88 (1), 187-212. doi:10.1007/s10472-019-09644-8

Laumanns M. 2002. Bayesian Optimization Algorithms for Multi-objective Optimization. Lecture Notes in Computer Science, 2439, 298-307.

Lyu W., P. Xue. 2018. An efficient Bayesian optimization approach for automated optimization of analog circuits. IEEE Transactions on Circuits and Systems. Part 1: Regular Papers, 65 (6), 1954-1967. doi:10.1109/TCSI.2017.2768826

Morice-Atkinson X. 2018. Learning from the machine: Interpreting machine learning algorithms for point- and extended-source classification. Monthly Notices of the Royal Astronomical Society, 481 (3), 4194-4205. doi:10.1093/mnras/sty2575

Nguyen T.D. 2018. Stable Bayesian optimization. International Journal of Data Science and Analytics, 6 (4), 327-339. doi:10.1007/s41060-018-0119-9

Pandita P. 2021. Computationally efficient bayesian optimization for multi-objective industrial applications. AIAA Scitech 2021 Forum, 1, 1-13. doi:10.2514/6.2021-1482

Pico-Valencia P. 2021. Bringing Machine Learning Predictive Models Based on Machine Learning Closer to Non-technical Users. Advances in Intelligent Systems and Computing, 1273, 3-15. doi:10.1007/978-3-030-59194-6_1

Popovic M. 2019. Formal verification of local and distributed python software transactional memories. Revue Roumaine des Sciences Techniques. Series Electrotechnique et Energetique, 64 (4), 423-428.

Пожидаева Н.А. 2023. Применение байесовской стратегии определения процесса для первоочередной оптимизации (на примере банковского сектора). Экономика: вчера, сегодня, завтра, 13 (3), 530-535, doi: 10.34670/AR.2023.83.23.085

Sharma P. 2021. Role of machine learning and deep learning in securing 5G-driven industrial IoT applications. Ad Hoc Networks, 123, 102685. doi:10.1016/j.adhoc.2021.102685

Sheikh H.M. 2022. Bayesian optimization for mixed-variable, multi-objective problems. Structural and Multidisciplinary Optimization, 65 (11), 1-14. doi:10.1007/s00158-022-03382-y

Smirnova V.S., Shalamov V.V., Efimova V.A., Filchenkov A.A. 2020. Hyperparameter optimization based on combining a priori and a posteriori knowledge of the classification task. Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics, 20 (6), 828-834.

Subasi A. 2020. Practical Machine Learning for Data Analysis Using Python. Practical Machine Learning for Data Analysis Using Python, 1-520. doi: 10.1016/B978-0-12-821379-7.00008-4

INFORMATION ABOUT AUTHORS / ИНФОРМАЦИЯ ОБ АВТОРАХ

CHERNYAGIN Anton S. – student.
E-mail: lyamin_bm@spbstu.ru
ЧЕРНЯГИН Антон Сергеевич – студент.
E-mail: a.s.chernyagin@gmail.com

SVETUNKOV Sergey G. – Professor, Doctor of Economic Sciences
E-mail: sergey@svetunkov.ru
СВЕТУНЬКОВ Сергей Геннадьевич – профессор, д.э.н.
E-mail: sergey@svetunkov.ru
ORCID: https://orcid.org/0000-0001-6251-7644

Статья поступила в редакцию 29.11.2023; одобрена после рецензирования 18.12.2023; принята к публикации 18.12.2023.

The article was submitted 29.11.2023; approved after reviewing 18.12.2023; accepted for publication 18.12.2023.