



INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT

INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT

Volume 2; Issue 1; 2024; Page No. 310-314

Received: 01-10-2023

Accepted: 09-11-2023

To analyze the optimization model of the OPLRA algorithm and the MP tree method

¹Purushottam Ji and ²Dr. Ajay Singh

¹Research Scholar, Department of Mathematics, Shri Krishna University Chhatarpur, Madhya Pradesh, India

²Associate Professor, Department of Mathematics, Shri Krishna University Chhatarpur, Madhya Pradesh, India

DOI: <https://doi.org/10.5281/zenodo.15078107>

Corresponding Author: Purushottam Ji

Abstract

Modern technological advancements have resulted in an unprecedented amount of data generation and storage. An important step is to summarize these datasets using basic quantitative models after spotting trends, patterns, and outliers. Our industrial partner Vattenfall is a real-world park designer, and they could really use a quick tool to assess design alternatives. They can comprehend the whole complexity of wind park design with the aid of these quick tools. The optimization model of the OPLRA algorithm is being utilized by this algorithm. It is important to include both numbers in order to make a fair comparison of those methods. Prior to using a heuristic iterative technique to determine the ideal number of regions for the regression model, the current OPLRA algorithm segmented the dataset using a MILP model.

Keywords: Optimization, model, OPLRA, algorithm, and Machine

Introduction

This optimization might be either convex or non-convex. Mathematics fields play an essential part in artificial intelligence and machine learning by studying the creation of crucial algorithms and models. We will emphasize the importance of optimizations, calculus, probability theory, and linear algebra in AI and ML, and we will provide examples of popular algorithms and strategies that utilize these areas. The goal of this study is to help researchers, practitioners, and students by giving a thorough review of the mathematical foundations of artificial intelligence and machine learning. In addition to improving one's capacity to build and tune ML and AI models, a thorough grasp of these mathematical underpinnings encourages creativity and makes it easier to find fresh answers to difficult challenges. Machine learning has relied on optimizations algorithms and formulations from its inception. Similarly, optimizations have benefited from machine learning, which has prompted the creation of novel optimizations methods to tackle the formidable problems inherent in ML applications. This mutual enrichment is still going strong, drawing top minds from both disciplines and creating an expanding body of

work at the interface of the two.

Because of their appealing theoretical features and broad practical application, optimizations techniques have risen to prominence in machine learning. Although methods that were suggested over twenty years ago are still being improved upon, a rational reevaluation of current assumptions and methods is necessary due to the growing complexity, size, and diversity of modern machine learning models. A first step towards that kind of reevaluation is taken up in this work.

Our industrial partner Vattenfall is a real-world park designer, and they could really use a quick tool to assess design alternatives. They can comprehend the whole complexity of wind park design with the aid of these quick tools, which allow them to evaluate a large number of potential design possibilities. Hundreds of different turbine types could be rapidly tested using a fast ML+MO method like the one we are suggesting here, and Vattenfall could even use this data to create a new turbine type that is particular to each location. Since this study aims to demonstrate that such tools may be created by combining ML and MO, it has accomplished its ultimate purpose.

There are a lot of challenging jobs and difficulties in computer science that need the use of computer programs, conventional procedures, and detailed instructions. The development of a self-driving automobile capable of object recognition is not a simple task, nor is it the creation of an enormously sophisticated desktop program or a dynamic game.

Literature Review

Snyman, J. *et al.* (2018) ^[1]. This book teaches scholars how to use gradient-only optimisation and gradient-only surrogates to handle challenging and discontinuous functions, stepping them out of the comfort zone of classical optimisation. Applications of this work range from automating the training of machine and deep learning models to discovering ideal structural forms. As far as we are aware, no other optimisation book addresses these topics. In May 2018, these concepts were presented in an invited PhD course at IMT Lille Douai, France. In March 2019, they were presented at the annual GPU Technology Conference in San Jose, Silicon Valley, followed by a talk on machine learning training.

Jalal, Ali *et al.* (2018) ^[2]. Many applications have shown the impressive performance of machine learning (ML) approaches. Nevertheless, most approaches rely on trial and error, which makes ML model creation a challenging and research-intensive process. An approach for optimizing ML model architecture and feature configurations using a supervised learning process is presented in this work. Two ML models, a deep neural network (DNN) and an adaptive neuro-fuzzy inference system (ANFIS), are optimized using the suggested method, which is based on a genetic algorithm (GA). When it comes to DNN optimisation questions, the chosen variables are the hidden layer count, activation function, and number of neurons.

England, Matthew. (2018) ^[3]. Symbolic computation's potential applications in artificial intelligence have received much attention, but the literature on its potential applications in the other direction is sparse. But new quantifier elimination findings show that, with enough example problems, Support Vector Machines and other machine learning methods may help Computer Algebra Systems perform better. In this review, we take a look at the authors' previous work as well as related uses of other mathematical tools. The precise answers desired by mathematical software could seem to be rendered useless by the intrinsically probabilistic character of machine learning techniques.

Sun, Shi liang *et al.* (2019) ^[4]. The fast advancement of machine learning has resulted in several theoretical discoveries and widespread applications across many domains. Researchers have paid a lot of attention to optimization since it is a crucial component of machine learning. Optimization techniques in machine learning are becoming more difficult to handle as both the quantity and complexity of available data continue to expand at an exponential rate. In machine learning, there has been a lot of

suggested work on improving optimization algorithms and tackling optimization issues in a sequential fashion. Great relevance lies in the systematic review and summary of optimization approaches from a machine learning standpoint, which may provide direction for advancements in both optimization and machine learning research. We begin by outlining the optimization issues with machine learning. After that, we will go over the basics of optimizations and the ways that it has evolved.

Gambell A, Claudio *et al.* (2019) ^[10]. Machine learning as optimizations models is introduced in this work, which also provides a literature review on the topic. The development of numerical optimizations methods, which have had a significant impact in several machine learning contexts, may improve these models. We focus on mathematical optimizations models that are applicable to deep neural networks, classification, regression, and clustering, as well as to novel applications in machine learning, empirical model learning, and machine teaching that are only starting to emerge. We highlight possible future research avenues and examine the benefits and drawbacks of various models.

Research Methodology

Two methods for implementing a piecewise regression algorithm are shown in this section. The optimization model of the OPLRA algorithm is being utilized by this algorithm. To maintain a happy medium between complicated models and accurate predictions, both variants use information criteria. How the two versions use the information criterion is the main difference between them.

Using post-processing the optimization model's output to determine the value of the criteria and use them to exit the algorithm, the first method iteratively selects the ideal number of regions. Here, the user may use the technique to fit data without ever having to set any hyper-parameters.

Here we see how Yang *et al.* (2016) ^[11] laid up the mathematical programming paradigm known as OPLRA. The model takes a multivariate dataset as input, using a specified variable to partition it into numerous segments. It then fits a linear function to each segment, determines the ideal regression coefficients and intercepts, and minimizes the total of the absolute deviation of the fitting. This section details every one of the formulation's indices, parameters, and variables: the regression intercept for feature m in area r , the regression coefficient for feature m in region r output expected for sample s within area r value at which feature m is considered broken for area r .

Data analysis

The mean absolute error (MAE) for each cross-validation run is presented in Table 1. The top performing strategy for each dataset is shown in bold. When looking at the MAE measure, Stat Tree comes out on top for seven out of fifteen instances. With 3 out of 15, Cubist is the runner-up, followed by MP tree and M5P with 2 out of 15, and CART with only one dataset. But that isn't a reliable measure of performance in and of itself.

Table 1: Cross-validation results using MAE

	StatTree	MPtree	Cubist	CART	M5P	CTree
Concrete	4.329	4.868	4.267	7.239	4.656	5.295
Cooling	1.175	0.891	0.938	2.400	1.210	1.403
Heating	0.367	0.354	0.347	2.011	0.693	0.665
Yacht	0.539	0.539	0.557	1.669	0.931	0.802
Bodyfat	0.183	5.282	0.205	1.356	0.373	0.911
Boston	2.568	4.644	2.587	3.234	2.501	3.014
Dee	0.313	0.975	0.316	0.381	0.316	0.356
Earthquake	7.345	12.427	7.294	8.223	7.273	7.884
Octane	0.391	0.805	0.384	0.602	0.464	0.591
Pharma	0.900	0.870	1.053	1.339	1.328	1.566
Plastic	1.226	1.230	1.229	1.658	1.234	1.410
Sensory	0.610	0.663	0.602	0.578	0.601	0.593
Wankara	0.972	3.605	1.000	3.213	0.977	1.574
Abalone	1.490	1.512	1.500	1.731	1.521	1.600
Speeding	4.143	4.243	4.188	4.524	4.239	4.581

The total prediction performance can be better understood with the help of a chart that compares the different methodologies.

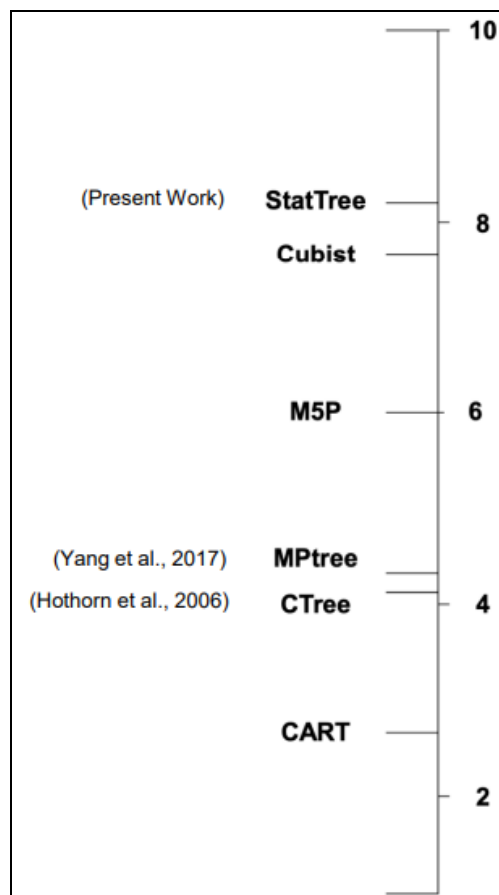


Fig 1: Visualization of the performance of the methods

Figure 1 facilitates a comparison of the algorithms' overall performance. There is a significant performance difference between StatTree and MPtree, suggesting that the new suggested technique addresses the shortcomings of the old one. When it comes to producing competitive outcomes, cubist is your only option. But because the two approaches' performances are so similar, we need to do a statistical test to see if the findings differ significantly.

You will only see a bar if the findings are significantly different from each other (the t-test rejects the null hypothesis). It is clear that CART and CTree are not

interchangeable in any of the cases studied. Eleven instances using M5P and eight cases using MP tree show a notable difference. In contrast, cubism distinguishes itself in just six instances.

the presence of a bar in Figure 2 b indicates that Stat Tree has earned a lower MAE score for the given case. On a consistent basis, the suggested method has beaten CART, CTree, and M5P. Despite sharing an optimizations model for node splitting with MPtree, StatTree produced superior MAE scores in 12 of the 15 instances after incorporating the F-test.

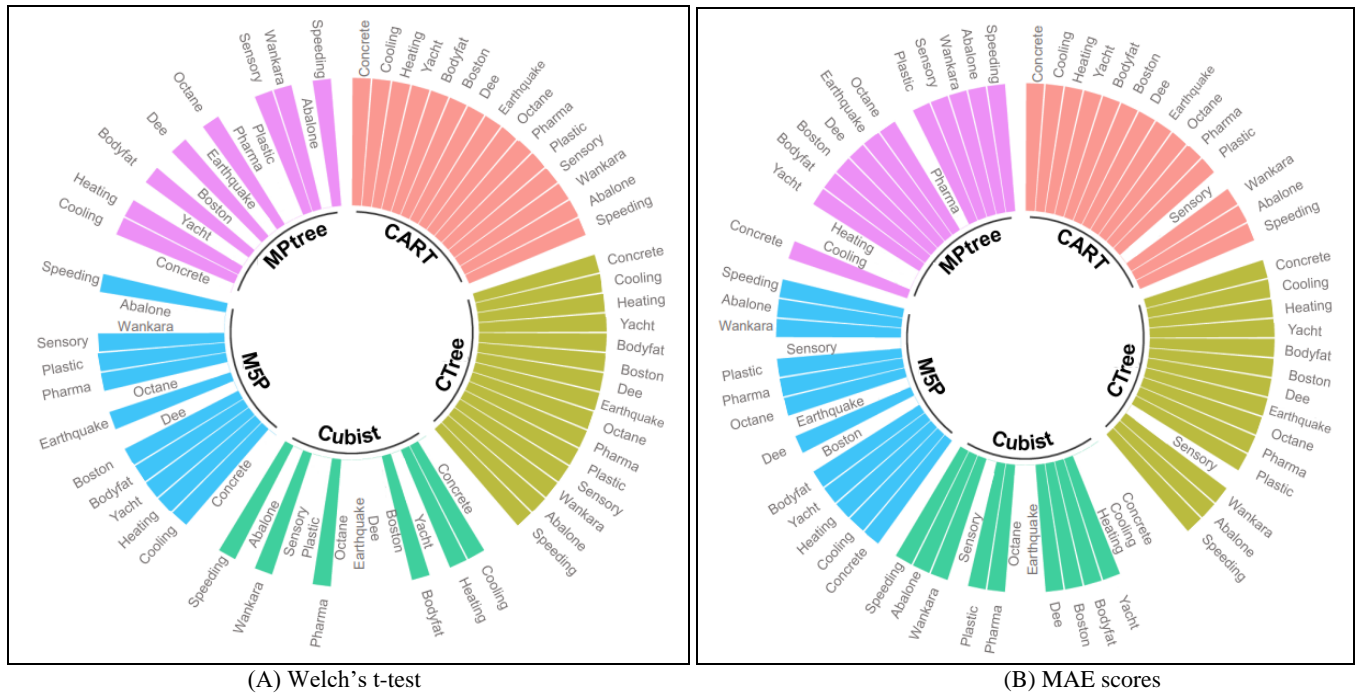


Fig 2: Visualization of the computational results of this work. Case (a) is a representation of the statistical analysis using Welch’s t-test. Case (b) is a representation of the MAE performance

It is important to include both numbers in order to make a fair comparison of those methods. Having a bar in both figures is beneficial for a specific case. In that instance, there's good reason to believe that StatTree has delivered superior error numbers compared to the rival and that the

MAE averages of all the CV runs are distinct. Another feature of this version is the ability to call GAMS from another programming language. The blue box components have all been implemented in Python, whereas the yellow ones have been implemented in GAMS.

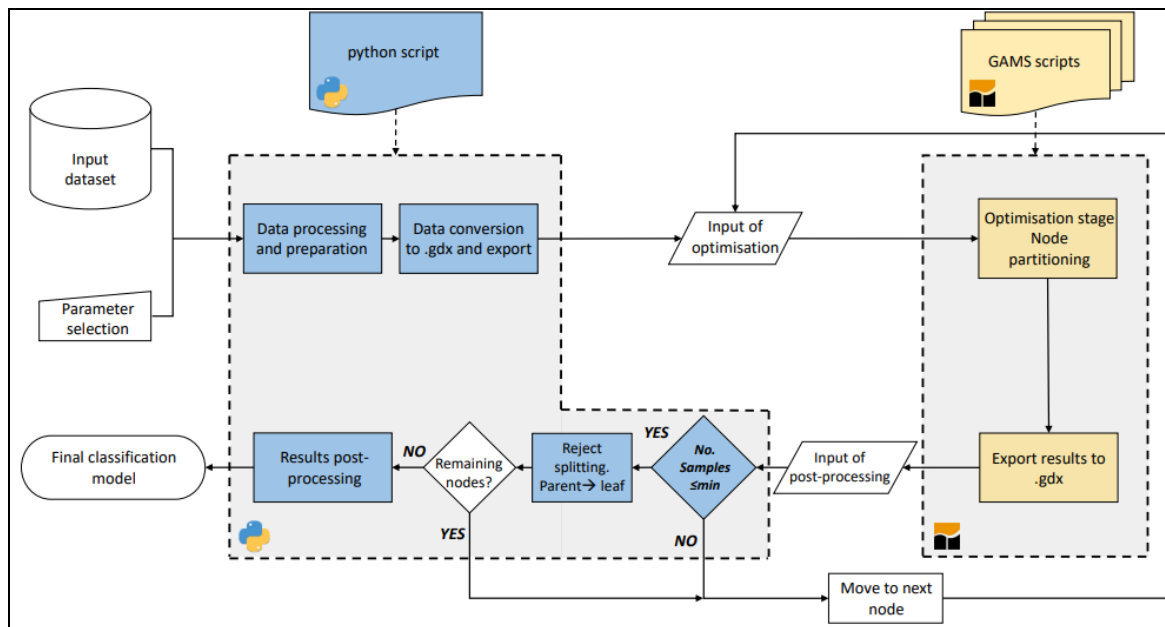


Fig 3: The implementation of the tree classification algorithm

The user must supply the input dataset and establish a value that determines the minimum number of samples permitted at each leaf node. They must also define the maximum number of classes per node using the parameter N, Processing the data and transforming them into the suitable format is the next stage in the implementation process. Again, a library named gdxpds is utilized to do the conversion, and the format utilized for this implementation

is.gdx. To make it easier to apply different machine learning methodologies to the developer's custom work, it is highly recommended to create a custom estimator by following the scikit-learn conventions and rules. This will ensure that the final implementation is compatible with all of the library's functionalities and utilities. Also, if future users are familiar with the scikit-learn library, they won't need any additional training to use this bespoke work.

Illustrative example

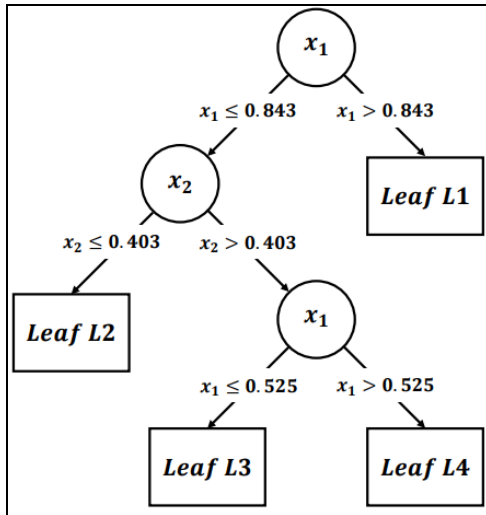


Fig 4: Illustrative example for the Optimal Decision Tree classifier. Figure (a) is a scatter plot of the examined dataset. Figure (b) is the constructed tree.

The ODT-generated tree structure. With just four-leaf nodes, the resultant tree is on the petite side. The laws of each leaf node that make up the overall classification model.

Table 2: Final classification model using the ODT algorithm

Leaf	Transformation	Ranges	
L1	$P_{L1} = 0.430 x_2$	purple: [0.101	0.366]
		yellow: [0.000	0.100]
		purple: [0.489	0.589]
L2	$P_{L2} = 0.306 x_1 + 0.654 x_2 + 0.840$	green: [0.000	0.488]
		yellow: [0.590	0.690]
		purple: [0.202	0.302]
L3	$P_{L3} = 0.464 x_1 + 0.513 x_2 + 0.320$	green: [0.000	0.100]
		yellow: [0.101	0.201]
		purple: [0.000	0.100]
L4	$P_{L4} = 0.568 x_1 + 0.386 x_2 + 0.823$	green: [0.101	0.201]
		yellow: [0.202	0.302]

Out of the seven nodes in the final classification model, four serve as terminals. A new pseudo-feature, P, is created from the multivariate data at each terminal node using a linear expression. In addition, the optimizations model's class ranges, as defined by that new feature, are stored in each leaf node.

Conclusion

Prior to using a heuristic iterative technique to determine the ideal number of regions for the regression model, the current OPLRA algorithm segmented the dataset using a MILP model. This optimization might be either convex or non-convex. Mathematics fields play an essential part in artificial intelligence and machine learning by studying the creation of crucial algorithms and models Using post-processing the optimization model's output to determine the value of the criteria and use them to exit the algorithm, the first method iteratively selects the ideal number of regions Modern technological advancements have resulted in an unprecedented amount of data generation and storage. An important step is to summaries these datasets using basic

quantitative models after spotting trends, patterns, and outliers.

References

1. Snyman J, Wilke D. Practical Mathematical Optimization: Basic Optimization Theory and Gradient-Based Algorithms. Springer Optimization and Its Applications. 2018;133:1-15. doi:10.1007/978-3-319-77586-9.
2. Jalal A, Alowaidi AFA. Optimization: The Mathematical and Computational Approaches to Understanding the Medical Science Applications. Journal of Global Pharma Technology. 2018;10:28-34.
3. England M. Machine Learning for Mathematical Software. 2018. Available from: <https://arxiv.org/abs/1806.10920>.
4. Sun S, Cao Z, Zhu H, Zhao J. A Survey of Optimization Methods from a Machine Learning Perspective. 2019. Available from: <https://arxiv.org/abs/1906.06821>.
5. Hamdia K, Zhuang X, Rabczuk T. An Efficient Optimization Approach for Designing Machine Learning Models Based on Genetic Algorithm. Neural Computing and Applications. 2021;33:277-290. doi:10.1007/s00521-020-05035-x.
6. Zhang C, Yao M, Chen W, Zhang S, Chen D, Wu Y. Gradient Descent Optimization in Deep Learning Model Training Based on Multistage and Method Combination Strategy. Security and Communication Networks. 2021;2021:9956773. doi:10.1155/2021/9956773.
7. Talekar S. The Role of Mathematics in Machine Learning. 2023;38:1062-1073.
8. He Y-H. Machine-Learning Mathematical Structures. 2021. Available from: <https://arxiv.org/abs/2101.06317>.
9. Naurzybayev B, Sakysh B, Zhanar A, Pak N, Karipzhanova A, Urazbaeva K. Using Machine Learning to Analyze the Learning Process for Solving Mathematical Problems. International Journal of Interactive Mobile Technologies (IJIM). 2022;16:114-124. doi:10.3991/ijim.v16i21.36065.
10. Behnam M, Dey A, Gambell T, Rajendran R. Rethinking manufacturing and distribution networks in MedTech. McKinsey & Company. 2019.
11. Yang B, Wang Y, Qian PY. Sensitivity and correlation of hypervariable regions in 16S rRNA genes in phylogenetic analysis. BMC bioinformatics. 2016;17:1-8.

Creative Commons (CC) License

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY 4.0) license. This license permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.