Multi-objective Optimisation in Machine Learning

730066594

ABSTRACT

This report explores multi-objective optimisation of hyperparaemters: minimum child weight,maximum depth and number of boost rounds for classification based machine learning method XGBoost using NSGA-2 and SPEA-2 multi-objective optimisers .The dataset used is a legacy data of 178 types of wines and their alcohol, ash and proline content. The fitness function for NSGA-2 and SPEA-2 are maximising accuracy and macro average. The model tries to predict alcohol content in a wine given the ash and proline amount.

1 INTRODUCTION

Machine Learning models can be supervised or unsupervised. Unsupervised machine learning models and their hyperparameters act as a black box problem[2].Thus, optimising them is currently an open challenge. This report utilises a supervised Machine Learning to analyse efficiency of optimising the hyper-parameters of the machine learning method XGBoost[1].XGBoost is an ensemble learning method based on decision trees. It sequentially builds trees to correct errors of the previous ones.This involves optimising a regularised objective function, combining tree predictions and minimizing loss. The parameters chosen for optimisation are ,minimum child weight,maximum depth and number of boost rounds. The minimum child weight parameter dictates the minimum sum of instance weights required for a child node during the tree-building process. This leads to the algorithm regulating the complexity of the individual trees, helping prevent over fitting. A higher minimum child weight results in more conservative splitting decisions which enhances generalization towards unseen data.

The maximum depth hyperparameter controls the maximum depth of each decision tree in the ensemble. More number of trees can capture intricate patterns but they also pose a risk of overfitting to the training data. maximum depth acts as a regularization mechanism, limiting the depth of the trees and promoting a balance between model complexity and generalization.

The number of boost rounds parameter defines the total number of boosting rounds, indicating how many trees are sequentially added to the ensemble. Increasing number of boost rounds generally improves model accuracy, as the algorithm has more opportunities to correct errors and refine predictions. However, a higher number of boosting rounds also requires more computational resources and time for training.

Balancing these hyperparameters is essential for achieving optimal model performance. It involves a trade-off between capturing complex patterns in the data, preventing overfitting, and managing computational resources. Careful experimentation and tuning of these parameters are essential for tailoring XGBoost to the specific characteristics of the dataset and the objectives of the modeling task.

To demonstrate utility of Multi objective optimisation of hyper parameters[5], This report uses a small dataset of 178 elements to demonstrate the efficiency of multi objective optimisers in finding pareto-optimal solutions and how they can help in fine tuning machine learning models with pareto-optimal objectives such as maximum accuracy and maximum macro average .

2 BACKGROUND LITERATURE REVIEW

Hyperparameter Optimization (HPO) involves understanding a solution's behavior on unseen data. This is achieved through a three-way split of the data into training, validation, and test sets. MOHPO however, involves multiple conflicting objectives.This makes the evaluation more complex. The multi-objective hyper parameter optimisation (MOHPO)[5] is a developing field of study. It involves optimising the hyperparameters of machine learning methods[3] to fine tune the machine learning models. A simple approach is to scalarise the different objectives using a scalarising function such as weighted Chebyshev. This converts the multi objective problem to a single objective problem. This method however assumes a priori knowledge of how different objectives interact with each other. The weights are decided a priori as well. This can lead the optimisation to a result which is already expected.The goal of Diversity of optimal solutions is not achieved using this approach.

Thus, other multi-objective optimisers[4] are needed to achieve the goals of:

Convergence: The proximity to the true Pareto front
Diversity: The coverage of the Pareto front
Uniformity: The evenness of the distribution of thesolutions
Cardinality: The number of solutions

Multi objective evolutionary algorithm NSGA2 is the most popular MOEA for optimising binary classifications. NSGA-II employs a Pareto dominance-based strategy. It initially utilises a non-dominated sorting to establish a coarser ranking of the population. Next,with help of iterative steps NSGA-II classifies non-dominated solutions and assigns them to available classes. It then progressively removes them from consideration. Each class solutions are ranked based on their extremeness in each objective with the best in each objective attaining the highest rank. The remaining solutions are then prioritized using crowding distance.

NSGA-II excels in scenarios with two objectives However, it encounters challenges when dealing with a higher number of objectives. The non-dominated sorting becomes less discriminative which leads to potential breakdowns in solution diversity. The softwares Dragonfly and Hypermapper are industry-standard MOHPO. They are being used to solve real world problems such as Dragonfly in aerospace sector, Hypermapper for tuning High performance computing applications and simulation-based design scenarios, where optimizing design parameters can lead to improved performance. Dragonfly is inspired by the collective behavior of dragonflies. This is a nature-inspired optimization library that excels in multi-objective hyperparameter optimization. It uses asynchronous particle swarm optimization to efficiently navigate the hyperparameter space. Dragonfly adapts well to the dynamic nature of optimization problems which makes it suitable for real-world scenarios. The incorporation of surrogate models also enhances its scalability. This allows it to explore large and complex search spaces. Hypermapper has been designed to tackle complex optimization tasks. It focuses on surrogate-assisted optimization for multi-objective hyperparameter tuning. It's strength lies in handling expensive and black-box objective functions. It does that by using a Bayesian optimization framework. It uses surrogate models to guide the search towards promising regions of solutions. It supports parallel evaluations.

3 EMPIRICAL WORK PLAN

The empoirical work invovled:
Formulating a machine learning problem to solve
Choosing a Dataset,The type of dataset decides the machine learning method and optimiser to use
Choosing a classification machine learning method
Choosing a multi-objective optimiser
Choosing hyperparmeters of machine learning method
Estimating the hyperparameters using the chosen multi objective optimiser

3.1 Machine Learning problem

To predict the alcohol content in a given wine, Based on its ash and proline content.

3.2 Dataset

Predictable classification based datasetwhich contains alcohol, proline and ash content of 178 different types of wines was used. The data set was obtained from Huggingface.

3.3 Machine Learning method

XGBoost was chosen as the classification machine learning method. XGBoost is easy to implement and its hyperparameters are simple to understand.

3.4 Multi-Objective optimisers

To analyse different multi objective optimisers performance in machine learning hypoerparameter optimisation.Two algorithms were tested. EAsimple with NSGA2 based selection and EAsimple with SPEA2 selection. EAsimple is a evolutionary algortihm with crossover and mutation. This was chosen due to the simplicity of the algorithm.

3.5 Hyperparameters

The Hyperparameters minimum child weight,maximum depth and number of boost rounds were chosen as they have conflicting influence on the accuracy and macro averages.

3.6 Estimating Hyperparameters using EAsimple(selection: NSGA2 and SPEA2)

Split the data into training and testing set. Define a function called 'evaluate' which runs the machine learning method XGBoost using training set with variable parameters minimum child weight,maximum depth and number of boost rounds. All other parameters are default. The function returns the accuracy and macro average of the model developed by comparing the results of model using training and testing dataset. Setup EAsimple from DEAP library. The fitness function for EAsimple is the 'evaluate'. The population is generated by specifying different values of minimum child weight,maximum depth and number of boost rounds. For sake of simplicity, the hyperparameters are generated randomly as a set of valid values. For Example: minimum child weight is an floating point number between 0-10, maximum depth is an integer between 1-100, number of boost rounds are an integer between 1-100.

4 EXPECTED RESULTS

The possible search space of selected hyper-parameters in specified range are: 10^5. Utilising NSGA2 and SPEA2, the best accuracy and best macro weight should be achieved fast in limited number of results as data-set for machine learning model is small. The pareto front so obtained should be a selection of best results.This will highlight the utility of MOHPO compare to traditional HPO which is computationally expensive and does not fulfill our pareto-optimal soltuion criterias.

5 EXPERIMENTATION RESULTS

The most optimal solution of best accuracy and best macro average is found in less than 15 generations(lambda =10,initial population = 20) for both NSGA2 and SPEA2 based selections. However, due to the way XGBoost hyper-parameters are solved in 'evaluate', the accuracy values are either 0 or 0.9722.The hyper-paremeters can go beyond the valid range accepted by XGBoost, Thus a minimum value is specified for each hyper-parameter.Therefore, (max) accuracy and (max) macro average pareto-optimal solution plot are just two points. The EAsimple algorithm is able to find multiple values of hyperparameters which have same accuracy and macro averages.

6 CONTEXTUALISATION OF RESULTS

The results showcase the efficiency of MOHPO. In just 10 generations, the MOEAs are able to find the set of most opitmal values for the given dataset. However, the challenges of MOEAs in machine learning do exist. For example, the output of MOEAs can be invalid to be input as hyperparameters of machine learning models. This can lead to loss of diversity in solutions.

num round	Max Depth	Min Child Wt	Accuracy	(Mcr)Average
56.07	2.66	-0.86	0.972	0.957
56.21	-0.48	1.15	0.972	0.957
50.33	13.55	2.76	0.972	0.957
122.71	4.70	8.47	0.972	0.957
50.43	-0.62	2.59	0.972	0.957
-32.68	6.85	6.96	0.972	0.957
-77.49	6.91	6.77	0.972	0.957
34.28	8.29	11.71	0.972	0.957
38.15	7.94	6.23	0.972	0.957
-23.35	7.09	5.56	0.972	0.957
48.99	8.73	7.46	0.972	0.957
13.16	3.84	6.06	0.972	0.957
36.14	5.05	6.88	0.972	0.957
33.48	7.17	7.17	0.972	0.957
28.99	3.80	11.50	0.972	0.957
136.78	5.68	5.51	0.972	0.957
58.74	5.51	5.06	0.972	0.957
54.80	4.66	4.94	0.972	0.957
90.15	5.56	6.57	0.972	0.957
73.22	7.46	3.05	0.972	0.957

Table 1: NSGA2 selection[initial population = 20,gen =10]

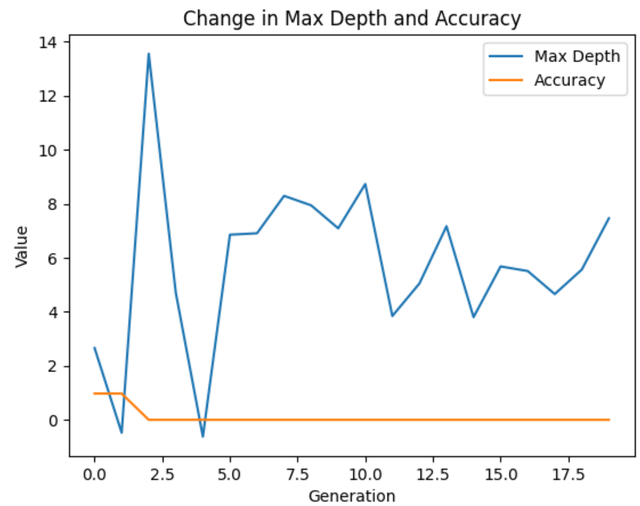

Figure 1: Change in max depth and accuracy with generations of NSGA2

num round	Max Depth	Min Child Wt	Accuracy	(Mcr)Average
140.13	5.57	1.26	0.972	0.957
60.91	-0.81	1.37	0.972	0.957
-10.13	1.43	1.26	0.972	0.957
184.18	7.95	25.86	0.972	0.957
35.87	7.84	4.48	0.972	0.957
83.28	7.47	9.45	0.972	0.957
55.12	8.15	5.81	0.972	0.957
2.53	5.46	9.85	0.972	0.957
23.91	4.13	7.02	0.972	0.957
-45.25	10.59	7.64	0.972	0.957
177.45	8.23	7.68	0.972	0.957
90.21	3.39	7.16	0.972	0.957
-3.17	9.38	6.25	0.972	0.957
17.37	7.33	9.53	0.972	0.957
83.95	5.13	9.39	0.972	0.957
142.43	15.46	5.16	0.972	0.957
33.25	12.21	8.92	0.972	0.957
27.93	1.94	5.40	0.972	0.957
38.19	9.38	2.50	0.972	0.957
1.05	9.00	3.02	0.972	0.957

Table 2: SPEA2 selection[initial population = 20,gen =10]

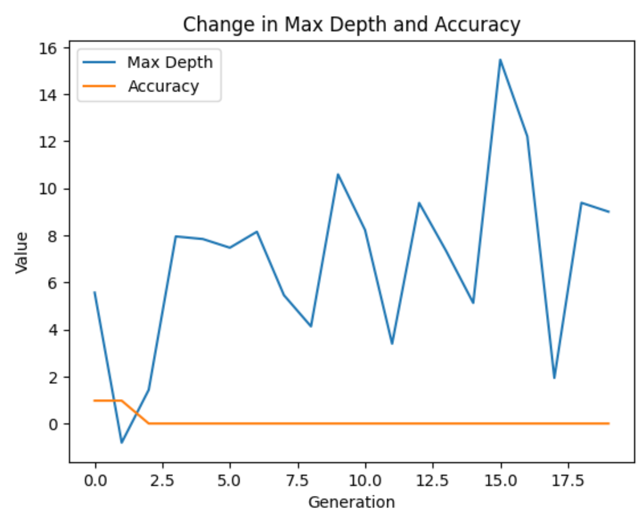

Figure 2: Change in max depth and accuracy with generations of SPEA2

7 CONCLUSION

The results achieved with a limited number of generations have shown the promise of Multi-Objective Hyperparameter Optimization (MOHPO) over traditional methods (HPO)

The report highlights the intricate trade-offs inherent in fine-tuning hyper-parameters and emphasizes the significance of navigating conflicting hyper-parameters such as minimum child weight, maximum depth, and boost rounds. These hyper-parameters play a pivotal role in preventing over-fitting, capturing intricate patterns, and optimizing computational resources.

Future areas of interest include exploring the scalability of Multi-Objective Evolutionary Algorithms (MOEAs) to larger datasets and more complex models. Addressing challenges related to the validity of output hyperparameters and potential diversity loss in solutions.

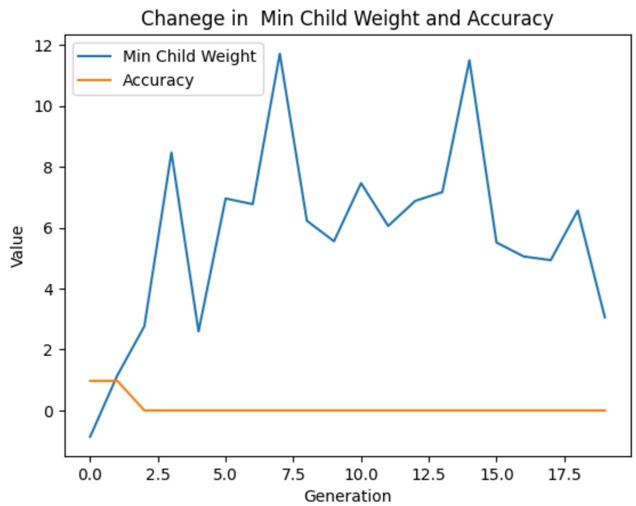

Figure 3: Change in minimum child weight and accuracy with generations of NSGA2

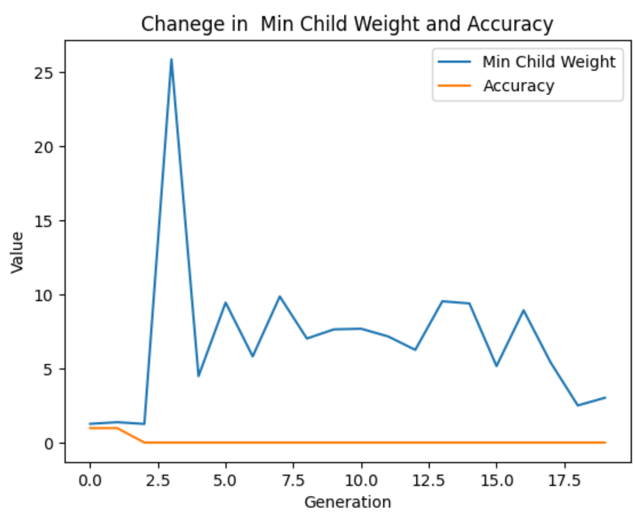

Figure 4: Change in minimum child weight and accuracy with generations of SPEA2

This research contributes to advancing the application of multi-objective optimization in tailoring machine learning models, fostering efficiency, effectiveness, and applicability in real-world scenarios.

REFERENCES

[1] Lothar Thiele Eckart Zitzler, Dimo Brockhoff. [n. d.]. The hypervolume indicator revisited:On the design of Pareto-compliant indicators via weighted integration In. *Evolutionary Multi-Criterion Optimization* 4403 ([n. d.]), 862–876.

[2] Florian Karl et al. 2023. Multi-Objective Hyperparameter Optimization in MachineLearning—An Overview. *ACM Transactionson Evolutionary Learning* 3, 4 (Dec. 2023), 1–50. https://doi.org/10.1145/3610536

[3] Daniel Horn. 2010. *Multi-Objective Analysis of Machine Learning Algorithms Using Model-Based Optimization Techniques.* Technische Universität Dortmund.

[4] Daniel Horn and bernd Bischl. 2016. *Multi-objective parameter configuration of machine learning algorithms using model-based optimization.* Number 1-8. IEEE, LosAlamitos,CA.

[5] Giulia DeSalvo Afshin Rostami zadeh and Ameet Talwalkar Lisha Li, Kevin G.Jamieson. 2017. Maintaining Discrete Probability Distributions Optimally. In *Hyperband:A novel bandit based approach to hyperparameter optimization*, Vol. 18. Journal of Machine Learning Research, 52.