BBO Competition Report

Benedictus Kent Rachmat
Diego Andrés Torres Guarín
Submission ID: 670441 and 670628 (Documentation)

February 2024

1 Approach

For this competition we decided to use the technique described in the paper that we presented, titled *Practical Bayesian Optimization of Machine Learning Algorithms*[1]. It was compelling for us to take the concepts and formulas that we learned and put them into practice. As described in the paper, the key idea is to approximate the objective function (in this case the performance of an ML model) by using a Gaussian Process (GP). This meta-model will then be used to find promising areas of the hyperparameter space, which can lead to both a better fit of the objective function and a better performing ML model. This dilemma is quite resembling of the exploitation exploration trade-off in reinforcement learning.

We follow the advice mentioned in the paper. We use the Matern(5/2) kernel (provided by the sklearn library), and the expected improvement acquisition function (EI), given by the following formula:

$$a_{\mathrm{EI}}(x) = \sigma(x)\left[\gamma(x)\Phi(\gamma(x)) + \mathcal{N}(\gamma(x); 0, 1)\right], \tag{1}$$

with $\gamma(x) = (f(x_{\text{best}}) - \mu(x))/\sigma(x)$.

One of the biggest challenges was designing an algorithm that would work for any type of hyperparameter (categorical, binary, integer, etc). We decided to use the 'range' variable provided in the 'api_config' to rescale the integer and float variables between 0 and 1. The boolean variables were converted to integers, and the categorical variables converted to an integer range between 0 and n_{cat}-1.

Similarly, we encountered the challenge of optimizing the acquisition function to find the next move $x^* = \operatorname{argmin} a(x)$. Since we didn't have much time, we employed a brute force approach, by sampling at least N points from the search space and evaluating the acquisition function in each of them. The best $n_{\text{suggestions}}$ points were returned in the 'suggest function'.

2 Experiments

Our experimentation focused on refining the algorithm's performance. Initial tests without variable scaling yielded lower results. Implementing scaling improved outcomes, supporting its inclusion. We also increased the number of points sampled to leverage the meta-model's rapid evaluation, which directly improved performance. The adoption of Gaussian Process (GP) with Expected Improvement (EI) was crucial. It outperformed random search by using a probabilistic model for smarter sampling, leading to faster convergence. In comparison to TURBO and PySOT, GP EI maintained parity, benefitting from its balance of exploration and exploitation within the search space.

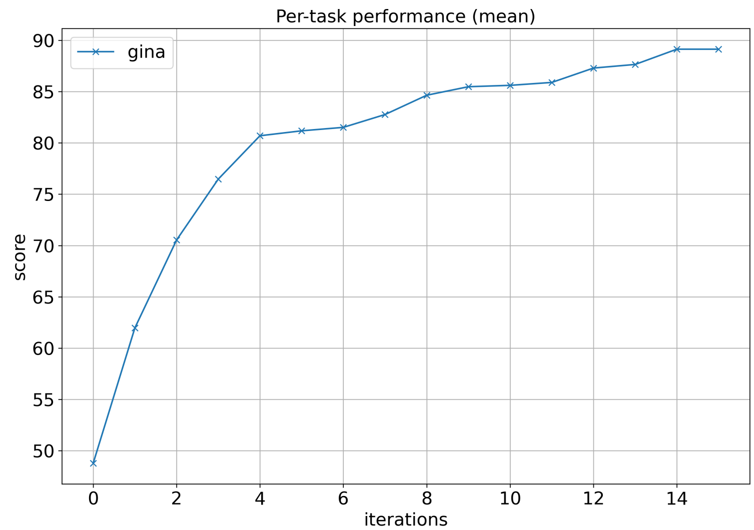
Table 1: Experiment results with various sample sizes (N).

N	**Result**
10	91.54
50	91.27
200	91.59
500	92.14
1000	**93.36**
2000	91.41
5000	92.59

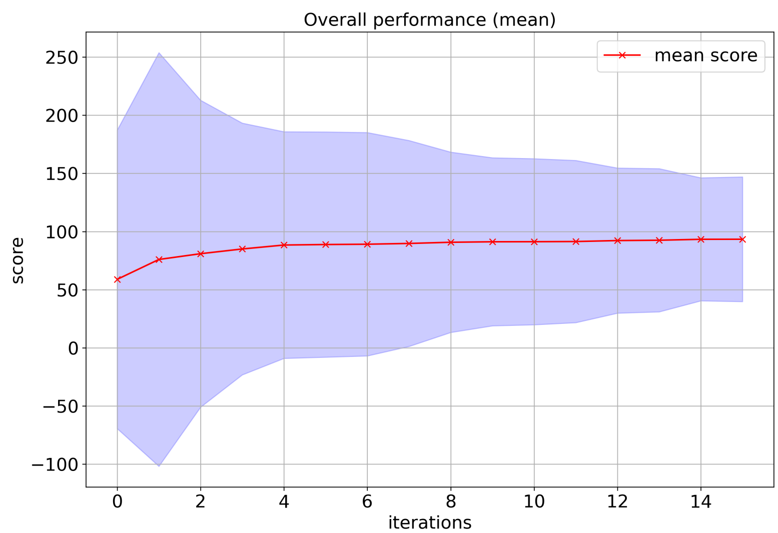
References

[1] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

A Supplementary Material (N=1000)



(a) Per-task performance (mean) over iterations, showing the trend of individual task performance with increasing number of iterations.



(b) Overall performance (mean) over iterations, indicating the mean score achieved across all tasks at each iteration.

Figure 1: Performance graphs showing the trend of scores over the course of iterations. The per-task graph (a) displays individual task improvements, while the overall graph (b) shows the aggregate performance trend.