Higher Performance for AutoML: The Benefit of Various Ensemble Bayesian Optimization Strategy

Jiajun Wu[1], Manliang Cao[2], Liping Shan[1], and Qing Yang[1](✉)

[1]Data Intelligence Department, Du Xiaoman (Beijing) Science Technology Co., Ltd.
Building 4, West District, yard 10, East Xibeiwang Road, Haidian District, Beijing
{wujiajun,shanliping,yangqing}@duxiaoman.com
[2]School of Computer Science, Fudan University
220 Handan Road, Shanghai, China
17110240029@fudan.edu.cn

Abstract

In AutoML tasks, the Bayesian Optimization (BO) based methods, which often resort to various surrogate models to explore a good trade off between the exploitation and exploration, have shown great effectiveness for hyper-parameter optimization. Most of these methods use a single pattern, however, each of these model contains the common limitations: 1) easy to get the suboptimal solution; 2) time-consuming for optimizing the model when meeting high dimension features. To get better results for AutoML tasks, in this paper, we propose different ensembles Bayesian models and try to explore these models, which adopt the advantages of different surrogate models so as to make these models to complement each other and alleviate these limitations. The experimental results on the BBO datasets show the effectiveness of our motivations. Moreover, we can find that the hybrid method can generally get better scores. In the mix of models, we also find that turbo and pysot based hybrid strategy gets the best performance. The code is public available at `https://github.com/SupUnicorn/bbo_challenge_4th`.

1 Introduction

In AutoML [9, 18, 8] tasks, in order to get the optimal solutions of various tasks, the hyper-parameter optimization process plays an important role. Among these hyper-parameter optimization based methods, Bayesian Optimization (BO) [6, 16] based methods show strong advantages to this area. Generally, the BO based methods are black box process for different tasks and typically take a lot of time to get good hyper-parameters. The performance of BO algorithm can be improved by defining the surrogate model and acquisition function of the algorithm.

Recently, different surrogate models have shown their effectiveness for choosing the hyper-parameters of AutoML tasks. Most of these methods try to find a good tradeoff between the exploitation and exploration, which are two classical sampling strategy for getting global optimal solution. TPE [20, 12], known as Tree-Structured Parzen Estimator, is a method of learning the hyper-parameter Model with the GMM (Gaussian Mixture Model) [14, 21]. RBF interpolation [11, 7] is one of the most popular methods for approximating the general dimensions of discrete data. Gaussian process [15, 17] is a random process with joint Gaussian distribution for any finite number of random variables when seeking the desired hyper-parameters [10]. Although the above surrogate models get some effectiveness, these methods contain several limitations: (1) Since Bayesian optimization is to use Bayesian theorem to estimate the posterior distribution of the objective function, and it is easy to continuously get the local optimal solution when involving the tradeoff between exploitation and

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

exploration[13]; (2) When meeting the high dimension features, these methods are likely to take lots of time for the solutions.

To solve the above limitations and get better results for BBO competition, we propose various ensemble models [3, 19] strategy which combines the advantages of the above surrogate models. Specifically, in order to get the global optimal solution, our goal is to achieve a better hyper-parameter enumeration in each batch optimization through balanced exploration and exploitation. The strategy proposed in this paper can combine the advantages of different surrogate models and acquisition functions, and at the same time find a balance in the exploitation and exploration, so as to improve the efficiency of hyper-parameter optimization. Finally, our proposed strategy shows competitive results with the BBO datasets.

2 The proposed approach

The method proposed in this paper uses the hybrid surrogate model, puts the hyper-parameter groups proposed by different base Bayesian optimizers into the same optimization space, and uses different surrogate models and acquisition functions to pick up new hyper-parameters.

When the same surrogate model and acquisition function are used to push the hyper-parameter beyond a certain number of steps, the new hyper-parameter often falls into the local optimal situation. In the engineering implementation of the surrogate algorithms, pysot [4] and turbo [5] have some optimizations in dealing with local optimality, but they also tend to pay more attention to development instead of exploration. This is because in their exploration methods, the value of exploration is relatively random, which lacks priori. Our approach will leverage different algorithms to enhance exploration without losing exploitability.

In Hybrid Bayesian optimization, we initialize two or more different Surrogate models. On each batch, different models select new hyper-parameters through their own acquisition function, and then update the mapping relationship between hyper-parameters and metrics after training and validation, at the same time, different surrogate models will update their own model by using all mapping relationships.

Using the same mapping space for different surrogate models, but using different acquisition functions when collecting new hyper-parameters, can make different approaches more explorative. The basic goal of our motivation is to balance the exploration and the exploitation aspects of the algorithm. The pseudocode of the algorithm is shown in the Algorithm 1.

Algorithm 1 Hybrid Surrogate Model Optimization

```
Input: $f_X, \chi, S_1, S_2$
initialize $surrogate_1$,$surrogate_2$,$H = \{\}$
for each $batch \in batches$ do
    $p_1(y|x, H) \leftarrow FITMODEL(surrogate_1, H)$
    $p_2(y|x, H) \leftarrow FITMODEL(surrogate_1, H)$
    $h_1 \leftarrow \arg\max_{x \in \chi} S_1(x, p_1(y|x, H), batch\_size/2)$;
    $h_2 \leftarrow \arg\max_{x \in \chi} S_2(x, p_2(y|x, H), batch\_size/2)$;
    $metrics = f_X(h_1 \cup h_2)$
    $H = H \cup \{(h_1 \cup h_2), metrics\}$
end for
return $hp \leftarrow \arg\max_{metrics} H$
```

3 Experiments

3.1 Experimental setup

Datasets In this paper we use the BBO datasets, These data belong to the traditional machine learning data set, including binary classification, multiple classification and regression data. The

Table 1: The experimental results

	Original			Hybrid		
Dataset	**hyperopt**	**pysot**	**turbo**	**hpo+pysot**	**hpo+turbo**	**pysot+turbo**
All	95.679	97.824	97.972	97.935	98.172	**98.795**
iris	94.580	96.261	95.621	95.682	95.312	**96.759**
boston	98.926	99.669	100.393	99.184	100.286	**100.511**
breast	98.086	97.946	97.320	**99.614**	98.795	98.946

value range of the hyper-parameter is specified by the Bayesmark toolkit; The experiment was carried out using a toolkit provided by blackbox.

Baselines This paper we apply several surrogate models of Bayesian optimization based method as our baselines. They are TPE method, which uses **hyperopt** [1] to support TPE's optimization algorithm, RBF method which applies **pysot** packages optimization processes and the **turbo** is used for Gaussian process of bayesian optimization.

Experimental details TPE, RBF and GP belong to different surrogate models, and we use different acquisition functions to propose new hyper-parameters. When combining the two different algorithms, the batch definition is divided into two parts and only the hyper-parameter of $batch\_size/2$ is proposed respectively. During the optimization of the surrogate model, each model will update the mapping of all the hyper-parameters and metrics, which is equivalent to its exploratory being considered based on the experience of other models, and it maintains its exploitation at the same time. In the process of trying out different methods, we have also adopted the meta-based hyper-parameter tuning method, shrinkage method combined with Thompson Sampling [2], etc., and the final method is used the ensemble Bayesian optimization method.

3.2 Experimental results

We conduct experiments on different algorithms based on the data provided by Bayesmark, and compare the results of three basic algorithms and different combinations of these three algorithms. It can be seen that the hybrid approach generally has better results than using a single surrogate model, and the stronger the base model is, the better the fitting ability of the combined model is. These effective performances certify that all of these surrogate models can complement each other, and we find that the ensemble model based on **turbo** and **pysot** method achieves higher results.

4 Conclusion

We use the method of hybrid surrogate model to improve the effect of Bayesian optimization. By comparing with the original individual model, we can find that the hybrid method can generally get better scores. In the mix of models, we find that **turbo** and **pysot** based hybrid results are the best choice, and we used this code for the final version of the submission.

Acknowledgement

This work is supported by the company Du Xiaoman, which offers us plenty of hardware resources so as to make us to implement various ensemble model possible. We thanks very much NeurIPS and Twitter for giving us the precious opportunity to take part in this great competition.

References

[1] Bergstra, J., Yamins, D., Cox, D.D.: Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In: Proceedings of the 12th Python in science conference. vol. 13, p. 20. Citeseer (2013)

[2] Chapelle, O., Li, L.: An empirical evaluation of thompson sampling. In: Advances in neural information processing systems. pp. 2249–2257 (2011)

[3] Dietterich, T.G., et al.: Ensemble learning. The handbook of brain theory and neural networks **2**, 110–125 (2002)

[4] Eriksson, D., Bindel, D., Shoemaker, C.A.: pysot and poap: An event-driven asynchronous framework for surrogate optimization. arXiv preprint arXiv:1908.00420 (2019)

[5] Eriksson, D., Pearce, M., Gardner, J., Turner, R.D., Poloczek, M.: Scalable global optimization via local bayesian optimization. In: Advances in Neural Information Processing Systems. pp. 5496–5507 (2019)

[6] Frazier, P.I.: A tutorial on bayesian optimization. In: arXiv preprint arXiv:1807.02811 (2018)

[7] Gutmann, H.M.: A radial basis function method for global optimization. In: Journal of global optimization. vol. 19, pp. 201–227. Springer (2001)

[8] He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowledge-Based Systems p. 106622 (2020)

[9] He, Y., Lin, J., Liu, Z., Wang, H., Li, L.J., Han, S.: Amc: Automl for model compression and acceleration on mobile devices. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 784–800 (2018)

[10] McGibbon, R.T., Hernández, C.X., Harrigan, M.P., Kearnes, S., Sultan, M.M., Jastrzebski, S., Husic, B.E., Pande, V.S.: Osprey: Hyperparameter optimization for machine learning. Journal of Open Source Software **1**(5), 34 (2016)

[11] Neumaier, A.: Complete search in continuous global optimization and constraint satisfaction. In: Acta numerica. vol. 13, pp. 271–369. Cambridge University Press, The Edinburgh Building, Cambridge CB 2 2 RU … (2004)

[12] Nguyen, H.P., Liu, J., Zio, E.: A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by tree-structured parzen estimator and applied to time-series data of npp steam generators. Applied Soft Computing **89**, 106116 (2020)

[13] Raisch, S., Birkinshaw, J., Probst, G., Tushman, M.L.: Organizational ambidexterity: Balancing exploitation and exploration for sustained performance. Organization science **20**(4), 685–695 (2009)

[14] Rasmussen, C.: The infinite gaussian mixture model. Advances in neural information processing systems **12**, 554–560 (1999)

[15] Rusmassen, C., Williams, C.: Gaussian process for machine learning. the MIT Press (2005)

[16] Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: A review of bayesian optimization. In: Proceedings of the IEEE. vol. 104, pp. 148–175. IEEE (2015)

[17] Wilson, J.T., Borovitskiy, V., Terenin, A., Mostowsky, P., Deisenroth, M.P.: Efficiently sampling functions from gaussian process posteriors. In: arXiv preprint arXiv:2002.09309 (2020)

[18] Wong, C., Houlsby, N., Lu, Y., Gesmundo, A.: Transfer learning with neural automl. In: Advances in Neural Information Processing Systems. pp. 8356–8365 (2018)

[19] Zhang, C., Ma, Y.: Ensemble machine learning: methods and applications. Springer (2012)

[20] Zhao, M., Li, J.: Tuning the hyper-parameters of cma-es with tree-structured parzen estimators. In: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI). pp. 613–618. IEEE (2018)

[21] Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. vol. 2, pp. 28–31. IEEE (2004)