Bayesian Optimization for Category Space

Jun Jin

jinbob@yeah.net

ABSTRACT

Hyper parameter optimization is widely used in AI areas. Hyper parameter usually means the value controls the whole learning process, but itself cannot be learned or tunned in training process. Hyper parameter is very important because it will greatly affect the learning result. A good hyper parameter set can lead to a much better result or cost much less training time, instead a bad hyper parameter usually will end in local optimum, or even failed to converge.

Hyper parameters can be many difference kinds of types, it could be in the model itself (depth, node counts, etc..), or it could be in the algorithm (learning rate, optimizer, etc..). Different models or algorithms usually need different hyper parameters, even the same model/algorithm can use different hyper parameters to achieve better results. So hyper parameter exists in different part of the training process, some of the hyper parameter is described in a category. It usually means that the parameter can only be chosen in a range. This kind of parameter has some properties, for this special kind of hyper parameter we proposed a common method here to optimize it. By using this method we turn the category problems into Real searching space to achieve a better result.

CCS CONCEPTS

• Theory of computation•Theory of computation~Theory and algorithms for application domains•Theory of computation~Theory and algorithms for application domains~Theory of randomized search heuristics

KEYWORDS

Hyper Parameter Optimization, Bayesian Optimization, Category Optimization

1 Introduction

With the development of the technologies, the model become more and more complex, this results the hyper parameter also become very large and hard to tune. Traditionally we can use grid search or random search to find the best parameter, but in practice it is not feasible due to it will cost too much time. We want to find an efficient algorithm that it could find the better result in the limited time. Bayesian Optimization is the widely used as the black-box hyper parameter optimization algorithm[1]. It assume that the hyper parameter could be defined by $x^* = \text{argmin } f(x)$, f here refers to the relationship between the reward and the hyper parameters(x). Bayesian Optimization predicts the probability distribution of the function by the value of the sample point. Gaussian Process is usually used as prior probability.

1.1 Category Space

Hyper parameters optimized by Bayesian Optimization algorithm need to be defined in R, but for the category search space, it is not continuous, so it cannot be optimized directly. we need to extend the space from category to R first. But there are a lot of points cannot be sampled in R space. In this case we can simply combine these values into the adjacent categories and treat them as the same value. But this will introduce inconvenience and harm to the Bayesian Optimization algorithm. Image the bad case, the algorithm always sampled in the same category, it waste time and meaningless. We want the sample point to be far away enough from each other in order not to be in the same category. Inspired by [2] we proposed the idea to add penalty on the sample points according to the distance for the category space. After adding the penalty, hyper parameter can be described as $x^* = \text{argmin } f(x)$ $\xi(d)$. The penalty could be computed as [2], or could be a function ξ to the distance d, d is category bound. There are 2 advantages by adding penalization, the first is penalization can enlarge search space, it tends to explore more unknown spaces. Second it tries to avoid being resampled.

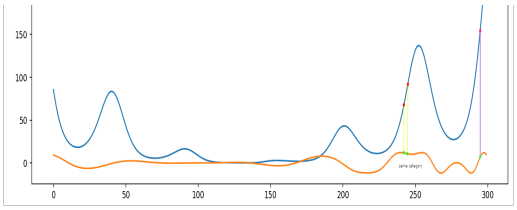

Figure 1: adjacent sample points in the same category

As shown in Fig1, the orange line represents for the acquisition function, and the blue line is f function. The green point is the sample point that belongs to the same category. by adding the penalization, the purple point is selected as sample point at last.

1.2 Ensemble

Ensemble methods are also used to improve the performance of the algorithm. Beside the averaging and voting strategies which are common for the Bayesian Optimization algorithm. We proposed an implement method for the Category space. As described above, we have introduced local penalization strategy to keep the sample points to be far away from each other. But this still will happen especially in the later iterations. Because in this stage, the sample point range has been greatly reduced, so even added the penalization, the sampled points could be still adjacent to the same category. This usually means that the searching has already entered the local optimal area, we need to jump out this region to expand the searching area, so another acquisition function is introduced in another point of view to get a new sample value as implement. The new acquisition function could be very different from the previous one so as to optimize the function from other parts. when updating the sample points and reward, both acquisition functions should be called to ensure it up to date.

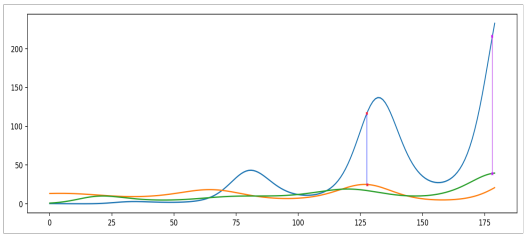

Figure 2: implement from another acquisition function

As shown in fig2. The orange line is the main acquisition function while the green line is the implement function. Suppose the red sample points is already duplicate with previous ones (in the same category), so the pink sample point from the implement function is chosen to use, in this illustration, it jumps out the local optimal area, and find the better rewards.

2 Experiments and Results

We applied the method mentioned above for the Dataset from AIAC 2021. The Data is test on Data-2 and Data-30 locally.

	Normalized Score	Normalized Time
Ensemble Penalization	1	1
Ensemble	0.79	0.968
Penalization	0.8	1.028

Table 1: Normalized Score and Time

As shown in the table, both penalization and ensemble strategy can provide about 20% improvements for the category search space while it doesn't consume too much time. Both strategy can provide equal or better performance.

3 Conclusion

We proposed a common method for category search space hyper parameter optimization using Bayesian optimization algorithm. The method is to using Bayesian optimization in real space other than category space to abide the continuous. Then add penalization on the sample points to let these points being far away from each other as much as possible, in order to be not in the same category. We also proposed an ensemble method to improve performance, not like traditional ensemble strategy as averaging or voting on the acquisition function. We proposed an implement method. After entering the local optimum area, due to searching space is not continuous, duplicated optimization in the same category is meaningless, so when this happens, we introduce another acquisition function to optimize the function from other parts to enlarge the searching space, and jump out the local optimum area. After ablation experiments, both penalization strategy and the ensemble strategy can provide about 20% score improvement independently. And the strategy won't introduce negative effects because it only take effect when the sample point are within the same category or too close to each other. So it can provide equal or better performance for the category space by using Bayesian optimization algorithm.

REFERENCES

[1] *Eric Brochu, Vlad M. Cora and Nando de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning arXiv:1012.2599v1 [cs.LG] 12 Dec 2010*

[2] *Javier Gonz´alez , Zhenwen Dai , Philipp Hennig , Neil Lawrence. Batch Bayesian Optimization via Local Penalization arXiv:1505.08052v4 [stat.ML] 15 Oct 2015*

[3] Eduardo C. Garrido-Merch´an, Daniel Hern´andez-Lobato, Dealing with Categorical and Integer-valued Variables in Bayesian Optimization with Gaussian Processes arXiv:1805.03463v2 [stat.ML] 22 May 2018

[4] Yang Li† , Yu Shen†§, Wentao Zhang† , Yuanwei Chen† , Huaijun Jiang†§, Mingchao Liu† Jiawei Jiang‡ , Jinyang Gao◦ , Wentao Wu∗ , Zhi Yang† , Ce Zhang‡ , Bin Cui† OpenBox: A Generalized Black-box Optimization Service arXiv:2106.00421v2 [cs.LG] 6 Jun 2021

[2] *Alexander I. Cowen-Rivers, Wenlong Lyu∗, Rasul Tutunov. Zhi Wang,, Antoine Grosnit, Ryan Rhys Griffiths, Alexandre Max Maravel, Hao Jianye, Jun Wang, Jan Peters, Haitham Bou-Ammar, An Empirical Study of Assumptions in Bayesian Optimisation, arXiv:2012.03826v5 [cs.LG] 24 Sep 2021*