Hi Nick,

Today, I want to explain you the idea of hyper parameter optimization and the algorithms we used for the optimization: grid search, random search and Bayesian methods. I will mainly focus on the Bayesian method.

The aim of hyper parameter optimization in machine learning is to find the hyper parameters of a given machine learning algorithm that return the best performance as measured on a validation set metric. The problem with this optimization is that evaluating the object function (evaluate the performance) to find the score is extremely expensive.

Grid search and random search are two algorithms that slightly better than manual tuning because we set up a grid of model hyper parameters and run the train-predict -evaluate cycle automatically. However, even these methods are relatively inefficient because they do not choose the next hyper parameters to evaluate based on previous results. Grid and random search are **completely uninformed by past evaluations**, and as a result, often spend a significant amount of time evaluating "bad" hyper parameters. It would **keep searching across the entire range of numbers of estimator**s even though it's clear the optimal answer lies in small region.

Now you must want to ask why don't we take the past evaluations into account and find the most promising hyper parameters. This is actually the idea of Bayesian method. The general idea of Bayesian method is to build a probability model of the objective function and use it to select the most promising hyper parameters to evaluate in the true object functions and update the probability model iteratively. In the literal, the probability model is called a surrogate model for the objective function and is represented as p(y|x). The surrogate is **much easier to optimize** than the objective function and Bayesian methods work by finding the next set of hyper parameters to evaluate on the actual objective function by selecting hyper parameters that perform best on the surrogate function. Then we use the new results to update the surrogate model. After a few iterations the surrogate model could be a well estimation of the true object functions. The algorithms can be summarized as the following four steps.

1) Build a surrogate probability model(usually Gaussian Process) of the objective function
2) Find the hyper parameters that perform best on the surrogate
3) Apply these hyper parameters to the true objective function
4) Update the surrogate model incorporating the new results
5) Repeat steps 2–4 until max iterations or time is reached

I hope you get a rough idea about the Bayesian Optimization and can apply this algorithm in your tasks.

Have a good day.

Best,
Zhixuan