Hi Nick,

Currently, to tune hyperparameters for unsupervised learning the industry use Grid search, Random search and Bayesian Optimization.

Grid search is easy to understand and implement. For example, our hyperparameter A has 2 options, hyperparameter B has 3 options, and hyperparameter C has 5 options, then all of our hyperparameter combinations have 2 * 3 * 5 That is 30 combinations, we need to train these 30 combinations and find the best solution among them. For continuous values, we also need to sample at equal intervals. In fact, it’s not necessary to train all these 30 combinations to get a global optimal solution, and the amount of calculation is very large, and it is easy to get exploded. Obviously, It is not an efficient parameter tuning method.

The industry-recognized Random search’s effect is better than Grid search. Random search is actually a random search. For example, in the previous scenario, there are 2 options for A, 3 options for B, and 5 options for C. We can randomly combine values in A, B, and C into a new hyperparameter combination to train. Although there are random factors, random search may appear to have particularly poor results or may have particularly good results. Generally, the maximum result will be greater when the number of attempts is the same as Grid search. Of course, the variance is also larger, but this does not affect the final result. It can be optimized when Random Search is implemented, filter the possible combination of hyperparameters that have appeared to avoid repeated calculations.

In fact, Grid search and Random search are very common and effective methods. But when the computing resources are limited, maybe they are not better than modeling engineers. The Bayesian Optimization is very probably better than modeling engineers. First of all, the Bayesian optimization uses the Bayesian formula: Posterior= Likelihood * Prior. It requires that there are already several sample points, and the Gaussian process regression (assuming that the hyperparameters is the joint Gaussian distribution) Calculate the posterior probability distribution of the first n points and get the expected mean and variance of each hyperparameter at each point. The mean represents the final expected effect of this point. The larger the mean is, the larger final expectation of the model is. The variance represents the uncertainty of the effect of this point. The larger the variance is, the more uncertain whether this point is likely to reach the maximum expectation is.

The Bayesian optimization procedure is as follows.

```
for $n = 1, 2, . . .$ do
    select new $x_{n+1}$ by optimizing acquisition function $\alpha$ $x_{n+1} = \arg\max_x \alpha(x; D_n)$
    query objective function to obtain $y_{n+1}$
    augment data $D_{n+1} = \{D_n, (x_{n+1}, y_{n+1})\}$
    update statistical model
end for
```

If you have any further question, please let me know.

Best,

Jack