towardsdatascience.com

Hyper Parameter Tuning with Randomised Grid Search - Towards Data Science

Sun Weiran

3–4 minutes

Why is Randomised Grid Search better and how to use it?

Jewel Changi Singapore — beautiful isn't it?

> Randomised grid search is very useful in finding near-optimal hyper parameters for any machine learning models.
>
> **Rules of thumb:** with 60 iterations, 95% of the time, best 5% sets

of parameters can be found, regardless of grid size.

What is Grid Search?

Grid search is a brutal search algorithm to find the optimal set of parameters.

Let’s assume we are building a binary random forest classification model and will use AUC to evaluate it. After all the data processing and feature engineering, we need to tune the parameters to reach maximum accuracy.

After some primary analysis, we decide to select parameters from ranges of values, for example:

Example: parameter grid

This is a **parameter grid** which consists of **all the possible combinations of parameters.** 36 (2*3*3*2) potential sets of parameters can be drawn from this grid.

Grid search will compute accuracy score resulted from each set of parameters and select the one yielding best results. In the below example, grid search will output set 4 since it yields highest AUC.

Example: AUC score for each parameter set in the grid

Why Grid Search fails?

Grid search guarantees finding the optimal parameter set in the grid. However, **size of the grid explodes exponentially with more parameters or more choices for parameters adding to the grid.**

Using above example, let’s tune one more parameter ‘min_impurity_decrease’ with 5 candidates and add one choice for all the existing parameters. By widening search area a little bit, the grid expands to 720 (3*4*4*3*5) sets, 20 times larger than the origin grid! If the original grid search takes 20 minutes, this grid will cost nearly 7 hours to complete.

Hence, when we are tuning complex models with many parameters, grid search is not very feasible in many situations.

Randomised Grid Search is a great alternative to Grid Search

Instead of exhaustive searching, **randomised grid search randomly draws one set of parameters from the grid to compute accuracy score at every iteration.** Number of iterations can be adjusted and the best set in all iterations will be the output.

Apparently, randomised grid search consumes less computation power as number of iteration is controlled. However, how many iterations are needed to find a good enough parameter set?

> **Rules of Thumb:** with 60 iterations, 95% of the time, best 5% sets of parameters can be found, regardless of grid size.

Simple Math Proving:

1. Let's denote *n* as the number of iterations.
2. After *n* iterations, probability of none of the best *p*% sets of parameters is drawn from the grid is $(1-p)^n$
3. Therefore, probability of at least one of the best *p*% sets of parameters is drawn from the grid after *n* iterations is $1-(1-p)^n$
4. If we want to have the best 5% sets of parameters with 95% confidence, solving the above equation, $1-(1–5\%)^n = 95\%$, n ≈ **59**

   From this proving, we can also conclude that the recommended number of iterations is independent of grid size.

Checklist for number of iterations for randomised grid search

For readers' convenience, please refer to the following table for the recommended numbers of iterations for different confidence and performance thresholds

top x%	Confidence	no. of iterations
1%	99%	460

1%	95%	300
3%	99%	150
3%	95%	100
5%	99%	90
5%	95%	60
7%	99%	65
7%	95%	40
10%	99%	45
10%	95%	30

Recommended number of iterations

Thanks for reading!