AML A1 - Hyperparameter Optimization

Irina-Mona Epure

1 Introduction

Big tech companies such as LaTech often have large amounts of customer-generated data, which they could leverage through predictive machine learning techniques. However, every machine learning algorithm has multiple hyperparameters that can influence its performance. Optimizing these hyperparameters manually is a tedious task, but, fortunately, there are multiple techniques for automating this process.
This report seeks to compare the efficiency of two of the most popular and easy to implement hyperparameter optimization techniques: grid search and random search. As an initial hypothesis of this experiment, it is expected that random search will perform better than grid search, as it has been shown in various specialty literature [2].

2 Background

This section provides some theoretical background into the two hyperparameter optimization techniques analyzed in this report. Additionally, some information is given about a third technique called Bayesian optimization.
The first optimization technique, **grid search** is based on selecting a discrete set of values to be evaluated for each hyperparameter and then training and evaluating a model for each possible combination of these hyperparameters [1].
Random search is based on simply providing some ranges for hyperparameter definition and then randomly selecting a value within that range. Alternatively, a set of possible values can be declared for categorical and conditional hyperparameters to be randomly generated from. Random search has been shown to be more efficient than grid search [1, 2].
Finally, **Bayesian optimization** has been shown to have better accuracy and be faster than the two previous methods mentioned. This technique creates a probabilistic model of the function whose minimum we are trying to find. It then uses this model to decide on new points to evaluate the function at and reduces uncertainty. This method is better at evaluating complex functions and can do so in fewer iterations [1, 3, 4].

3 Experiments

For this experiment, five different datasets were provided, consisting of the input data described by multiple features, and the output data in the form of a label from 0 to 9 for each record. The

databases have different sizes: 200, 800, 1500, 3000, and 5000 records. Each of them is split in a train, validation, and test set, representing 70%, 20%, and 10% of the entire dataset, respectively. A classification algorithm was trained and optimized for these datasets. The Support Vector Machine provided by sklearn.svm.SVC was used as a classifier. The hyperparameters optimized for it were C (regularzation parameter), *kernel*, *degree*, and *gamma*. The values considered for them were the following:

- C took real values in the set 0.01, 0.1, 1, 10, 100, 1000 for grid search and in the range [0.01, 1000] for random search,
- *kernel* took categorical values in the set 'linear', 'poly', 'rbf', 'sigmoid',
- *degree* took integer values in the set 1, 3, 5, 7, 10, 15, 20 for grid search in the range [1, 20] for random search,
- *gamma* took categorical values in the set 'auto', 'linear'.

The accuracy of the optimal hyperparameter configurations found for the five datasets were measured with *sklearn.metrics.accuracy_score* and included in 1. The entirety of the code used for the implementation of this experiment is linked in the footnotes [1].

Table 1: The accuracy score obtained with the two optimization methods on the test set of each dataset.

Dataset	**Accuracy with grid search**	**Accuracy with random search**
0	0.7	0.7
1	0.725	0.725
2	0.7067	0.752
3	0.76	0.9933
4	0.75	0.75

4 Discussion and conclusions

From the accuracy scores obtained during this experiment, we can see that grid search and random search perform just as well on datasets 0, 1, and 4. However, random search has a higher accuracy for datasets 3 and 4. For the latter, it actually has a massive improvement in accuracy.

As a conclusion to this experiment, it appears that random search performs as well or better than grid search in the case of any of the five datasets. Therefore, it is the better hyperparameter optimization method for performing Support Vector Machine classification on the given datasets.

[1] https://drive.google.com/file/d/1zOMbdeA-xsMLlSqMbTCdNlDA7KOd52yT/view?usp=sharing

References

[1] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[2] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[3] Martin Pelikan, David E Goldberg, Erick Cantú-Paz, et al. Boa: The bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99*, volume 1, pages 525–532. Citeseer, 1999.

[4] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.