

Wednesday: Hyperparameter Tuning

Agenda

5 min: Overview

50 min: Suggested Resources

1 hr 5 min: Exercise

Specific Learning Outcomes

- I can perform several feature engineering methods before applying logistic regression modelling.
- I can understand the meaning of hyperparameters.
- I can explain how hyperparameters differ from parameters.
- I can understand the importance of using hyperparameters.
- I can optimize and tune hyperparameters using Grid Search and Randomized Search.

Main Resources

In this session, we will focus on learning **Hyperparameter Tuning**. This is the automatic optimization of the hyperparameters of a machine learning model. These hyperparameters are configuration variables that are external to the model and whose values cannot be estimated from data. That is to say, they can't be learned directly from the data in standard model training. They're almost always specified by the machine learning engineer prior to training.

For simpler understanding, we might also define a hyperparameter as ***a parameter whose value is set before the learning process begins*** with the goal of searching across various **hyperparameter** configurations to find a configuration that results in the best performance.

Such hyperparameters are the second type of parameters that belong to a machine learning model. The first type of parameters are the parameters that are learned through a machine learning model. These first type of parameters are internal to the model and their value can be estimated from data.

Below are some examples of model hyperparameters;

1. The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
2. The C and sigma hyperparameters for support vector machines.
3. The k in k-nearest neighbors.

Why are hyperparameters important?

- They help define higher-level concepts about the model, such as its complexity and ability to learn.
- With the right parameters, they can eliminate the chances of overfitting and underfitting

These hyperparameters might address model design questions such as:

- What degree of polynomial features should I use for my linear model?
- What should be the maximum depth allowed for my decision tree?
- How many trees should I include in my random forest?

How do we perform hyperparameter tuning?

There are several approaches to hyperparameter tuning:

1. Manual: Select hyperparameters based on intuition/experience/guessing, train the model with the hyperparameters, and score on the validation data. Repeat the process until you run out of patience or are satisfied with the results.

2. Grid Search: Try out a bunch of hyperparameters from a given set of hyperparameters, and see what works best. Set up a grid of hyperparameter values and for each combination, train a model and score on the validation data. In this approach, every single combination of hyperparameters values is tried which can be very inefficient! It can be easily parallelized however, it's computationally expensive.

3. Random search: Try out a bunch of random hyperparameters from a uniform distribution over some hyperparameter space, and see what works best. Set up a grid of hyperparameter values and select random combinations to train the model and score. The number of search iterations is set based on time/resources. It can be easily parallelized. Just as simple as grid search, but a bit better performance. While it gives better performance than Grid Search, it is still just as computationally intensive.

4. Automated Hyperparameter Tuning: Use of methods such as Gradient Descent, Bayesian Optimization, or evolutionary algorithms to conduct a guided search for the best hyperparameters.

Suggested Resources

- Demystifying Hyper - Parameter Tuning. [[Link](https://towardsdatascience.com/demystifying-hyper-parameter-tuning-acb83af0258f) [_\(https://towardsdatascience.com/demystifying-hyper-parameter-tuning-acb83af0258f\)](https://towardsdatascience.com/demystifying-hyper-parameter-tuning-acb83af0258f).]
- Hyperparameter Tuning. [[Link](https://www.geeksforgeeks.org/ml-hyperparameter-tuning/) [_\(https://www.geeksforgeeks.org/ml-hyperparameter-tuning/\)](https://www.geeksforgeeks.org/ml-hyperparameter-tuning/).]
- Hyperparameter Optimization in Machine Learning Models. [[Link](https://www.datacamp.com/community/tutorials/parameter-optimization-machine-learning-models) [_\(https://www.datacamp.com/community/tutorials/parameter-optimization-machine-learning-models\)](https://www.datacamp.com/community/tutorials/parameter-optimization-machine-learning-models).]
- Limitations of Hyperparameter Tuning. [[Link](https://sigopt.com/blog/common-problems-in-hyperparameter-optimization/) [\(https://sigopt.com/blog/common-problems-in-hyperparameter-optimization/\)](https://sigopt.com/blog/common-problems-in-hyperparameter-optimization/).]

Exercise

- Python Programming: Hyperparameter Tuning (Logistic Regression) [[Link \(https://colab.research.google.com/drive/1ggX6Fy5XKCQLBCafB4TTzRGCVTEZ0AGW?usp=sharing\)](https://colab.research.google.com/drive/1ggX6Fy5XKCQLBCafB4TTzRGCVTEZ0AGW?usp=sharing)]

The following two exercises contain hyperparameter tuning problems for other supervised learning algorithms. These should provide you with an idea of how the two methods Grid Search and Random search can be used for optimization. Once we go through the respective algorithms, we will work out those problems.

- Python Programming: Grid Search. [[Link \(https://colab.research.google.com/drive/1yFcLacRxAHouQzqkishK0prAKd6gQGT1?usp=sharing\)](https://colab.research.google.com/drive/1yFcLacRxAHouQzqkishK0prAKd6gQGT1?usp=sharing)]
- Python Programming: Random Search. [[Link \(https://colab.research.google.com/drive/1vwFO9hHu5MKii_kWtnaiU05yrql4Bypt?usp=sharing\)](https://colab.research.google.com/drive/1vwFO9hHu5MKii_kWtnaiU05yrql4Bypt?usp=sharing)]

"Data matures like wine, applications like fish." ~ James governor