Hyperparameter Optimization

Milana M. Wolff

April 23, 2024

1 Introduction

In this assignment, we optimize relevant hyperparameters for a small selection of classification models using the wine quality dataset. This widely used dataset contains a variety of physicochemical input features, such as wine density and acidity, along with expert ratings for red Vinho Verde wines. We approach the hyperparameter optimization problem by selecting three classifiers with competitive performance under default hyperparameter configurations: Ridge, bagging, and random forest. We then conduct hyperparameter optimization with grid search cross-validation and nested resampling (3 outer folds, 10 inner folds per hyperparameter configuration).

2 Dataset Description

The dataset used for this assignment contains physicochemical quantitative input features and sensory quantitative output features (i.e., an expert wine score) for the red variant of the Portuguese "Vinho Verde" wine [1]. The dataset includes 1599 observations and eleven input features, including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content. According to the UC Irvine Machine Learning Repository website, "the classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones)", with a total of 1319 observations rated as 5 or 6 and a mere 28 observations rated with the highest and lowest scores (3 and 8) [2]. This robust dataset includes no missing values to be imputed. We use the eleven listed input features to predict the wine quality measurement as the target during hyperparameter optimization.

3 Experimental Setup

We use Python 3.10.12 (GCC 11.4.0) in a Jupyter/interactive Python notebook on Google Colaboratory, as well as a functionally identical pure Python file

Classifier	Ridge	Bagging	Random Forest
Numerical Hyperparameter Ranges	alpha: (1.0, 5.0) tol: (0.0001, 1.0) max_iter: (100, 100000)	n_estimators: (100, 10000) max_samples: (0.01, 1.0) max_features: (0.01, 1.0)	n_estimators: (100, 10000) max_depth: (2, 10)
Numerical Hyperparameter Priors	log-uniform (all)	log-uniform (all)	log-uniform (all)
Categorical Hyperparameters	'svd', 'cholesky', 'lsqr', 'sparse_cg'		'gini', 'entropy', 'log_loss'

running on the Beartooth/Teton cluster (default Python version 3.7.16 (GCC 11.2.0)) for computational efficiency. After importing the `scikit-learn` package, we use Ridge, bagging, and random forest classifier models. We optimize the alpha, tolerance, solver, and maximum iteration hyperparameters for the Ridge classifier models; the maximum samples (as a proportion of total samples), number of estimators, and maximum feature hyperparameters for the bagging classifier models; and the number of estimators, splitting criterion, and maximum tree depth for the random forest classifier models.
The ranges and priors for numerical hyperparameters and the options for categorical hyperparameters are tabulated below.

We use a Bayesian search approach with the `BayesSearchCV()` method provided by `scikit-learn`. We load the wine data using the `pandas` library and use all eleven features for classification without additional pre-processing steps. For each machine learning algorithm listed above, we use a nested resampling approach with a three-fold outer cross-validation and ten-fold inner cross validation. We use balanced accuracy as a scoring criterion in the Bayesian search.

4 Results

For each outer loop of the cross-validation, we report the average results of the inner 10-fold cross validation, the generalization score, the balanced accuracy on the outer CV test dataset, and the best score on the left-out data for the inner CV yielding the best estimator. These results are tabulated for the Ridge, bagging, and random forest classifiers below. The Bayesian optimization approach yielded the same results for each outer loop.

The hyperparameter configuration for the best estimator for the Ridge classifier was: `alpha`=1.267774542235981, `max_iter`=22377, solver='cholesky' and

Classifier	Ridge	Bagging	Random Forest
Generalization Score	0.236 ± 0.022		
Score on Left Out Data (inner CV)	0.2318		
Score on Test Data (outer CV)	0.2322		

`tol`=0.002318281928932147. Notably, the `tol` hyperparameter has no effect if the Cholesky solver is used.

5 Code

`https://github.com/COSC5557/hyperparameter-optimization-mwolff2021-1`

References

[1] In: (). URL: `http://www.vinhoverde.pt/en/`.

[2] Paulo Cortez, A. Cerdeira, F. Almeida, et al. "Wine Quality". In: (2009). DOI: https://doi.org/10.24432/C56S3T.