Hyperparameter Optimization

COSC5557 – Practical Machine Learning

William Baumchen

12/8/2023

1 - Introduction

When considering solving a problem of some kind using machine learning, once some amount of data has been sourced and some machine learning model, or set of models, have been chosen, it's important to consider hyperparameter optimization. Hyperparameter optimization is the optimization of the hyperparameters, or meta-parameters, that define the behavior of the model as it fits to the given data. As such, by conducting hyperparameter optimization, often a much more accurate model can be found. In this report, several different machine learning models were developed and their hyperparameters optimized and then evaluated using the white wine quality data set, and from those models the 'best' was chosen.

2 – Dataset Description

The dataset used is the white wine quality dataset, containing 4,898 observations, with 11 features, all of which are real numbers taken from a continuous range. There is one 'target' feature, with seven possible classes. There are no missing values in the observations and target feature.

3 – Experimental Setup

In this exercise all computation was done utilizing MATLAB, more specifically the Statistics and Machine Learning Toolbox. Firstly, the dataset was split for sampling; a fifth of the original dataset was set aside for outer loop testing, and the remaining 80% of the data was kept for training. Then, two sets of hyperparameter and model selection optimization were carried out, one using classification models, and the other using regression models. Both models utilized constant-repartitioning 5-fold cross-validation and utilized Bayesian optimization with the objective function being the cross-validation error, different metrics being used in the two different cases. Both optimizations were given a budget of 300 iterations, about three times the default for the two functions used, in order to make sure the optimization could full explore the parameter space. For the classification optimization, the cross-validation classification error was utilized, and the models tested along with their hyperparameters are shown in Table 1, using the fitcauto function in MATLAB [1]. For the regression optimization, the objective function used was the cross-validated mean square error, and the models tested along with their hyperparameters are shown in Table 2, using the fitrauto function in MATLAB [2]. For both cases, once the optimal models and associated hyperparameters were found, 'baseline' models were created using the same type of model as the optimal models. Then, each model was tested using the above "Test" dataset, to find how well this optimization worked to improve the given model. Once all the error values and the predictions were found, the results were plotted confusion charts to graphically show the performance of the models.

Table 1 – Classification Models & Hyperparameters

"ensemble"	Method, NumLearningCycles, LearnRate, MinLeafSize
"kernel"	KernelScale, Lambda, Standardize, Coding (for three or more classes only)
"knn"	Distance, NumNeighbors, Standardize
"linear"	Lambda, Learner, Coding (for three or more classes only)
"nb"	DistributionNames, Standardize, Width
"net"	Activations, Lambda, LayerSizes, Standardize
"svm"	BoxConstraint, KernelScale, Standardize, Coding (for three or more classes only)
"tree"	MinLeafSize

Table 2 – Regression Models & Hyperparameters

"ensemble"	Method, NumLearningCycles, LearnRate, MinLeafSize
"gp"	Sigma, Standardize
"kernel"	Epsilon, KernelScale, Lambda, Standardize
"linear"	Lambda, Learner
"net"	Activations, Lambda, LayerSizes, Standardize
"svm"	BoxConstraint, Epsilon, KernelScale, Standardize
"tree"	MinLeafSize

4 – Results

After completing the optimization, which for the classification optimization took around 2090 seconds, and the regression optimization 10923 seconds, it was found that the best performing model and hyperparameters for the classification model set was a multiclass support vector machine model, with the Coding (ECOC) being onevsone, using a BoxConstraint of 18.66, and a KernelScale of 0.92892. The best performing model and associated hyperparameters for the regression model set was an ensemble model using the LSBoost Method, a

NumLearningCycles of 300, and a MinLeafSize of 2. The resulting test error using each optimization's respective metrics are shown below in Table 3.

Figure 1 below shows the resulting confusion chart for the optimal classification mode, while Figure 2 shows the same for the optimal regression model.

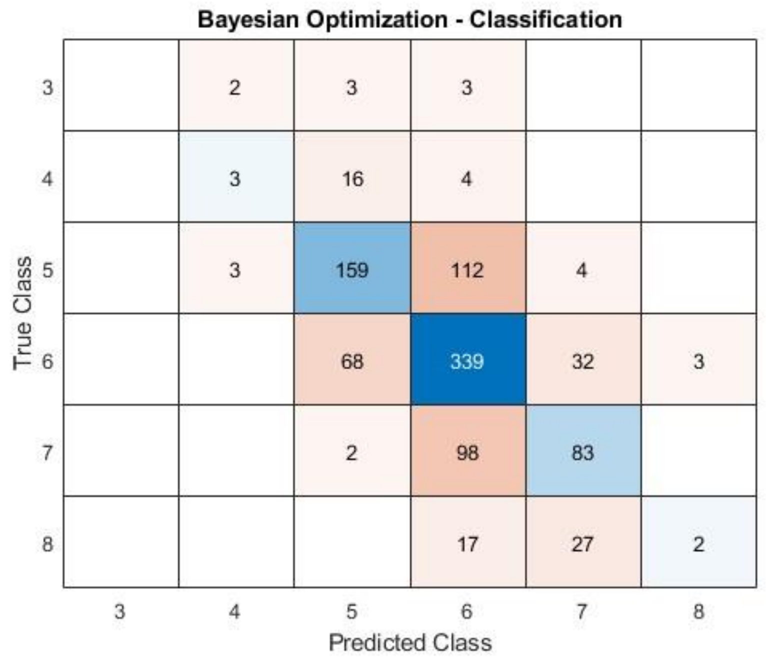

Figure 1 – Confusion Chart for Optimized Classification Model

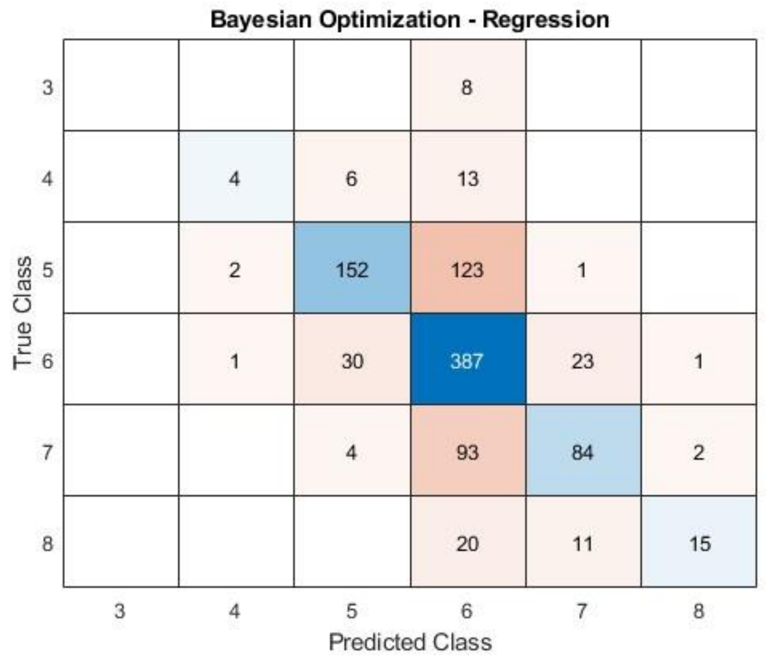

Figure 2 – Confusion Chart for Optimized Regression Model

Table 3 – Optimization Results, Comparison with Baseline models

Model	Metric on Test Data
Optimal Model Classification Loss	0.3436
Optimal Model Regression MSE	0.45088
Baseline Classification Loss	0.48228
Baseline Regression MSE	0.55609

As is clear, examining the given results, both of the optimal models outperformed the 'baseline' modes, demonstrating clearly that the optimization of hyperparameters allows for a clear increase in accuracy when fitting some model to a set of data.

References

[1] The MathWorks, Inc. . (n.d.). *fitcauto*. Automatically select classification model with optimized hyperparameters - MATLAB. https://www.mathworks.com/help/stats/fitcauto.html?s_tid=doc_ta

[2] The MathWorks, Inc. (n.d.). fitrauto. Automatically select regression model with optimized hyperparameters - MATLAB. https://www.mathworks.com/help/stats/fitrauto.html?s_tid=doc_ta