Machine Learning: Exercise Set VII (b)

Georgios Manos
csd4333

7 January 2023

1 Causal-Based Feature Selection

1.1 Implementation Details

I managed to implement both algorithms and wrapped them into a "select_features" function, which performs the forward-backward feature selection. The code is extremely slow due to the statistical test performed by the feature selection algorithm, something that could potentially be improved using threads, however for simplicity reasons I kept it as is.

Once again, I will be using standardization in order for the SVM classifier to converge. Of course, this is applied prior to the feature selection. Since we have standardization and feature selection on the preprocessing part, I will be using different preprocessing pipelines as different configurations. The configurations tried are shown and explained on the very top of the code. Essentially, for each classifier we have $P \cdot C$ different configurations to try, where P is the number of preprocessing pipelines and C is the number of different classifier configurations. For this exercise, I will be using 3 different preprocessing pipelines (1 for each a value of FS, always standardized data) for SVM with 6 different sets of hyperparameters (therefore 18 different SVM configurations to try) and 6 different preprocessing pipelines (with and without standardization, for all 3 different a values of FS) for Random Forest classifier with 4 different sets of hyperparameters (therefore 24 different RF configurations to try).

If you run the code, you will be seeing its progress printed on the terminal, along with the results of each configuration. I added skipping feature selection as a function argument instead of a configuration, to simplify rerunning the classifiers without feature selection. Otherwise, it would be best if that was also a preprocessing configuration argument.

Finally, I implemented stratified train-test splitting my self. I basically split the data into 2 classes, split each sub-dataset in 80-20 training-test set and merge both training sets into one training set (and the test sets respectively).

1.2 Results

1.2.1 Model selection with Feature Selection

With Feature Selection, the best configuration is SVM Classifier, with FS Significance Threshold a = 0.05, so the selected (standardized) features are 3, 13, 14, 19, 21, 26, 29, with C = 5 and linear kernel (gamma is ignored in linear kernels) with an average AUC score of 0.993 and hold-out set AUC 0.9977*.

1.2.2 Model selection without Feature Selection

To rerun the analysis without Feature Selection, change the flag in line 15 to True from False.

Without Feature Selection, the selected best configuration again is the SVM classifier (always with standardization, and FS threshold is ignored this time), with C = 1 and linear kernel. The estimated performance AUC score is 0.992 and the hold-out AUC is 0.999, which is slightly better to the one with feature selection. We can see that the performance drop resulting from feature selection is really small, but also the subset of features chosen by our feature selection algorithm (7 out 30 features) is relatively small. To further investigate the choices of our feature selection algorithm, as well as the effectiveness of our statistical test, we would need to dive deeper into our dataset and explore the columns.

1.2.3 Feature Selection only

Performing forward-backward feature selection on all data (with and without standardization) with a = 0.05 selects features 6, 15, 21, 23. The backward selection is already performed, so performing again the backward phase on set S results on the exact same set. This was expected, as the FB algorithm with a given significance threshold is known to always result on the same set. I also repeated the experiment for a=0.005, and the resulting set was 21, 23, 27, again on both S and S'. It is interesting how we got a different feature for lower a value, noting that reducing the significance threshold only results in less features being chosen overall but also different features marked as important.

*Once again, we see that the performance estimation is conservative as the hold-out set AUC is better than the CV estimation