CS5014, P2 - Classification

170008773

19th April 2018

1 Introduction

For this assignemdnt a classification system to classify certain colours from their optical reflectance spectroscopy readings was to be implemented.

2 The learning pipeline

Data description There were 180 samples in the binary set and 450 samples in the multiclass set. Both sets also had 921 features. All of the features consist of some intensity reading from the light reflecting of the surface at certain wavelengths.

2.1 Data preperation

data seperation Since the data dit not in seperate training and testing sets, it had to be seperated frist, into a testing and a training set. Even though the data contains a lot of features, the number of samples was relatively low. This meant that not a lot of data could be set aside for testing, but it still had to be large enough that the results would be representable. Eventually 25% of the data was set aside for testing. This was ligtly below a common rule of thumb of setting aside 30% of the data for testing.

Normalisation After the data was seperated, it was deemed favorable to normalise the data so that the scale wouldn't introduce additional biases duringg the rest of the process. A standard normalisation was applied to both the training and test data, according to the values of the training data.

2.2 Initial exploration

Getting a baseline A data set of these proportions is extremely hard to visualise effectively. Therefore a simple `LogisticRegression` classifier was used on the data to get a baseline accruacy. This was done to get a general sence of how complex the data was. If the accuracy of such a simple classifier with no aditional work was very high that would suggest that the data was not very complex and that some of the data could probably be pruned. If this was the case then models who could provide more information than just accurate classifications would have to be considered. On the other hand if this baseline was very low then this would have suggested that the data would have been very complex and more sopisticated methods with a bigger emphesis on accracy would have to have been considered. The logistic regression classifier achieved a `F1`

score of 1.0, meaning that it achieved perfect accuracy (for a deeper discussion of why `F1` was chosen see section **??**).

Visualisation Because of the number of features we elected to skip the visualisation step because it's impractical.

2.3 Feature selection

The fact that the baseline was this high means that we would need to judge the algorithms we would use on more than just their accuracy. We decided to focus on two other metrics besides accruacy: operation time and the ability to prune the feature set. This meant that that we would start out with the full feature set and use algorithms that have some mechanic available to tell us more about which features are important.

2.4 Selecting and training the model

2.4.1 Cross validation

We decided to select our model based on their performance metrics (to be discussed below) on k-fold cross validation. Because there is not a lot of data we elected to se $k = 5$ in this case, to make sure that the folds were still big enough to yeild significant results.

2.4.2 Metrics

For our accuracy score we decided to use the F1 score insert reference. We would need more to go on than this however. This was revealed by our baseline. We therefore decided that we also would consider the total operation time (that includes both training and testing) . Here we elected to measure the sum instead of the average because of insert reason. Eventually we also used a rating for model selection which we calculated as $\frac{\mu_s}{T}$ where μ_s is the mean of the score across the cross-validation folds and T is the total amount of seconds the algorithm took across all the cross-validation folds. This measure might not be very good to use in general circumstances but here is is still useful because all of our models had similar accuraccy. We used it because this would allow us to select a method that might do slightly worse than the other models but do it orders of magnitude factor, which turned out to be the case.

2.4.3 Models

To select our model we started out with several algorithms: logisitc regression, gradient boosting, random forests, decision tree and AdaBoost. All of these have ways of descriminating amoungst featueres. All of the models above, except for logistic regression provide a native way to rank features insert reference. In the case of logistic regression we used Recrusive Feature Elimination (RFE)insert reference.

2.5 Evaluating the model

	Algorithm	Mean score	Time on full set	Important features	Mean score on reduced set	Time on reduc
3	Decision tree	1.0	0.060660	1	1.0	0.006051
4	Adaboost	1.0	0.065143	1	1.0	0.011564
0	Logistic regression	1.0	0.063668	460	1.0	0.026572
2	Random Forests	1.0	0.091240	10	1.0	0.076235
1	Gradient Boosting	1.0	1.370301	69	1.0	0.210093

2.6 Discussing the results

3 Conclusions

word count: