10/06/2018

Pattern Recognition - Homework 3

Author: Ali Berrada

Program: MAIA 2017/19

Notes:

Among the attached scripts, there is the file “extras-gridsearch-with-sklearn.py”. This is a second and extra implementation for problem 3.2 that leverages the sklearn APIs to find best parameters using a grid search cross validation requiring only a handful of coding lines.

Problem 3.1

For preparing the data I use the script “crossvalidation-data-gen.py” to make a 5-fold cross-validation.

The original data is standardized by mean removal and variance scaling technique.

Then the data is split into 5 subsets: 2 subsets for training set (40%), 2 subsets for validation set (40%), and 1 subset for test set (20%).

Five different combinations of these subsets are taken in a way that each fold serves as test set.

The script “p1.py” finds, for each type of kernel, the optimal *v* for each cross validation set by looking for the *v* value that gives the highest AUC score over the validation set. The model with optimal *v* is then run over the test set and the AUC scores for the 5 cross validation set are averaged and their standard deviation computed.

The results are as follows:

Kernel	Cross Validation	Optimal *v*	AUC (Validation set)	Avg AUC (Test set)	Std Deviation (Test set)
Linear	1	0.52	0.7703	0.7681	0.0038
	2	0.54	0.7694		
	3	0.55	0.7759		
	4	0.54	0.7681		
	5	0.54	0.7741		
Polynomial	1	0.42	0.8122	0.8103	0.0038
	2	0.43	0.8031		
	3	0.47	0.8094		
	4	0.47	0.8181		
	5	0.46	0.8134		
RBF	1	0.34	0.8434	0.8389	0.0026
	2	0.3	0.8356		
	3	0.4	0.8406		
	4	0.4	0.8463		
	5	0.43	0.84		
Sigmoid	1	0.71	0.7575	0.7555	0.0037
	2	0.72	0.7619		
	3	0.78	0.7603		
	4	0.74	0.7566		
	5	0.74	0.7581		

Discussion

It is seen that the radial basis function (RBF) is best representative of the problem and outperforms other kernels. The standard error is also the lowest with RBF which shows great consistency in the scores across the 5-folds.

RBF nonlinearly maps the features space into a higher dimensional space to optimize the finding of the separating hyperplane. It outperformed the linear kernel as it can handle the nonlinearity relation between the classes and attributes, while against the polynomial kernel, RBF has less hyperparameters (which influence the complexity of the model selection) which means that with increasingly growing data, RBF is able to represent more complex relationships whereas the polynomial saturates at certain point.

Problem 2.2

The best model is created using the script “p2.py” and saved as “mymodel.tkl”.

5-folds and 10-folds cross validation were tested. To reduce the run time for best parameter search, the only kernel considered here is RBF since it led to highest scores previously.

Results with 5-folds:

cv=1 - best_v=0.3 - auc=0.845 - whole_data_auc=0.8801 - new_data_auc=0.7414

cv=2 - best_v=0.39 - auc=0.8506 - whole_data_auc=0.8514 - new_data_auc=0.8021

cv=3 - best_v=0.38 - auc=0.8462 - whole_data_auc=0.8538 - new_data_auc=0.8056

cv=4 - best_v=0.32 - auc=0.8456 - whole_data_auc=0.8719 - new_data_auc=0.7465

cv=5 - best_v=0.4 - auc=0.8438 - whole_data_auc=0.8484 - new_data_auc=0.8058

Results with 10-folds:

cv=1 - best_v=0.27 - auc=0.845 - whole_data_auc=0.8995 - new_data_auc=0.6996

cv=2 - best_v=0.27 - auc=0.84 - whole_data_auc=0.9008 - new_data_auc=0.7074

cv=3 - best_v=0.34 - auc=0.845 - whole_data_auc=0.8672 - new_data_auc=0.7613

cv=4 - best_v=0.39 - auc=0.8562 - whole_data_auc=0.8514 - new_data_auc=0.8034

cv=5 - best_v=0.39 - auc=0.8637 - whole_data_auc=0.8512 - new_data_auc=0.8054

cv=6 - best_v=0.34 - auc=0.8375 - whole_data_auc=0.8676 - new_data_auc=0.7696

cv=7 - best_v=0.32 - auc=0.8662 - whole_data_auc=0.8785 - new_data_auc=0.7427

cv=8 - best_v=0.31 - auc=0.835 - whole_data_auc=0.8775 - new_data_auc=0.7405

cv=9 - best_v=0.22 - auc=0.8588 - whole_data_auc=0.913 - new_data_auc=0.6622

cv=10 - best_v=0.44 - auc=0.8425 - whole_data_auc=0.8446 - new_data_auc=0.8079

NB: “new data” is self-generated data for test purpose only.