Answer for Assignment 2

MSBD 5002 Data Mining Course
2020-04-06

Q1 Comparison of Classifiers

Decision Tree

We summarize the performance of decision tree with different parameters in this table (keep 3 significant digits)

Criterion	Max Depth	Accuracy	Precision	Recall	F1	Training Time
gini	5	0.369	0.392	0.368	0.327	0.0942s
gini	10	0.713	0.763	0.716	0.726	0.103s
gini	15	0.832	0.842	0.833	0.835	0.125s
gini	20	0.867	0.869	0.867	0.867	0.126s
gini	25	0.872	0.872	0.872	0.872	0.126s
entropy	5	0.501	0.561	0.501	0.498	0.086s
entropy	10	0.799	0.804	0.799	0.800	0.118s
entropy	15	0.874	0.875	0.875	0.875	0.132s
entropy	20	0.871	0.871	0.872	0.871	0.151s
entropy	25	0.874	0.874	0.874	0.874	0.157s

From this table, we can see that with the same parameter of max_depth, entropy always performs better than gini. It's obvious that the larger the max_depth, the better performance of the classifier with higher accuracy, precison, recall and F1 but also with longer time to train. The best accuracy that the clssifier can achieve is about 0.874.

KNN and RandomForest

We summarize the performance of different classifiers in the following table. Notice we tune the n_neighbors(number of neighbors to use for K nearest neighbors queries) in the KNN and adjust the n_estimators(number of trees in the forest) in RandomForest. Then we get 6 different classifiers.

Because RandomForest is an ensemble methods that it need to train multiple base classifiers to combine, it needs much more time to train compared with KNN (2.734s is 15X of 0.187s). More number of trees in the forest, it needs more training time but can achieve better performance. However, in KNN, the training time becomes smaller when n_neighbors becomes larger. The best n_neighbors is 5 and the best accuracy that KNN achieves is about 0.952 which is lower than the performance of RandomForest's best accuracy 0.962.

Classifier	Accuracy	Precision	Recall	F1	Training Time
KNN -> n_neighbors = 2	0.947	0.948	0.949	0.947	0.198s
KNN -> n_neighbors = 5	0.952	0.952	0.952	0.952	0.189s
KNN -> n_neighbors = 8	0.948	0.949	0.948	0.948	0.187s
RandomForest-> n_estimators = 50	0.957	0.958	0.958	0.958	0.932s
RandomForest-> n_estimators = 100	0.960	0.961	0.961	0.961	1.813s
RandomForest-> n_estimators = 150	0.962	0.963	0.963	0.963	2.734s

In a word, RandomForest classifiers reduces the variance and has better performnce than KNN but need more time to train.

Q2 Implementation of Adaboost

We consider from two sides to find all best base classifiers, firstly we consider this form of classifier,

$$C(x) = \begin{cases} +1, x < v \\ -1, x \geq v \end{cases}$$

With v satisfying $v\%0.5 == 0$, it's obvious when $v = 2.5$ or $v = 8.5$, classifier achieves the lowest error rate 0.3, only misclassfying 3 samples. We set the two classifiers as C_1 and C_2:

$$C_1(x) = \begin{cases} +1, x < 2.5 \\ -1, x \geq 2.5 \end{cases} \qquad C_2\,(x) = \begin{cases} +1, x < 8.5 \\ -1, x \geq 8.5 \end{cases}$$

Then we consider a classifier with this form,

$$C(x) = \begin{cases} -1, x < v \\ +1, x \geq v \end{cases}$$

Similarly, we can find that the classifier with $v = 5.5$ has the smallest error rate of 0.4. We denote it by C_3:

$$C_3(x) = \begin{cases} -1, x < 5.5 \\ +1, x \geq 5.5 \end{cases}$$

By applying Adaboost algorithm, in the end, we get a strong classifier $C^*(x)$

$$C^*(x) = sign[1.7380493449176364 * C_1(x) + 2.07683056968926 * C_2(x) \\ + 2.2742999172498486 * C_3(x)]$$

that can achieve 0 error of classification where

$$C_1(x) = \begin{cases} +1, x < 2.5 \\ -1, x \geq 2.5 \end{cases} \qquad C_2\,(x) = \begin{cases} +1, x < 8.5 \\ -1, x \geq 8.5 \end{cases} \qquad \mathrm{C}_3(\mathrm{x}) = \begin{cases} -1, x < 5.5 \\ +1, x \geq 5.5 \end{cases}$$

The classification result of $C^*(x)$ is [1, 1, 1, -1, -1, -1, 1, 1, 1, -1]

More detailed information, please see in the source code.