PROJECT REPORT: C S 487/519 Applied Machine Learning I

Project 3: Compare classifiers in scikit-learn library

Written by

Omid Jafari

Department of Computer Science

New Mexico State University

October 2018

Preprocessing of REALDISP dataset:

This is a time series dataset of gym activity sensors information. There are 17 persons (a.k.a. subjects), each doing several gym activities. There are X sensors for the activities. Timestamp of the activities along with the sensor information and the activity number, which corresponds to the name of the activity, for each subject are stored in a separate file (i.e. subject1_ideal.log). The structure of the columns in each file is like the following:

Column 1: Timestamp in seconds
Column 2: Timestamp in microseconds
Column 3-15: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S1
Column 16-28: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S2
Column 29-41: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S3
Column 42-54: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S4
Column 55-67: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S5
Column 68-80: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S6
Column 91-93: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S7
Column 94-106: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S8
Column 107-119: [AccX, AccY, AccZ, GyrX, GyrY, Gyr, GyrZ, MagX, MagY, MagZ, Q1, Q2, Q3, Q4] of sensor S9
Column 120: Label (see activity set)

The goal of the classification task is to predict the activity number given the sensor data.

In order to preprocess the dataset, we first have to remove the missing values and the timestamp information since they are not going to participate in the classification process. It is important to note that having the timestamp values in the dataset will result in a perfect overfitting which we have to avoid.

Finally, we have to merge all of the information for the subjects into one single dataset.

Accuracy of the prediction for different classifiers:

We used a randomly chosen 75% of the data as the training set and the rest as the testing set.

Parameters							
	Eta	Iters	Seed	Gamma	C	n	P
Digits	0.001	20	1	0.2	1	1	1
REALDISP	0.001	20	1	0.2	1	1	1

Accuracy		
	Digits	**REALDISP**
Perceptron	93.0958%	99.9168%
Linear SVM	97.3274%	99.9236%
Non-linear SVM	8.4633%	74.422%
Decision Tree	84.1871%	99.96%
KNN	97.1047%	100%
Logistic Regression	94.2094%	97.869%

Running time of the different classifiers:

We used a randomly chosen 75% of the data as the training set and the rest as the testing set.

Parameters							
	Eta	Iters	Seed	Gamma	C	n	P
Digits	0.001	20	1	0.2	1	1	1
REALDISP	0.001	20	1	0.2	1	1	1

Running time (ms)				
	Digits		**REALDISP**	
	Training	**Prediction**	**Training**	**Prediction**
Perceptron	138	24	414132	0
Linear SVM	30	12	1360015	267
Non-linear SVM	322	61	6217501	36
Decision Tree	13	0	354072	0
KNN	3	50	45472	9558
Logistic Regression	149	9	524785	46

DecisionTreeClassifier code analysis:

Four strategies that the SKlearn library is using to pre-prune and post-prune are:

- Max-depth: The maximum depth of the tree. This value is a parameter defined in tree.py line 86.
- Min-samples-split: The minimum number of samples required to split an internal node. This value is a parameter defined in tree.py line 100.
- Min-samples-leaf: The minimum number of samples required to be at a leaf node. This value is a parameter defined in tree.py line 101.
- Min-impurity-decrease: A node will be split if this split induces a decrease of the impurity greater than or equal to this value. This value is a parameter defined in tree.py line 106.