Machine Learning (Assignment 4)

by

Ritesh Agarwal

Q1.) Radial basis function is implemented using Gaussian kernels. Three different bandwidths are tried and compared with standard logistic regression. The results for the same are in the tables below:

For bandwidths = 1, 0.5, .1 we have the following results:

Radial Basis Function('k': 10, 's': 1.0)	43.81111111	0.323505694
Radial Basis Function('k': 10, 's': 0.5)	39.58	1.021744246
Radial Basis Function('k': 10, 's': 0.1)	31.5	1.829796208

Radial Basis Function('k': 20, 's': 1.0)	38.908	1.140340356
Radial Basis Function('k': 20, 's': 0.5)	31.64	1.267352534
Radial Basis Function('k': 20, 's': 0.1)	25.68	1.071631973

Radial Basis Function('k': 50, 's': 1.0)	27.472	0.949941454
Radial Basis Function('k': 50, 's': 0.5)	26.49333333	0.60059305
Radial Basis Function('k': 50, 's': 0.1)	24.09876813	0.50789812

Now as the bandwidth decreases the error goes down but it takes a long time to run. So, there is a trade-off between execution time and accuracy, which must be chosen depending on the data set.

Accuracy also increases with increase in the centers.

As for Logistic Regression here ate results for two different splits:

Logistic Regression	24.33888889	0.220124394
Logistic Regression	23.592	0.509008641

As we can see Logistic Regression performs better than RBF and is faster than RBF as well.

Note: Run the code 'script_classify.py' to get the results

Q2.) I have taken banknote authentication dataset from UCI repository. Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.

Attribute Information:

1. variance of Wavelet Transformed image (continuous)
2. skewness of Wavelet Transformed image (continuous)
3. curtosis of Wavelet Transformed image (continuous)
4. entropy of image (continuous)
5. class (integer)

The dataset is included in the 'dataset' folder and is named 'banknote.csv'. Three learning methods implemented for this dataset are: Naïve Bayes, Logistic Regression and Neural Network. Now, I have applied multiple splits, t-test statistical significance and different k for cross validation. Results for the same are in the tables below for different meta parameters:

Without cross validation but with different splits:

train = 20%, test = 80%, runs = 100	Average Error	Standard Deviation
Naïve Bayes (columnones = False)	14.34545455	1.36203655
Logistic Regression	1.381818182	0.431773923
Neural Network (epochs = 100, step_size = .01, nodes = 32)	3.945454545	0.765260049

train = 40%, test = 60%, runs = 100	Average Error	Standard Deviation
Naïve Bayes (columnones = False)	16.13138686	0.989121068
Logistic Regression	1.058394161	0.356230704
Neural Network (epochs = 100, step_size = .01, nodes = 32)	2.226277372	0.786210746

train = 60%, test = 40%, runs = 100	Average Error	Standard Deviation
Naïve Bayes (columnones = False)	16.67883212	1.321326119
Logistic Regression	0.948905109	0.310159032
Neural Network (epochs = 100, step_size = .01, nodes = 32)	1.715328467	0.559604528

train = 80%, test = 20%, runs = 100	Average Error	Standard Deviation
Naïve Bayes (columnones = False)	16.17647059	1.03813131
Logistic Regression	0.955882353	0.157872972
Neural Network (epochs = 100, step_size = .01, nodes = 32)	2.132352941	0.624879227

With cross validation for different k values:

k=2, runs = 100	Average Error	Standard Deviation
Naïve Bayes (columnones = False)	15.11661808	0.634517332
Logistic Regression	0.714285714	0.236987901
Neural Network (epochs = 100, step_size = .01, nodes = 32)	2.186588921	0.419912527
k=3, runs = 100		
Naïve Bayes (columnones = False)	16.50199548	0.405463979
Logistic Regression	1.035071458	0.280880572
Neural Network (epochs = 100, step_size = .01, nodes = 32)	2.623845789	0.287970891
k=5, runs = 100		
Naïve Bayes (columnones = False)	15.3942933	0.738786946
Logistic Regression	0.874479098	0.221139475
Neural Network (epochs = 100, step_size = .01, nodes = 32)	1.7931785	0.386588095
k=10		
Naïve Bayes (columnones = False)	15.90394584	0.46619583
Logistic Regression	1.03438062	0.093846109
Neural Network (epochs = 100, step_size = .01, nodes = 32)	2.172220459	0.316954068

From the above tables we can see that Logistic regression performs really well compared to Naïve Bayes but Neural Network also performs well. Now Neural Network was tried with different step size parameters, epochs and number of hidden nodes, the output for the most efficient suited parameters is shown above. Also, k=5 performs the best for cross validation as taking a higher k or lower does not give us a very good result. t-test was performed on the two best algorithms for statistical significance and the threshold was taken to be 5% or 0.05 and if the p-value is less than the threshold the null hypothesis is rejected else the null hypothesis cannot be rejected. The two best algorithms are Logistic Regression and Neural Network.

Note : Run the code 'script_classify1.py' to get the results