Computer Science and Automation

Topics in Pattern Recognition

-Assignment–1-

Swapan Das
M.Tech ($2^{nd} year)CSA$
Indian Institute of Science, Bangalore

February 11, 2019

Contents

Chapter 1

:: Random Projection ::

1.1 Introduction

Implement random projection algorithm in projection.py file to reduce the dimension of all the three given data set(dolphins, pubmen and twitter). Let original dimension of a given data set is K. Using the random projection algorithm I created low dimensional data set for all the three given data set, for D = $\{2, 4, \cdots, \lceil K/2 \rceil\}$. See (Table : 1.1) for details.

1.2 Data set : twitter.csv

For twitter data set implement a twitter_test.py in twitter directory. Using this I read all sentence from the twitter.txt file convert them into feature vector and store in a new feature_vector.csv file in the twitter directory. To create low dimensional data set for twitter data and to perform all other operations, I used this feature_vector.csv file as orginal input.

1.3 Random projection algoritham :

Implemented in projection.py file in src directoty.

Step 1 : Create random matrix for various dimension d using normal distribution with ($\mu = 0$ *and* $\sigma = 1$).

Step 2 : Perform dot product between given data set and the random matrix to create a low dimensional projected data set. Store the newly created low dimensional data set for future use.

	Data set	Total low dim file	Directory	File name
1	dolphins	8	"*/data/Projected_data/dolphins/*"	dolphins_d.csv
2	pubmed	32	"*/data/Projected_data/pubmed/*"	pubmed_d.csv
3	twitter	750	"*/ddata/Projected_data/twitter/*"	twitter_d.csv

Table 1.1: Low-dimensional Projected data set for three given data set. The value of d different for different file.

Step 3 : Repeate step-1 and step-2 for all three given data set and for all values of d.

Chapter 2

Task 2 :: Bayes classifier and Nearest Neighbour classifier.

2.1 Bayes Classifier

Implement in : Implemented in bayes.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_2_bayes_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in "*/output_data/*" directory.

2.2 k-NN Classifier

Implement in : Implemented in nn.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

	Dimension	Test accuracy (%)
1	2	69.23
2	4	46.15
3	6	76.92
4	8	84.61
5	10	92.30
6	12	84.61
7	14	84.61
8	16	69.23
9	32	84.61

Table 2.1: Result of Bayes classifier for dolphins data set.

	Dimension	Test accuracy (%)
1	2	41.35
2	6	39.94
3	10	38.80
5	14	39.79
6	18	40.64
7	22	41.92
8	26	41.48
9	54	42.69
10	128	42.35

Table 2.2: Result of Bayes classifier for pubmed data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_2_nn_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in ”*/output_data/*” directory.

Chapter 3

Task 3 :: Cross-Validation technique, Measure accuracy and F1-score

3.1 Bayes Classifier using cross validation :

Implement in : Implemented in cross_validation_bayes.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_3_bayes_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in ”*/output_data/*” directory.

3.2 k-NN Classifier using cross validation :

Implement in : Implemented in cross_validation_NN.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_3_nn_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in ”*/output_data/*” directory.

Chapter 4

Task 4 :: Bayes classifier and Nearest Neighbour classifier using scikit-learn library

4.1 Bayes Classifier using scikit-learn :

Implement in : Implemented in bayes_CV_sklearn.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_4_bayes_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in ”*/output_data/*” directory.

4.2 k-NN Classifier using scikit-learn :

Implement in : Implemented in kNN_CV_sklearn.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_4_nn_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in ”*/output_data/*” directory.

Chapter 5

Task 5 :: Compare Task 3 and Task 4

Chapter 6

Task 6 :: Locality Sensitive Hashing(LSH)

Implement without using library.

6.1 Locality Sensitive Hashing Implementation:

Implement in : Implemented in lsh.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : It gives a hash table with all locally sensetive mappings of given data set and the hash functions used to create that hash table. And these are used in the classification.

Output file : No output to store.

Chapter 7

Task :: Classification with lsh and PCA

7.1 Classifier using lsh :

lsh is implemented by using the algorithm implemented in previous task.

Implement in : Implemented in classifier_using_lsh.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_7_lsh_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in ”*/output_data/*” directory.

7.2 Classifier using pca :

PCA is implemented by using standard library.

Implement in : Implemented in classifier_using_pca.py file of src directory.

Test on : All the given data set and all newly created low dimensional data set.

Result : Gives test accuracy, F1_score for macro, micro and weighted format. To calculate F1_score I used F1_score() method of metrics module.

Output file : The outputs are stored in various file with name as "Task_7_pca_datasetName.txt" file. datasetName is different for three different type of dataset. Each row in the output file contain a pair as (x,y), where x is the dimension of the data set and y define test accuracy in percentaage.
Each file is stored in ”*/output_data/*” directory.