Intro2MachineLearningHW01 Gökalp Ünsal

For the homework I used Python for programming as I did with my internship. In order to run the code without an error you need to have **pandas** and **NumPy** modules installed. In the command prompt you can simply give these commands to install:

```
pip install pandas

pip install numpy
```

I used pandas module to read the csv files, and then slicing them 25 or 14 rows. Then I converted the slices matrices to NumPy arrays to use NumPy's functions for arrays and matrices such as sum().

Variables

Data_set: the data set that has been read from the program.

Label_set: the label set.

Train_a, b …: 25*320 matrix that contains 25 of the images by pixels.

Test_a, b, c…: 14*320 slices of the data_set that contains the test data.

Label_a, b…: The labels of the train data.

Label_test_a, b…: The labels of the test data.

pcdA, B…: the probability of every pixel being a black one. (1*320)

hat_a, b…: the prediction of the test data. (14*1)

LabeledA, B…: The prediction of the trained data itself. (25*1)

Conf_all: The confusion matrix of the train data.

Conf_test_all: The confusion matrix of the test data.

Functions

Safe_log: replaces the log(0) parts as 0, because of the Loglikelihood.

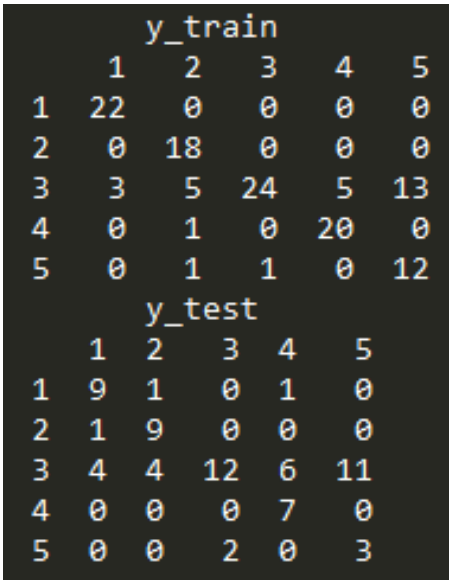
Pcd: Sums the black pixels of a column, then divides it to all the train data rows. Does this for all of the pixels to have the probability of every pixel being black. Returns an array pf 1*320.

G_score: Uses the Bernoulli equation of the pixel being black or white, then apply the loglikelihood to find the g(x). The prior probability is not important since it will be same for every g(x).

G_score_comparer: To predict the data, calculates its gScore with all the trained data. For every row, it receives the largest gScore, decides which letter it is and labels it with a number representing the letters. (1, 2, 3, 4, 5 => A, B, C, D, E) returns a vector of 25*1.

Conf_matrix: Creates a 5*5 matrix, iterates over the predicted data and increments the cells by the received label.

As the output I have the two confusion matrices with the train and the test data predictions. X being the original labels, and Y being the predicted labels.



y_train

	1	2	3	4	5
1	22	0	0	0	0
2	0	18	0	0	0
3	3	5	24	5	13
4	0	1	0	20	0
5	0	1	1	0	12

y_test

	1	2	3	4	5
1	9	1	0	1	0
2	1	9	0	0	0
3	4	4	12	6	11
4	0	0	0	7	0
5	0	0	2	0	3