CS520 : Project

Vinay Bysani (vkb20) — Nikita Balakrishnan (nb580)

May 2017

1. Algorithm Description:

Perceptron

Perceptron is one of earliest supervised algorithm. Peceptron can be called as a linear classifier. The algorithm would generate a line that best separates the labeled data to classify.

Perceptron can be mathematically expressed as :

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

w is a weight vector, x is input vector. b stands for bias. The bias pushes the decision boundary away from origin

MIRA

Mira is an online algorithm, an extension of Perceptron, where feature weights are learnt with additional loss function and margin constraint

Mira classifier is entirely a perceptron at classification level, but the way it trains differ. It makes use of validation data to choose from different weights, and chooses whichever weight gives best accuracy

Pseudo Code:(Used in the implementation)

1. For each C value(0.002, 0.004, 0.008), the following steps are iterated.
2. This would take each datum and iterate over all legal labels and find the best match/guess
3. if the guess is wrong, we would then modify the weight matrix. This is where MIRA varies with respect to perceptron. This update rule makes use of C value

4. We would then choose the weight matrix for that C, for which we get max accuracy

Naive Bayes

Naive bayes makes use of the Bayes Theorem

Bayes theorem is given by,

$$P(C_i|D) = \frac{P(C_i)P(D|C_i)}{P(D)} \tag{1}$$

In Naive Bayes, the conditional probability for each feature is calculated along with laplace smoothing with some k

And the guesses are made using

$$argmax_{C_i} logP(C_i) + logP(D|C_i) \tag{2}$$

The log-joint probability is given by the formula,

$$logP(C_i|D) = logP(C_i) + logP(D|C_i) \tag{3}$$

2. Features used :

1. Number of breaks in horizontal and vertical spaces
2. Number of inactive pixels (black pixels, pixels represented as a 0 in the matrix)
3. Splitting the image into four regions with a vertical and horizontal center line to get:
   (a) Percentage of active pixels above the horizontal center line
   (b) Percentage of active pixels below the horizontal center line
   (c) Percentage of active pixels to the right of the vertical center line
   (d) Percentage of active pixels to the left of the vertical center line

3. Performace of each algorithm (tabular data)

1. For digit classication, we have 5000 training data, 1000 test data
2. For face classication, we have 450 training data, 150 test data

Each of the algorithm (Perceptron, Naive Bayes, MIRA) have been trained on range of data points(10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%) and the following is observed.

1. Perceptron

Digits			
Percentage of total data	Training Data	Accuracy percentage	Time taken to train(sec)
10	500	56.6	6
20	1000	72.3	12
30	1500	71.5	19
40	2000	75.5	25
50	2500	76.5	33
60	3000	81.2	38
70	3500	77.5	44
80	4000	79.4	51
90	4500	77.9	56
100	5000	81.5	64

Faces			
Percentage of total data	Training Data	Accuracy percentage	Time taken to train(sec)
10	45	58.7	0.8
20	90	72	1.45
30	135	54.7	2.27
40	180	78.7	2.95
50	225	71.3	3.76
60	270	84	4.51
70	315	80	5.4
80	360	87.3	6.1
90	405	88	6.6
100	450	86.7	8.1

2. Naive Bayes

Digits			
Percentage of total data	Training Data	Accuracy percentage	Time taken to train(sec)
10	500	66.7	17.7
20	1000	71.9	18.1
30	1500	72.8	18.7
40	2000	74.4	19.2
50	2500	76	19.8
60	3000	76.3	20.3
70	3500	76.3	21
80	4000	76.7	21.4
90	4500	76.7	22
100	5000	77.5	22.9

Faces			
Percentage of total data	Training Data	Accuracy percentage	Time taken to train(sec)
10	45	60	3.1
20	90	74.7	3.5
30	135	84	3.9
40	180	89.3	4.3
50	225	87.3	4.7
60	270	89.3	5.2
70	315	89.3	5.7
80	360	88.7	6.2
90	405	88.7	6.7
100	450	90.7	7.4

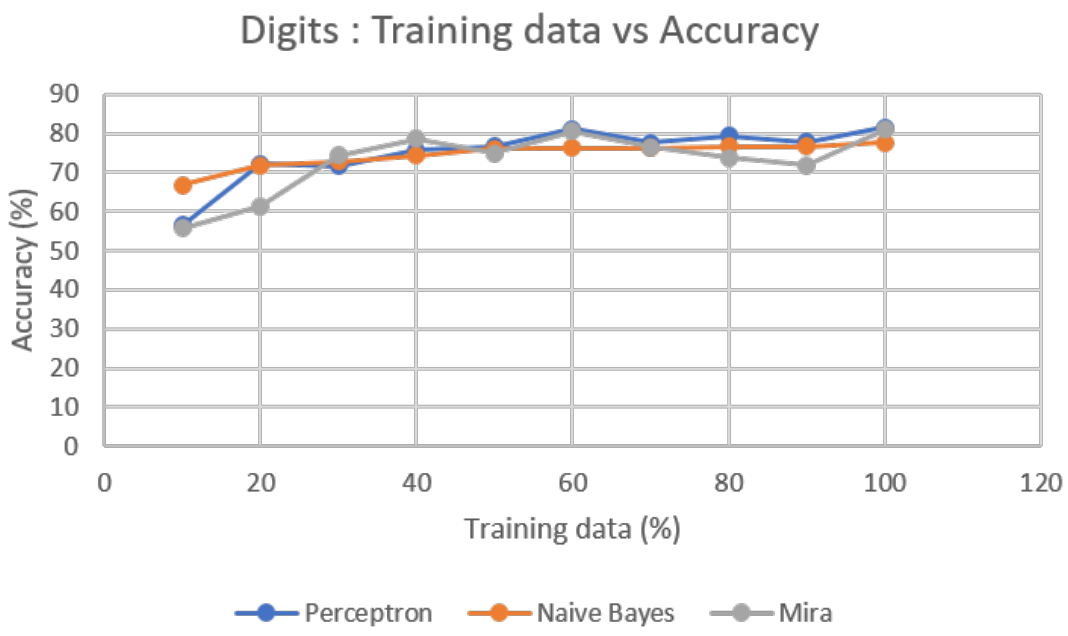
3. Margin Infused Relaxed Algorithm (Mira)

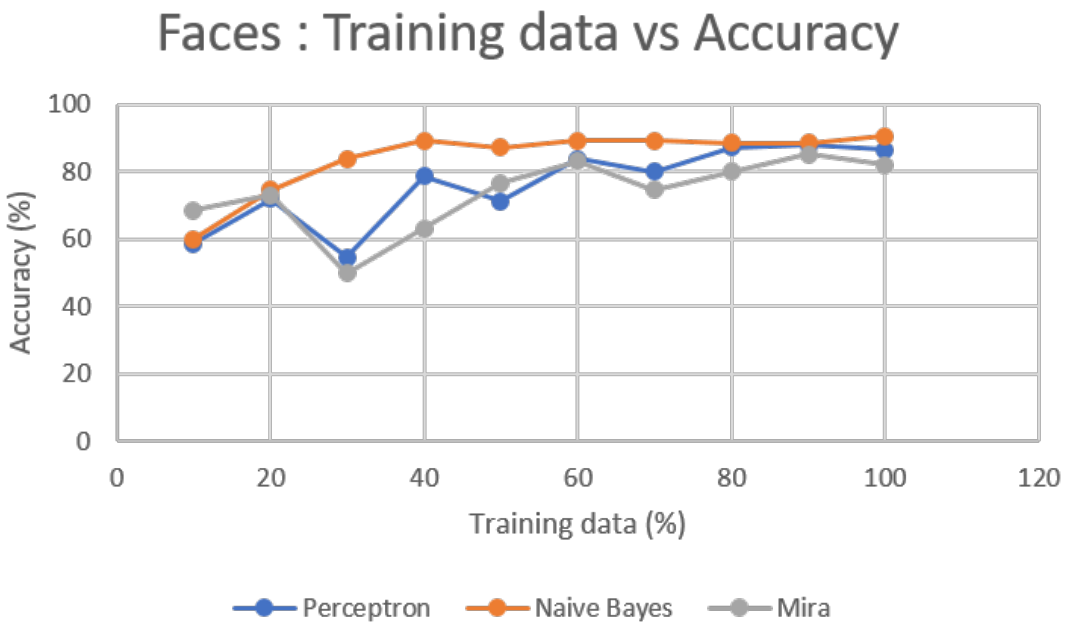
Digit			
Percentage of total data	Training Data	Accuracy percentage	Time taken to train(sec)
10	500	55.7	20
20	1000	61.3	40
30	1500	74.4	60
40	2000	78.5	78
50	2500	75	98
60	3000	80.5	120
70	3500	76.5	138
80	4000	73.8	157
90	4500	71.8	176
100	5000	81	199

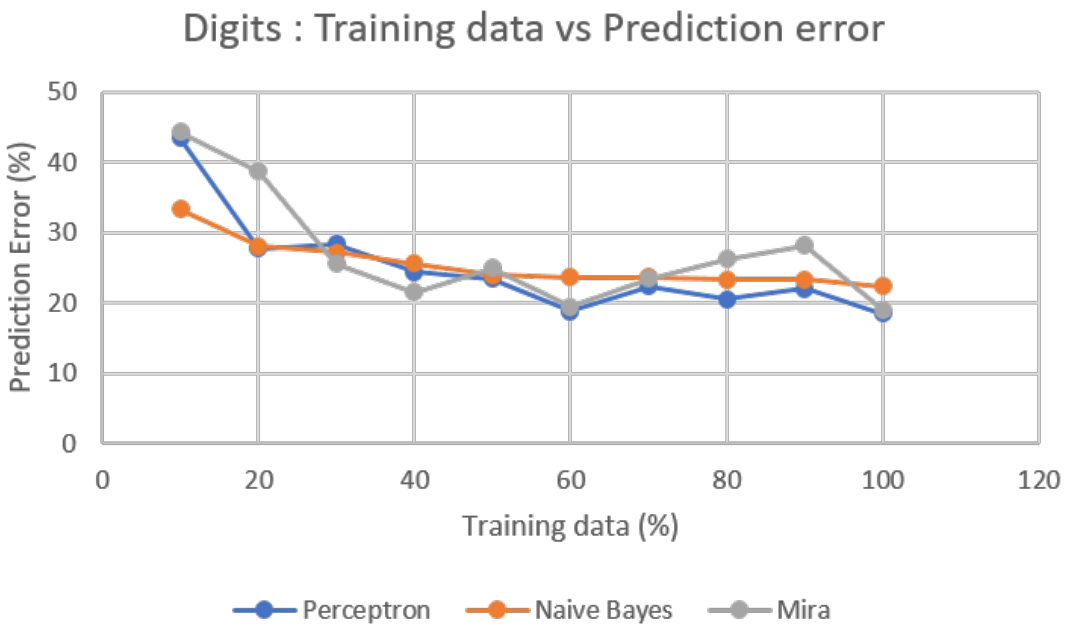
Faces			
Percentage of total data	Training Data	Accuracy percentage	Time taken to train(sec)
10	45	68.7	3.5
20	90	73	7.5
30	135	50	11
40	180	63.3	13
50	225	76.7	16
60	270	83.3	19
70	315	74.7	21
80	360	80	24
90	405	85.3	27
100	450	82	30

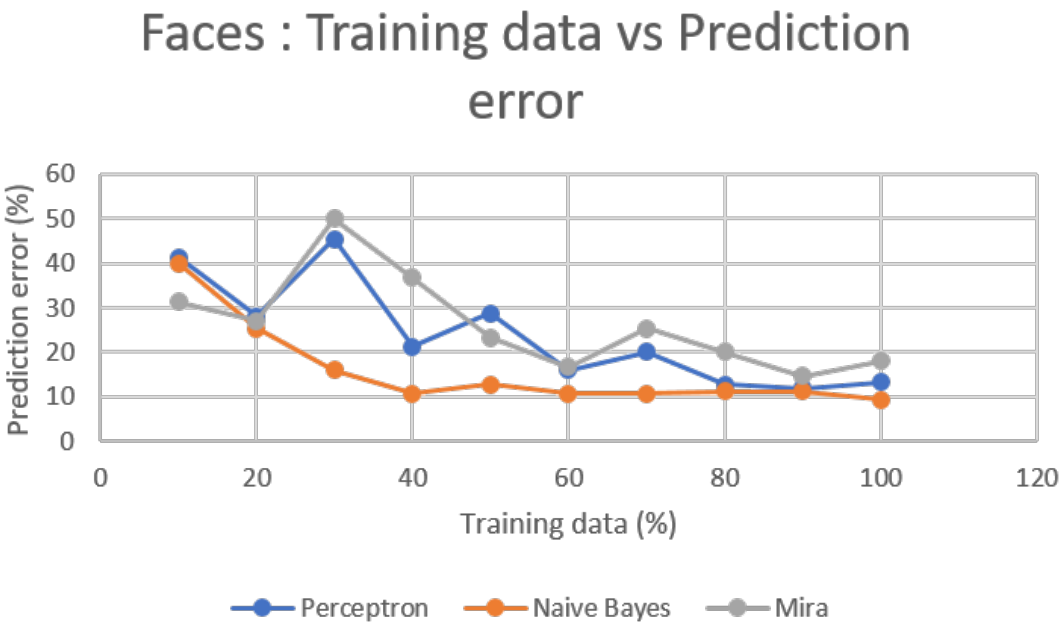
3. Performance of each algorithm compared (plots)

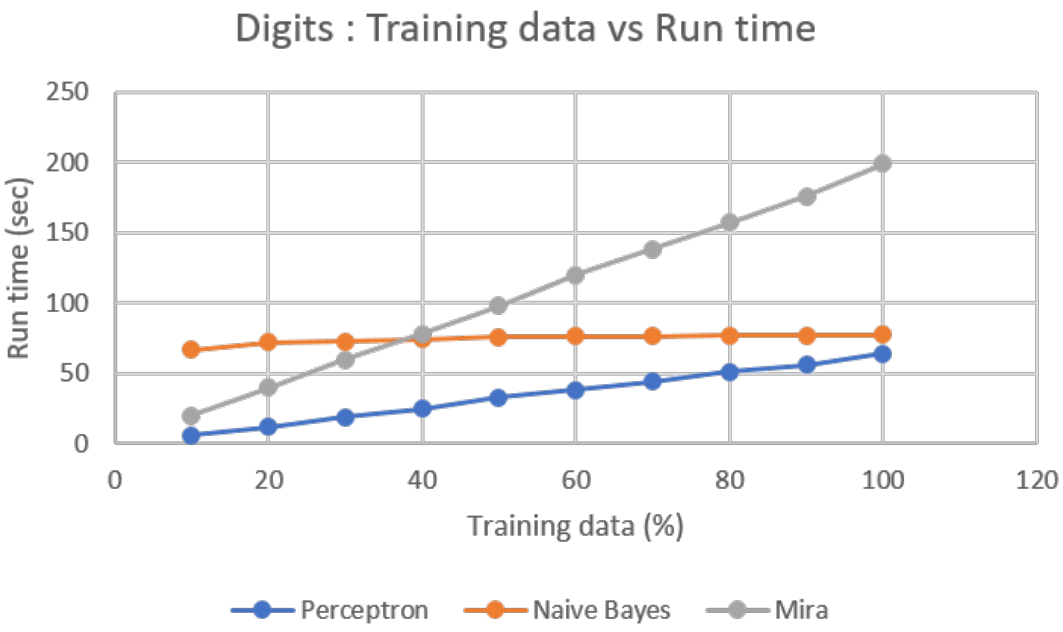
Accuracy for digits and faces are plotted against running time for each of algorithms(perceptron, naive Bayes, Mira). Run time against training data is plotted alike.

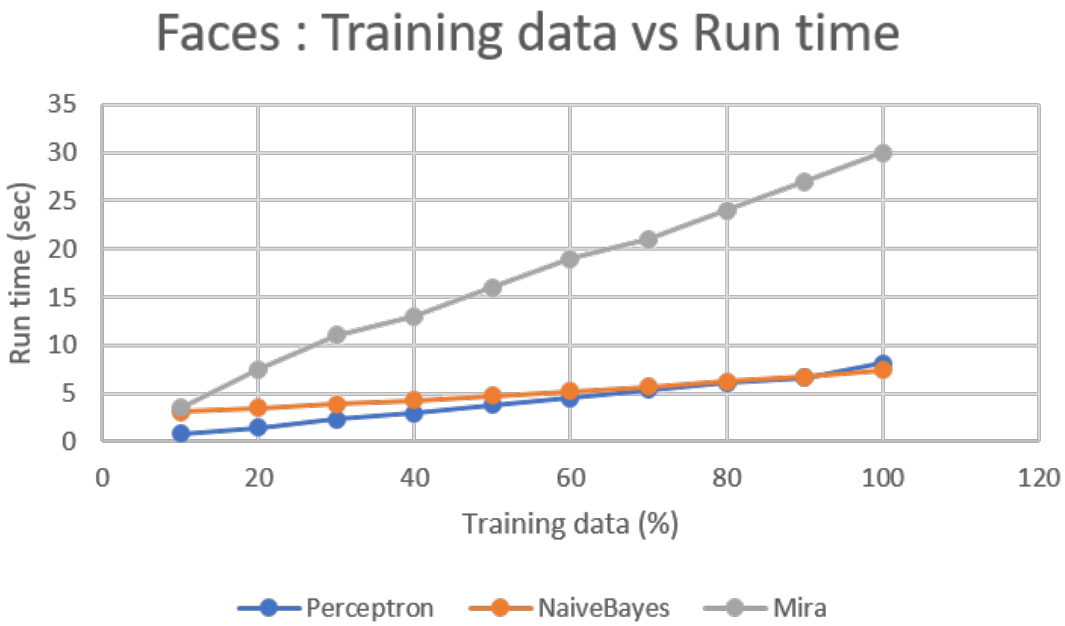



Faces : Training data vs Accuracy
Accuracy (%)
100
80
60
40
20
0
0
20
40
60
80
100
120
Training data (%)
Perceptron
Naive Bayes
Mira


Digits : Training data vs Prediction error
Prediction Error (%)
50
40
30
20
10
0
0
20
40
60
80
100
120
Training data (%)
Perceptron
Naive Bayes
Mira


Faces : Training data vs Prediction error
Prediction error (%)
60
50
40
30
20
10
0
0
20
40
60
80
100
120
Training data (%)
Perceptron
Naive Bayes
Mira


Digits : Training data vs Run time
Run time (sec)
250
200
150
100
50
0
0
20
40
60
80
100
120
Training data (%)
Perceptron
Naive Bayes
Mira



4. Analysis

Accuracy

For digits, Perceptron has given maximum accuracy compared to Naive Bayes and Mira. However the variations with other algorithms are quite low. Hence all of them produced good results, given huge training data that we had. Naive Bayes has the upper hand in accuracy with even low training data.

For faces, Naive Bayes accuracy is the best among the three. It has been consistent in learning with increase in training data.

Run time

For digits, Perceptron has performed very well in terms of run time, while Mira is worse.

For faces, both Perceptron and Naive Bayes were reasonable in terms of run time and Mira has taken 4 times more time than the other two algorithms at 100% training data.