September 21, 2016

Report

# Question 1

We use PLA.m to run the PLA algorithm; use PLA\_Pocket.m and test.m to run the PLA Pocket algorithm in this question.

When using PLA algorithm on this linear separable dataset, it converges after 8 iterations (Actually is 7 iteration, the last iteration is to make sure that it would not cause any error on this w.) which costs 0.003145 seconds time and gets the w=[1; 2.1028; -3.3960] for 0 error.

When using PLA Pocket algorithm on this linear separable dataset, it converges after 7 iterations which cost 0.003363 seconds time and get the w=[1; 2.1028; -3.3960] for 0 error.

It seems that both PLA Pocket algorithm and PLA algorithm are convergence, while the PLA algorithm is more efficient than PLA Pocket algorithm on this linear separable dataset. And they got the same results of w.

# Question 2

We use PLA.m to run the PLA algorithm; use PLA\_Pocket.m and test.m to run the PLA Pocket algorithm in this question.

When using PLA Pocket algorithm on this not linear separable dataset, it runs 1000 iteration which is equal to the T we set. And it costs 0.178508 seconds time to get the w=[0; 1.7225; -1.6323] with 2 errors (the error rate is 0.1).

When using PLA algorithm on this not linear separable dataset, it runs 1000 iteration which is equal to the T we set. And it costs 0.125961 seconds time to get the w=[0; 1.1666; -1.2378] with 3 errors (the error rate is 0.15).

It seems that both PLA Pocket algorithm and PLA algorithm are not convergence and the PLA Pocket algorithm is more efficient than PLA algorithm (on error rate) on this not linear separable dataset. And PLA Pocket algorithm got a better w.

# Question 3

We use PLA\_Pocket\_Zip.m, test.m and DataToXY.m to run the PLA Pocket algorithm in this question.

If we add ‘1’ to the X(D,:), the PLA Pocket algorithm converges after 9 iterations which costs 0.096187 seconds time to get the optimal w. Using this w to zip.test, we get 4 errors and the error rate is 0.0094.

If we do not add ‘1’ to the X(D,:), the PLA Pocket algorithm converges after 9 iterations which costs 0.091420 seconds time to get the optimal w. Using this w to zip.test, we get 5 errors and the error rate is 0.0118.

It is obviously that adding ‘1’ to the X(D,:) can get a better result. It makes the line of w that does not need to pass the origin point, just like adding a constant of a line.

# Question 4

We use PLA\_Features.m, test.m and DataToXY.m to run the PLA algorithm; use PLA\_Pocket\_Features.m, test.m and DataToXY.m to run the PLA Pocket algorithm in this question.

When using PLA algorithm on this dataset, it runs 1000 iteration which is equal to the T we set. It costs 7.882702 seconds time to get the w with 79 errors (the error rate is 0.0506.) on train dataset. Using this w to features.test, we get 33 errors and the error rate is 0.0778.

When using PLA Pocket algorithm on this dataset, it runs 1000 iteration which is equal to the T we set. It costs 9.009457 seconds time to get the optimal w with 5 errors (the error rate is 0.0032.) on train dataset. Using this w to features.test, we get 10 errors and the error rate is 0.0236.

Compared with question 3, the dataset with mean intensity and asymmetry of image is not linear separable because it is not convergence. It is better to use just intensity of 16\*16 grayscale image for machine learning.