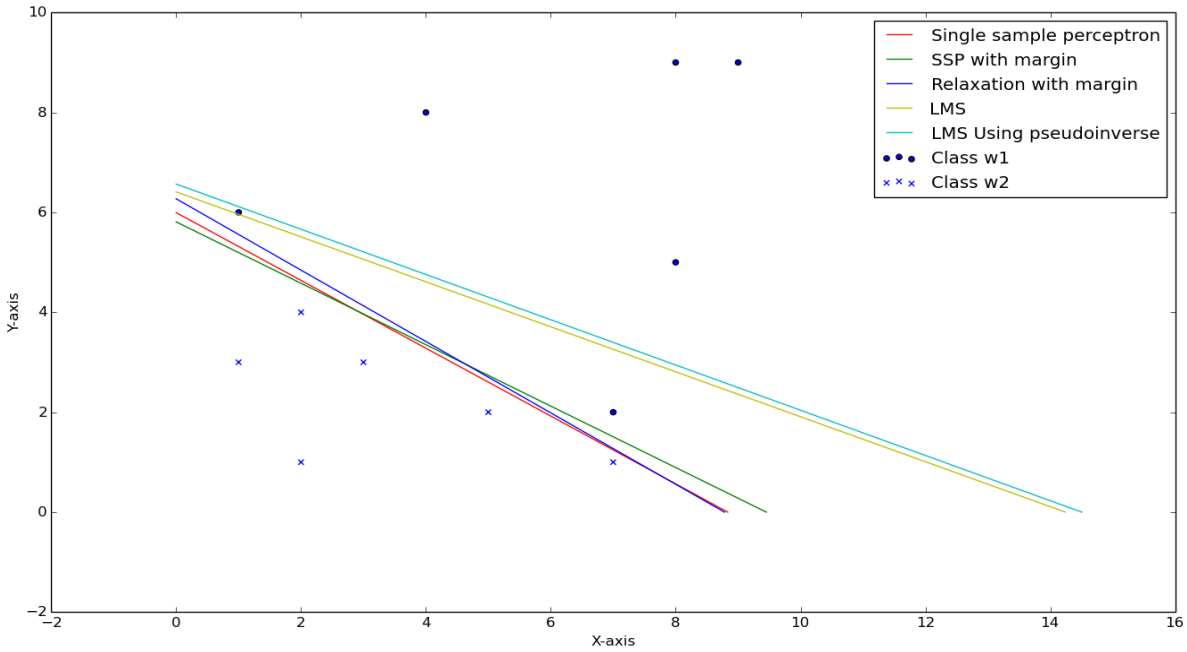
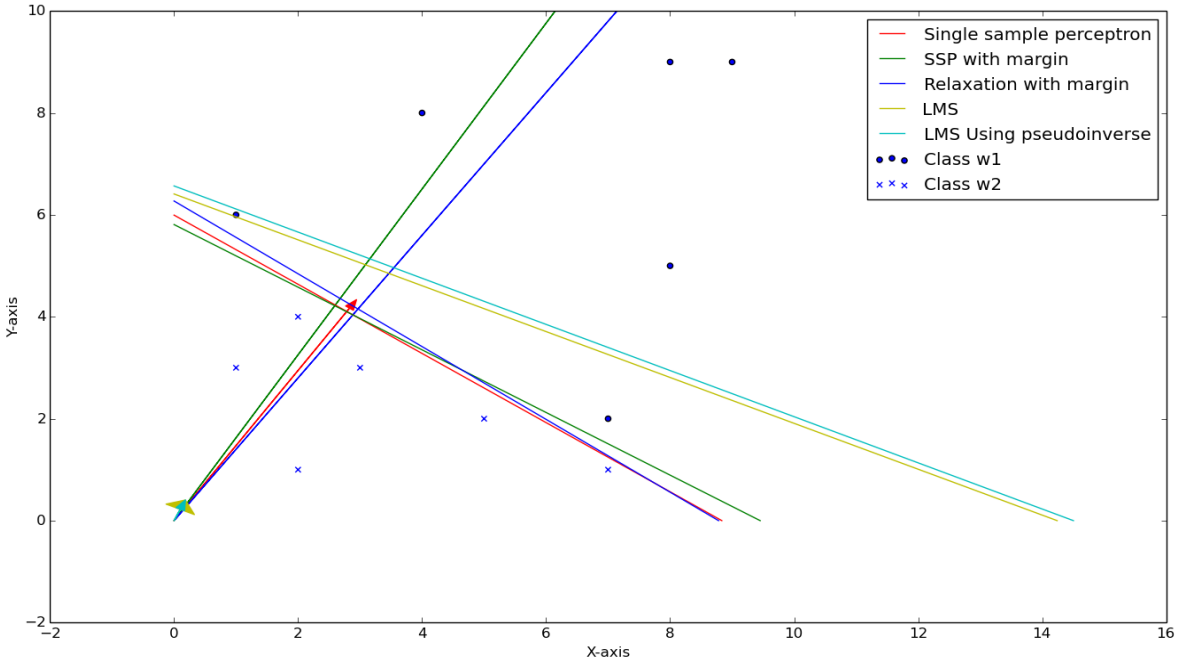
Graphs and Observations

A) Graphs1: Separating hyper planes:



Graphs2: Hyper planes with their normals:



In the weight vector for LMS algorithm is overlapped with LMS with pseudo inverse, to differentiate them arrow head width was increased for LMS weight vector

B) Test set creation and Accuracy:

Test samples created are : class w1: {(6,5),(4,4),(3,9)} and class w2: {(2,2),(4,1),(1,5)}

```
narendra@narendra-ThinkCentre-E73:~/Downloads$ python q2.py
Test samples:
[[ 1  6  5]
 [ 1  4  4]
 [ 1  3  9]
 [-1 -2 -2]
 [-1 -4 -1]
 [-1 -1 -5]]
No.of misclassified using ssp 0
No.of misclassified using ssp with margin 0
No.of misclassified using relaxation with margin 0
No.of misclassified using LMS 1
```

This set is linearly separable so all algorithm given 100 % accuracy except LMS algorithm. It has misclassified the point (4,4) as class w2. So its accuracy is 83.33%

C) Initial weight vectors and respective convergence time:

A good initial heuristic is to start with “Difference of average of positive input vectors and average of negative input vectors”. We can justify that as the final weight vector is a linear combination of input vectors.

For given samples the initial weight vector is (0, 2.38, 4.17) and to check convergence dependence on initial weights two other weight vectors considered, (1, 1, 1) and (300, 200, 100)

The results are like below:

Vector/algorithm	Perceptron	Perceptron with margin	Relaxation with margin	LMS
(300, 200, 100)	0.003799	0.004741	0.07350	0.44514 (45,967 iterations)
(1, 1, 1)	0.0009210	0.00144	0.01832	0.21397 (18,127 iterations)
(0,2.38, 4.17)	0.00088	0.000556	0.01121	0.17170 (18,127 iterations)

*Units of time is seconds

From the above table we can infer

Vectors in the order of respective convergence time: (300,200,100) > (1, 1, 1) > (0, 2.38, 4.17)

Algorithms in the order of respective convergence time: LMS > Relaxation with margin > perceptron with margin > Perceptron

D) Margins and convergence time:

With initial weight vector as (1, 1, 1) the below table gives the dependence of convergence time on margins

Margin/ Algorithm	Relaxation with margin	Perceptron with margin
10	0.01759	0.0066
50	0.11041	0.03489
100	0.3883	0.0786

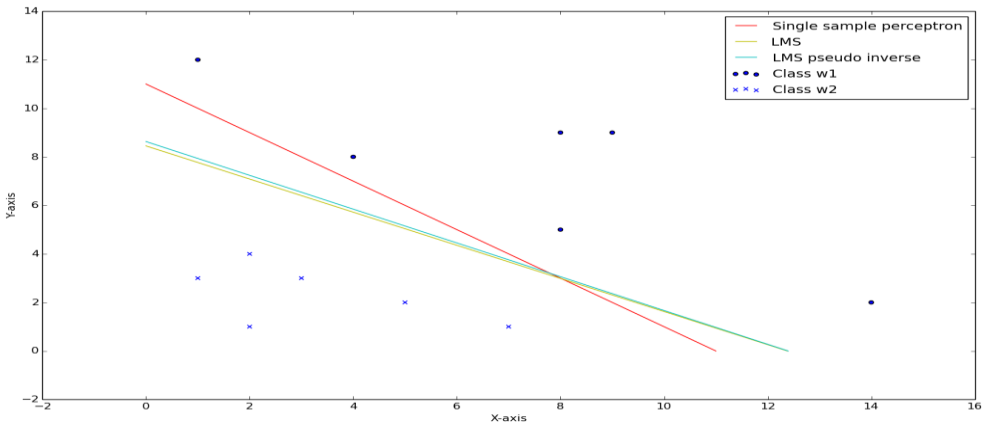
From above table we can infer that “the more the margin the more the convergence time”

LMS VS Perceptron

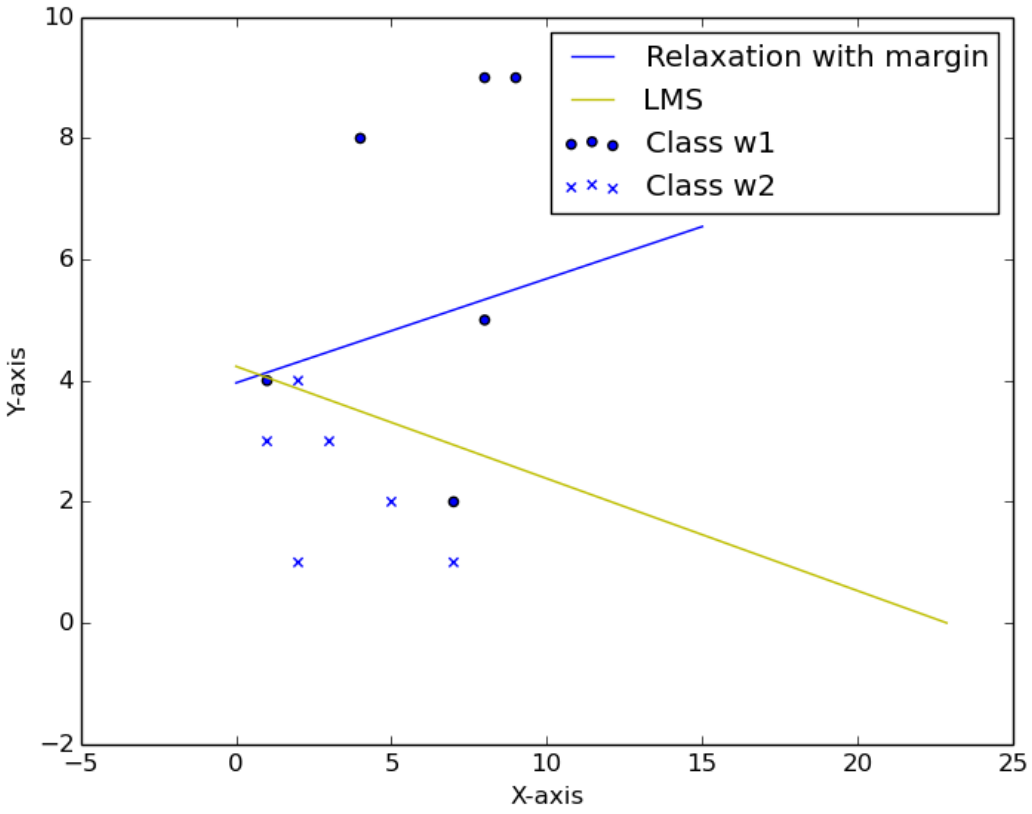
6) samples are modified such that the solutions align

Changes : from class 1 : (1,6) has changed to (1,12) and (7,2) has changed to (14,2)

```
modified samples such that soltuions of Perceptron and LMS align
[[ 1  1 12]
 [ 1 14  2]
 [ 1  8  9]
 [ 1  9  9]
 [ 1  4  8]
 [ 1  8  5]
 [-1 -2 -1]
 [-1 -3 -3]
 [-1 -2 -4]
 [-1 -7 -1]
 [-1 -1 -3]
 [-1 -5 -2]]
```



7)



LMS and Relaxation with margin giving nearest solution