Assignment 2

201364176

Noor Pratap Singh

Problem 1

If a2 is any solution vector, then ||a 2 (k+1) - a || is smaller than || a(k) - a 2 (k)||

Now as per our algorithm

a(k+1) = a(k) + y k. We can add a scale factor α and rewrite this equation

a(k+1) - α a 2 = (a(k) - α a 2) + y k
||a(k+1) - α a 2 || 2 = || a(k) - α a 2 || 2 + 2(a(k) - α a 2) t y k + ||y k || 2

Because y k was misclassified, a t (k)yk <= 0

||a(k+1) - α a 2 || 2 <= || a(k) - α a 2 || 2 + 2 α a 2 t y k + ||y k || 2

Because a 2 t y k is strictly positive, the second term will overpower the third term.if α is large enough

For example, let β be the maximum pattern vector length (β = max i || y i || 2), and γ be the smallest inner product of the solution vector with any pattern vector (γ = min i [a 2 t y i] > 0).

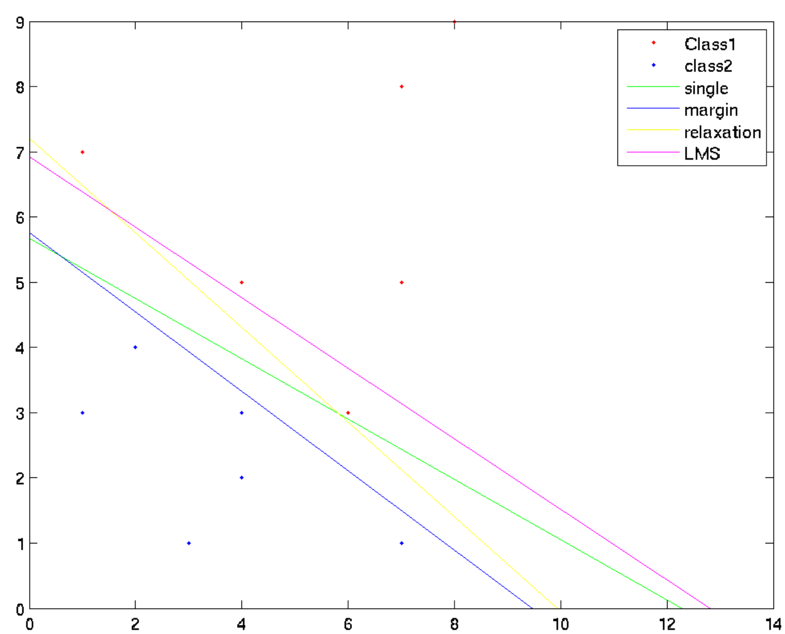
We now have the inequality:
||a(k+1) - α a 2 || 2 <= || a(k) - α a 2 || 2 - 2 αγ + β 2
Choosing α = β 2 / γ gives:
||a(k+1) - α a 2 || 2 <= || a(k) - α a 2 || 2 - β 2
So, the squared distance between a(k) and α a 2 is reduced by at least β 2 after each correction.

After k corrections, we have:
$||a(k+1) - \alpha a 2 || 2 <= ||a(1) - \alpha a 2 || 2 - k \beta 2$

The squared distance can not be negative, so there must be at most k 0 corrections, where $k 0 = ||a(1) - \alpha a 2 || 2 / \beta 2$.
The weight vector that results after the corrections are done must classify all samples correctly, because there are a finite number of corrections that only occur at each misclassification.
In summary, k 0 is a bound on the number of corrections, and there will always be a finite number of corrections using the fixed - increment rule if the samples are linearly separable.

Problem 2

I) The graph is given below



Legend is self explanatory

II)The weight vectors have been initially randomized. But the farther the initial weight factor from my solution region the more time it took to converge towards the final solution. It was also dependent on my value of *eta*.

III)From the figure it is quite evident that when margin was added no point actually lies on the solution vector(normal to solution vector). While in others it is possible.

IV)

Single Sample Perceptron

Algorithm:

```
begin
init a, k = 0
do
  k = (k + 1) mod n
  if y k is misclassified by a
     then a = a + y k
  until all patterns classified
return a
end
```

As shown above in a single iteration we update our *a* based on only 1 mis-classification at a time. At each iteration we again find new misclassified based on updated *a* and we run the above untill all samples have been classfied.

Note Solution will come only for linearly separable case.

As far as implementation is concerned it was done in Matlab and we first noramlised the points. As soon as a misclassified sample is found we update *a* and move to next iteration.

We found a separating boundary after the process but in some case points were lying on the boundary.

Single Sample Perceptron with margin

Algorithm:

begin initialize a, criterion θ, margin b, η(·), k = 0
do k ← k + 1
  if a t y k + b < 0 then a ← a − η(k)y k
    until a t y k + b ≤ 0 for all k
  return a
end

The above algorithm is same as single sample but we have just added the condition that a t y k + b < 0 instead of a t y k < 0.

All the analyis and results are similar except that no point lies on the boundary because of the margin.

Relaxation with margin

Algorithm:

begin initialize a, η(·), k = 0
do k ← k + 1
  if y k is misclassified then a ← a + η(k)(b−transp(a)y)*y k/mod(yk)^2
    until all patterns properly classified
return a
end

Widrow-Hoff Procedure

Algoirthm:

begin initialize a, b, criterion θ, η(·), k = 0
do k ← k + 1
    a ← a + η(k)(b k − a t y k)y k
    until η(k)(b k − a t y k)y k < θ
return a
end