1/4/2019 Recap: Convergence of perceptron Algorithm

Perceptron criterion:

$$\{X\} \to \{y\}$$

$$\downarrow$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \to \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \\ 1 \end{bmatrix}$$

$$\text{If } a^t y > 0 \Rightarrow y \in \omega_1$$
$$< 0 \Rightarrow y \in \omega_2.$$

uniform criterion fn:

for all the samples $a^t y > 0$ the weight vector a is correctly classified, otherwise it is mis-classified and then we should update the weight vector $a(k)$ to $a(k+1)$.

we are interested to find the weight vector 'a'.

$J(a)$ has to be minimum.

$a(0)$ – arbitrary

$a(k+1) = a(k) - \eta(k) \nabla J(a(k))$ and perceptron criterion

$$J_p(a) = \sum_{\forall y \text{ misclassified}} -a^t y$$

$$\boxed{\begin{aligned} &a(0) - \text{arbitrary} \\ &a(k+1) = a(k) + \eta(k) \cdot \sum_{\forall y \text{ misclassified}} y \end{aligned}}$$

let's Consider this sample-by-sample.

$$y^1, y^2, y^3, \ldots y^k \ldots y^n$$

k^{th} sample misclassified

$$\begin{cases} a(0) & - \text{arbitrary} \\ a(k+1) = a(k) + \eta y^k. \end{cases}$$

sequential algorithm

To demonstrate that the above sequential algorithm Converge let's consider the two-dimensional case

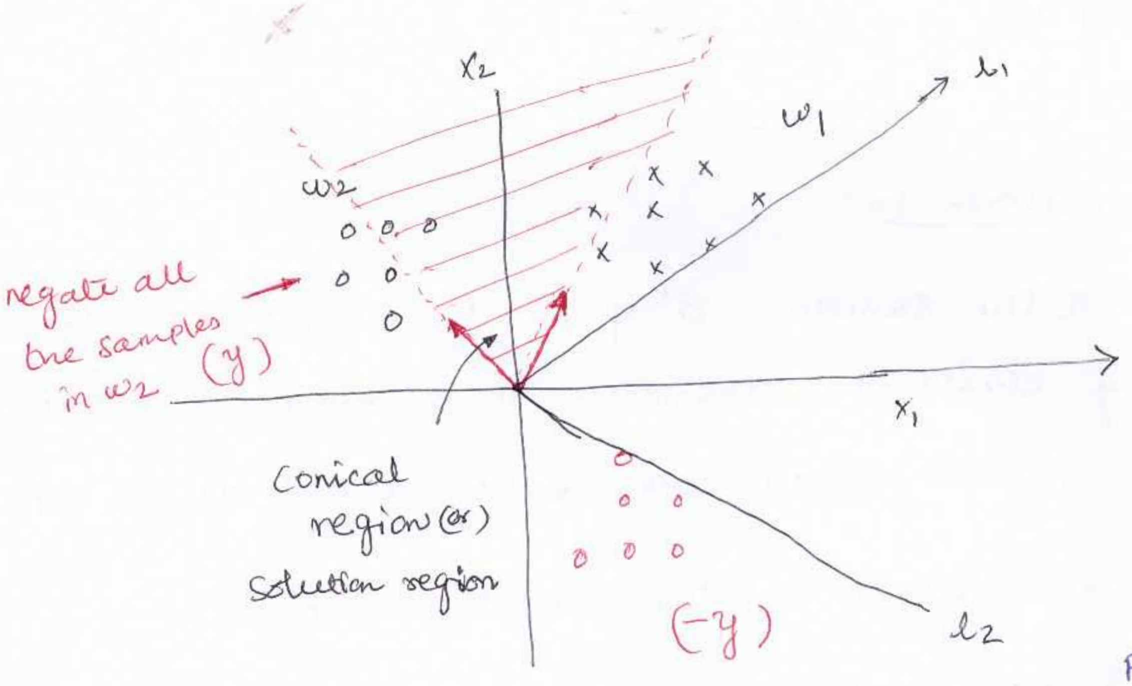

Fig. 1

weight vector 'a' is orthogonal to the decision surface. In 2-D it is nothing but a line. what are stright lines which actually separates these two classes?

I could have some limiting cases. two lines l_1 and l_2 Any line that lies in between these two limiting lines l_1 and l_2 which properly separates these two classes without error.

Now the weight vectors are orthogonal to the decision boundary.

Any weight vectors (a) lies within the conical region is solving our purpose. The conical region is the solution region. Our weight vectors should lie within this solution region. When the algorithm converges the weight vectors should lie within the solution region.

Algorithm Illustration:

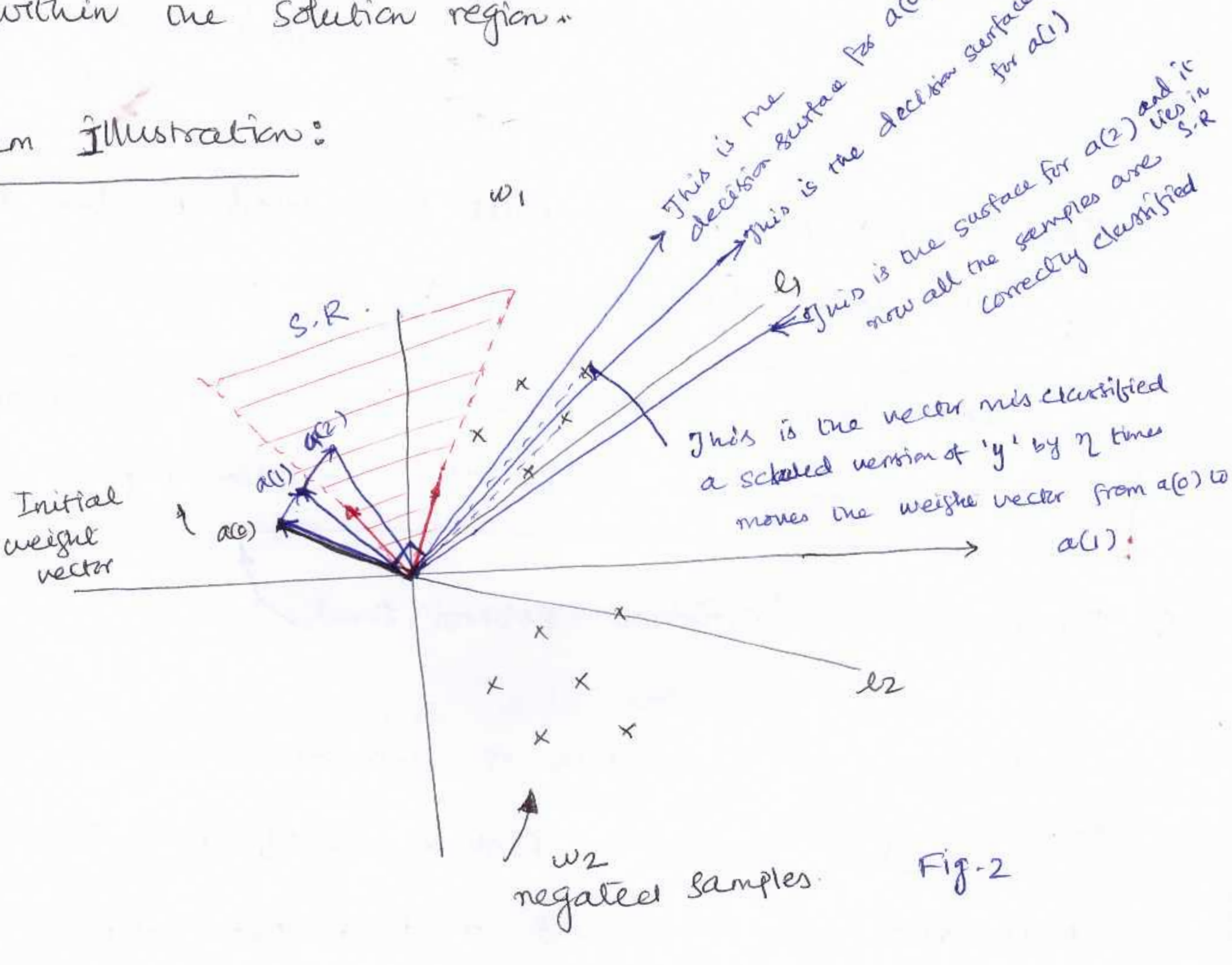

Fig-2

The initial weight vector a(0) mis-classify the '3' samples in ω_1. The decision surface corresponding to the weight vectors a(0) which is drawn in blue line.

according to the algorithm

$$a(k) = a(k-1) + \eta \sum y \quad \forall y \text{ misclassified.}$$

$$\boxed{a(k) = a(k-1) + \eta\, y}$$

This vector 'y' is scaled by a factor 'η' in the direction of 'y' and added with previous weight vector $a(k-1)$

The weight vector $a(0)$ will be moved in the direction of (mis-classified) vector 'y' by 'η' times.

and finally when the algorithm converges the weight vectors lies within the solution region.

(*) The perceptron criterion ensures that

But There is a problem of generalization?

This leads to Risk in classification. To minimize this risk, we should restrict the solution region some where as the safe region [sub region of solution region]. That means we should ensure the weight vector 'a' should lie in safe region. [Refer Fig. 3]

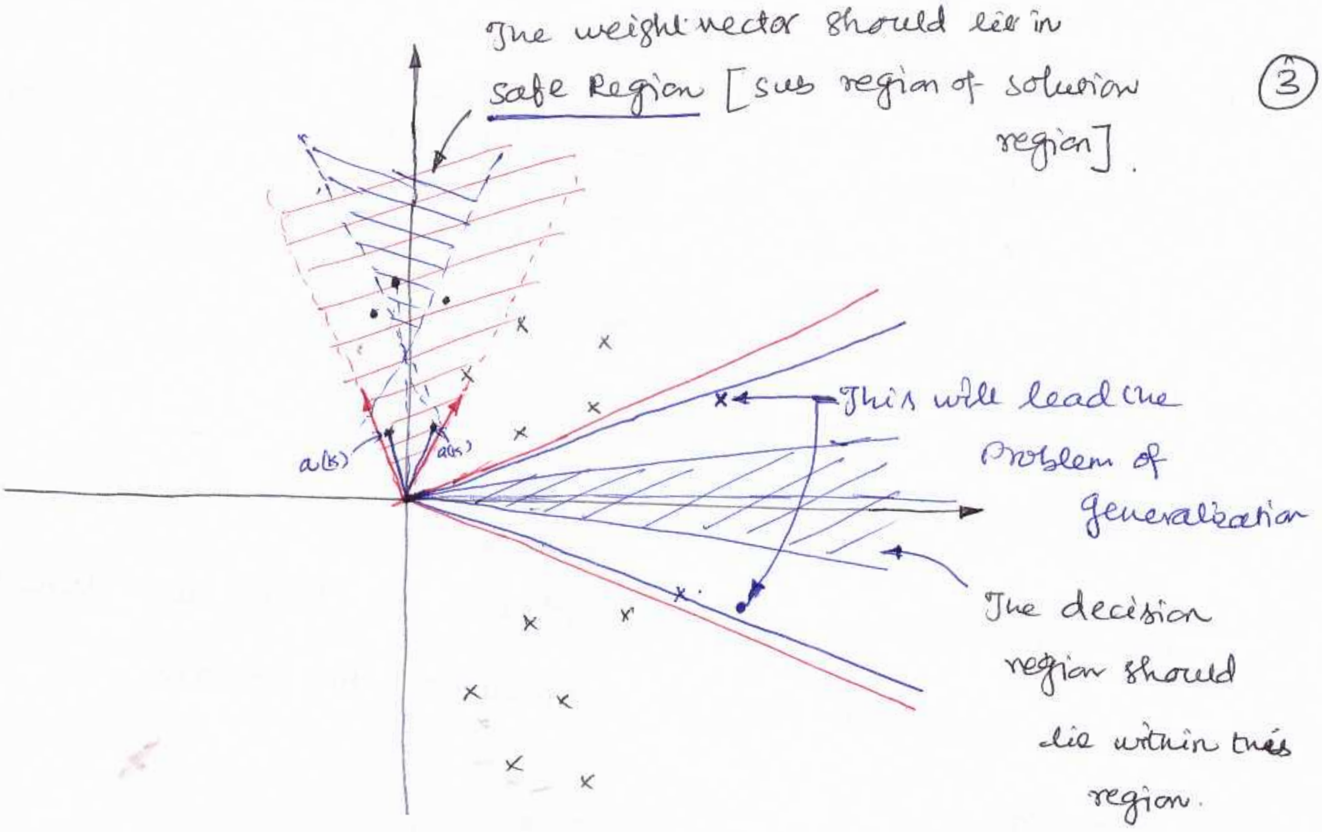

Fig. 3

This can be ensured by the rule

$$\boxed{a^t y > b.} \text{ instead of } a^t y > 0.$$

margin

I would say now, any y which satisfies $a^t y \geq b$ margin then it is safely classified. If it is $\not> 0$ it is properly classified, but it is not safe.

With this, we can ensure that the weight vector should lie on the safe region.

The perceptron criterion is not only the criterion function to design a linear classifier. One of the criteria function can be defined as follows.

Relaxation criterion:

$$J_r(a) = \frac{1}{2} \sum_{\forall y \text{ misclassified}} \frac{(a^t y - b)^2}{\|y\|^2}$$

Relaxation criterion

For minimization of this criterion fn $J_r(a)$ we use the same gradient descent procedure. to obtain the weight vector 'a'.

$$\nabla J_r(a) = \sum \frac{(a^t y - b)^2}{\|y\|^2} \cdot$$

$$= \sum_{\substack{\forall y \\ \text{misclassified}}} \frac{(a^t y - b)}{\|y\|^2} \cdot y$$

$a(0)$ – arbitrary

$$a(k+1) = a(k) + \eta \sum_{\substack{\forall y \\ \text{misclassified.}}} \frac{b - a^t y}{\|y\|^2} \cdot y$$

Sequential version of this

$$a(0) \text{ - arbitrary}$$

$$a(k+1) = a(k) + \eta \frac{b - a^t(k)\, y^k}{\|y^k\|^2} \cdot y^k.$$

here, the samples are considered one-after-another. The moment, when we find the vector 'y' is misclassified, we should update the weight vector

It can be noted that whether I use perceptron criteria (or) relaxation procedure, in both cases., the convergences gauranteed If the classes are linearly separable.

otherwise, the algorithm never converge. we can make use of these algorithms, only we know for sure the classes are linearly separable.

However, If we ~~do not know~~ are not sure the classes are linearly separable (or) not, still we can design linear classifier, with minimum error.