Lecture 5: Linear Classification

Classification: Given labelled data

$$(x_i, y_i), \quad i = 1, .., n$$

x_i: feature vector ← y_i: label (discrete value)

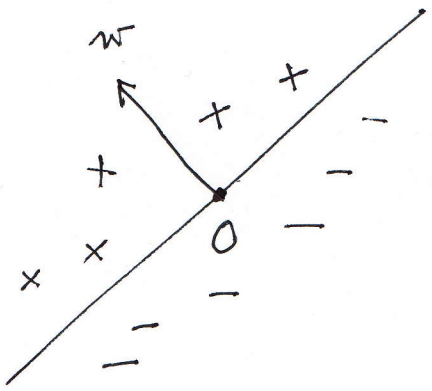
find a rule to predict y for unseen x.

For linear classification:

- assume y is $+1$ or -1
- a classification rule is a hyperplane that separates $+1$ from -1

How to represent this hyperplane?

* For now, assume hyperplane is through origin



w = normal vector to the hyperplane.

For a point x on the + side,

$$\langle w, x \rangle > 0$$

For a point x on the − side,

$$\langle w, x \rangle < 0$$

So:

(1) Classification rule: represented by a vector w

(2) Given a test example x, output its label as:

$$y = \text{sign}(\langle w, x \rangle)$$

(3) Given training data, find a w corresponding to a hyperplane that largely has the $(x_i, +)$ training points on one side, and the $(x_i, -)$ ones on another.

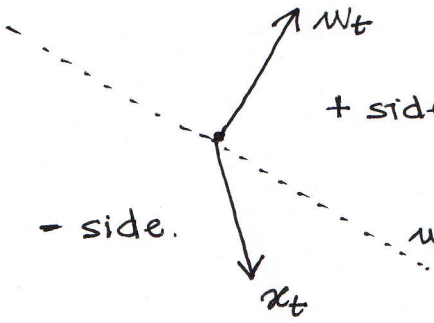
The Perceptron Algorithm:

1. Initially, $w_1 = 0$.
2. For $t = 1, 2, 3, \ldots$
   If $y_t \langle w_t, x_t \rangle \leq 0$ then
   $w_{t+1} = w_t + y_t x_t$
   Else
   $w_{t+1} = w_t$

Examples arrive in a sequence:
$(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t), \ldots$

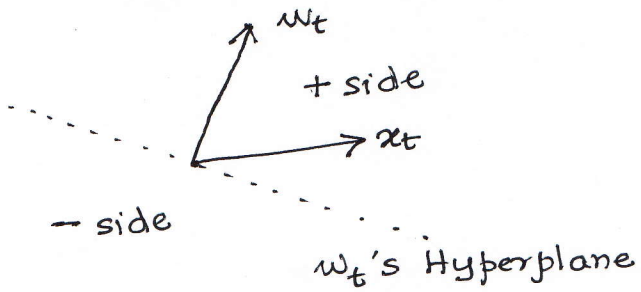
* What does the condition "$y_t \langle w_t, x_t \rangle \leq 0$" mean?

- Case 1: $y_t = 1$. Then: $\langle w_t, x_t \rangle \leq 0$



* x_t's label is +
* yet angle b/w x_t, and $w_t \geq 90°$, so x_t is not on the correct side of the hyperplane corresponding to w_t.

- Case 2: $y_t = -1$. Then: $\langle w_t, x_t \rangle \geq 0$



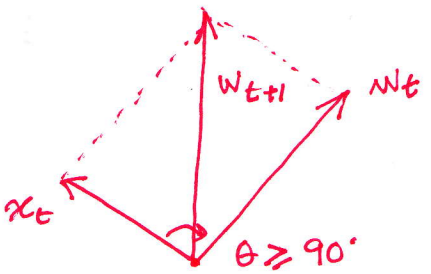
* x_t's label is −
* yet angle b/w x_t and w_t is $\leq 90°$, so x_t is not on the correct side of the hyperplane corresponding to w_t

In both cases, $y_t \langle w_t, x_t \rangle \leq 0$ means:

w_t predicts the label of x_t incorrectly

Geometric interpretation of: "$w_{t+1} = w_t + y_t x_t$".

Case 1: $y_t = 1$; $w_{t+1} = w_t + x_t$

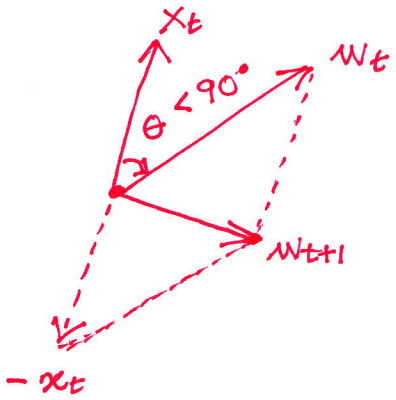


w_{t+1} moves "closer to" x_t

or

x_t moves towards the + side of the decision boundary (hyperplane)

Case 2: $y_t = -1$; $w_{t+1} = w_t - x_t$.



w_{t+1} moves "away from" x_t

or

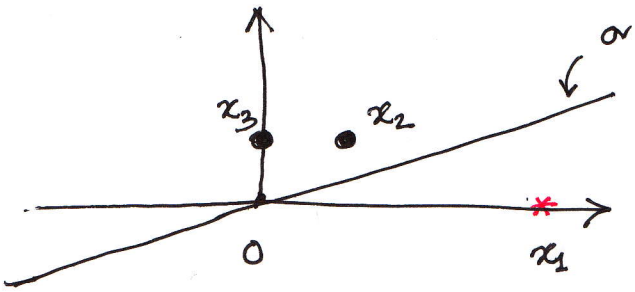
x_t moves towards the − side of the decision boundary hyperplane.

In both cases, we are moving towards the "correct solution".

Exercise: Suppose w_t makes a mistake on (x_t, y_t), and we update w_{t+1} as $w_{t+1} = w_t + y_t x_t$. Is it possible for w_{t+1} to also make a mistake on (x_t, y_t)?

Example:

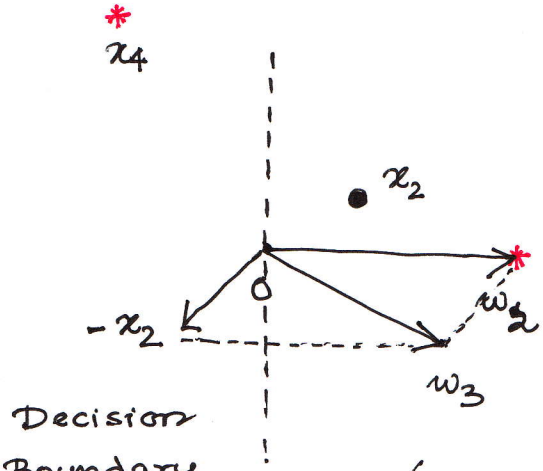
Training Data: $((4,0), 1), ((1,1), -1), ((0,1), -1), ((-2,-2), 1)$



or separating hyperplane for the input data points

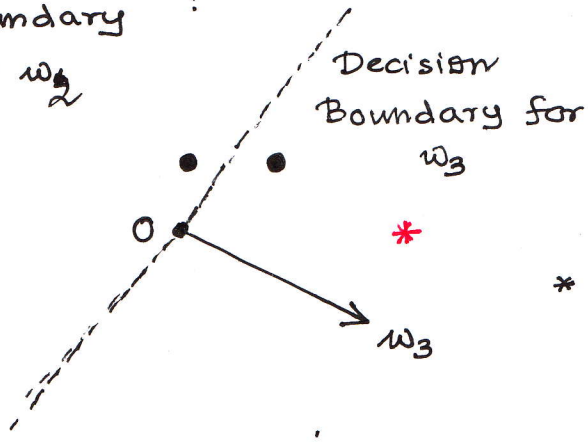
Round 1:

* $w_1 = 0$
* $y_t \langle w_t, x_t \rangle$ for $t = 1$
  $= 0$ as well.
* $w_2 = w_1 + y_1 x_1$
  $= (4,0)$



Round 2:

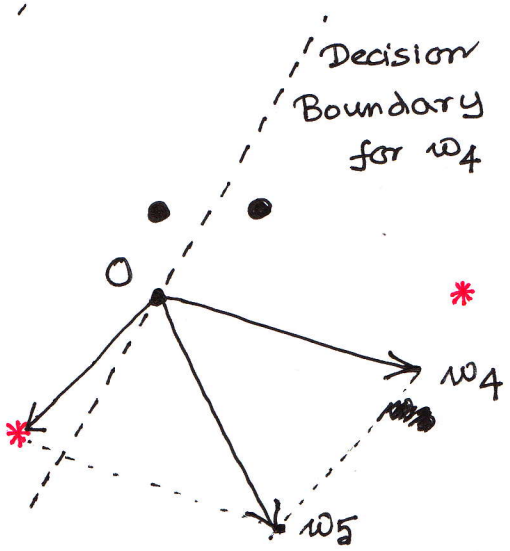
* $y_2 \langle w_2, x_2 \rangle < 0$
* So $w_3 = w_2 + y_2 x_2$
  $= (4,0) - (1,1) = (3,-1)$



Round 3:

* $y_3 \langle w_3, x_3 \rangle > 0$

Correct. $w_4 = w_3 = (3,-1)$

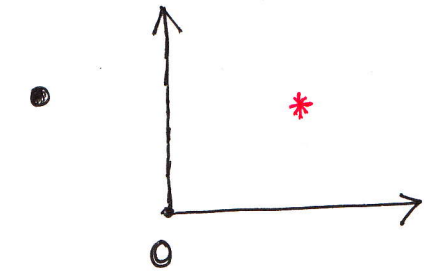


Round 4:

* $y_4 \langle w_4, x_4 \rangle < 0$
* So $w_5 = w_4 + y_4 x_4$
  $= (3,-1) + (-2,-2)$
  $= (1,-3)$

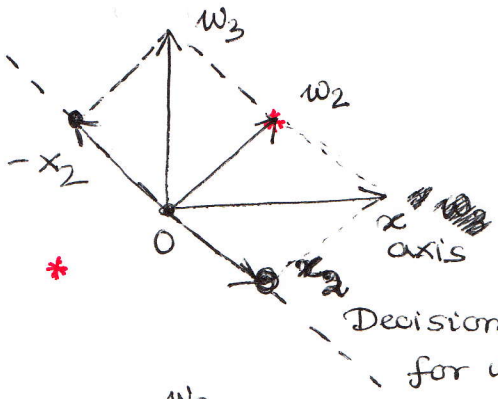
Example 2: Training data:

((1, 1), 1), ((1, -1), -1), ((-1, 1), -1), ((-1, -1), 1)



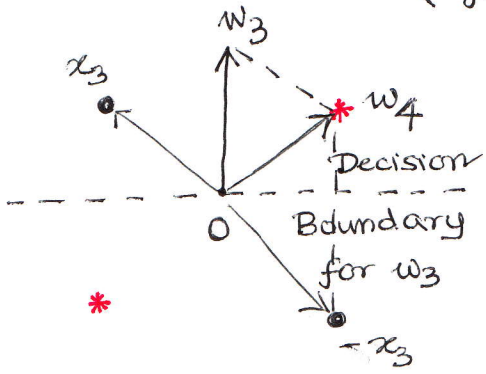
Initially, $w_1 = 0$. In round 1,

* $y_1 \langle w_1, x_1 \rangle \leq 0$, so mistake.
* $w_2 = w_1 + y_1 x_1 = (1, 1)$



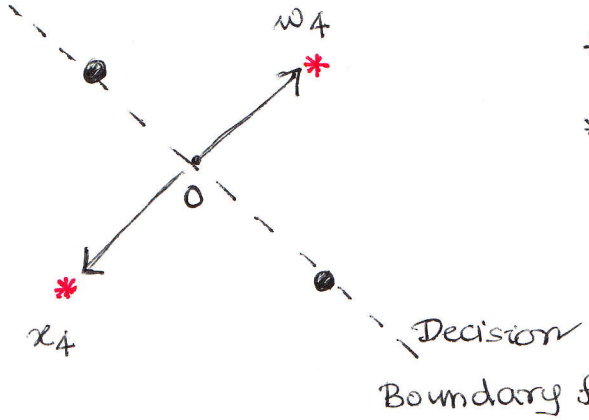
Round 2:

* $y_2 \langle w_2, x_2 \rangle \leq 0$
* $w_3 = w_2 + y_2 x_2 = (1, 1) - (1, -1)$
  $= (0, 2)$



Round 3:

* $y_3 \langle w_3, x_3 \rangle \leq 0$
* $w_4 = w_3 + y_3 x_3 = (0, 2) - (-1, 1)$
  $= (1, 1)$

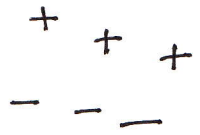


Round 4:

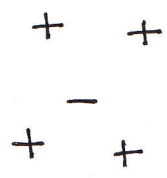
* $y_4 \langle w_4, x_4 \rangle \leq 0$.
* $w_5 = w_4 + y_4 x_4 = (1, 1) + (-1, -1)$
  $= 0$

When does Perceptron Converge?

Linear separability: there exists a hyperplane separating the + labelled data from the −.



Linearly Separable



Not Linearly Separable

Measure of Separability: Margin.

For a vector w, and training set S, margin of w wrt S is:

$$\gamma = \min_{(x,y) \in S} \frac{|\langle w, x \rangle|}{\|w\|}$$

Example: $S = ((1, -1), 1), ((-1, -1), 1), ((0.01, 0), 1), ((-1, 0), -1)$

$w = (1, 0)$

* What is the margin of w wrt S? $\|w\| = 1$.

* $\frac{|\langle w, x_1 \rangle|}{\|w\|} = 1$

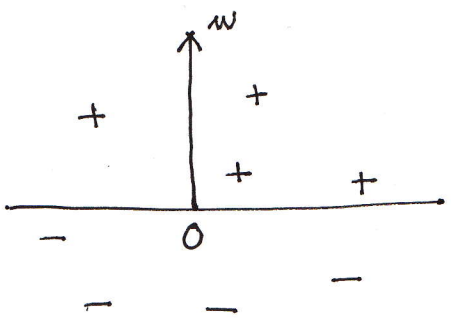
$\frac{|\langle w, x_2 \rangle|}{\|w\|} = 1$

$\frac{|\langle w, x_3 \rangle|}{\|w\|} = 0.01$

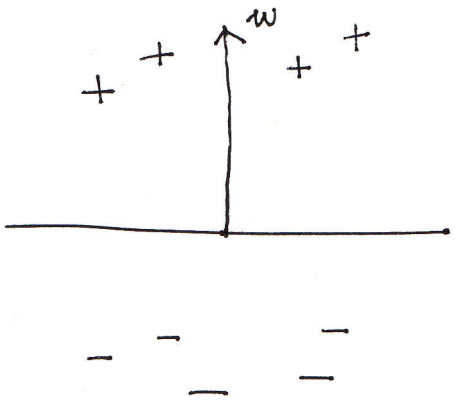
$\frac{|\langle w, x_4 \rangle|}{\|w\|} = 1$.

So, margin = 0.01

Geometrically:



Low Margin Data



High Margin Data

Theorem: If the training data is linearly separable with margin γ, and if $\|x_i\| \leq 1$ for all i in the training set, then, perceptron makes $\leq 1/\gamma^2$ mistakes.

Note:
1. Lower margin $\Rightarrow$ more mistakes.
2. May need >1 pass over training data to get a classifier with no mistakes.

Proof: Let w^* be a linear separator with margin γ on the training set s.t. $\|w^*\| = 1$.

Fact 1: If there is a mistake at round t, $\langle w_{t+1}, w^* \rangle \geq \langle w_t, w^* \rangle + \gamma$

Proof: On a mistake,

$w_{t+1} = w_t + y_t x_t$. So:

$$\langle w_{t+1}, w^* \rangle = \langle w_t + y_t x_t, w^* \rangle = \langle w_t, w^* \rangle + \underbrace{y_t \langle x_t, w^* \rangle}_{\geq \gamma}$$

(Separable wrt $w^* \Rightarrow y_t \langle x_t, w^* \rangle > 0$,

also, $|\langle x_t, w^* \rangle| \geq \gamma$ by definition of margin)

Fact 2: If there is a mistake at round t,

$$\|w_{t+1}\|^2 \leq \|w_t\|^2 + 1.$$

$$\|w_{t+1}\|^2 = \|w_t + y_t x_t\|^2 = \|w_t\|^2 + \underbrace{y_t^2 \|x_t\|^2}_{\leq 1, \text{ as } \|x_t\|^2 \leq 1.} + \underbrace{2 y_t \langle w_t, x_t \rangle}_{\leq 0, \text{ as there is a mistake at round } t}$$

$$\leq \|w_t\|^2 + 1.$$

Suppose there are K mistakes.

After K mistakes,

$$\|w_{t+1}\|^* \leq \sqrt{K}.$$

$$\langle w_{t+1}, w^* \rangle \geq \gamma K$$

Now: $\cos(\text{Angle b/w } w_{t+1}, w^*) = \dfrac{\langle w_{t+1}, w^* \rangle}{\|w^*\| \, \|w_{t+1}\|} \leq 1.$

So, $\dfrac{\gamma K}{\sqrt{K}} \leq 1 \Rightarrow K \leq \dfrac{1}{\gamma^2}$ (Proved).

What if data is not linearly separable?

Ideally, we want to find a linear separator that makes the minimum number of mistakes on training data, but this is NP Hard.

But perceptron (and some other linear classification algorithms) still work when data is almost linearly separable — that is, there are a few mistakes, close to the decision boundary.

However, perceptron will never converge to a single w if the data is not linearly separable, as we make more passes over training data.

Voted and Averaged Perceptron:

Perceptron:

Initially, $m = 1$; $w_1 = 0$

For $t = 1, 2, 3, \ldots$

If $y_t \langle w_m, x_t \rangle \leq 0$ then

$w_{m+1} = w_m + y_t x_t$

$m = m + 1$

Output w_m

Voted Perceptron:

Initially, $m = 1$, $w_1 = 0$, $c_m = 1$

For $t = 1, 2, 3, \ldots$

If $y_t \langle w_m, x_t \rangle \leq 0$ then:

$w_{m+1} = w_m + y_t x_t$

$m = m + 1$

$c_m = 1$

Else:

$c_m = c_m + 1$.

Output $(w_1, c_1), (w_2, c_2), \ldots, (w_m, c_m)$

c_m is a count of how long w_m survived as a classifier

How to classify test example x?

Output:

$$\text{sign}\left(\sum_{i=1}^{m} c_i \, \text{sign}(\langle w_i, x \rangle) \right)$$

A problem with voted perceptron is that we have to store all the classifiers. To solve this, we use the Averaged Perceptron.

Averaged Perceptron uses the same algorithm as voted perceptron, but the classification rule is different.

Averaged Perceptron Classification Rule for test example x:

$$\text{Output} \quad \text{sign}\left(\left\langle \sum_{i=1}^{m} c_i w_i, x \right\rangle\right)$$

Compare with voted perceptron:

$$\text{Output} \quad \text{sign}\left(\sum_{i=1}^{m} c_i \, \text{sign}\langle w_i, x\rangle\right)$$

Example: If $c_1 = c_2 = c_3 = 1$, then:

Averaged rule: $\text{sign}(\langle w_1 + w_2 + w_3, x\rangle)$

Voted rule: Majority sign of $\langle w_1, x\rangle, \langle w_2, x\rangle, \langle w_3, x\rangle$

Perceptron Example:

Training data: $((4,0),1), ((1,1),-1), ((0,1),-1), ((-2,-2),1)$

Initially, $w_1 = 0$, $m = 1$, $c_m = 1$

Round 1: $y_1 \langle w_1, x_1\rangle = 0$.

So: $m = 2$, $c_m = 1$, $w_2 = w_1 + y_1 x_1 = (4,0)$

Round 2: $y_2 \langle w_2, x_2\rangle \leq 0$.

So: $m = 3$, $c_m = 1$, $w_3 = w_2 + y_2 x_2 = (3,-1)$

Round 3: $y_3 \langle w_3, x_3 \rangle > 0$

so: $c_3 = 2$.

Round 4: $y_4 \langle w_3, x_4 \rangle \leq 0$

so: $c_4 = 1$, $m = 4$, $w_4 = w_3 + y_4 x_4 = (1, -3)$

Output:

$((4, 0), 1)$ $((3, -1), 2)$ $((1, -3), 1)$

w_2 c_2 w_3 c_3 w_4 c_4

$w_1 = (0, 0)$ (always)

$c_1 = 1$ (but its value doesn't matter)

Voted Perceptron Rule:

$$\text{sign}\left[\text{sign}\left(\left\langle \begin{bmatrix}0\\0\end{bmatrix}, x\right\rangle\right) + \text{sign}\left(\left\langle \begin{bmatrix}4\\0\end{bmatrix}, x\right\rangle\right) + 2\,\text{sign}\left(\left\langle \begin{bmatrix}3\\-1\end{bmatrix}, x\right\rangle\right) + \text{sign}\left(\left\langle \begin{bmatrix}1\\-3\end{bmatrix}, x\right\rangle\right)\right]$$

x = test example.

Averaged Perceptron Prediction Rule:

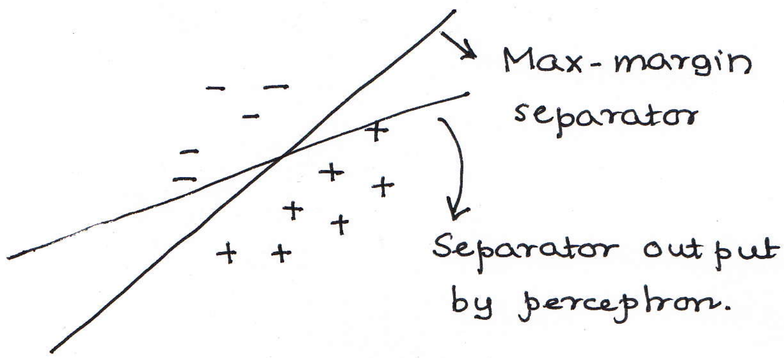
$$\text{sign}\left[\left\langle \begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}4\\0\end{bmatrix} + 2*\begin{bmatrix}3\\-1\end{bmatrix} + \begin{bmatrix}1\\-3\end{bmatrix}, x\right\rangle\right]$$

$$= \text{sign}\left[\left\langle \begin{bmatrix}11\\-5\end{bmatrix}, x\right\rangle\right]$$

Exercise: Is the voted perceptron decision boundary a hyperplane? What about the averaged perceptron decision boundary?

Margins:

- Data linearly separable with a margin.
  - * Perceptron stops once a separator is found, but we may want to compute the max-margin separator.



Max-margin separators are computed by Support Vector Machines (SVMs).

Linear Classification by a Hyperplane not thru Origin

We can transform this problem to linear classification by a hyperplane through the origin.

Original Problem:

Training data: (x_i, y_i), $i = 1, .., n$, $x_i \in R^d$, $y_i \in \{-1, 1\}$

Separating hyperplane: $a^T x + b \geq 0$

$(a \in R^d, \; b = \text{a scalar})$

Transform it to:

Training data: (z_i, y_i), $i = 1, .., n$, $z_i = \begin{bmatrix} 1 \\ x_i \end{bmatrix} \in R^{d+1}$, $y_i \in \{-1, 1\}$