Lecture 5: Linear Classification

Classification: Given labelled data

\[ (x_i, y_i), i = 1, \ldots, n \]

feature \rightarrow \downarrow \label{discrete value}

vector

find a rule to predict \( y \) for unseen \( x \).

For linear classification:

- assume \( y \) is +1 or -1
- classification rule is a hyperplane that separates +1 from -1

How to represent this hyperplane?

* For now, assume hyperplane is through origin

\[ w \]

\[ w \] = normal vector to the hyperplane.

\[ \langle w, x \rangle > 0 \]

For a point \( x \) on the + side,

\[ \langle w, x \rangle < 0 \]

For a point \( x \) on the - side,

So:

1. Classification rule: represented by a vector \( w \)
2. Given a test example \( x \), output its label as:
   \[ y = \text{sign} \left( \langle w, x \rangle \right) \]
3. Given training data, find a \( w \) that largely
   \[ \text{separates the } (x_i, +) \text{ training points on one side,} \]
   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
   
   \[ w_{t+1} = w_t + y_t \cdot x_t \]
   
   Else
   
   \[ w_{t+1} = w_t \]

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

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

- **Case 2:** \( y_t = -1 \). Then: \( \langle w_t, x_t \rangle > 0 \)

   * \( x_t \)'s label is +
   * Yet angle b/w \( x_t \) and \( w_t \) is \( > 90^\circ \) so \( x_t \) is not on the correct side of the hyperplane corresponding to \( w_t \).

   * \( w_t \) 's Hyperplane

   - side
   
   \[ w_t \]

   + side

   \[ x_t \]

   - side

   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: \( W_{t+1} = W_t + \gamma_t x_t \).

**Case 1:** \( \gamma_t = 1; \) \( W_{t+1} = W_t + x_t \)

- \( W_{t+1} \) moves "closer to" \( x_t \)
- or
- \( x_t \) moves towards the + side of the decision boundary (hyperplane)

**Case 2:** \( \gamma_t = -1; \) \( W_{t+1} = W_t - x_t \)

- \( W_{t+1} \) 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 \( W_t \) as \( W_{t+1} = W_t + \gamma_t x_t \). Is it possible for \( W_{t+1} \) to also make a mistake on \((x_t, y_t)\)?
Example:

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

Round 1:
* \(w_1 = 0\)
* \(y_t < w_t, x_t >\) for \(t = 1\)
  
  \[= 0\] as well.

* \(w_2 = w_1 + y_1 x_4\)
  
  \[= (4,0)\]

Round 2:
* \(y_2 < w_2, x_2 >\) \(\ast < 0\)
* So \(w_3 = w_2 + y_2 x_2\)
  
  \[= (4,0) - (1,1) = (3,-1)\]

Round 3:
* \(y_3 < w_3, x_3 >\) \(\ast > 0\)

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

Round 4:
* \(y_4 < w_4, x_4 >\) \(\ast < 0\)
* So \(w_5 = w_4 + y_4 x_4\)
  
  \[= (3,-1) + (-2,-2)\]

  \[= (1,-3)\]
Example 2: Training data:

\[(1, 1), (1, -1), (-1, 1), (-1, -1)\]

Initially, \(w_1 = 0\). In round 1,

* \(\gamma_1 < w_1, x_1 \geq 0\), so mistake.
* \(w_2 = w_1 + \gamma_1 x_1 = (1, 1)\)

Round 2:

* \(\gamma_2 < w_2, x_2 \geq 0\)
* \(w_3 = w_2 + \gamma_2 x_2 = (1, 1) - (1, -1) = (0, 2)\)

Round 3:

* \(\gamma_3 < w_3, x_3 \geq 0\)
* \(w_4 = w_3 + \gamma_3 x_3 = (0, 2) * (-1, 1) = (1, 1)\)

Round 4:

* \(\gamma_4 < w_4, x_4 \geq 0\)
* \(w_5 = w_4 + \gamma_4 x_4 = (1, 1) + (-1, -1) = 0\)
When does Perceptron Converge?

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

\[ \begin{align*}
+ & + + \\
- & - - \\
\end{align*} \]

Linearly Separable

\[ \begin{align*}
+ & + + \\
- & - \\
\end{align*} \]

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 \( S \)? \( \|w\| = 1 \).

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

So, margin = 0.01

* \[ \frac{|\langle w, x_2 \rangle|}{\|w\|} = 1 \]

* \[ \frac{|\langle w, x_3 \rangle|}{\|w\|} = 0.01 \]

* \[ \frac{|\langle w, x_4 \rangle|}{\|w\|} = 1 \]
Geometrically:

- Low Margin Data
- High Margin Data

Theorem: If the training data is linearly separable with margin $\gamma$, and if $\|\mathbf{x}_i\| \leq 1$ 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 $\mathbf{w}^*$ be a linear separator with margin $\gamma$ on the training set s.t. $\|\mathbf{w}^*\| = 1$.

Fact 1: If there is a mistake at round $t$, $\langle \mathbf{w}_t + \mathbf{x}_t, \mathbf{w}^* \rangle = \langle \mathbf{w}_t, \mathbf{w}^* \rangle + \gamma$

Proof: On a mistake,

$\mathbf{w}_{t+1} = \mathbf{w}_t + y_t \mathbf{x}_t$.

So:

$\langle \mathbf{w}_{t+1}, \mathbf{w}^* \rangle = \langle \mathbf{w}_t + y_t \mathbf{x}_t, \mathbf{w}^* \rangle = \langle \mathbf{w}_t, \mathbf{w}^* \rangle + y_t \langle \mathbf{x}_t, \mathbf{w}^* \rangle$

(Separable wrt $\mathbf{w}^*$ $\Rightarrow y_t \langle \mathbf{x}_t, \mathbf{w}^* \rangle > 0$, $\geq \gamma$

also, $|\langle \mathbf{x}_t, \mathbf{w}^* \rangle| > \gamma$ by definition of margin)
Fact 2: If there is a mistake at round $t$,

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

$$||w_{th}||^2 = ||w_t + y_t x_t||^2 = (||w_t||^2 + y_t^2 ||x_t||^2 + 2y_t \langle w_t, x_t \rangle \leq 1, \text{ as } ||x_t||^2 \leq 1. \text{ a mistake at round } t$$

Suppose there are $K$ mistakes.

After $K$ mistakes,

$$||w_{th}||^* \leq \sqrt{K}.$$ 

$$\langle w_{th}, w^* \rangle \geq \gamma K$$

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

So, \(\frac{\gamma K}{\sqrt{K}} \leq 1 \Rightarrow K \leq \frac{1}{\gamma^2}\) (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.
**Noted and Averaged Perceptron:**

**Perceptron:**

Initially, \( m = 0 \); \( w_1 = 0 \)

For \( t = 1, 2, 3, \ldots \)

If \( y_t < w_m^t \cdot x_t \leq 0 \) then

\[
\begin{align*}
\omega_{m+1} &= \omega_m + y_t x_t \\
\omega_m &= \omega_m + y_t x_t \\
m &= m + 1
\end{align*}
\]

Output \( \omega_m \)

**Voted Perceptron:**

Initially, \( m = 1 \), \( \omega_1 = 0 \), \( c_m = 1 \)

For \( t = 1, 2, 3, \ldots \)

If \( y_t < w_m^t \cdot x_t \leq 0 \) then:

\[
\begin{align*}
\omega_{m+1} &= \omega_m + y_t x_t \\
m &= m + 1 \\
c_m &= 1
\end{align*}
\]

Else:

\[
\begin{align*}
c_m &= c_m + 1
\end{align*}
\]

Output \((\omega_1, c_1), (\omega_2, c_2), \ldots, (\omega_m, c_m)\)

\( c_m \) is a count of how long \( \omega_m \) survived as a classifier

**How to classify test example \( x \)?**

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

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

Compare with voted perceptron:

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

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

- **Averaged rule:** $\text{sign} \left( \langle w_1 + w_2 + w_3, x \rangle \right)$
- **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: \[ y_3 < w_3, x_3 > > 0 \]
so: \[ c_3 = 2. \]

Round 4: \[ \gamma_4 < w_3, x_4 > \leq 0 \]
so: \[ c_4 = 1, m = h, w_4 = w_3 + \gamma_4 x_4 = (1, -3) \]

Output:
\[
\begin{array}{cccc}
(4, 0), & 1 ) & (3, -1), & 2 ) & (1, -3), & 1 ) \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\begin{array}{cccc}
w_2 & c_2 & w_3 & c_3 & w_4 & c_4 \\
\end{array}
\end{array}
\]

\[ w_1 = (0, 0) \] (always)
\[ c_1 = 1 \] (but its value doesn't matter)

**Noted Perceptron Rule:**

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

\[ x = \text{test example}. \]

**Averaged Perceptron Prediction Rule:**

\[ \text{sign} \left[ < \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \end{bmatrix} + 2 \times \begin{bmatrix} -1 \\ -3 \end{bmatrix}, x > \right] \]

\[ = \text{sign} \left[ < \begin{bmatrix} -11 \\ -5 \end{bmatrix}, x > \right] \]

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: \((x_i, y_i), \ i = 1, \ldots, n, \ x_i \in \mathbb{R}^d, \ y_i \in \{-1, 1\}\)
Separating hyperplane: \(a^T x + b > 0\)

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

**Transform it to:**
Training data: \((z_i, y_i), \ i = 1, \ldots, n, \ z_i = [1, x_i] \in \mathbb{R}^{d+1}, \ y_i \in \{-1, 1\}\)
Separating hyperplane: \([a_1, a_x] \in \mathbb{R}^{d+1}\)