Lecture 5: Linear Classification

Classification: Given labelled data

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

- feature \rightarrow \text{vector}
- label (discrete value)

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.

\[w = \text{normal vector to the hyperplane.}\]

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 \(w\) that largely
has \(\langle x_i, + \rangle\) training points on one side,
and the \(\langle x_i, - \rangle\) ones on another.
The Perceptron Algorithm:

1. Initially, \( w_1 = 0 \).
2. For \( t = 1, 2, 3, \ldots \):
   \[
   \text{If } y_t \langle w_t, x_t \rangle \leq 0 \text{ then}
   \begin{align*}
   w_{t+1} &= w_t + y_t x_t \\
   \text{Else}
   w_{t+1} &= w_t
   \end{align*}
   \]

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

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

- Side: \( w_t \)', Hyperplane: \( w_t \)'

  - Side.

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

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

- Side: \( w_t \)', Hyperplane: \( w_t \)'

  - Side.

  * \( x_t \)'s label is -
  * \( x_t \)'s angle b/w \( x_t \) and \( w_t \) is \( \leq 90^\circ \); \( x_t \) 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: "\( w_{t+1} = w_t + \gamma_t x_t \)."

**Case 1:** \( \gamma_t = 1; \quad w_{t+1} = w_t + x_t \)

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

**Case 2:** \( \gamma_t = -1; \quad w_{t+1} = w_t - x_t \)

\( w_{t+1} \) 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 \( w_{t+1} = w_t + \gamma_t x_t \). 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:**
- \(\mathbf{w}_1 = 0\)
- \(y_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle \) for \(t = 1\)
  - \(= 0\) as well.
- \(\mathbf{w}_2 = \mathbf{w}_1 + y_1 \mathbf{x}_1\)
  - \(= (4, 0)\)

---

**Round 2:**
- \(y_2 \langle \mathbf{w}_2, \mathbf{x}_2 \rangle \) \(< 0\)
- So \(\mathbf{w}_3 = \mathbf{w}_2 + y_2 \mathbf{x}_2\)
  - \(= (4, 0) - (1, 1) = (3, -1)\)

---

**Round 3:**
- \(y_3 \langle \mathbf{w}_3, \mathbf{x}_3 \rangle \) \(> 0\)
- Correct. \(\mathbf{w}_4 = \mathbf{w}_3 = (3, -1)\)

---

**Round 4:**
- \(y_4 \langle \mathbf{w}_4, \mathbf{x}_4 \rangle \) \(< 0\)
- So \(\mathbf{w}_5 = \mathbf{w}_4 + y_4 \mathbf{x}_4\)
  - \(= (3, -1) + (-2, -2)\)
  - \(= (1, -3)\)
Example 2: Training data:

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

Initially, \(w_1 = 0\). In round 1,
* \(y_1 < w_1, x_1 \geq 0\), so mistake.
* \(w_2 = w_1 + y_1 x_1 = (1, 1)\)

Round 2:
* \(y_2 < w_2, x_2 \geq 0\)
* \(w_3 = w_2 + y_2 x_2 = (1, 1) - (1, -1) = (0, 2)\)

Round 3:
* \(y_3 < w_3, x_3 \geq 0\)
* \(w_4 = w_3 + y_3 x_3 = (0, 2) - (-1, 1) = (1, 1)\)

Round 4:
* \(y_4 < w_4, x_4 \geq 0\)
* \(w_5 = w_4 + y_4 x_4 = (1, 1) + (-1, -1) = 0\)
When does Perceptron Converge?

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

\[
\begin{array}{cccc}
+ & + & + \\
- & - & -
\end{array}
\]

Linearly Separable

Not Linearly Separable

Measure of Separability: Margin.

For a vector \( \mathbf{w} \), and training set \( S \), margin of \( \mathbf{w} \) wrt \( S \) is:

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

Example: \( S = \{ (1, -1), 1), (\mathbf{-1}, -1), 1), (\mathbf{0.01}, 0), 1), (\mathbf{-1}, 0), -1) \)

\( \mathbf{w} = (1, 0) \)

What is the margin of \( \mathbf{w} \) wrt \( S \)? \( \| \mathbf{w} \| = 1 \).

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

So, margin = 0.01

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

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

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

```
+ + + w
+ + +
+ + +
- 0 -
- - -
```

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 \( \leq \frac{1}{\gamma^2} \) 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 \( \gamma \) on the training set s.t. \( \|w^*\| = 1 \).

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

**Proof:** On a mistake,
\[
w_{t+1} = w_t + y_t x_t . \text{ So:}
\]
\[
\langle w_{t+1}, w^* \rangle = \langle w_t + y_t x_t, w^* \rangle = \langle w_t, w^* \rangle + y_t \langle x_t, w^* \rangle
\]

(Separable wrt \( w^* \Rightarrow y_t \langle x_t, w^* \rangle > 0 \), \( \geq \gamma \)

Also, \( |\langle x_t, w^* \rangle| \geq \gamma \) by definition of margin)
Fact 2: If there is a mistake at round $t$,
\[ \|w_{t+1}\|^2 \leq \|w_t\|^2 + 1. \]

\[ \|w_{t+1}\|^2 = \|w_t + y_t x_t\|^2 = \|w_t\|^2 + y_t^2 \|x_t\|^2 + 2y_t \langle w_t, x_t \rangle \leq 1, \text{ as } \|x_t\|^2 \leq 1. \]

\[ \leq \|w_t\|^2 + 1. \]

Suppose there are $K$ mistakes.
After $K$ mistakes,
\[ \|w_{t+1}\|^u \leq \sqrt{K}. \]
\[ \langle w_{t+1}, w^\star \rangle \geq \gamma K \]

Now: \[ \cos (\text{Angle b/w } w_{t+1}, w^\star) = \frac{\langle w_{t+1}, w^\star \rangle}{\|w^\star\| \|w_{t+1}\|} \leq 1. \]

So, \[ \frac{\gamma K}{\sqrt{K}} \leq 1 \Rightarrow K \leq \frac{1}{\gamma^2} \quad \text{(Proved).} \]

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 \( t = 1, 2, 3, \ldots \)
\[
\text{If } \ w_t \cdot x_t \leq 0 \text{ then } \\
\quad \begin{align*}
\quad w_{m+1} &= w_m + y_t x_t \\
\quad m &= m + 1
\end{align*}
\]
Output \( w_m \)

Voted Perceptron:
Initially, \( m = 1; \ w_1 = 0, c_m = 1 \)
For \( t = 1, 2, 3, \ldots \)
\[
\text{If } \ w_t \cdot x_t \leq 0 \text{ then: } \\
\quad \begin{align*}
\quad w_{m+1} &= w_m + y_t x_t \\
\quad m &= m + 1 \\
\quad c_m &= 1
\end{align*}
\]
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:
\[
\text{sign} \left( \sum_{i=1}^{m} c_i \text{ sign}(w_i \cdot x) \right)
\]


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 \( \mathbf{x} \):**

\[
\text{Output: } \text{sign} \left( \sum_{i=1}^{m} c_i \langle \mathbf{w}_i, \mathbf{x} \rangle \right)
\]

Compare with voted perceptron:

\[
\text{Output: } \text{sign} \left( \sum_{i=1}^{m} c_i \text{sign} \langle \mathbf{w}_i, \mathbf{x} \rangle \right)
\]

**Example:** If \( c_1 = c_2 = c_3 = 1 \), then:

Averaged rule: \( \text{sign} \left( \langle \mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3, \mathbf{x} \rangle \right) \)

Voted rule: Majority sign of \( \langle \mathbf{w}_1, \mathbf{x} \rangle, \langle \mathbf{w}_2, \mathbf{x} \rangle, \langle \mathbf{w}_3, \mathbf{x} \rangle \)

**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:** \( y_1 \langle \mathbf{w}_1, \mathbf{x}_1 \rangle = 0 \).

So: \( m = 2, c_m = 1, \mathbf{w}_2 = \mathbf{w}_1 + y_1 \mathbf{x}_1 = (4,0) \)

**Round 2:** \( y_2 \langle \mathbf{w}_2, \mathbf{x}_2 \rangle \leq 0 \).

So: \( m = 3, c_m = 1, \mathbf{w}_3 = \mathbf{w}_2 + y_2 \mathbf{x}_2 = (3,-1) \)
Round 3: \( y_3 < w_3, x_3 > > 0 \)
so: \( c_3 = 2 \).

Round 4: \( y_4 < w_3, x_4 > \leq 0 \)
so: \( c_4 = 1 \), \( m = 6 \), \( w_4 = w_3 + y_4 x_4 = (1, -3) \)

Output: \[( (4, 0), 1 ) \quad ( (3, -1), 2 ) \quad ( (1, -3), 1 ) \]
\[ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \]
\[ w_2 \quad c_2 \quad w_3 \quad c_3 \quad w_4 \quad c_4 \]

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

**Noted Perceptron Rule:**
\[
\text{sign} \left( \frac{\text{sign}(<0, x>) + \text{sign}(<4, x>) + 2 \text{sign}(<-3, x>) \quad + \text{sign}(<-3, x>)]}{2}
\]

\( x \) = test example.

**Averaged Perceptron Prediction Rule:**
\[
\text{sign}(<0 + 4 + 2 \times -3 + -3, x>)
\]
= \[
\text{sign}(<-0.5, x>)
\]

**Exercise:** Is the voted perceptron decision boundary a hyperplane?

What about the averaged perceptron decision boundary?