The Perceptron Algorithm
Announcements

1. P2 (Perceptron) will be out tmr
Recap on PCA

T/F: we need to center the dataset before we run PCA
Recap on PCA

T/F: we need to center the dataset before we run PCA

Q: How to pick the parameter K in PCA?

\[ XX^T = U \Lambda U^T \]
Perceptron, 1957

Predecessor of deep networks.

Separating two classes of objects using a linear threshold classifier.

Frank Rosenblatt @ Cornell!
"Later perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted."

Today

Objective: learn our first (binary) classification algorithm and understand why it works
Outline

1. Linear binary Classifier

2. Algorithm

3. Proof of why it works
Linear classifier

Binary classification setting: $x \in \mathbb{R}^d, y = \{-1, +1\}$
Linear classifier

Binary classification setting: \( x \in \mathbb{R}^d, y = \{-1, +1\} \)

Hyperplane \( H = \{x : w^\top x + b = 0\} \)
Linear classifier

Binary classification setting: \( x \in \mathbb{R}^d, y = \{-1, +1\} \)

Hyperplane \( H = \{x : w^T x + b = 0\} \)

\( w \): weight vector, wlog assume \( \|w\|_2 = 1 \)
Linear classifier

Binary classification setting: $x \in \mathbb{R}^d, y = \{-1, +1\}$

Hyperplane $H = \{x : w^T x + b = 0\}$

- $w$: weight vector, wlog assume $\|w\|_2 = 1$
- $b$: bias term; $|b|$ determines the distance of the hyperplane to origin
Binary classification setting: $x \in \mathbb{R}^d, y = \{-1, +1\}$

A Hyperplane defines a binary linear classifier 

$\text{sign}(w^T x + b)$
Setting

We often assume data \( \{x_i, y_i\}_{i=1}^n \) is linearly separable,

\[ y_i \in \{-1, +1\} \]

i.e., \( \exists w^*, b^* \), such that

\[ \text{sign}((w^*)^T x_i + b^*) = \text{sign}(y_i), \forall i \]
Setting

We often assume data \( \{x_i, y_i\}_{i=1}^n \) is linearly separable,

i.e., \( \exists w^*, b^* \), such that

\[
\text{sign}((w^*)^T x_i + b^*) = \text{sign}(y_i), \forall i
\]

Or equivalently,

\[
y_i((w^*)^T x_i + b^*) > 0, \forall i
\]

\[
\forall i \left( (w^*)^T x_i + b^* \right) < 0
\]
Linear classifier

Absorbing the bias term into the feature vector

\[ w^T x + b = \begin{bmatrix} w \\ b \end{bmatrix}^T \begin{bmatrix} x \\ 1 \end{bmatrix} \]

\[ \text{sign} (w^T x + b) = \text{sign} (\hat{w}^T \hat{x}) \]
Linear classifier

Absorbing the bias term into the feature vector

\[ w^\top x + b = \begin{bmatrix} w \\ b \end{bmatrix}^\top \begin{bmatrix} x \\ 1 \end{bmatrix} \]

Throughout the semester, we will assume feature \( x \) in default contains the constant 1
Outline

1. Linear binary Classifier

2. Algorithm

3. Proof of why it works
The learning protocol

Consider the **online learning** setting where every iteration $t$, a pair $(x_t, y_t)$ shows up

For $t = 0 \rightarrow \infty$
The learning protocol

Consider the online learning setting where every iteration $t$, a pair $(x_t, y_t)$ shows up

For $t = 0 \rightarrow \infty$

- New feature $x_t$ shows up
The learning protocol

Consider the **online learning** setting where every iteration $t$, a pair $(x_t, y_t)$ shows up.

For $t = 0 \rightarrow \infty$

- New feature $x_t$ shows up
- Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
The learning protocol

Consider the **online learning** setting where every iteration $t$, a pair $(x_t, y_t)$ shows up.

For $t = 0 \to \infty$

- New feature $x_t$ shows up
- Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
- Check if $\hat{y}_t = y_t$
The learning protocol

Consider the **online learning** setting where every iteration $t$, a pair $(x_t, y_t)$ shows up

For $t = 0 \rightarrow \infty$

<table>
<thead>
<tr>
<th>New feature $x_t$ shows up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$</td>
</tr>
<tr>
<td>Check if $\hat{y}_t = y_t$</td>
</tr>
<tr>
<td>Alg updates $w_{t+1}$</td>
</tr>
</tbody>
</table>
The learning protocol

Consider the **online learning** setting where every iteration $t$, a pair $(x_t, y_t)$ shows up.

For $t = 0 \rightarrow \infty$

- New feature $x_t$ shows up
- Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
- Check if $\hat{y}_t = y_t$
- Alg updates $w_{t+1}$

Goal: make # of mistakes $\sum_{t=0}^{\infty} 1(\hat{y}_t \neq y_t)$ as small as possible
The learning protocol

Consider the **online learning** setting where every iteration $t$, a pair $(x_t, y_t)$ shows up

For $t = 0 \rightarrow \infty$

- New feature $x_t$ shows up
- Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
- Check if $\hat{y}_t = y_t$
- Alg updates $w_{t+1}$

Perceptron tells us how to do this update!

Goal: make # of mistakes $\sum_{t=0}^{\infty} 1(\hat{y}_t \neq y_t)$ as small as possible
The Algorithm

Initialize $w_0 = 0$

For $t = 0 \rightarrow \infty$

- New feature $x_t$ shows up
- Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
- Check if $\hat{y}_t = y_t$
The Algorithm

Initialize \( w_0 = 0 \)

For \( t = 0 \to \infty \)

- New feature \( x_t \) shows up
- Alg makes a prediction \( \hat{y}_t = \text{sign}(w_t^T x_t) \)
- Check if \( \hat{y}_t = y_t \)
- Alg updates \( w_{t+1} = w_t + 1(\hat{y}_t \neq y_t)y_tx_t \)
The Algorithm

Initialize $w_0 = 0$

For $t = 0 \rightarrow \infty$

New feature $x_t$ shows up
Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
Check if $\hat{y}_t = y_t$
Alg updates $w_{t+1} = w_t + 1(\hat{y}_t \neq y_t)y_t x_t$

Case 1: $\hat{y}_t = y_t$, $w_{t+1} = w_t$
The Algorithm

Initialize $w_0 = 0$

For $t = 0 \to \infty$

\[ w_{t+1} = w_t + \mathbf{1}(\hat{y}_t \neq y_t)y_t x_t \]

New feature $x_t$ shows up
Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
Check if $\hat{y}_t = y_t$

Case 1: $\hat{y}_t = y_t$, $w_{t+1} = w_t$

Case 2: $\hat{y}_t \neq y_t$ (e.g., $\hat{y}_t = -1, y_t = 1$)

$w_{t+1} = w_t + \mathbf{1}(\hat{y}_t \neq y_t)y_t x_t$

$\mathbf{1}(\hat{y}_t \neq y_t)y_t x_t$

$w_{t+1} X_e - w_t X_e$

$X_e^T X_e > 0$

$w_t X_e$ was negative
The Algorithm

Initialize $w_0 = 0$

For $t = 0 \rightarrow \infty$

New feature $x_t$ shows up
Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
Check if $\hat{y}_t = y_t$
Alg updates $w_{t+1} = w_t + 1(\hat{y}_t \neq y_t)y_t x_t$

Case 1: $\hat{y}_t = y_t$, $w_{t+1} = w_t$

Case 2: $\hat{y}_t \neq y_t$ (e.g., $\hat{y}_t = -1, y_t = 1$)

$w_{t+1}^T x_t - w_t^T x_t = (x_t^T x_t)$
The Algorithm

Initialize $w_0 = 0$

For $t = 0 \rightarrow \infty$

<table>
<thead>
<tr>
<th>New feature $x_t$ shows up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$</td>
</tr>
<tr>
<td>Check if $\hat{y}_t = y_t$</td>
</tr>
<tr>
<td>Alg updates $w_{t+1} = w_t + 1(\hat{y}_t \neq y_t)y_t x_t$</td>
</tr>
</tbody>
</table>

Case 1: $\hat{y}_t = y_t$, $w_{t+1} = w_t$

Case 2: $\hat{y}_t \neq y_t$ (e.g., $\hat{y}_t = -1, y_t = 1$)

$$w_{t+1}^T x_t - w_t^T x_t = (x_t^T x_t)$$

Value of $w_{t+1}^T x_t$ is increased (the correct progress)
The Algorithm

Initialize $w_0 = 0$

For $t = 0 \rightarrow \infty$

New feature $x_t$ shows up
Alg makes a prediction $\hat{y}_t = \text{sign}(w_t^T x_t)$
Check if $\hat{y}_t = y_t$
Alg updates $w_{t+1} = w_t + 1(\hat{y}_t \neq y_t)y_t x_t$

Case 1: $\hat{y}_t = y_t$, $w_{t+1} = w_t$

Case 2: $\hat{y}_t \neq y_t$ (e.g., $\hat{y}_t = -1, y_t = 1$)

$w_{t+1}^T x_t - w_t^T x_t = (x_t^T x_t)$

Value of $w_{t+1}^T x_t$ is increased (the correct progress)

Q: what happens when $\hat{y}_t = 1, y_t = -1$
A Geometric explanation

When we make a mistake, i.e., $y_t(w_t^T x_t) < 0$ (e.g., $y_t = -1$, $w_t^T x_t > 0$)
A Geometric explanation

When we make a mistake, i.e., $y_t(w_t^T x_t) < 0$ (e.g., $y_t = -1$, $w_t^T x_t > 0$)

Q: What does $w^*$ look like?
A Geometric explanation

When we make a mistake, i.e., $y_t(w_t^T x_t) < 0$ (e.g., $y_t = -1$, $w_t^T x_t > 0$)

Q: What does $w^*$ look like?

$w_{t+1} = w_t + y_t x_t$

$\text{sign}(w^* x_t) < 0$
A Geometric explanation

When we make a mistake, i.e., $y_t(w_t^T x_t) < 0$ (e.g., $y_t = -1$, $w_t^T x_t > 0$)

Q: What does $w^*$ look like?
A Geometric explanation

When we make a mistake, i.e., $y_t(w_t^T x_t) < 0$ (e.g., $y_t = -1$, $w_t^T x_t > 0$)

Q: What does $w^*$ look like?
A Geometric explanation

When we make a mistake, i.e., \( y_t(w_t^\top x_t) < 0 \) (e.g., \( y_t = -1, \ w_t^\top x_t > 0 \))

We should track how the \( \cos(\theta_t) \) is changing:

\[
\cos(\theta_t) = \frac{w_t^\top w^*}{\|w_t\|_2}
\]

Q: What does \( w^* \) look like?
Outline

1. Linear binary Classifier
2. Algorithm
3. Proof of why it works
Main theorem
Main theorem

Theorem of Perceptron:

Assume $\|x_t\|_2 \leq 1, \forall t$. If there exists $w^*$ with $\|w^*\|_2 = 1$, such that $y_t(x_t^T w^*) \geq \gamma > 0, \forall t$, where $\gamma$ is a margin.
Main theorem

Theorem of Perceptron:

Assume $\|x_t\|_2 \leq 1, \forall t$. If there exists $w^*$ with $\|w^*\|_2 = 1$, such that $y_t(x_t^T w^*) \geq \gamma > 0, \forall t,$

then:

$$\sum_{t=0}^{\infty} 1(\hat{y}_t \neq y_t) \leq 1/\gamma^2$$
Proof of the theorem

\[ \cos(\theta_t) = \frac{w_t^\top w^*}{\|w_t\|_2} \]
Proof of the theorem

\[ \cos(\theta_t) = \frac{w_t^T w^*}{\|w_t\|_2} \]

Assume we make a mistake at \( x_t \), track how the denominator and numerator change.
Proof of the theorem

1. Track $w_t^TW^*$
Proof of the theorem

1. Track $w_t^T w^*$

$$w_{t+1}^T w^* = (w_t + y_t x_t)^T w^*$$
Proof of the theorem

1. Track $w_t^T w^*$

$$w_{t+1}^T w^* = (w_t + y_t x_t)^T w^*$$

$$= w_t^T w^* + y_t x_t^T w^*$$

$$\gamma_t (w^T x_t) \geq \delta$$
Proof of the theorem

1. Track $w_{t+1}^T w^*$

$$w_{t+1}^T w^* = (w_t + y_t x_t)^T w^*$$

$$= w_t^T w^* + y_t x_t^T w^*$$

$$\geq w_t^T w^* + \gamma$$
Proof of the theorem

1. Track $w_t^T w^*$

\[
\begin{align*}
w_{t+1}^T w^* &= (w_t + y_t x_t)^T w^* \\
&= w_t^T w^* + y_t x_t^T w^* \\
&\geq w_t^T w^* + \gamma
\end{align*}
\]

Whenever we make a mistake, $w_t^T w^*$ at least increased by $\gamma$
Proof of the theorem

2. Track $w_t^T w_t$

\[ w_{t+1}^T w_{t+1} = (w_t + y_t x_t)^T (w_t + y_t x_t) = \] 

\[ = w_t^T w_t + 2 w_t^T (x_t y_t) + x_t^T x_t \]

\[ \leq w_t^T w_t + 1 \]

Discuss this derivation in small group for 5 minutes!
Proof of the theorem

3. What is $\cos(\theta_t) = \frac{w_t^T w^*}{\sqrt{w_t^T w_t}}$ if we have made $M$ mistakes?

After make $M$ mistakes:
Proof of the theorem

3. What is $\cos(\theta_t) = \frac{w_t^T w^*}{\sqrt{w_t^T w_t}}$ if we have made $M$ mistakes?

After make $M$ mistakes:

$$w_t^T w^* \geq M \gamma$$
Proof of the theorem

3. What is $\cos(\theta_t) = \frac{w_t^T w^*}{\sqrt{w_t^T w_t}}$ if we have made $M$ mistakes?

After make $M$ mistakes:

$$w_t^T w^* \geq M\gamma$$

$$w_t^T w_t \leq M$$

$$1 \geq \frac{w_t^T w^*}{\|w_t\|^2} \geq \frac{m\gamma}{\sqrt{m}}$$
Proof of the theorem

3. What is \( \cos(\theta_t) = \frac{w_t^T w^*}{\sqrt{w_t^T w_t}} \) if we have made \( M \) mistakes?

After make \( M \) mistakes:

\[
\begin{align*}
    w_t^T w^* &\geq M\gamma \\
    w_t^T w_t &\leq M \\
    \cos(\theta_t) &\geq (M\gamma)/\sqrt{M} = \sqrt{M\gamma}
\end{align*}
\]
Proof of the theorem

3. What is $\cos(\theta_t) = \frac{w_t^T w^*}{\sqrt{w_t^T w_t}}$ if we have made $M$ mistakes?

After make $M$ mistakes:

$$w_t^T w^* \geq M\gamma$$

$$w_t^T w_t \leq M$$

$$1 \geq \cos(\theta_t) \geq \frac{(M\gamma)/\sqrt{M}}{\sqrt{M}} = \sqrt{M\gamma}$$

$$\Rightarrow M \leq \frac{1}{\gamma^2}$$
Summary

The Perceptron algorithm:

1. Binary classification algorithm, runs in online mode, makes update when makes a mistake

   (See lecture note for how to apply Perceptron on a static dataset)

2. Total # of mistakes is bounded by a constant \((1/\gamma^2)\)