10-607 Computational Foundations for Machine Learning

Perceptron Mistake Bound

Instructor: Pat Virtue

Plan

Perceptron Background

- ML Data, Tasks, Notation
- Vectors, dot products
- Geometry with linear functions
- Perceptron (and the sign() function)
- Decision boundaries and errors

Perceptron Algorithm

Perceptron Mistake Bound Theory

- Background: projections, distances, and margin

ML Data, Tasks, Notation

Notation alert!

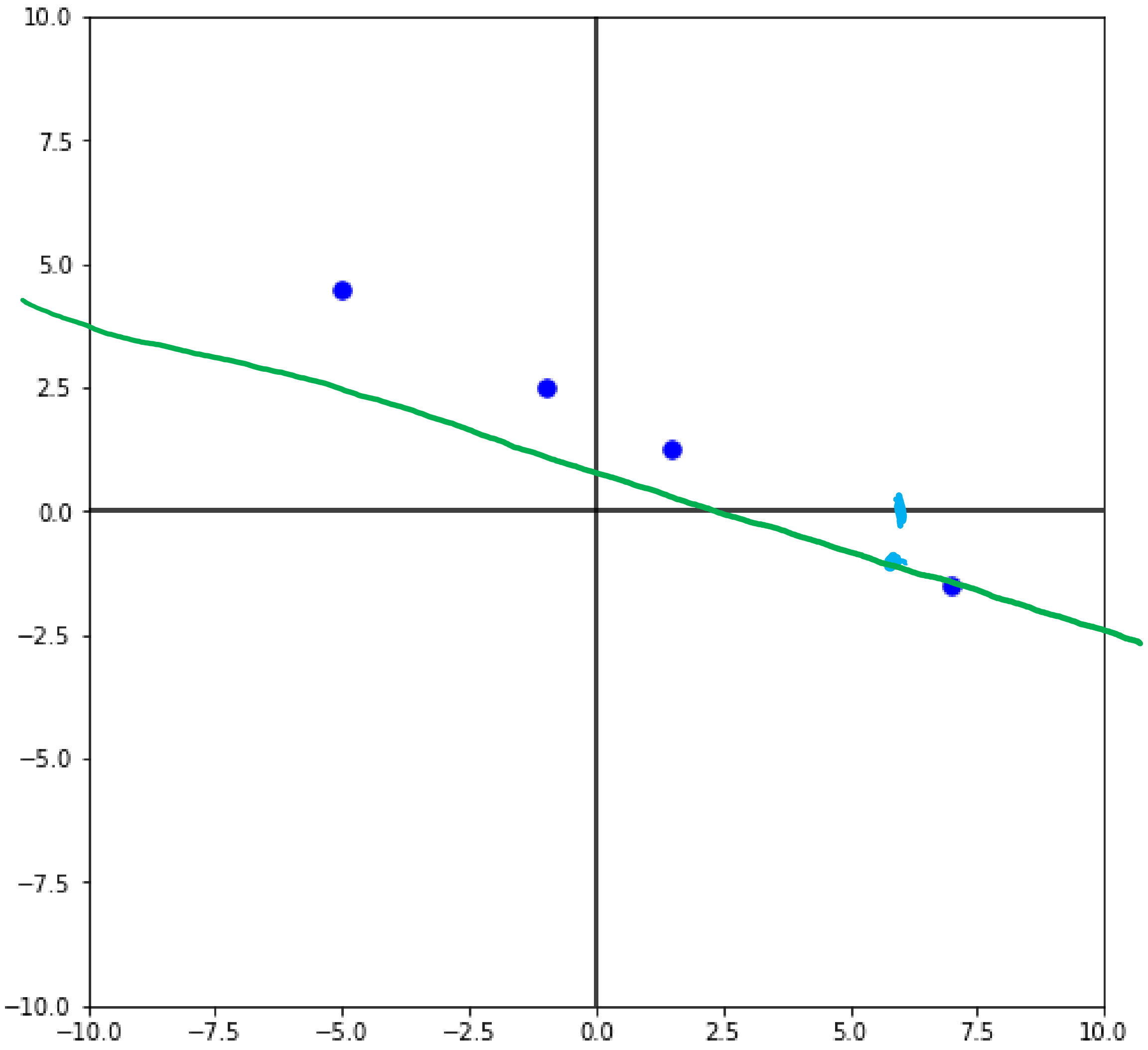
Regression

$$\mathcal{D} = \{(x^{(i)}, \quad y^{(i)})\}_{i=1}^{4}$$
$$= \{(-1, \quad 2.5),$$
$$(\quad 7, \ -1.5),$$
$$(-5, \quad 4.5),$$
$$(1.5, \quad 1.25)\}$$

$y = f(x)$

$= w x + b$

parameters



ML Data, Tasks, Notation

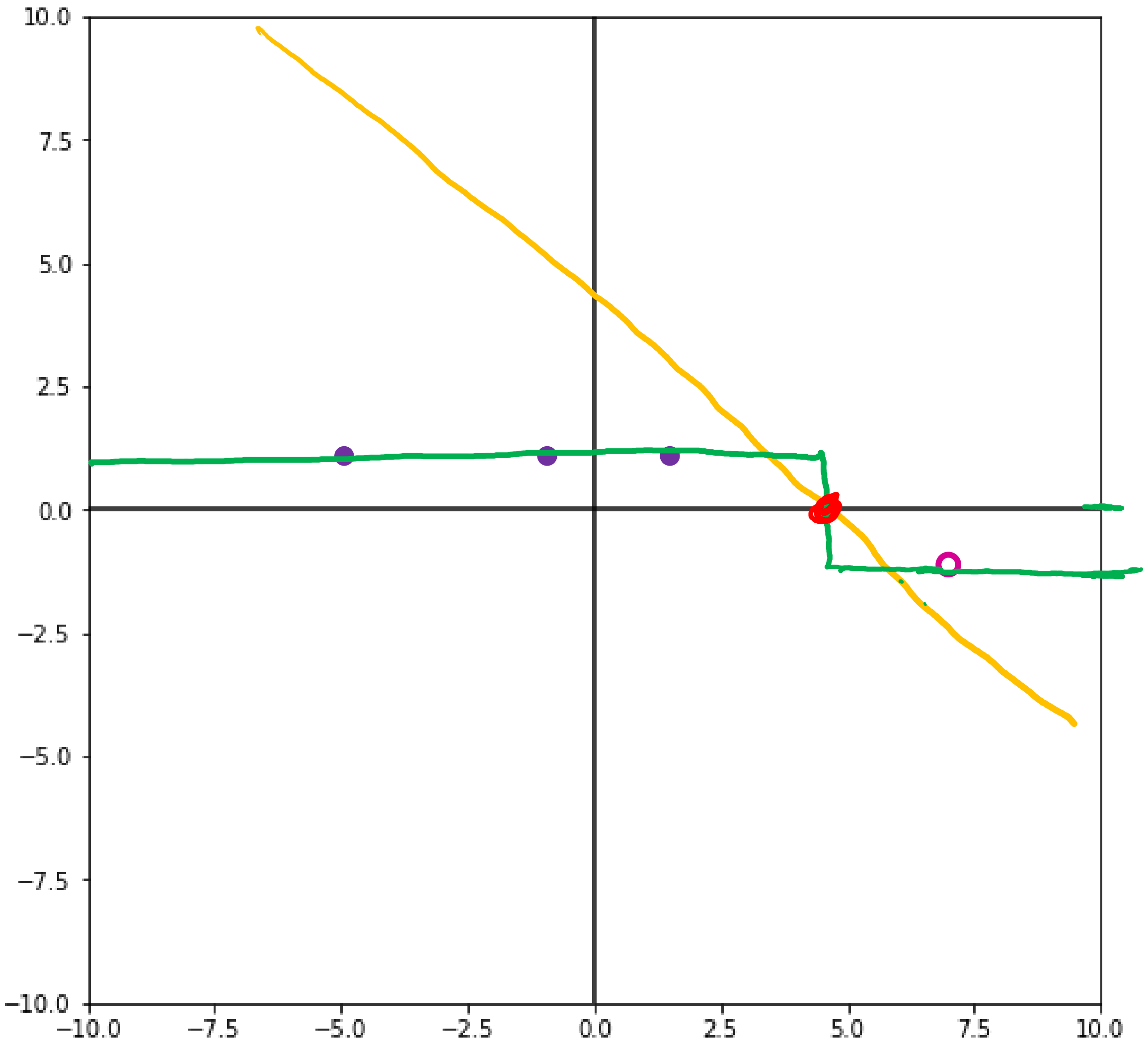
Classification

$$\mathcal{D} = \{(x^{(i)},\ y^{(i)})\}_{i=1}^{4}$$
$$= \{(-1,\ 1),$$
$$(7,\ -1),$$
$$(-5,\ 1),$$
$$(1.5,\ 1)\}$$

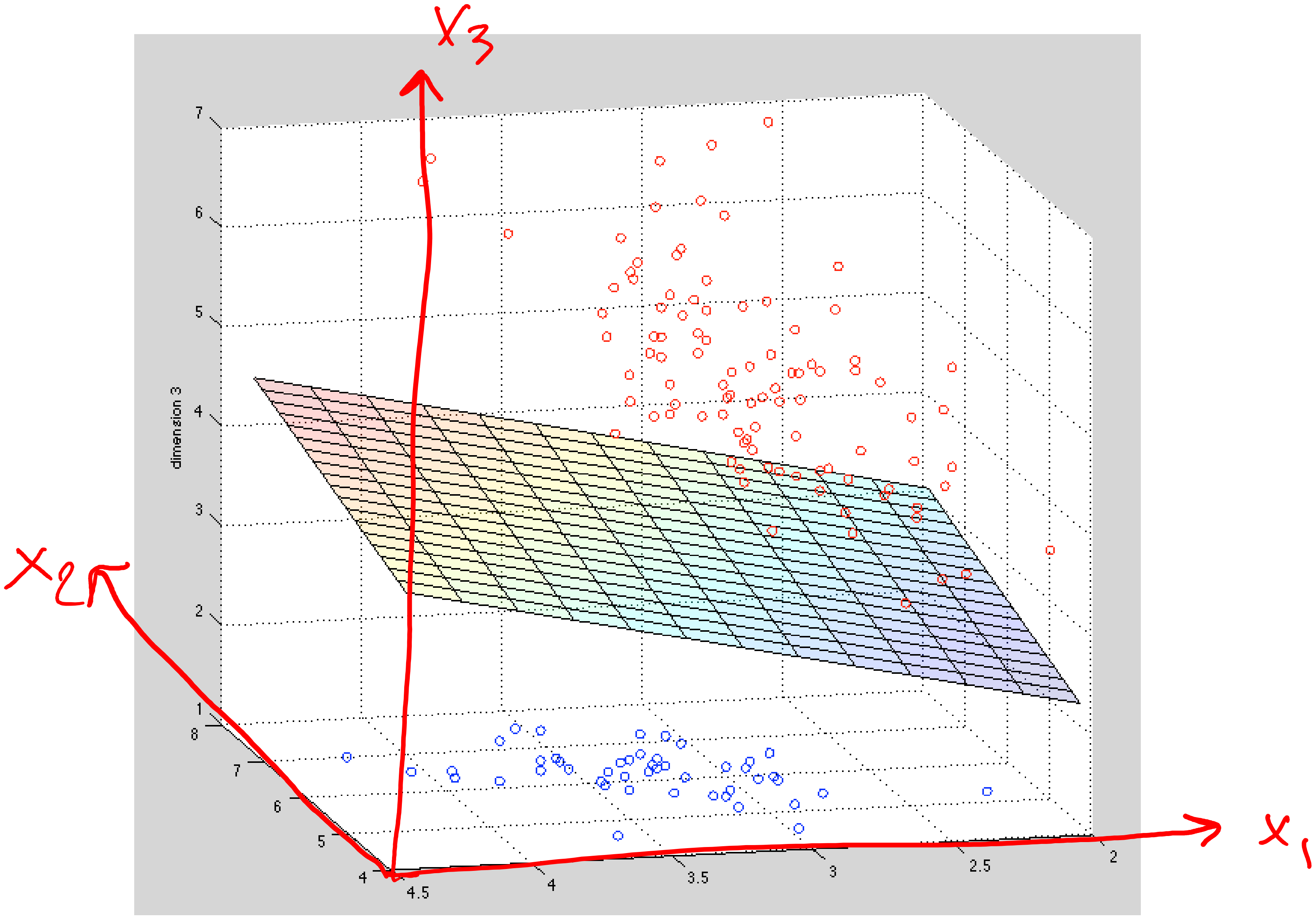
$z = wx + b$

$y = \text{sign}(wx + b)$

$$\text{sign}(z) = \begin{cases} +1 & \text{if } z \geq 0 \\ -1 & \text{o.w.} \end{cases}$$



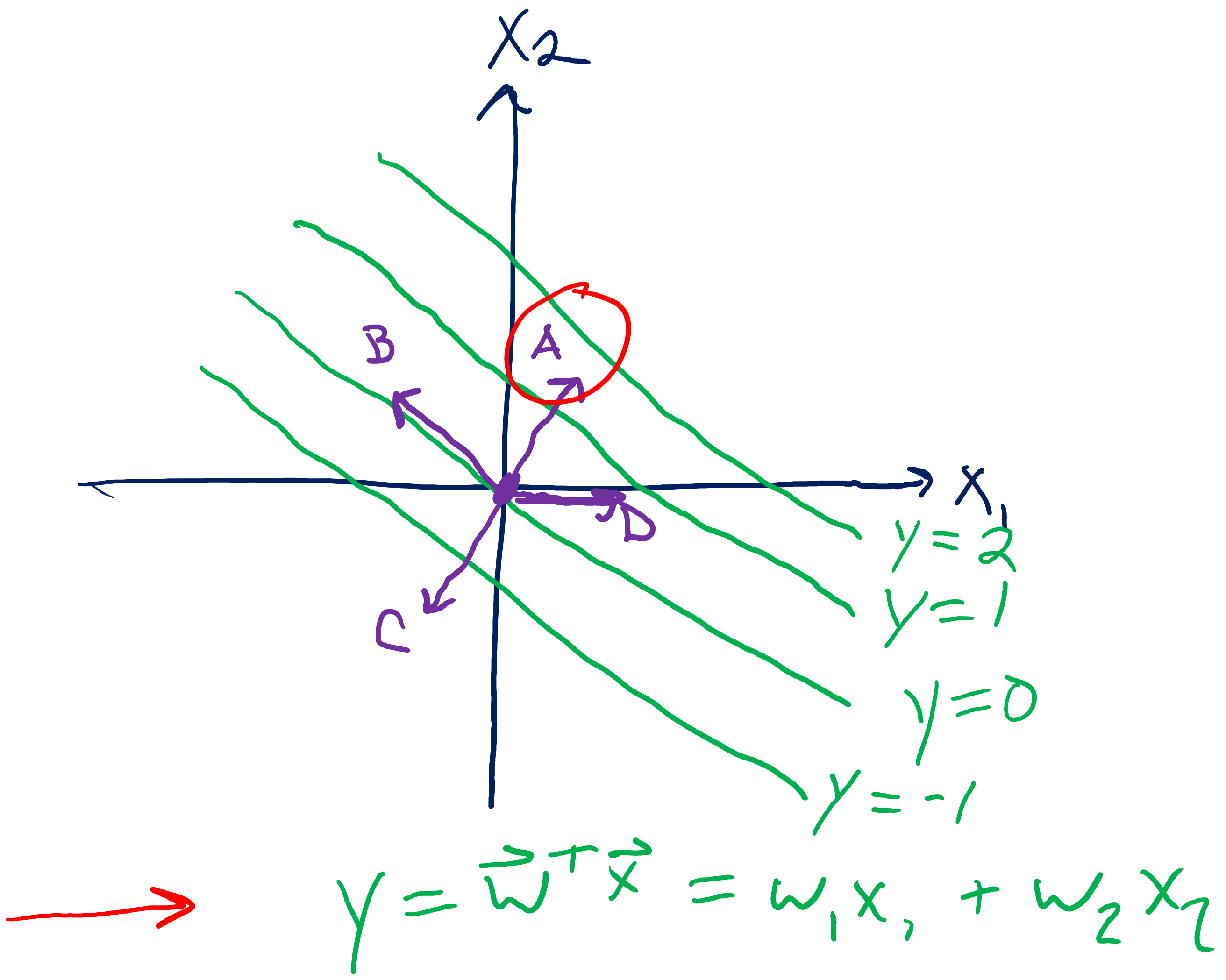
Perceptron



Previous Poll

Which is the correct vector $\boldsymbol{w}$?

A.
B.
C.
D.
E. I don't know



Previous Exercise

Geometry

Draw a picture of the region corresponding to:

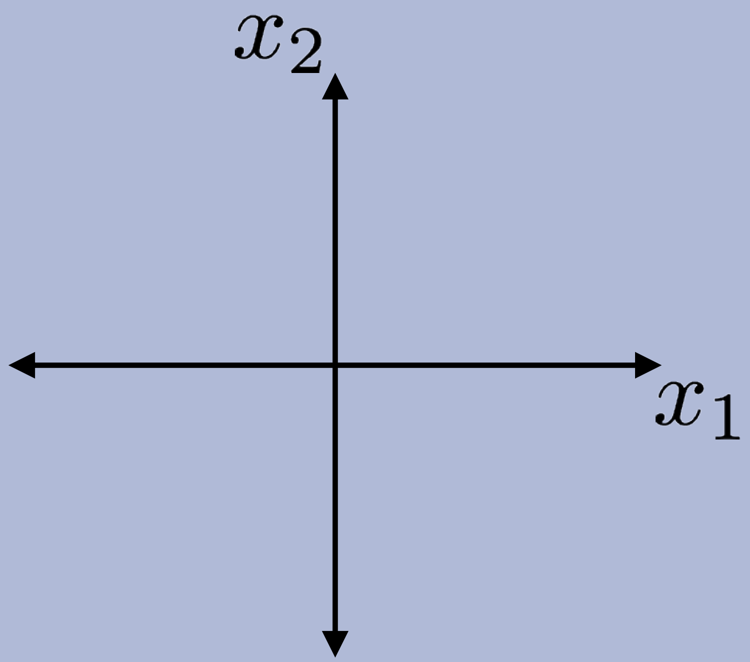
$$w_1 x_1 + w_2 x_2 + b > 0$$

where $w_1 = 2, w_2 = 3, b = 6$

Draw the vector

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

Answer Here:



Slide credit: CMU MLD Matt Gormley

Notation alert!
$\mathbb{R}^M$

Vectors

$\mathbf{u} \in \mathbb{R}^M$

$\mathbf{v} \in \mathbb{R}^M$ Note we assume these are both column vectors

Multiplication of vectors of the same length

Dot product (inner product):

- $y = \mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \sum_{i=1}^{M} u_i \, v_i$ $y \in \mathbb{R}$
- A dot product is an inner product (inner product is a more general term)

Outer product:

- $Y = \mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T$ $Y \in \mathbb{R}^{M \times M}$ $Y_{i,j} = u_i v_j$

Vectors

$\mathbf{u} \in \mathbb{R}^M$

Vector magnitude

$$|\mathbf{u}| = \sqrt{\sum_i^M u_i^2} = \left(\sum_i^M u_i^2\right)^{1/2} = (\ ?\ )^{1/2}$$

L2 norm (Euclidean norm) (More norms later)

$$\|\mathbf{u}\|_2 = \sqrt{\sum_i^M u_i^2} = \left(\sum_i^M u_i^2\right)^{1/2} = \left(\mathbf{u}^T\mathbf{u}\right)^{1/2}$$

L2 norm squared

$$\|\mathbf{u}\|_2^2 = \sum_i^M u_i^2 = \mathbf{u}^T\mathbf{u}$$

Linear and Affine

Notation alert!
$\forall$

Linear combination (of a set of terms)

Multiplying each term by a scalar and adding the results

e.g. Given a set of terms $\mathcal{S} = \{x_1, x_2, x_3\}$, where $x_i \in \mathbb{R} \; \forall \; i \in \{1 \dots 3\}$
$w_1 x_1 + w_2 x_2 + w_3 x_3$ is a linear combination of $\mathcal{S}$ if $w_i \in \mathbb{R} \; \forall \; i \in \{1 \dots 3\}$

e.g. Given a set of terms $\mathcal{S} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$, where $\boldsymbol{v}_i \in \mathbb{R}^M \; \forall \; i \in \{1 \dots 3\}$
$w_1 \boldsymbol{v}_1 + w_2 \boldsymbol{v}_2 + w_3 \boldsymbol{v}_3$ is a linear combination of $\mathcal{S}$ if $w_i \in \mathbb{R} \; \forall \; i \in \{1 \dots 3\}$

Linear and Affine

Affine combination (of a set of terms)

Affine allows for an additional scalar term to be added to a linear combination. Often called an offset or bias term

$w_1 x_1 + w_2 x_2 + w_3 x_3 + b$, where $b \in \mathbb{R}$

$w_1 \boldsymbol{v}_1 + w_2 \boldsymbol{v}_2 + w_3 \boldsymbol{v}_3 + b$, where $b \in \mathbb{R}$

Linear vs Affine Models

What linear usually means depends on the domain:

Linear algebra:

Linear usually means strictly linear

Geometry, algebra:

Linear usually means affine

Machine learning:

Linear usually means affine but we often transform affine to strictly linear to make the linear algebra and notation easier

Shapes in higher dimensions

What are these linear (affine) shapes called for 1-D, 2-D, 3-D, N-D input?

$$y = \boldsymbol{w}^T \boldsymbol{x} + b$$

$$\boldsymbol{w}^T \boldsymbol{x} + b = 0$$

$$\boldsymbol{w}^T \boldsymbol{x} + b \geq 0$$

Poll 1

Consider a data set with $\mathbf{x} \in \mathbb{R}^3$ and $y \in \{-1, +1\}$.

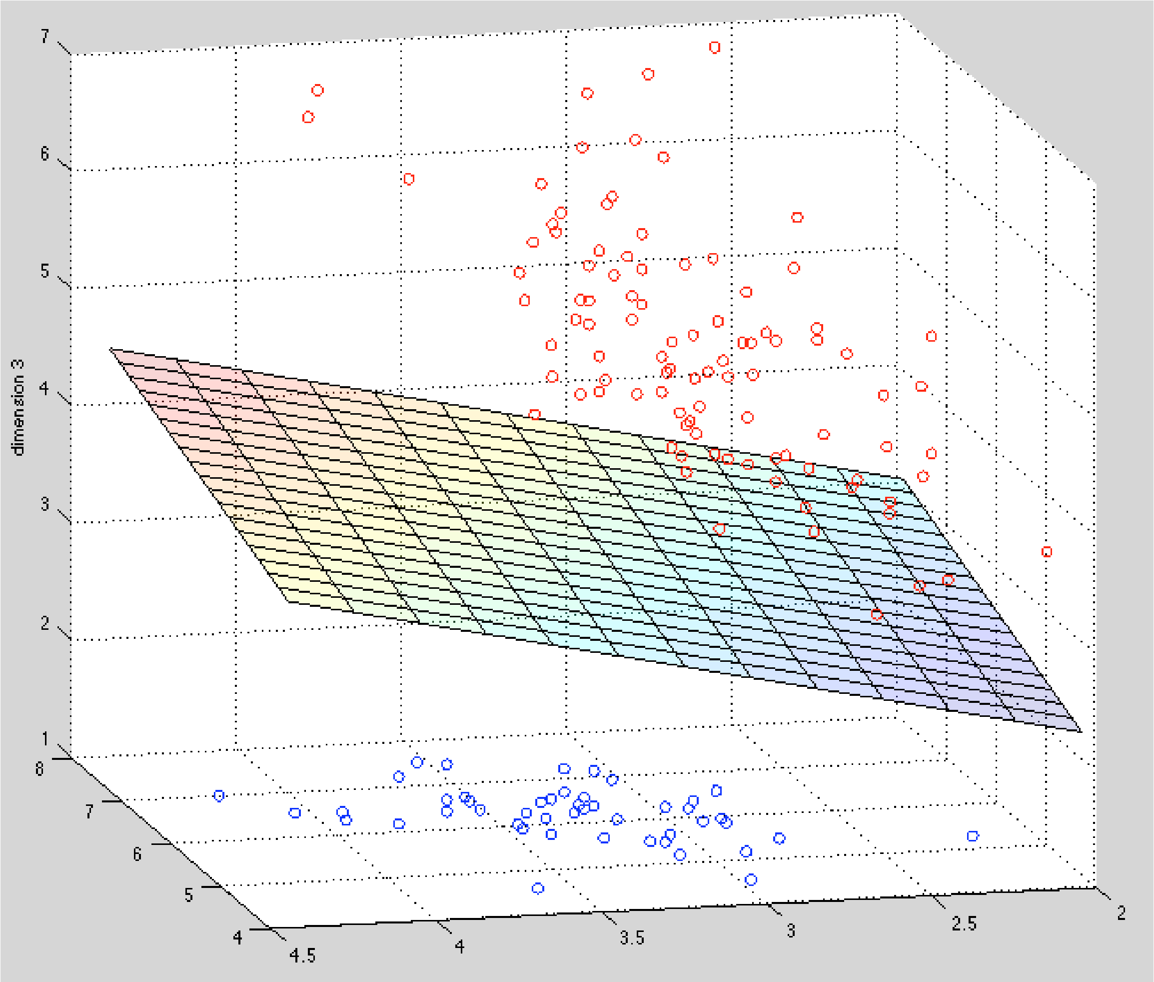
Which equation represents the plane shown here?

A. $y = w_1 x_1 + w_2 x_2 + b$

B. $0 = w_1 x_1 + w_2 x_2 + b$

C. $y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$

D. $0 = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$



Perceptron

$$sign(z) = \begin{cases} 1 & if\ z > 0 \\ 0 & if\ z = 0 \\ -1 & if\ z < 0 \end{cases}$$

A perceptron hypothesis function is simply the sign of a linear (affine) combination of the input:

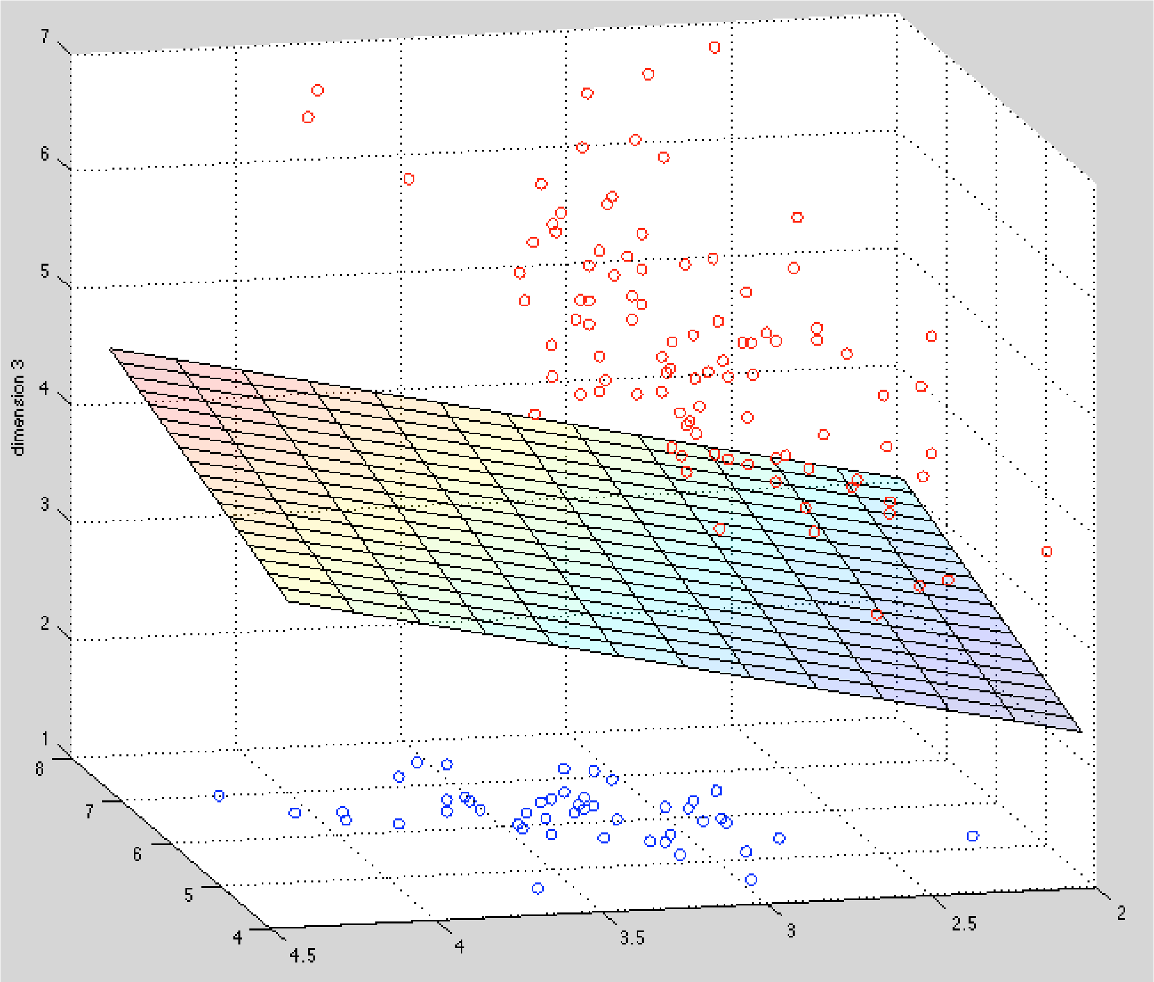
$$\hat{y} = h(\boldsymbol{x}) = sign(\boldsymbol{w}^T \boldsymbol{x} + b)$$

Dot Product

How does the dot product relate to a linear decision boundary?

Dot Product

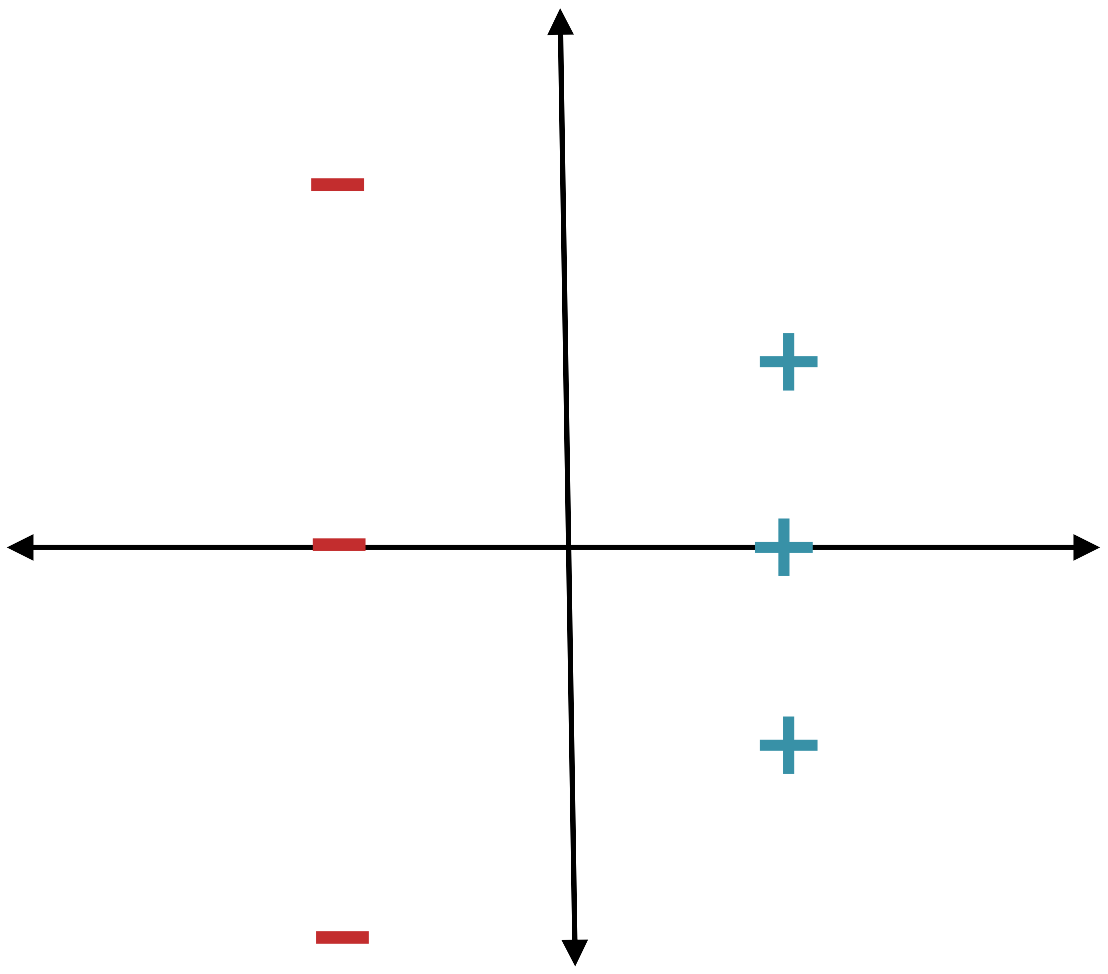
How does the dot product relate to linear decision boundary?



Exercise

Perceptron Algorithm

Sketch of algorithm



Perceptron Algorithm

Sketch of algorithm

Initialize $\boldsymbol{w} = \mathbf{0}$

Loop

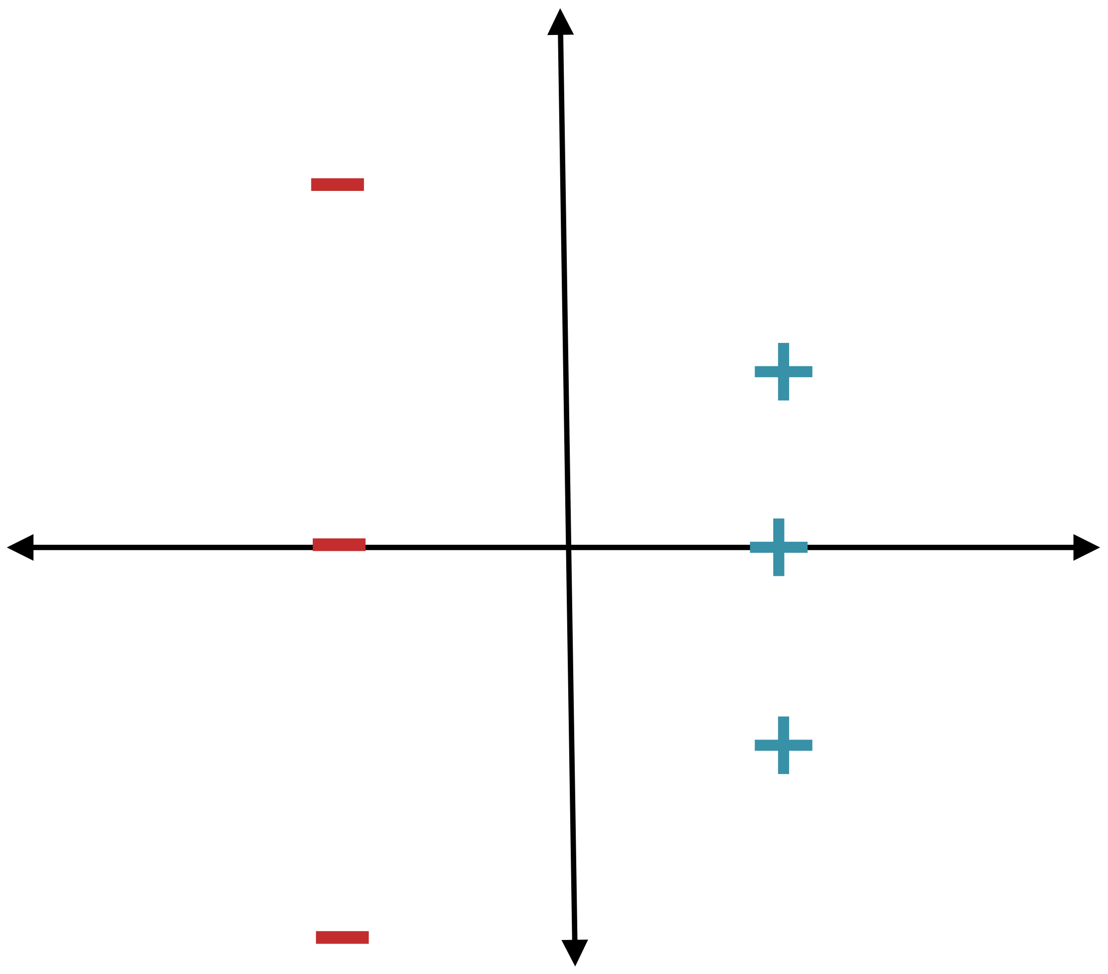
Given a point $\boldsymbol{x}$ predict $\hat{y} = sign\left(\boldsymbol{w}^T \boldsymbol{x}\right)$

Given actual label y:

If $y \neq \hat{y}$

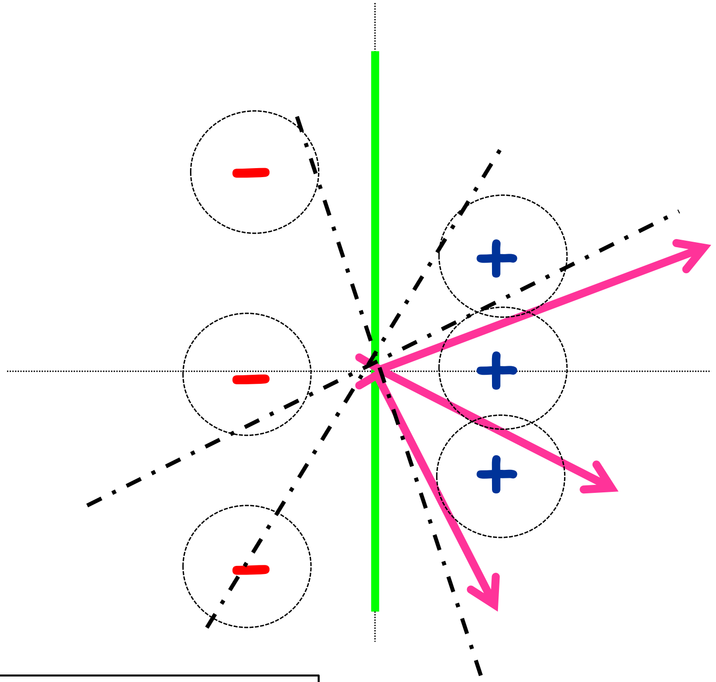
If $y = +1, \quad \boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{x}$

If $y = -1, \quad \boldsymbol{w} \leftarrow \boldsymbol{w} - \boldsymbol{x}$



Perceptron Algorithm: Example

Example: $(-1,2) -$ ✗
$(1,0) +$ ✓
$(1,1) +$ ✗
$(-1,0) -$ ✓
$(-1,-2) -$ ✗
$(1,-1) +$ ✓



Perceptron Algorithm: (without the bias term)

- Set t=1, start with all-zeroes weight vector w_1.
- Given example x, predict positive iff $w_t \cdot x \geq 0$.
- On a mistake, update as follows:
  - Mistake on positive, update $w_{t+1} \leftarrow w_t + x$
  - Mistake on negative, update $w_{t+1} \leftarrow w_t - x$

$w_1 = (0,0)$

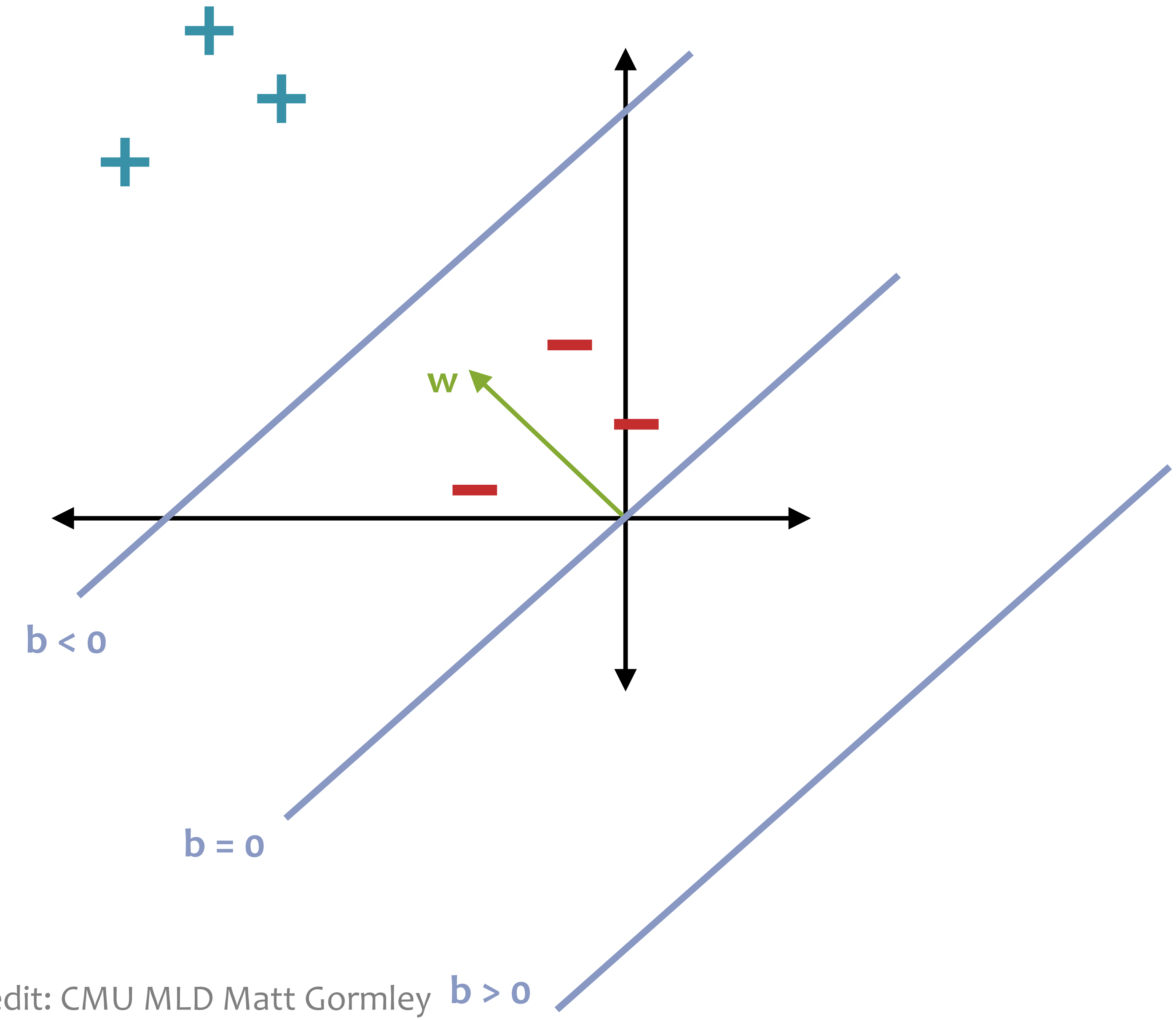
$w_2 = w_1 - (-1,2) = (1,-2)$

$w_3 = w_2 + (1,1) = (2,-1)$

$w_4 = w_3 - (-1,-2) = (3,1)$

Slide credit: CMU MLD Nina Balcan

Intercept Term



Q: Why do we need an intercept term?

A: It shifts the decision boundary off the origin

Q: What should happen to b during the perceptron algorithm

A: Two cases

1. Increasing b shifts the decision boundary towards the negative side
2. Decreasing b shifts the decision boundary towards the positive side

Slide credit: CMU MLD Matt Gormley

Perceptron Algorithm

Sketch of algorithm

Initialize $\boldsymbol{w} = \mathbf{0}$

Loop

Given a point $\boldsymbol{x}$ predict $\hat{y} = sign(\boldsymbol{w}^T \boldsymbol{x})$

Given actual label y:

If true $y \neq \hat{y}$

If true $y = +1, \quad \boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{x}$

If true $y = -1, \quad \boldsymbol{w} \leftarrow \boldsymbol{w} - \boldsymbol{x}$

Perceptron Algorithm

Sketch of algorithm

Initialize $\boldsymbol{\theta} = \mathbf{0}$

Loop

  Given a point $\boldsymbol{x}$ predict $\hat{y} = sign(\boldsymbol{\theta}^T \boldsymbol{x})$

  Given actual label y:

  If true $y \neq \hat{y}$

    If true $y = +1, \quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{x}$

    If true $y = -1, \quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \boldsymbol{x}$

Perceptron Algorithm

Learning for Perceptron also works if we have a fixed training dataset, D.

Algorithm 1 Perceptron Learning Algorithm

procedure PERCEPTRON($\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$)
    $\boldsymbol{\theta} \leftarrow \mathbf{0}$ ▷ Initialize parameters
    while not converged **do**
        for $i \in \{1, 2, \ldots, N\}$ **do** ▷ For each example
            $\hat{y} \leftarrow \text{sign}(\boldsymbol{\theta}^T \mathbf{x}^{(i)})$ ▷ Predict
            if $\hat{y} \neq y^{(i)}$ **then** ▷ If mistake
                $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y^{(i)} \mathbf{x}^{(i)}$ ▷ Update parameters
    return $\boldsymbol{\theta}$

Slide credit: CMU MLD Matt Gormley

Perceptron Algorithm

Learning for Perceptron also works if we have a fixed training dataset, D.

Algorithm 1 Perceptron Learning Alg

1: **procedure** PERCEPTRON($\mathcal{D} = \{(\mathbf{x}$

2: $\boldsymbol{\theta} \leftarrow \mathbf{0}$

3: **while** not converged **do**

4: **for** $i \in \{1, 2, \ldots, N\}$ **do** ▷ For each example

5: $\hat{y} \leftarrow \text{sign}(\boldsymbol{\theta}^T \mathbf{x}^{(i)})$ ▷ Predict

6: **if** $\hat{y} \neq y^{(i)}$ **then** ▷ If mistake

7: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y^{(i)} \mathbf{x}^{(i)}$ ▷ Update parameters

8: **return** $\boldsymbol{\theta}$

Implementation Trick: same behavior as our "*add on positive mistake and subtract on negative mistake*" version, because $y^{(i)}$ takes care of the sign

Plan

Perceptron Background

- ML Data, Tasks, Notation
- Vectors, dot products
- Geometry with linear functions
- Perceptron (and the sign() function)
- Decision boundaries and errors

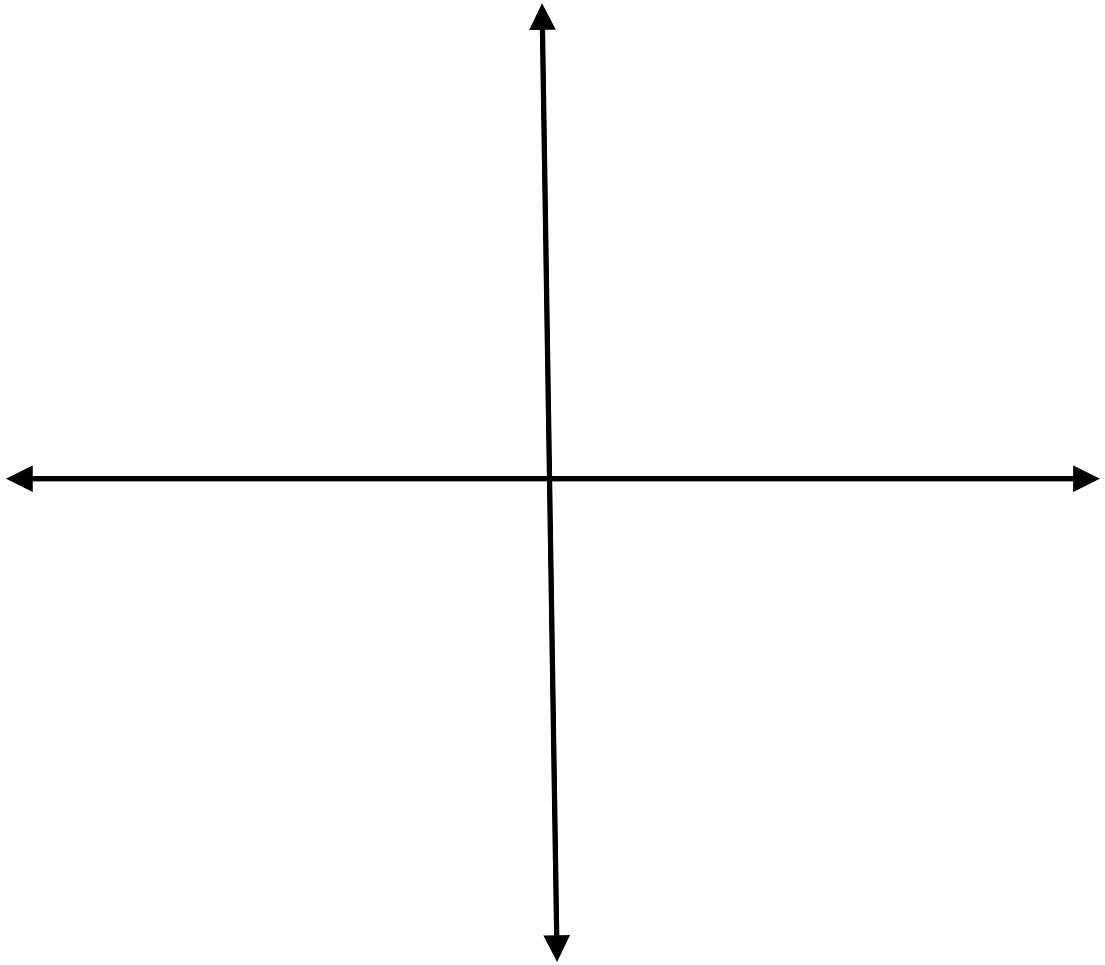
Perceptron Algorithm

Perceptron Mistake Bound Theory

- Background: projections, distances, and margin

Projection

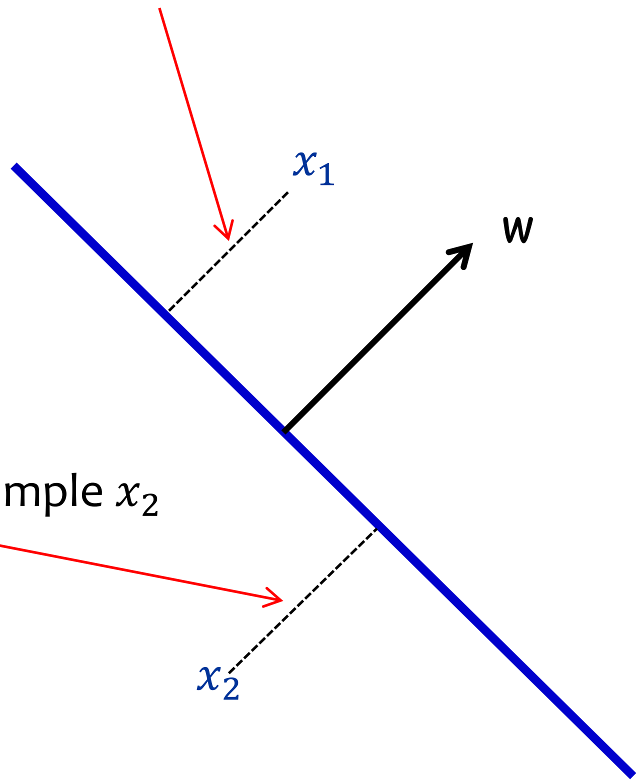
Projection of $\mathbf{u}$ on to $\mathbf{v}$



Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$ (or the negative if on wrong side)

Margin of positive example x_1



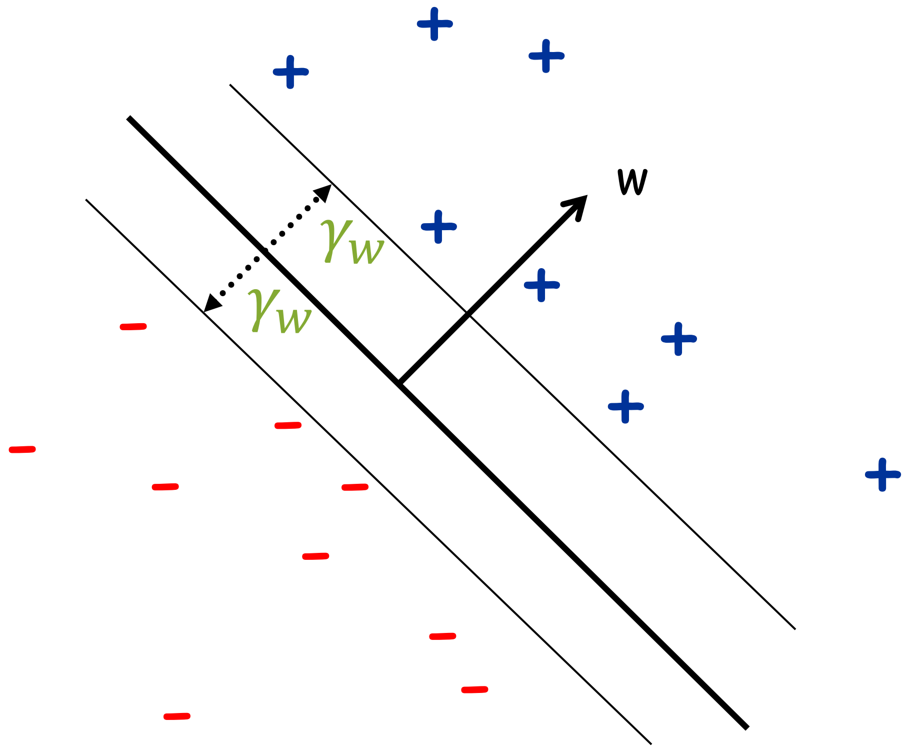
Margin of negative example x_2

Slide credit: CMU MLD Nina Balcan

Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$ (or the negative if on wrong side)

Definition: The margin γ_w of a set of examples S wrt a linear separator w is the smallest margin over points $x \in S$.



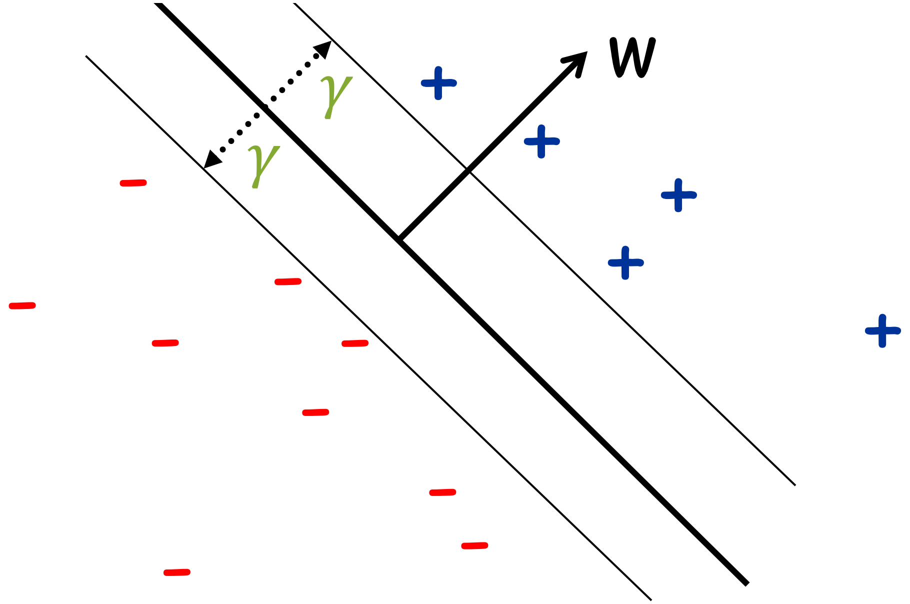
Slide credit: CMU MLD Nina Balcan

Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$ (or the negative if on wrong side)

Definition: The margin γ_w of a set of examples S wrt a linear separator w is the smallest margin over points $x \in S$.

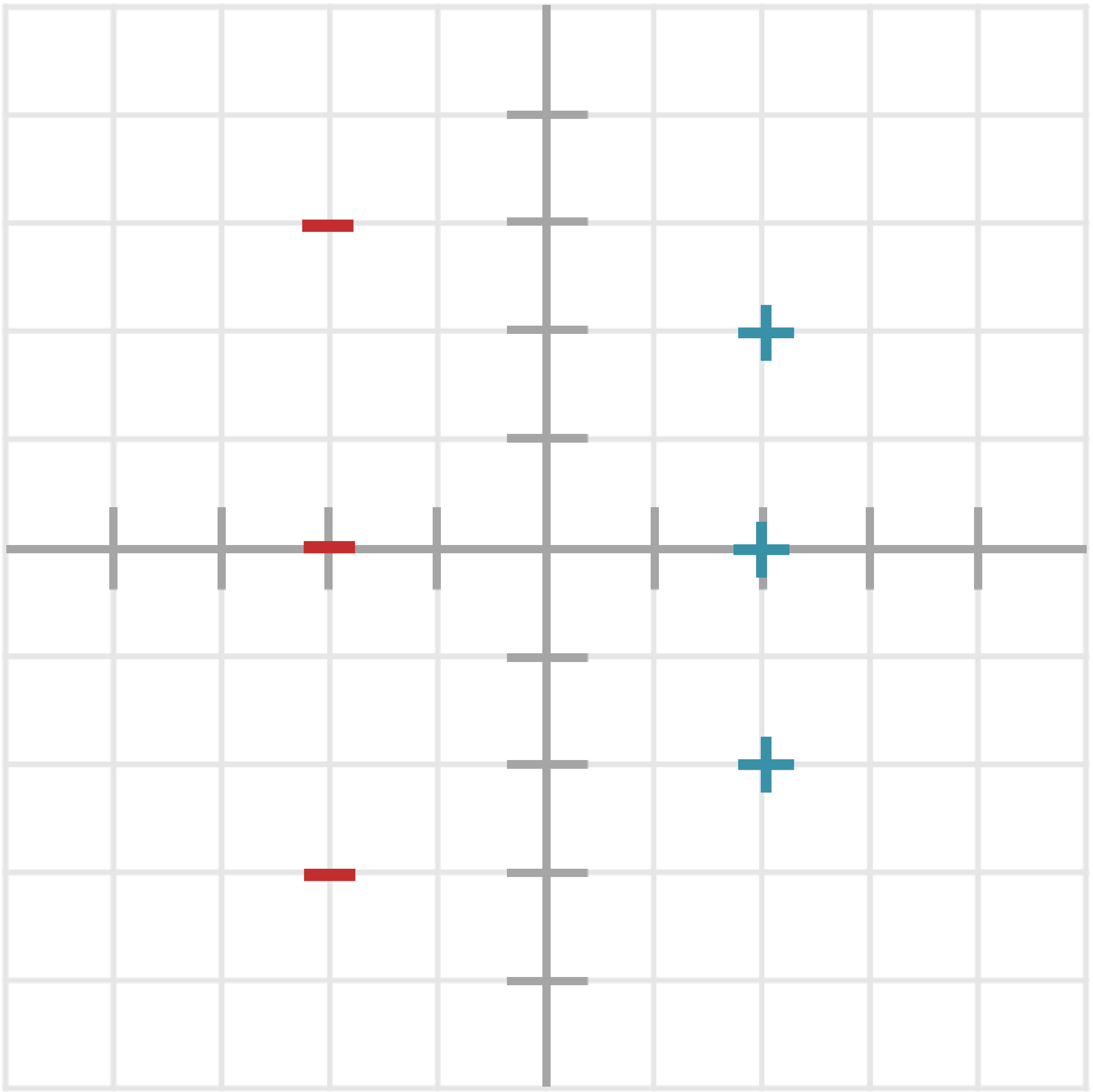
Definition: The margin γ of a set of examples S is the maximum γ_w over all linear separators w.



Slide credit: CMU MLD Nina Balcan

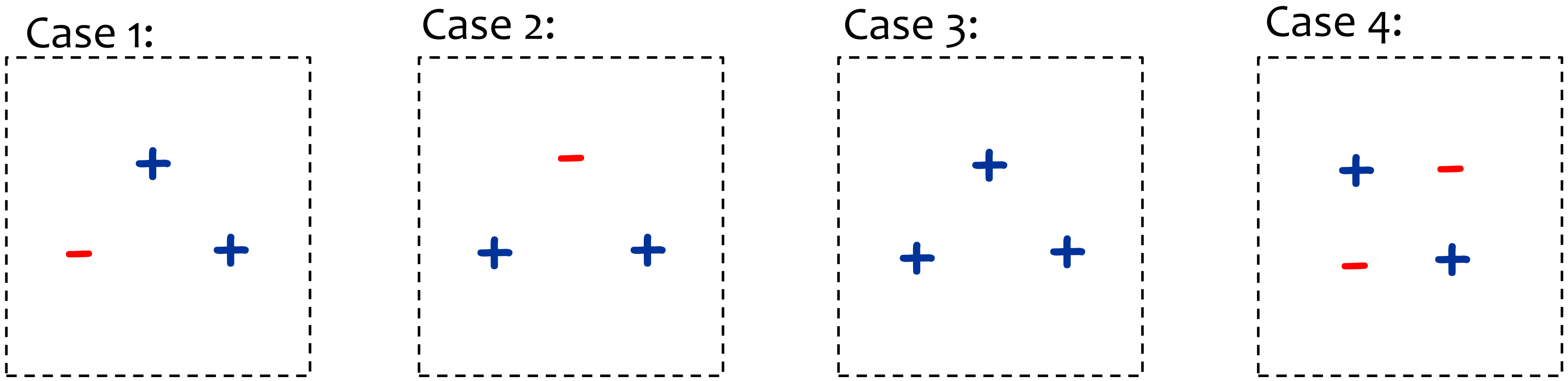
Geometric Margin

What is the margin for this dataset?



Linear Separability

Def: For a **binary classification** problem, a set of examples S is **linearly separable** if there exists a linear decision boundary that can separate the points



Slide credit: CMU MLD Matt Gormley

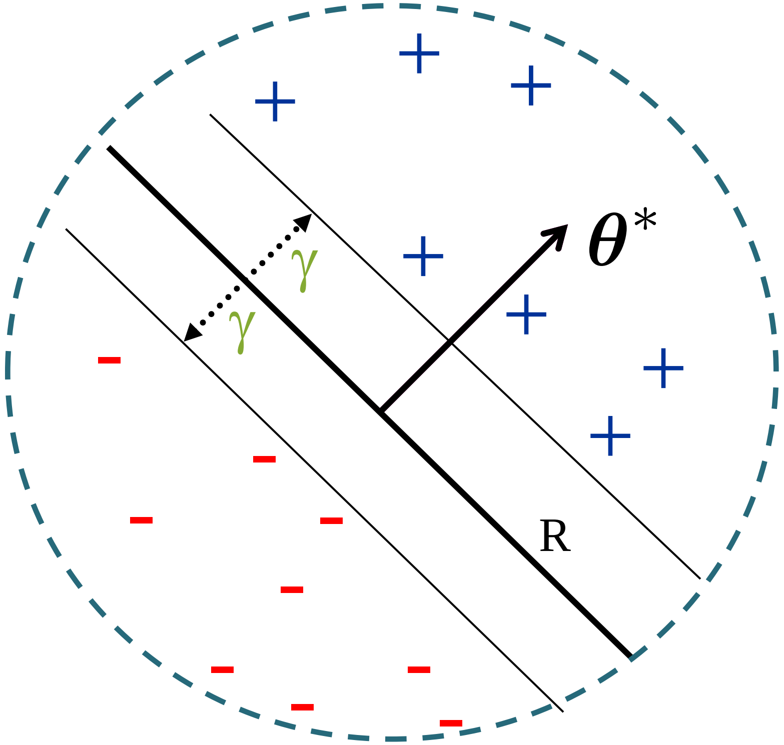
ANALYSIS OF PERCEPTRON

Analysis: Perceptron

Perceptron Mistake Bound

Guarantee: If data has margin γ and all points inside a ball of radius R, then Perceptron makes $\leq (R/\gamma)^2$ mistakes.

(Normalized margin: multiplying all points by 100, or dividing all points by 100, doesn't change the number of mistakes; algo is invariant to scaling.)



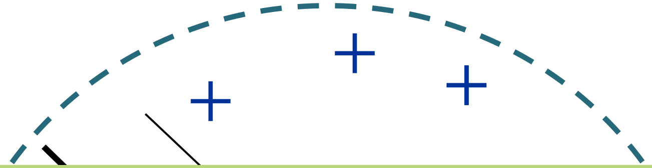
Slide credit: CMU MLD Nina Balcan and Matt Gormley

Analysis: Perceptron

Perceptron Mistake Bound

Guarantee: If data has margin γ and all points inside a ball of radius R, then Perceptron makes $\leq (R/\gamma)^2$ mistakes.

(Normalized margin: multiplying all points by 100, or dividing all points by 100, doesn't change the number of mistakes; algo is invariant to scaling.)



Def: We say that the (batch) perceptron algorithm has **converged** if it stops making mistakes on the training data (perfectly classifies the training data).

Main Takeaway: For **linearly separable** data, if the perceptron algorithm cycles repeatedly through the data, it will **converge** in a finite # of steps.

Slide credit: CMU MLD Nina Balcan and Matt Gormley

Analysis: Perceptron

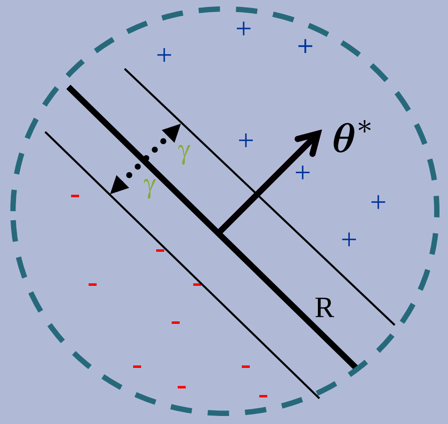
Perceptron Mistake Bound

Theorem 0.1 (Block (1962), Novikoff (1962)).
Given dataset: $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$.
Suppose:

1. *Finite size inputs:* $||x^{(i)}|| \leq R$
2. *Linearly separable data:* $\exists \boldsymbol{\theta}^*$ *s.t.* $||\boldsymbol{\theta}^*|| = 1$ *and* $y^{(i)}(\boldsymbol{\theta}^* \cdot \mathbf{x}^{(i)}) \geq \gamma, \forall i$

Then: The number of mistakes made by the Perceptron algorithm on this dataset is

$$k \leq (R/\gamma)^2$$



Slide credit: CMU MLD Nina Balcan and Matt Gormley

Analysis: Percept

Common Misunderstanding:

The **radius** is **centered at the origin,** not at the center of the points.

Perceptron Mistake Boun

Theorem 0.1 (Block (1962), Novikoff (19

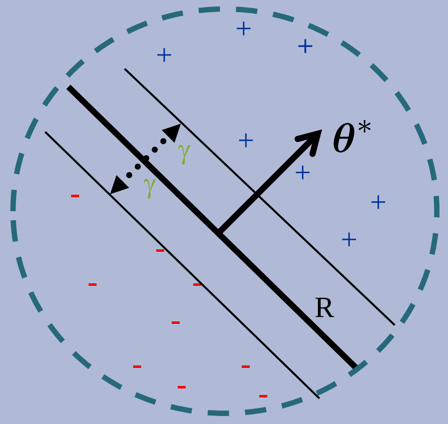
Given dataset: $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$

Suppose:

1. *Finite size inputs:* $||x^{(i)}|| \leq R$
2. *Linearly separable data:* $\exists \boldsymbol{\theta}^*$ *s.t.* $||\boldsymbol{\theta}^*|| = 1$ *and* $y^{(i)}(\boldsymbol{\theta}^* \cdot \mathbf{x}^{(i)}) \geq \gamma, \forall i$

Then: The number of mistakes made by the Perceptron algorithm on this dataset is

$$k \leq (R/\gamma)^2$$

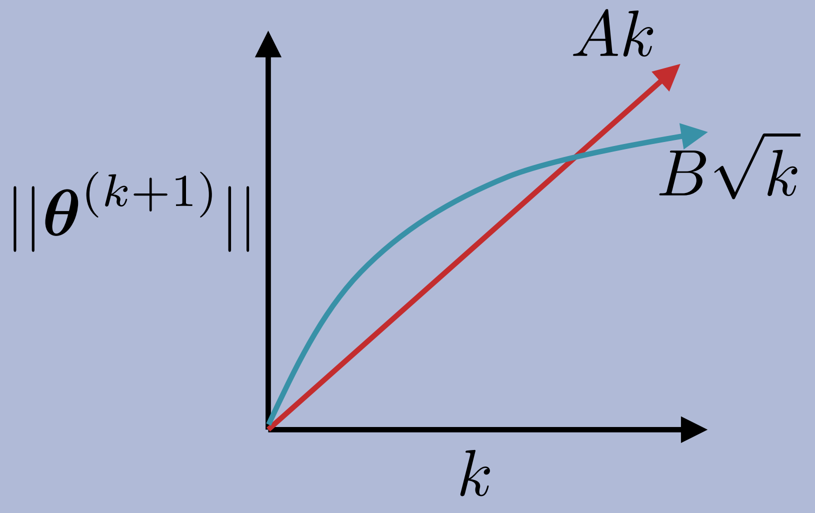


Analysis: Perceptron

Proof of Perceptron Mistake Bound:

We will show that there exist constants A and B s.t.

$$Ak \leq ||\boldsymbol{\theta}^{(k+1)}|| \leq B\sqrt{k}$$



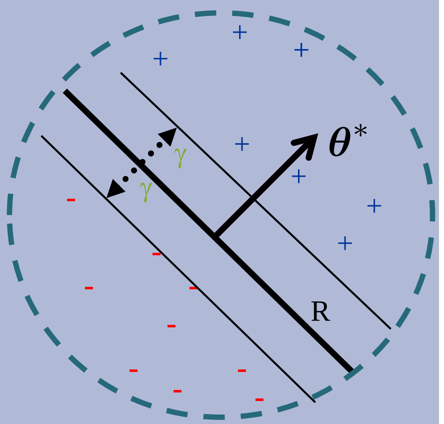
Analysis: Perceptron

Theorem 0.1 (Block (1962), Novikoff (1962)).
Given dataset: $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$.
Suppose:

1. *Finite size inputs:* $||x^{(i)}|| \leq R$
2. *Linearly separable data:* $\exists \boldsymbol{\theta}^*$ s.t. $||\boldsymbol{\theta}^*|| = 1$ *and* $y^{(i)}(\boldsymbol{\theta}^* \cdot \mathbf{x}^{(i)}) \geq \gamma, \forall i$

Then: The number of mistakes made by the Perceptron algorithm on this dataset is

$$k \leq (R/\gamma)^2$$



Algorithm 1 Perceptron Learning Algorithm (Online)

procedure PERCEPTRON($\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots\}$)
    $\boldsymbol{\theta} \leftarrow \mathbf{0}, k = 1$ ▷ Initialize parameters
    for $i \in \{1, 2, \ldots\}$ **do** ▷ For each example
        if $y^{(i)}(\boldsymbol{\theta}^{(k)} \cdot \mathbf{x}^{(i)}) \leq 0$ **then** ▷ If mistake
            $\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)} + y^{(i)}\mathbf{x}^{(i)}$ ▷ Update parameters
            $k \leftarrow k + 1$
    return $\boldsymbol{\theta}$

Slide credit: CMU MLD Nina Balcan and Matt Gormley

Analysis: Perceptron

Proof of Perceptron Mistake Bound:

Part 1: for some A, $Ak \leq ||\boldsymbol{\theta}^{(k+1)}||$

$$\boldsymbol{\theta}^{(k+1)} \cdot \boldsymbol{\theta}^* = (\boldsymbol{\theta}^{(k)} + y^{(i)}\mathbf{x}^{(i)})\boldsymbol{\theta}^*$$

by Perceptron algorithm update

$$= \boldsymbol{\theta}^{(k)} \cdot \boldsymbol{\theta}^* + y^{(i)}(\boldsymbol{\theta}^* \cdot \mathbf{x}^{(i)})$$

$$\geq \boldsymbol{\theta}^{(k)} \cdot \boldsymbol{\theta}^* + \gamma$$

by assumption

$$\Rightarrow \boldsymbol{\theta}^{(k+1)} \cdot \boldsymbol{\theta}^* \geq k\gamma$$

by induction on k since $\theta^{(1)} = \mathbf{0}$

$$\Rightarrow ||\boldsymbol{\theta}^{(k+1)}|| \geq k\gamma$$

since $||\mathbf{w}|| \times ||\mathbf{u}|| \geq \mathbf{w} \cdot \mathbf{u}$ and $||\theta^*|| = 1$

Cauchy-Schwartz inequality

Slide credit: CMU MLD Matt Gormley

Analysis: Perceptron

Proof of Perceptron Mistake Bound:

Part 2: for some B, $||\boldsymbol{\theta}^{(k+1)}|| \leq B\sqrt{k}$

$$||\boldsymbol{\theta}^{(k+1)}||^2 = ||\boldsymbol{\theta}^{(k)} + y^{(i)}\mathbf{x}^{(i)}||^2$$

by Perceptron algorithm update

$$= ||\boldsymbol{\theta}^{(k)}||^2 + (y^{(i)})^2||\mathbf{x}^{(i)}||^2 + 2y^{(i)}(\boldsymbol{\theta}^{(k)} \cdot \mathbf{x}^{(i)})$$

$$\leq ||\boldsymbol{\theta}^{(k)}||^2 + (y^{(i)})^2||\mathbf{x}^{(i)}||^2$$

since kth mistake $\Rightarrow y^{(i)}(\boldsymbol{\theta}^{(k)} \cdot \mathbf{x}^{(i)}) \leq 0$

$$= ||\boldsymbol{\theta}^{(k)}||^2 + R^2$$

since $(y^{(i)})^2||\mathbf{x}^{(i)}||^2 = ||\mathbf{x}^{(i)}||^2 = R^2$ by assumption and $(y^{(i)})^2 = 1$

$$\Rightarrow ||\boldsymbol{\theta}^{(k+1)}||^2 \leq kR^2$$

by induction on k since $(\theta^{(1)})^2 = 0$

$$\Rightarrow ||\boldsymbol{\theta}^{(k+1)}|| \leq \sqrt{k}R$$

Analysis: Perceptron

Proof of Perceptron Mistake Bound:

Part 3: Combining the bounds finishes the proof.

$$k\gamma \leq ||\boldsymbol{\theta}^{(k+1)}|| \leq \sqrt{k}R$$

$$\Rightarrow k \leq (R/\gamma)^2$$

The total number of mistakes must be less than this

Slide credit: CMU MLD Matt Gormley