

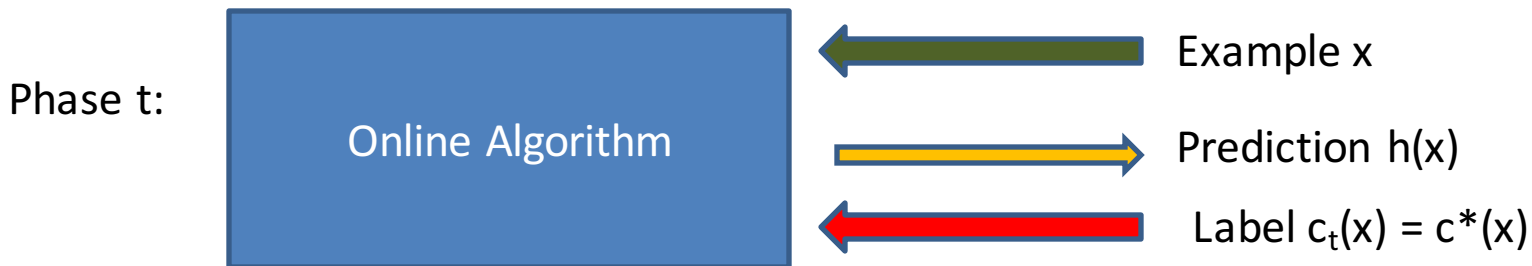
Online Algorithms: Perceptron and Winnow

Outline

- Online Model
- Linear Separator
- Perceptron
 - Realizable case
 - Unrealizable case
- Winnow

Online Model

- Example arrive sequentially
- Need to make a prediction
 - Afterwards observe the outcome
- No distributional assumptions
- Goal: Minimize the number of mistakes



Example: Learning OR of literals

- Inputs: (z_1, \dots, z_n)
- Literals : x_1, \bar{x}_1
- OR functions:
 $x_1 \vee \bar{x}_4 \vee x_7$
- Realizable case:
 - $C^*(z)$ is an OR
- ELIM algorithm:
 - Initialize: $L = \{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$
 - Time t , receive
 - $z = (z_1, \dots, z_n)$
 - Predict $OR(L, z)$
 - Receive $c^*(z)$
 - If error (has to be negative)
 - delete from L the positive literals in z .

What is the MAXIMUM number of mistakes?

Learning Linear Separators

- Input $\{0,1\}^d$ or \mathbb{R}^d
- Linear Separator
 - weights w in \mathbb{R}^d and threshold θ
 - hypothesis $h(x)=+1$ iff
$$\langle w, x \rangle = \sum w_i x_i \geq \theta$$
- Simplifying assumptions:
 - $\theta=0$ (add coordinate x_0 such that $x_0=1$ always)
 - $\|x\|=1$

Perceptron - Algorithm

- Initialize $\mathbf{w}_1 = (0, \dots, 0)$
- Given example \mathbf{x}_t ,
 - predict positive iff $\langle \mathbf{w}_t, \mathbf{x}_t \rangle \geq 0$
- On a Mistake t : $\mathbf{w}_{t+1} = \mathbf{w}_t + c_t(\mathbf{x}) \mathbf{x}_t$,
 - Mistake on negative (i.e., $c^*(\mathbf{x}) = +1$): $\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{x}_t$.
 - Mistake on positive (i.e., $c^*(\mathbf{x}) = -1$): $\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{x}_t$.

Perceptron - motivation

- **False Negative**

- $c_t(x) = +1$
- $\langle w_t, x_t \rangle$ negative
- after update

$$\begin{aligned} & \langle w_{t+1}, x_t \rangle \\ &= \langle w_t, x_t \rangle + \langle x_t, x_t \rangle \\ &= \langle w_t, x_t \rangle + 1 \end{aligned}$$

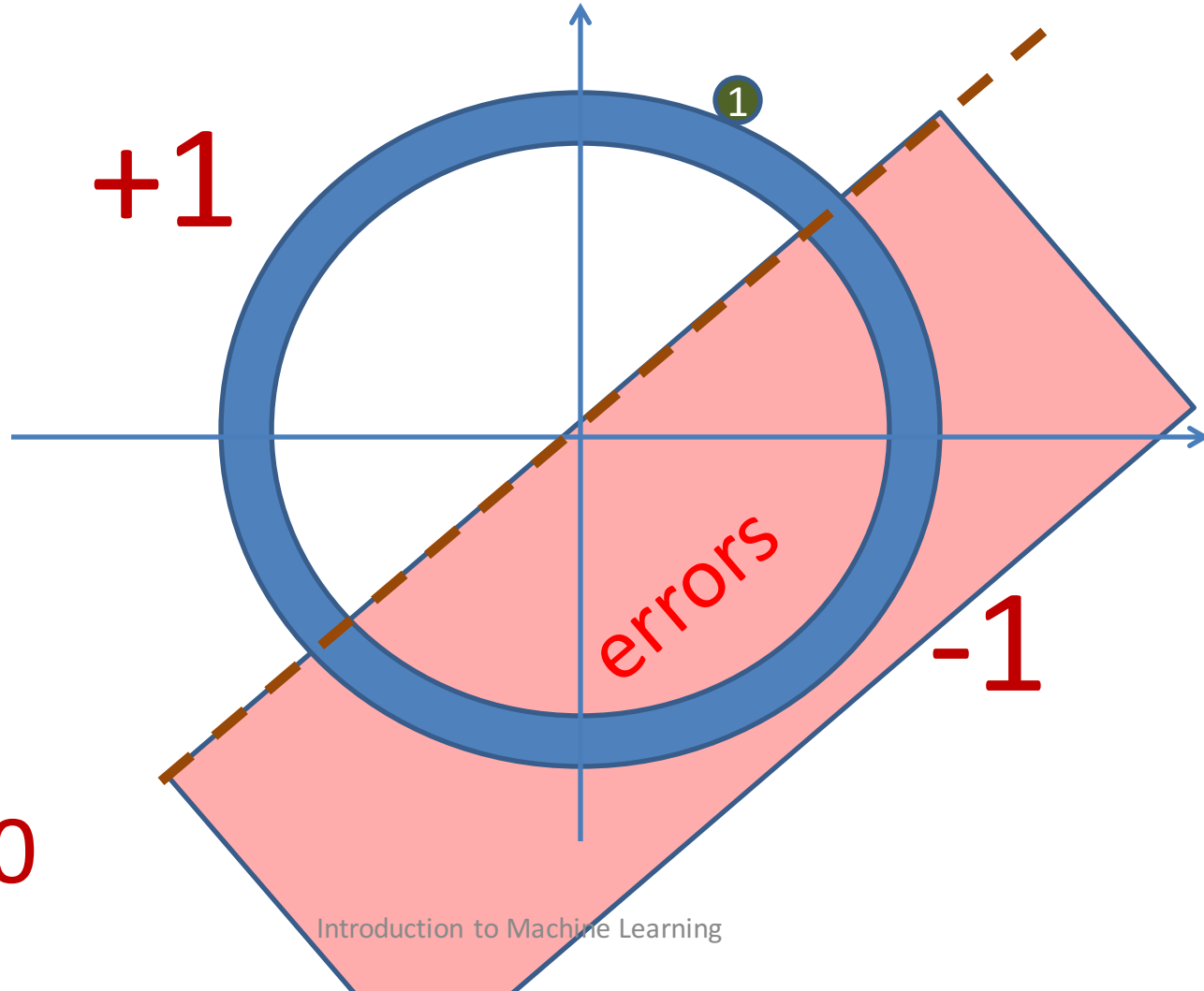
- **False Positive**

- $c_t(x) = -1$
- $\langle w_t, x_t \rangle$ positive
- after update

$$\begin{aligned} & \langle w_{t+1}, x_t \rangle \\ &= \langle w_t, x_t \rangle - \langle x_t, x_t \rangle \\ &= \langle w_t, x_t \rangle - 1 \end{aligned}$$

Perceptron Example

$$w_1 = (0,0)$$
$$w_2 = (0,0)$$

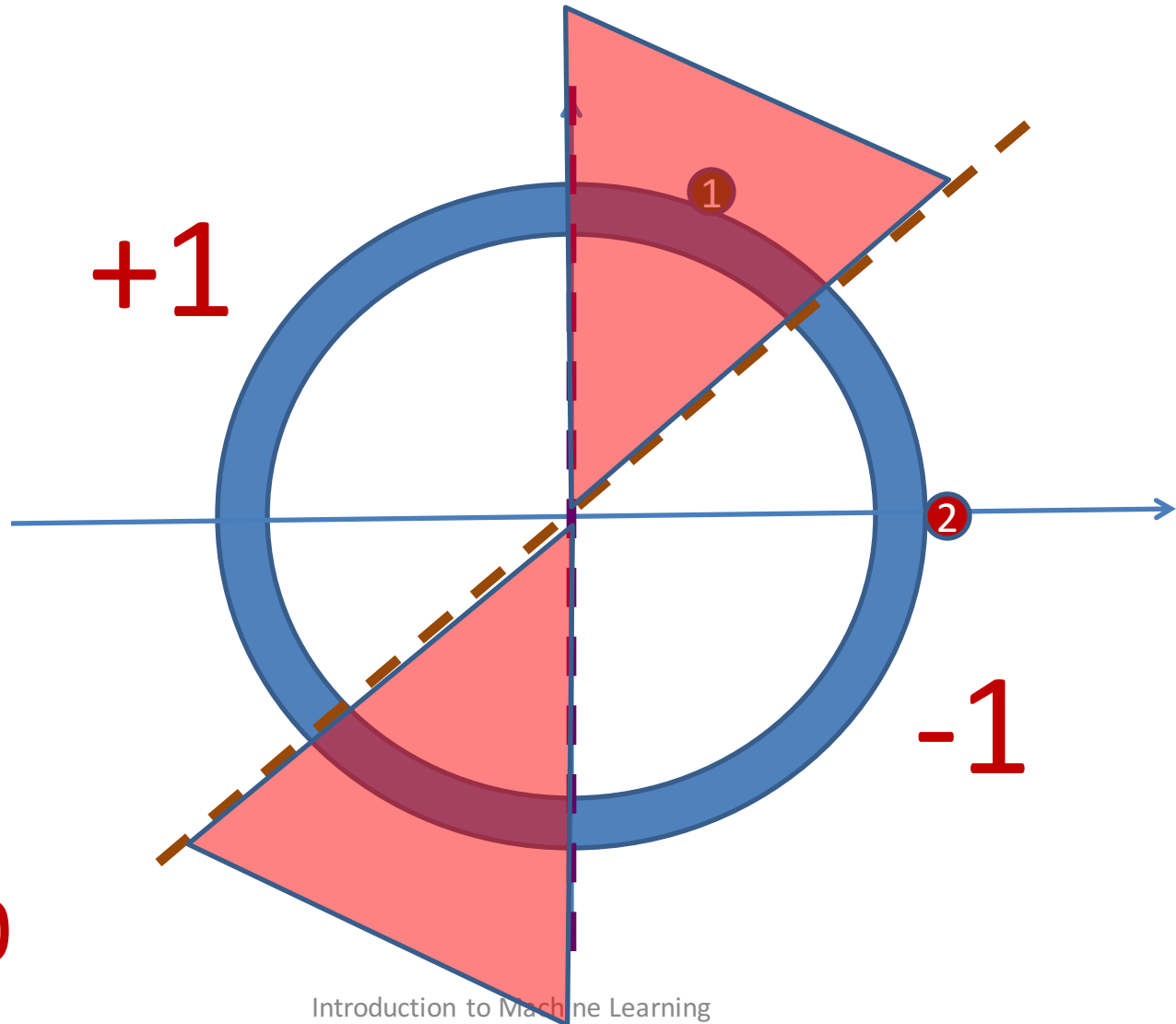


$$x_1 - x_2 = 0$$

10/11/2013

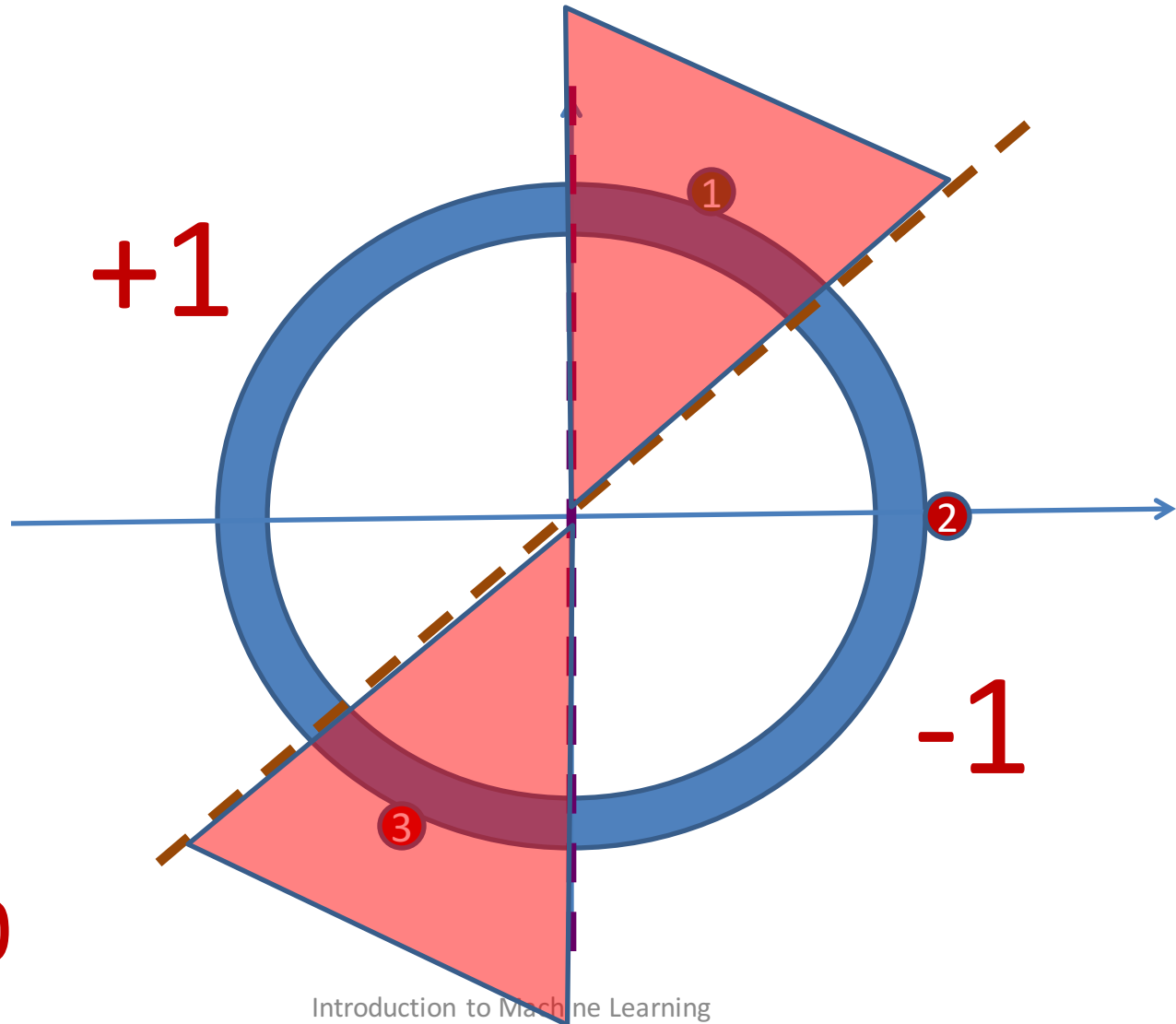
Perceptron Example

$$w_1 = (0,0)$$
$$w_2 = (0,0)$$
$$w_3 = (-1,0)$$



Perceptron Example

$$\begin{aligned}w_1 &= (0,0) \\w_2 &= (0,0) \\w_3 &= (-1,0)\end{aligned}$$



$$x_1 - x_2 = 0$$

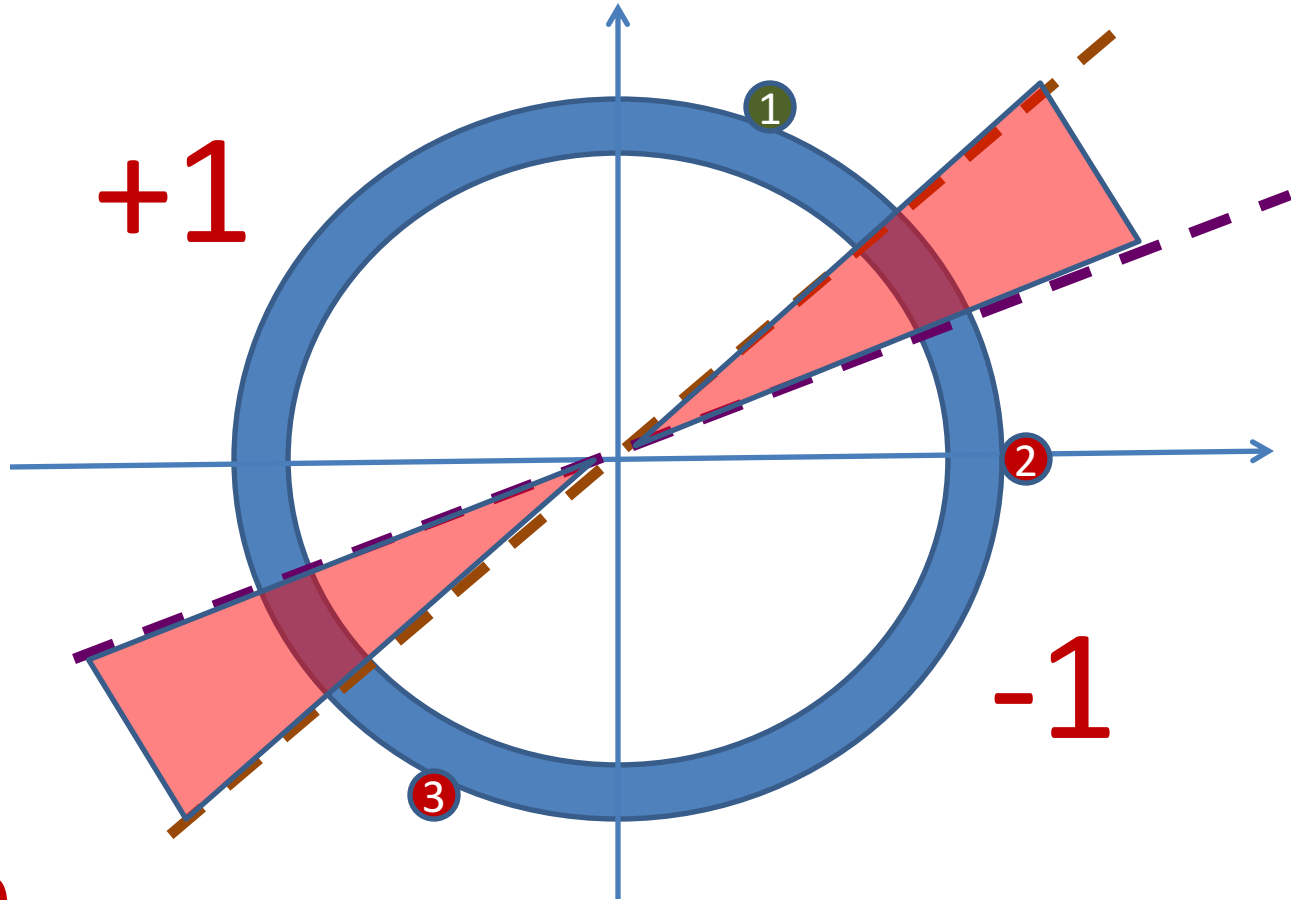
Perceptron Example

$$w_1 = (0,0)$$

$$w_2 = (0,0)$$

$$w_3 = (-1,0)$$

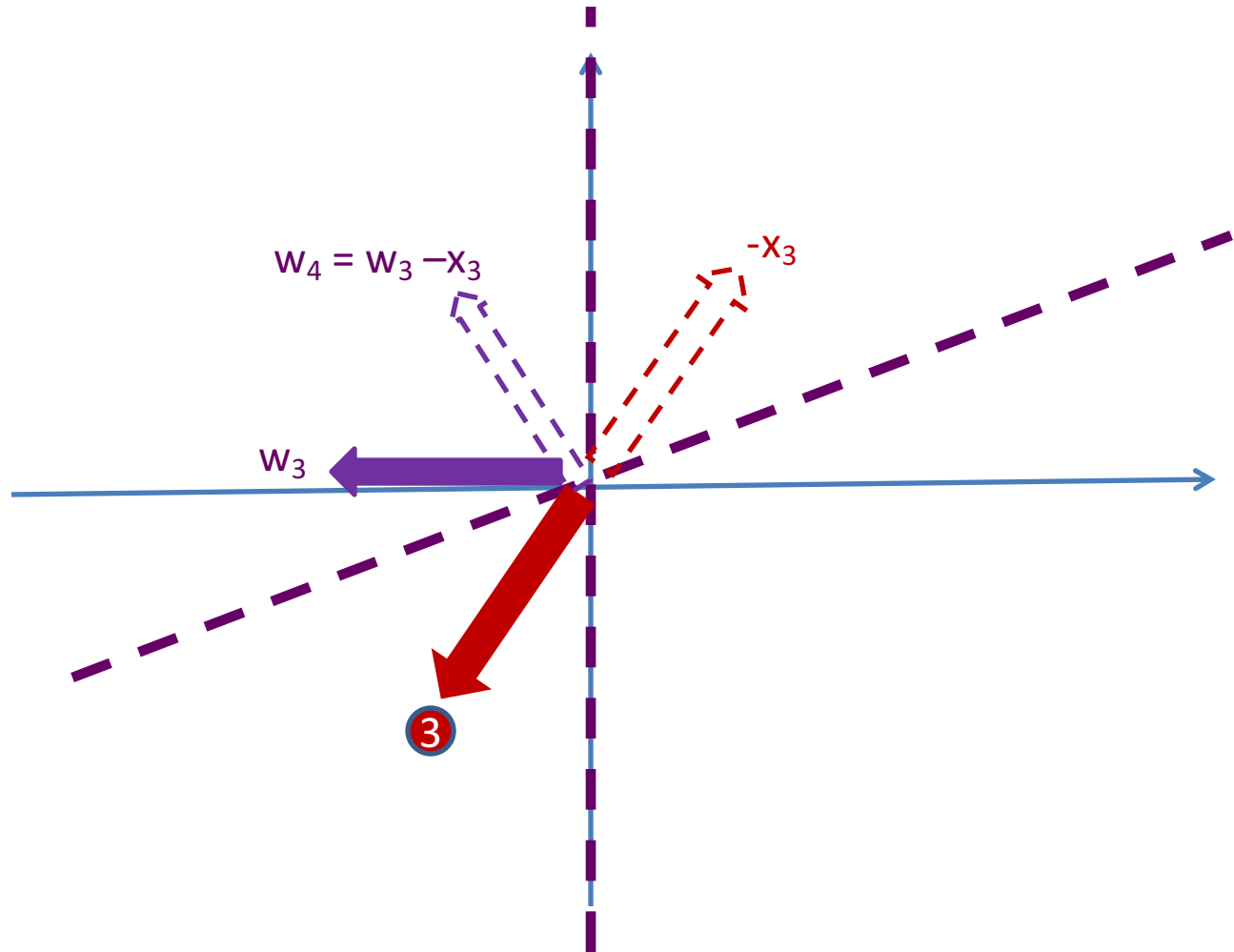
$$w_4 = (-0.2, +0.6)$$



$$x_1 - x_2 = 0$$

Perceptron - Geometric Interpretation

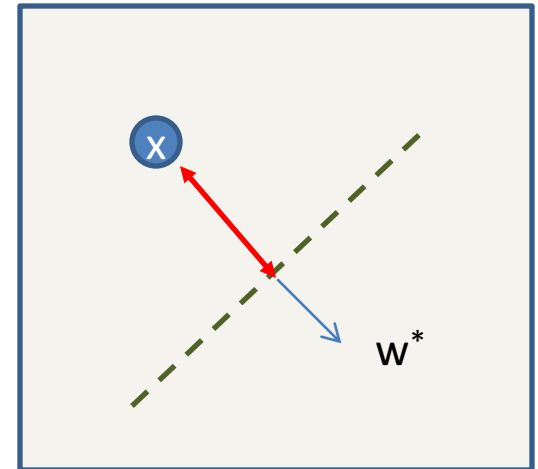
$w_1 = (0,0)$
 $w_2 = (0,0)$
 $w_3 = (-1,0)$
 $w_4 = (-0.2, +0.6)$



Perceptron - Analysis

- target concept $c^*(x)$ uses w^* and $\|w^*\| = 1$
- Margin γ :
 - For any x in S

$$\gamma = \min_{x \in S} \frac{|\langle x, w^* \rangle|}{\|x\|}$$



- **Theorem:** *Number of mistakes $\leq 1/\gamma^2$*

Perceptron - Performance

Claim 1:

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

Assume $c^*(x) = +1$

$$\langle w_{t+1}, w^* \rangle =$$

$$\langle (w_t + x), w^* \rangle =$$

$$\langle w_t, w^* \rangle + \langle x, w^* \rangle \geq$$

$$\langle w_t, w^* \rangle + \gamma$$

Similar for $c^*(x) = -1$

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

Assume $c^*(x) = +1$

$$\|w_{t+1}\|^2 =$$

$$\|w_t + x\|^2 =$$

$$\|w_t\|^2 + 2\langle w_t, x \rangle + \|x\|^2 \leq$$

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

Since x is a mistake $\langle w_t, x \rangle$ is negative.

Similar for $c^*(x) = -1$

Perceptron - performance

Claim 3: $\langle w_t, w^* \rangle \leq \|w_t\|$


$$\langle w_t, w^* \rangle \leq \langle w_t, \frac{w_t}{\|w_t\|} \rangle = \|w_t\|$$

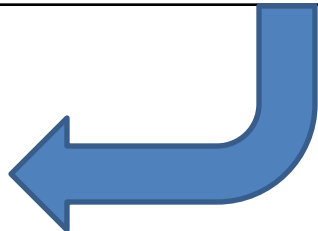
Completing the proof

• After M mistakes:

$$\langle w_{M+1}, w^* \rangle \geq \gamma M \quad (\text{claim 1})$$

$$\|w_{M+1}\|^2 \leq M \quad (\text{claim 2})$$


$$\gamma M \leq \langle w_{M+1}, w^* \rangle \leq \|w_{M+1}\| \leq \sqrt{M}$$


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

Perceptron

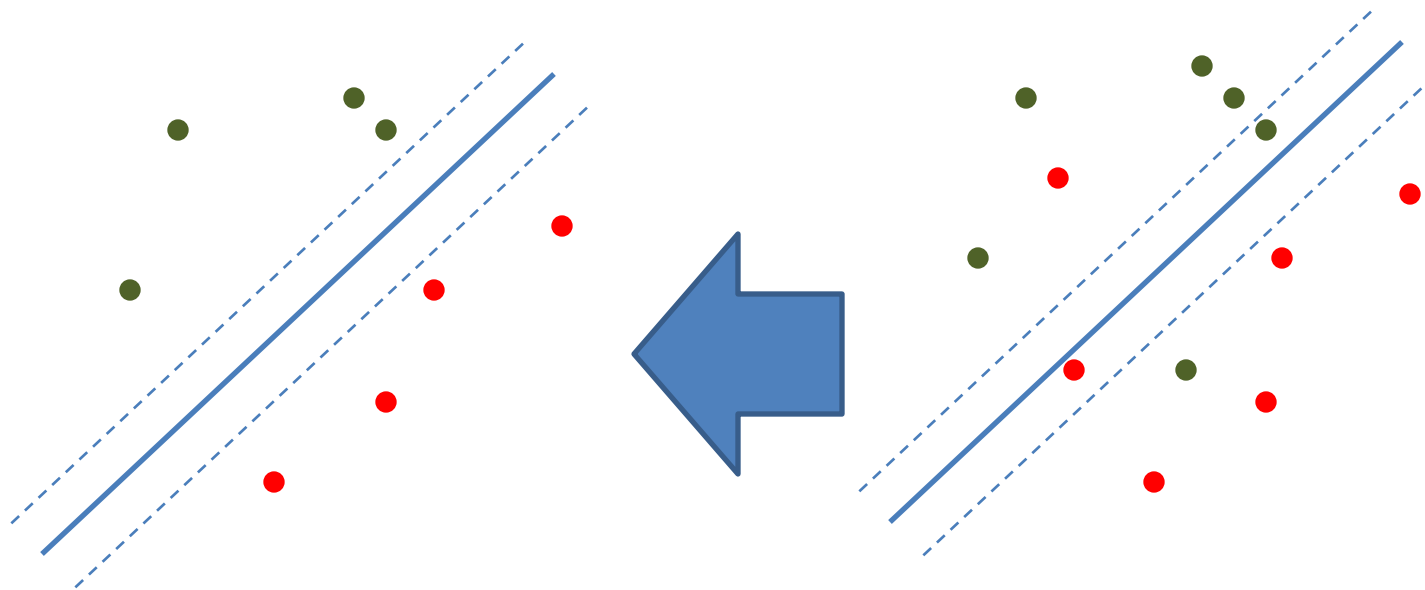
- Guaranteed convergence
 - realizable case
- Can be very slow (even for $\{0,1\}^d$)
- Additive increases:
 - problematic with large weights
- Still, a simple benchmark

Perceptron – Unrealizable case

Motivation

Realizable case

Unrealizable case



Hinge Loss

Motivation

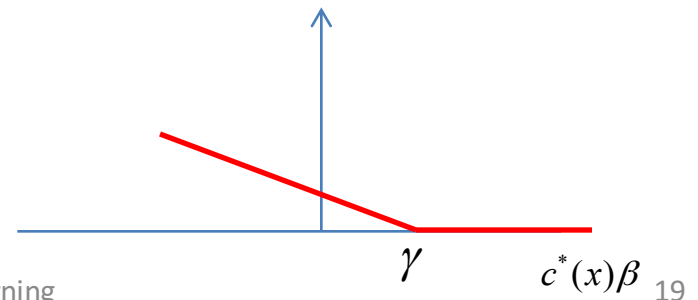
- “Move” points to be realizable
 - with margin γ
- correct points
 - both classification and margin
 - zero loss
- mistake points
 - even just margin
 - loss is the distance

Definition

- Assume $\langle x, w \rangle = \beta$
- Hinge Loss with margin γ :

$$\max\left\{0, 1 - \frac{c^*(x)\beta}{\gamma}\right\}$$

- Compare to Error: $c^*(x)\beta < 0$



Perceptron - Performance

- Let TD_γ = total distance
 $\sum_i \max\{0, \gamma - c^*(x)\beta_i\}$, where $\beta_i = \langle x_i, w^* \rangle$
- Claim 1': $\langle w_{M+1}, w^* \rangle \geq \gamma M - TD_\gamma$
- Claim 2: $\|w_{t+1}\|^2 \leq \|w_t\|^2 + 1$
- Bounding the mistakes:

$$\sqrt{M} \geq \gamma M - TD_\gamma \quad \Rightarrow \quad M \leq \frac{1}{\gamma^2} + \frac{2}{\gamma} TD_\gamma$$

Winnow

Winnow –motivation

- Updates
 - multiplicative vs additive
- Domain
 - $\{0,1\}^d$ or $[0,1]^d$
 - we will use $\{0,1\}^d$
- Weights
 - non-negative
 - monotone function
- Separation
 - $c^*(x)=+1: \langle w^*, x \rangle \geq \theta$
 - $c^*(x)=-1: \langle w^*, x \rangle \leq \theta - \gamma$
 - $\theta > 1$
 - part of the input
- Remarks:
 - normalizing x in L_∞ to 1

Winnow - Algorithm

- parameter $\beta > 1$
 - we will use $\beta = 1 + \gamma/2$
- Initialize $\mathbf{w} = (\mathbf{1}, \dots, \mathbf{1})$
- predict $h(\mathbf{x}) = +1$ iff
$$\langle \mathbf{w}, \mathbf{x} \rangle \geq \theta$$
- For a mistake:
- False Positive (**demotion**)
 - $c^*(\mathbf{x}) = -1, h(\mathbf{x}) = +1$
 - for every $x_i = 1$: $w_i = w_i / \beta$
- False Negative (**promotion**)
 - $c^*(\mathbf{x}) = +1, h(\mathbf{x}) = -1$
 - for every $x_i = 1$: $w_i = \beta w_i$

Winnow - intuition

- Demotion step
 - target negative
 - hypothesis positive
- Before update
$$\langle w, x \rangle = \alpha \geq \theta$$
- After the update:
$$\langle w, x \rangle = \alpha / \beta < \alpha$$
- Decrease in $\sum w_i$
 - at least $(1 - \beta^{-1})\theta$
- Promotion step
 - target positive
 - hypothesis negative
- Before update
$$\langle w, x \rangle = \alpha < \theta$$
- After the update:
$$\langle w, x \rangle = \alpha \beta > \alpha$$
- Increase in $\sum w_i$
 - at most $(\beta - 1)\theta$

Winnow - example

- Target function:
- $w^*=(2,2,0,0)$
- $\theta=2$, $\beta=2$
- What is the target function?
 - $x_1 \vee x_2$
 - monotone OR
- $w_0=(1,1,1,1)$
- $x_1=(0,0,1,1)$ $c_t(x_1)=-1$
 - $w_1=(1,1, \frac{1}{2}, \frac{1}{2})$
- $x_2=(1,0,1,0)$ $c_t(x_2)=+1$
 - $w_2=(2,1, 1, \frac{1}{2})$
- $x_3=(0,1,0,1)$ $c_t(x_3)=+1$
 - $w_3=(2,2, 1, 1)$

Winnow - Theorem

- **Theorem** (realizable case)

Number of mistakes bounded by

$$O\left(\frac{1}{\gamma^2} \frac{d}{\theta} + \frac{\ln \theta}{\gamma^2} \sum_{i=1}^d w_i^*\right)$$

- **Corollary:** For $\theta=d$ we have $O\left(\frac{\ln d}{\gamma^2} \sum_{i=1}^d w_i^*\right)$

Winnow - Analysis

- Mistakes
 - u promotion steps
 - v demotion steps
 - mistakes = u+v

- Lemma 1: due to sum equalities before, and since weights are positive

$$v \leq \frac{\beta}{\beta - 1} \frac{d}{\theta} + \beta u$$

- Lemma 2: $w_i \leq \beta\theta$

We won't promote unless w_i by itself is not enough to pass theta

- Lemma 3:
after u prom.
and v demo.
exists i

$$\log w_i \geq \frac{\theta u - (\theta - \gamma)v}{\sum_{i=1}^d w_i} \log \beta$$

- Proof of theorem

Winnow vs Perceptron

Perceptron

- Additive updates
 - slow for large d
 - slow large weights
- Non-monotone
 - natural
- Simple Algorithm
- Margin scale $L_2(w^*)L_2(x)$

Winnow

- Multiplicative updates
 - handles large d nicely
 - ok with large weights
- Monotone
 - need to make monotone
 - flip non-monotone attributes
- Simple Algorithm
- Margin scale $L_1(w^*)L_\infty(x)$
- Additional factor $\log d$
 - for $\theta=d$

Summary

Linear Separators

- Today: Perceptron and Winnow
- Next week: SVM
- 2 weeks: Kernels
- 3 weeks: Adaboost

Brief history:

- Perceptron
 - Rosenblatt 1957
- Fell out of favor in 70s
 - representation issues
- Reemerged with Neural nets
 - late 80s early 90s
- Linear separators:
 - Adaboost and SVM
- The immediate future: deep learning