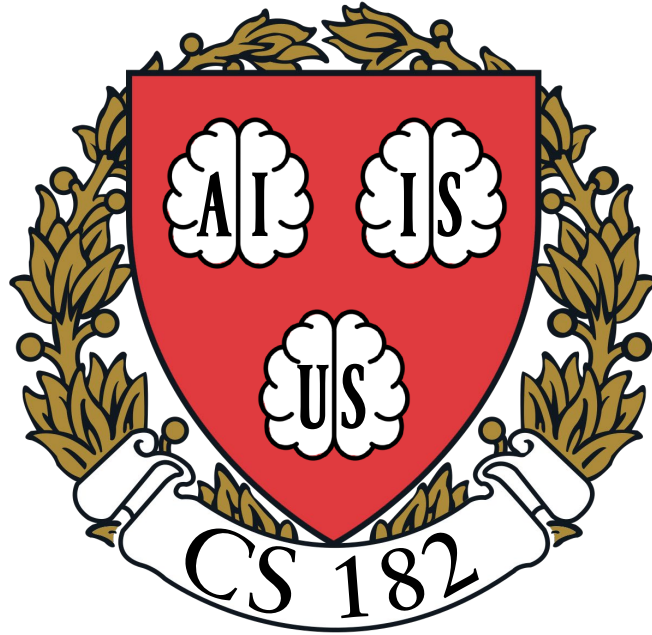


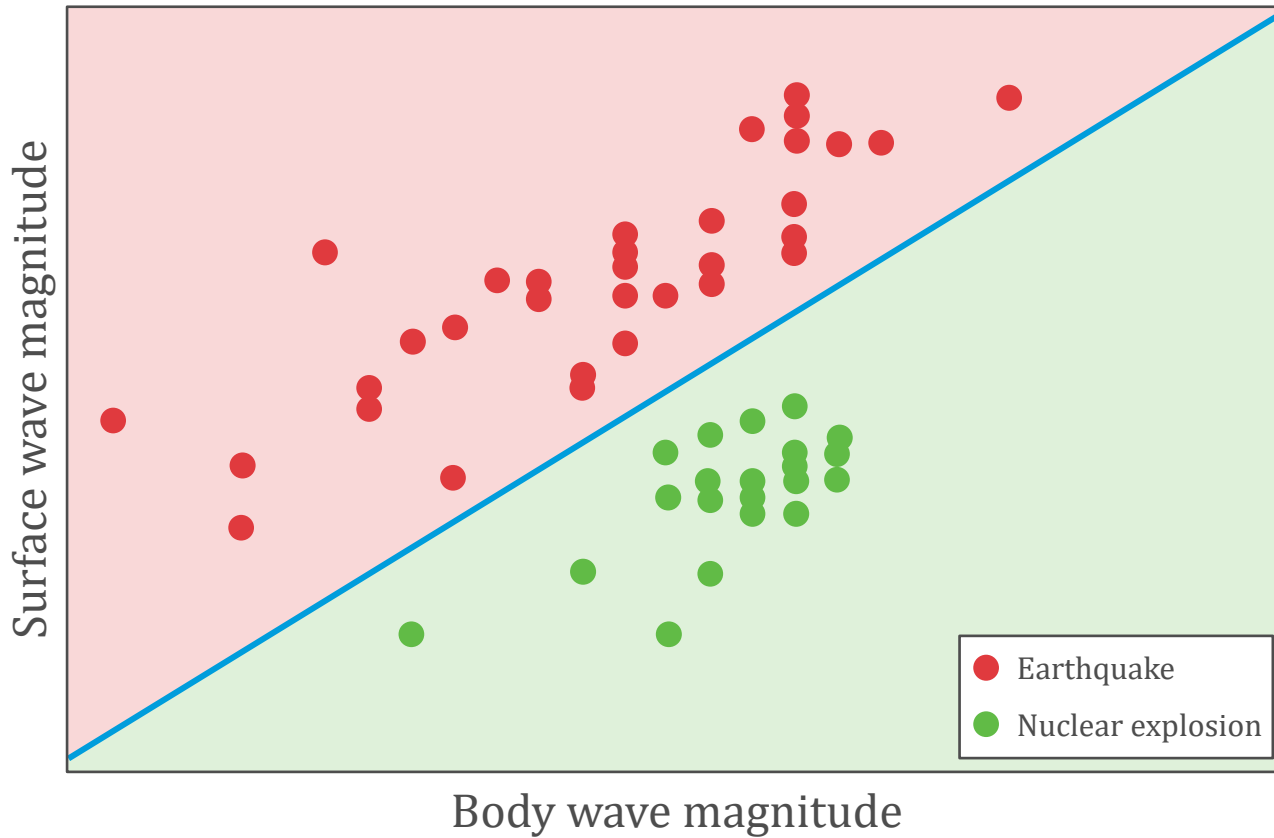


Fall 2021 | Lecture 17

Linear Classification

Ariel Procaccia | Harvard University

LINEAR CLASSIFICATION



Earthquakes and nuclear explosions in the Middle East and Asia, 1982-1990

LINEAR CLASSIFICATION

- A hypothesis is defined by

$$h_{\mathbf{w}}(\mathbf{x}) = \text{Threshold}(\mathbf{w} \cdot \mathbf{x})$$

where $\text{Threshold}(z) = +1$ if $z \geq 0$ and
 $\text{Threshold}(z) = -1$ if $z < 0$

- A **linear separator** can be found via a linear feasibility program

find $\mathbf{w}$

$$\text{s.t. } \forall i \in \mathcal{D}^+, \mathbf{w} \cdot \mathbf{x}^{(i)} \geq 0$$

$$\forall i \in \mathcal{D}^-, \mathbf{w} \cdot \mathbf{x}^{(i)} \leq -\epsilon$$

- But we want to learn **online**

The New York Times

July 13, 1958

Electronic 'Brain' Teaches Itself

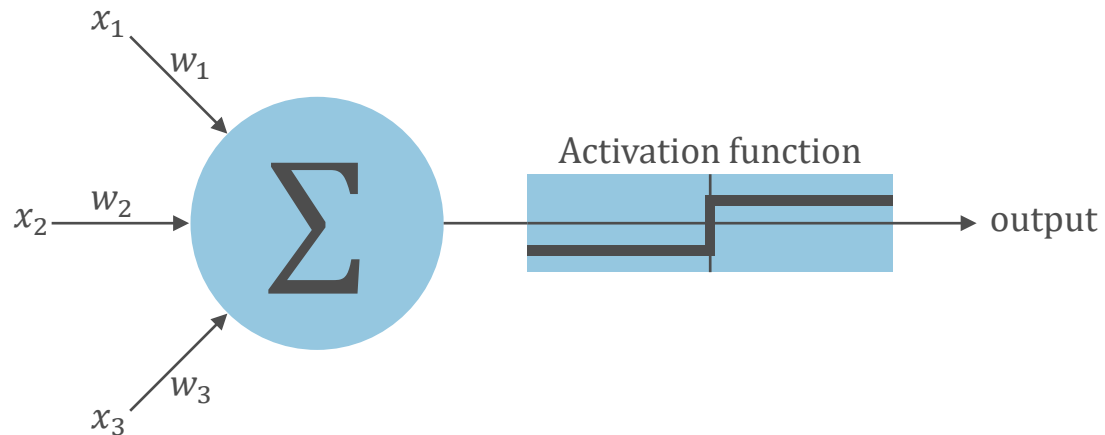
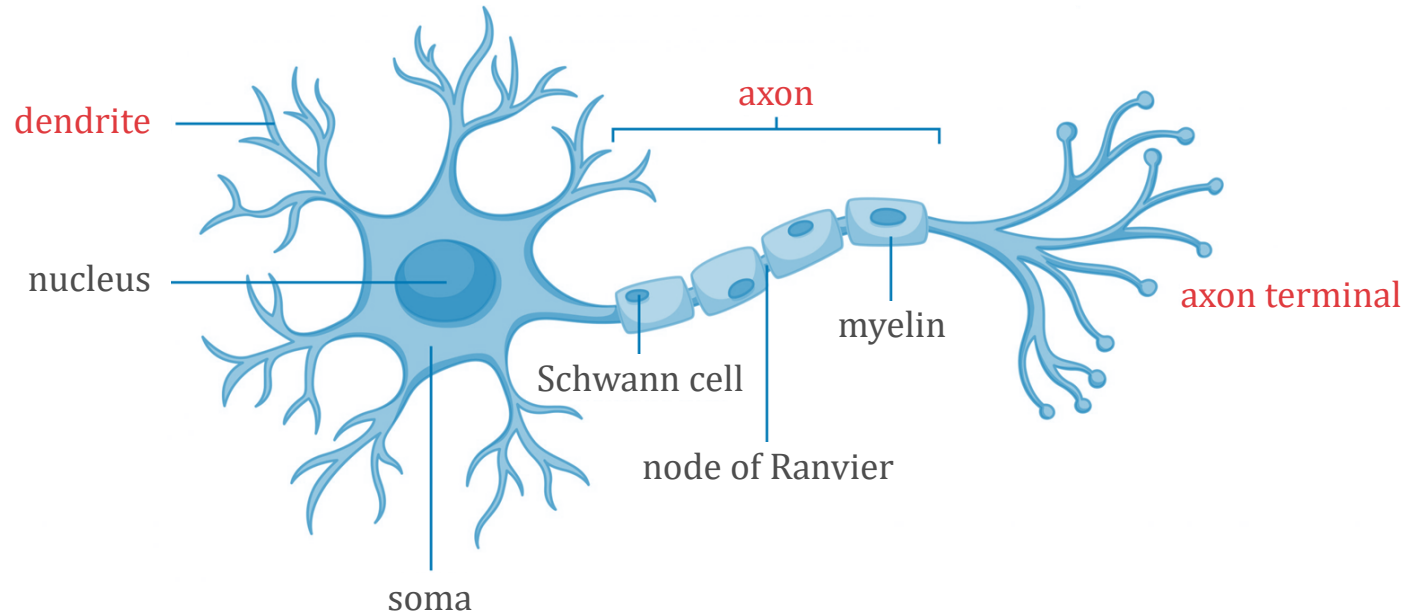
The Navy last week demonstrated the embryo of an electronic computer named the Perceptron which, when completed in about a year, is expected to be the first non-living mechanism able to "perceive, recognize and identify its surroundings without human training or control." Navy officers demonstrating a preliminary form of the device in Washington said they hesitated to call it a machine because it is so much like a "human being without life."

recognize the difference between right and left, almost the way a child learns.

When fully developed, the Perceptron will be designed to remember images and information it has perceived itself, whereas ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons, Dr. Rosenblatt said, will be able to recognize people and call out their names. Printed pages, longhand letters and even

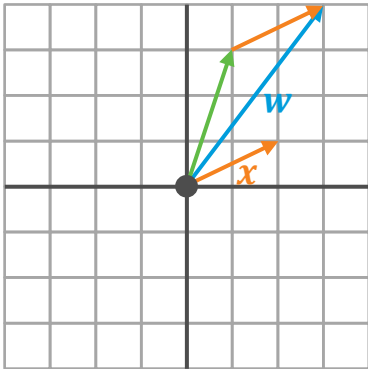
ELECTRONIC BRAIN



PERCEPTRON

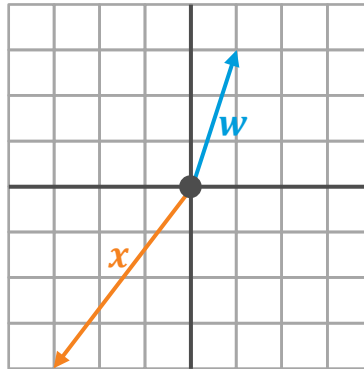
The Perceptron learning rule: For each example $(\mathbf{x}, y)$, classify $\hat{y} = \text{Threshold}(\mathbf{w} \cdot \mathbf{x})$ and, If $\hat{y} \neq y$, update $\mathbf{w} = \mathbf{w} + y \cdot \mathbf{x}$

Stage k



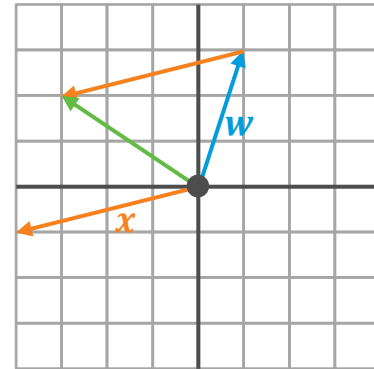
$$\begin{aligned}\mathbf{w} &= (3, 4) \\ (\mathbf{x}, y) &= ((2, 1), -1) \\ \mathbf{w} \cdot \mathbf{x} &= 10 \\ \mathbf{w} &= (3, 4) - (2, 1) = (1, 3)\end{aligned}$$

Stage $k + 1$



$$\begin{aligned}\mathbf{w} &= (1, 3) \\ (\mathbf{x}, y) &= ((-3, -4), -1) \\ \mathbf{w} \cdot \mathbf{x} &= -15 \\ \mathbf{w} &\text{ is unchanged}\end{aligned}$$

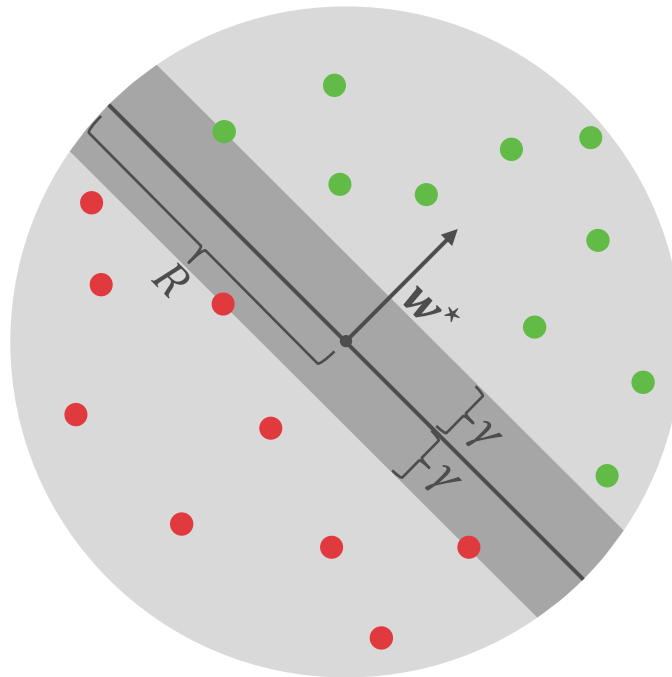
Stage $k + 2$



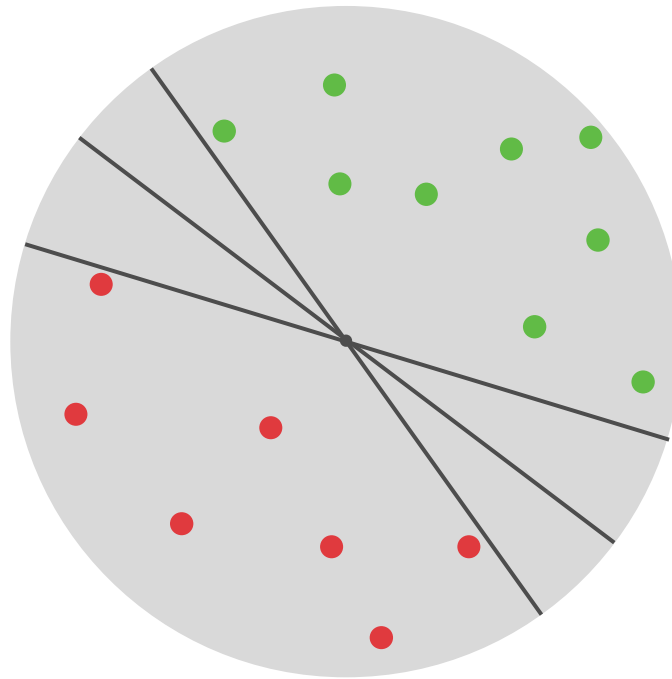
$$\begin{aligned}\mathbf{w} &= (1, 3) \\ (\mathbf{x}, y) &= ((-4, -1), 1) \\ \mathbf{w} \cdot \mathbf{x} &= -7 \\ \mathbf{w} &= (1, 3) + (-4, -1) = (-3, 2)\end{aligned}$$

PERCEPTRON MISTAKE BOUND

Theorem: Given a dataset $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^m$, if $\|\mathbf{x}^{(i)}\| \leq R$ for all i , and there exists $\mathbf{w}^*$ such that $\|\mathbf{w}^*\| = 1$ and $y^{(i)}(\mathbf{w}^* \cdot \mathbf{x}^{(i)}) \geq \gamma$ for all i , then the number of mistakes made by the Perceptron is at most $(R/\gamma)^2$

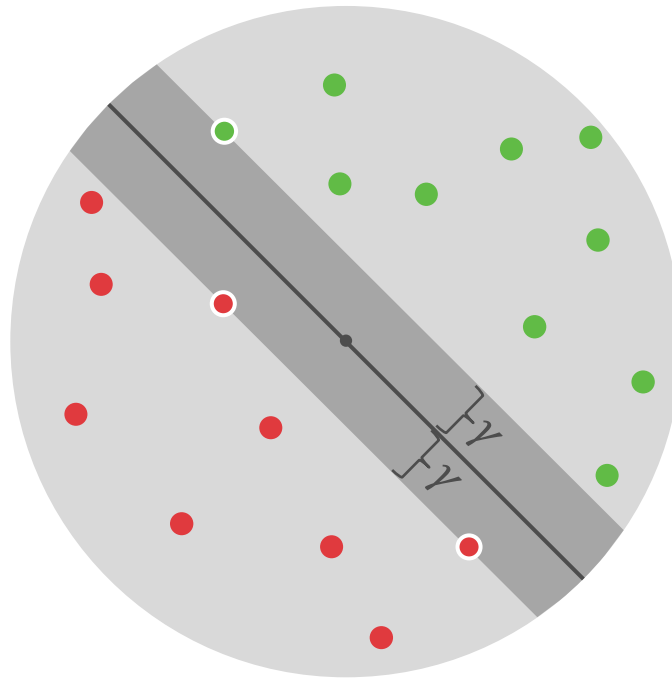


GENERALIZATION



Which separator would you expect to generalize better?

SUPPORT VECTOR MACHINES



Find a **maximum margin** separator, where the margin is defined by **support vectors**

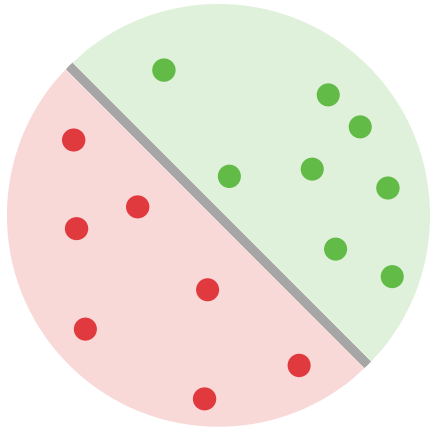
SUPPORT VECTOR MACHINES

- Through tedious derivations one can find that, to maximize the margin, it suffices to solve the quadratic program

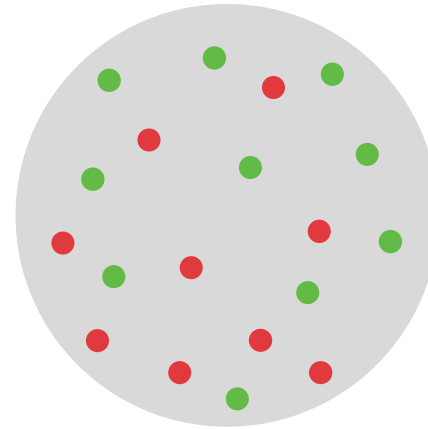
$$\begin{aligned} \max_{\alpha} \quad & \sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y^{(j)} y^{(k)} (\mathbf{x}^{(j)} \cdot \mathbf{x}^{(k)}) \\ \text{s.t.} \quad & \sum_j y^{(j)} \alpha_j = 0 \\ & \forall j, \alpha_j \geq 0 \end{aligned}$$

- We can recover the separator from this program's solution
- It holds that $\alpha_j \neq 0$ only for support vectors $\mathbf{x}^{(j)}$

NON-SEPARABLE DATA



Linearly
separable



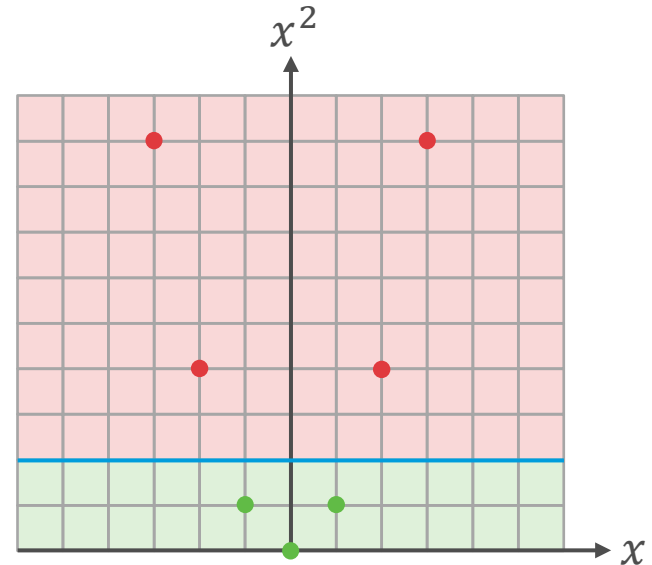
Not linearly
separable

Poll 1: Would the Perceptron algorithm converge on non-separable data? **Maybe**

APPROACH 1: HIGHER DIMENSIONS

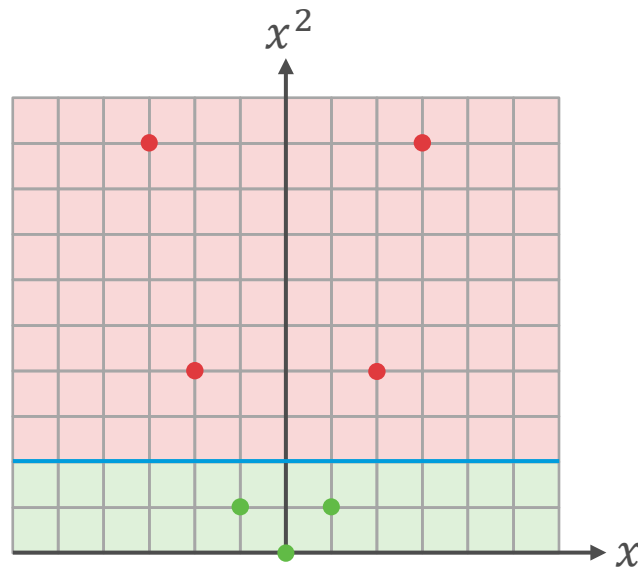


Not linearly
separable



Linearly
separable

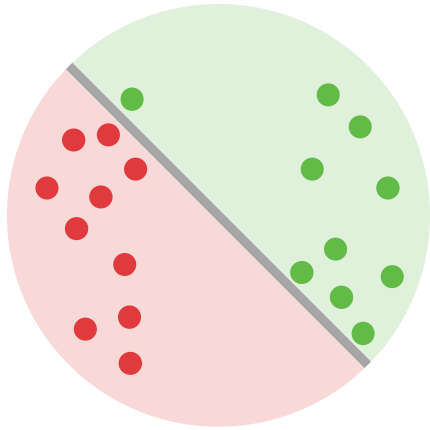
APPROACH 1: HIGHER DIMENSIONS



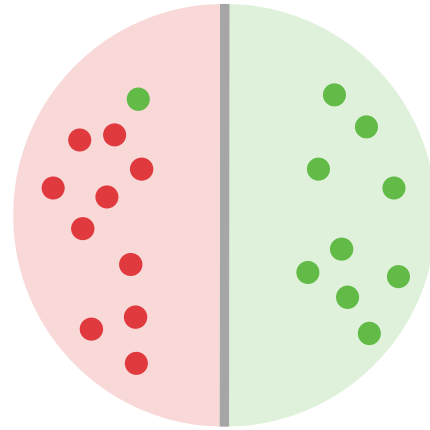
Poll 2: What is the set of positively labeled points on the line that the above separator corresponds to?

- $[0, \sqrt{2}]$
- $[\sqrt{2}, \infty]$
- $[-\sqrt{2}, \sqrt{2}]$ ✓
- $[-\infty, -\sqrt{2}] \cup [\sqrt{2}, \infty]$

APPROACH 2: SOFT MARGIN



No mistakes
small margins

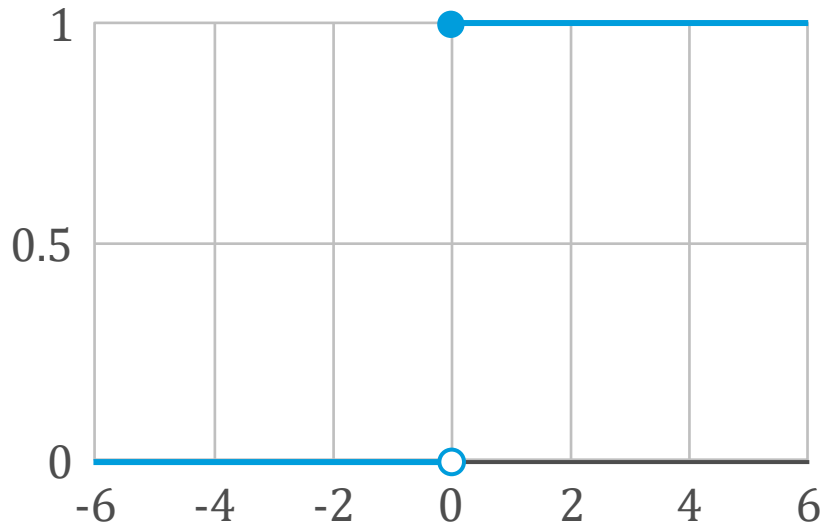


One mistake
Large margins

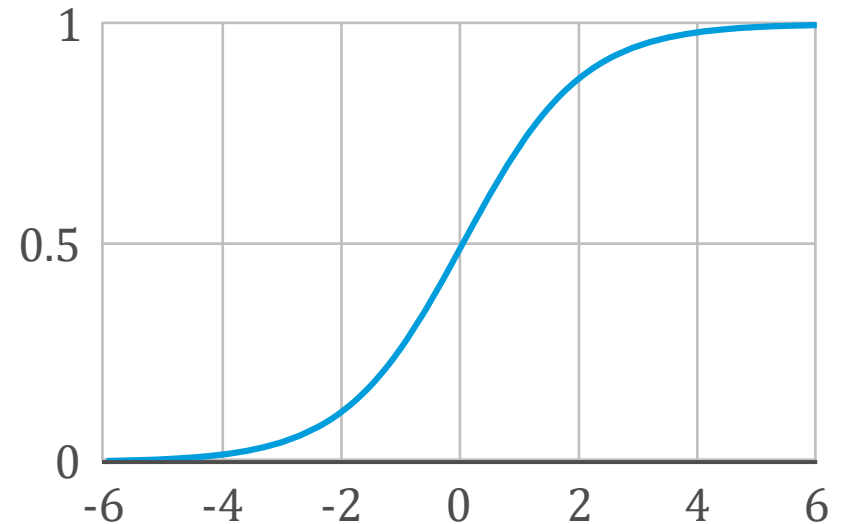
It's possible to enable soft margins by introducing slack variables into the SVM objective and constraints

LOGISTIC ACTIVATION FUNCTION

Another approach for enabling soft margins:



Threshold (step) function
(0/1 version)



Logistic (sigmoid) function
 $f(z) = 1/(1 + e^{-z})$

LOGISTIC REGRESSION

- True labels are $y \in \{0,1\}$
- Denote the logistic function by

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- A hypothesis is defined by

$$h_{\mathbf{w}}(\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}}$$

- Interpreted as the probability that the label of $\mathbf{x}$ is 1

LOGISTIC REGRESSION

- Probability of observing $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ under $\mathbf{w}$ is

$$\prod_i \left[\sigma(\mathbf{w} \cdot \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left(1 - \sigma(\mathbf{w} \cdot \mathbf{x}^{(i)})\right)^{1-y^{(i)}} \right]$$

- The **log-likelihood function** is

$$\begin{aligned} LL(\mathbf{w}) &= \log \left(\prod_i \left[\sigma(\mathbf{w} \cdot \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left(1 - \sigma(\mathbf{w} \cdot \mathbf{x}^{(i)})\right)^{1-y^{(i)}} \right] \right) \\ &= \sum_i \left[y^{(i)} \log \left(\sigma(\mathbf{w} \cdot \mathbf{x}^{(i)}) \right) + (1 - y^{(i)}) \log \left(1 - \sigma(\mathbf{w} \cdot \mathbf{x}^{(i)}) \right) \right] \end{aligned}$$

- Our goal is to maximize the concave function $LL(\mathbf{w})$

WORDS OF WISDOM

Hebrew - detected



English



מה ששנוא עליך אל
תעשה לחברך

Do not treat others
the way you do not
want to be treated



 *Verified*

[Open in Google Translate](#) • [Feedback](#)

GRADIENT ASCENT, IN DETAIL

- If we take steps in the direction of the gradient, we eventually reach a local maximum
- Since our problem is convex, a local maximum is a global maximum
- The gradient ascent update step is

$$w_j^{(t+1)} = w_j^{(t)} + \alpha_t \frac{\partial LL(\mathbf{w}^{(t)})}{\partial w_j^{(t)}}$$

where α_t is the learning rate

GRADIENT ASCENT, IN DETAIL

- One can verify that $\frac{\partial}{\partial z} \sigma(z) = \sigma(z)(1 - \sigma(z))$
- For a single example $(\mathbf{x}, y)$, we have

$$\begin{aligned}\frac{\partial LL(\mathbf{w})}{\partial w_j} &= \frac{\partial}{\partial w_j} y \log \sigma(\mathbf{w} \cdot \mathbf{x}) + \frac{\partial}{\partial w_j} (1 - y) \log(1 - \sigma(\mathbf{w} \cdot \mathbf{x})) \\&= \left[\frac{y}{\sigma(\mathbf{w} \cdot \mathbf{x})} - \frac{1 - y}{1 - \sigma(\mathbf{w} \cdot \mathbf{x})} \right] \frac{\partial}{\partial w_j} \sigma(\mathbf{w} \cdot \mathbf{x}) \\&= \left[\frac{y}{\sigma(\mathbf{w} \cdot \mathbf{x})} - \frac{1 - y}{1 - \sigma(\mathbf{w} \cdot \mathbf{x})} \right] \sigma(\mathbf{w} \cdot \mathbf{x})(1 - \sigma(\mathbf{w} \cdot \mathbf{x})) x_j \\&= \left[\frac{y - \sigma(\mathbf{w} \cdot \mathbf{x})}{\sigma(\mathbf{w} \cdot \mathbf{x})(1 - \sigma(\mathbf{w} \cdot \mathbf{x}))} \right] \sigma(\mathbf{w} \cdot \mathbf{x})(1 - \sigma(\mathbf{w} \cdot \mathbf{x})) x_j \\&= [y - \sigma(\mathbf{w} \cdot \mathbf{x})] x_j\end{aligned}$$

- We conclude that

$$\frac{\partial LL(\mathbf{w})}{\partial w_j} = \sum_{i=1}^n [y^{(i)} - \sigma(\mathbf{w} \cdot \mathbf{x}^{(i)})] x_j^{(i)}$$