CS 383: Machine Learning

Prof Adam Poliak

Fall 2024

11/05/2024

Lecture 20

Perceptron as a neural network

Biological model of a neuron

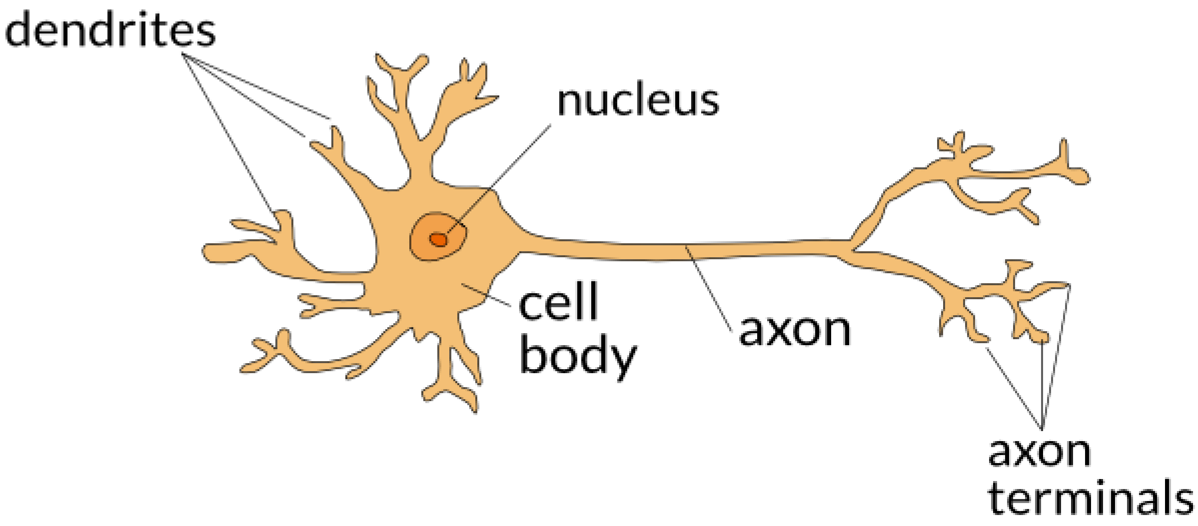


Image: modified from “Applied Go”

Perceptron as a neural network

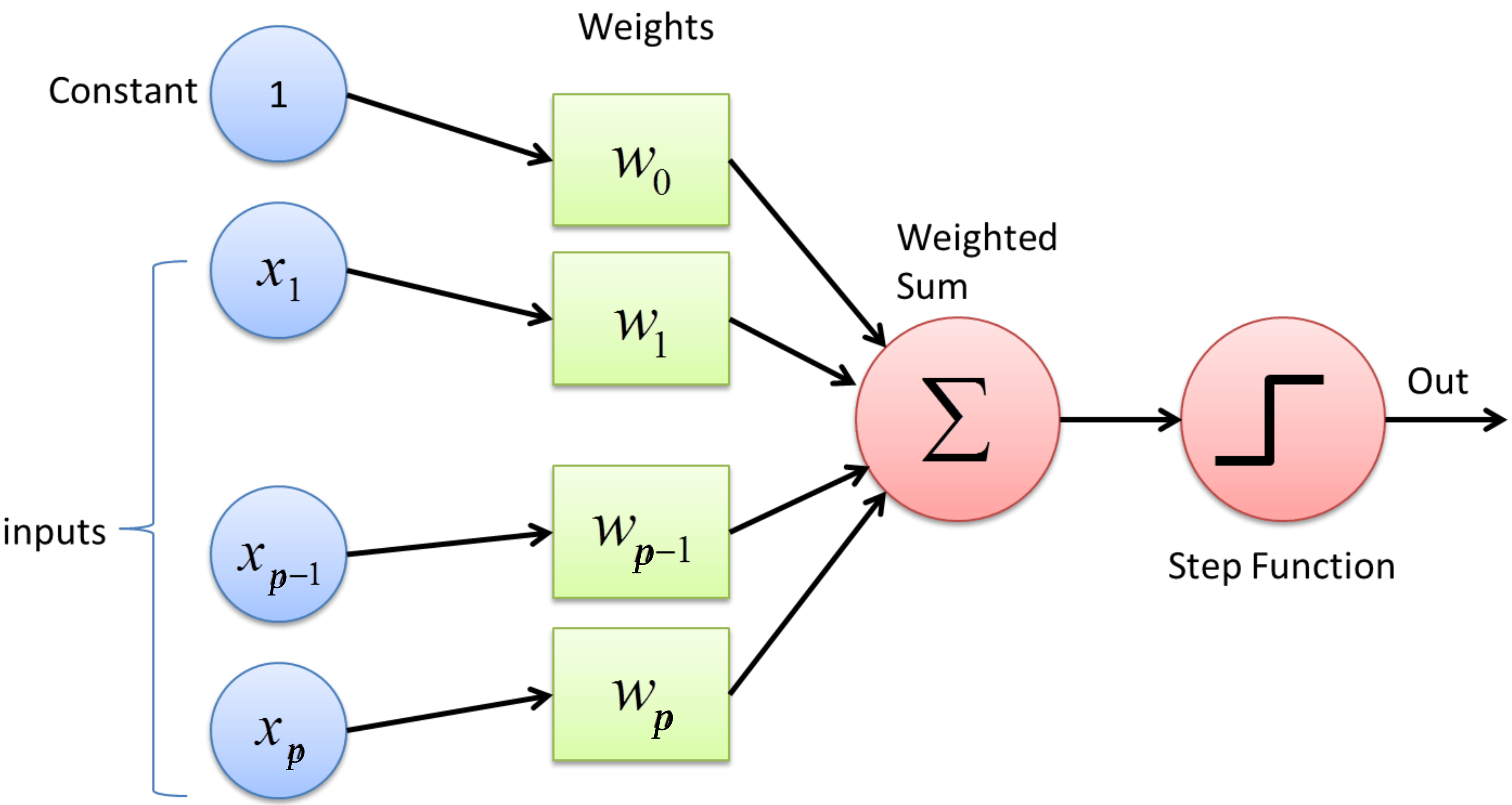


Image: modified from "Towards Data Science"

Perceptron Algorithm: Making a prediction

$y \in \{-1, +1\}$

$h(\vec{x}) = \text{sign}(\vec{w} \ * \mathrm{x})$

If $\vec{w} \ * \mathrm{x} > 0, \Rightarrow \ \ \hat{y} = +1$

If $\vec{w} \ * \mathrm{x} < 0, \Rightarrow \ \ \hat{y} = -1$

Perceptron Algorithm: updating weights

Set $\vec{w}$ to 0-vector

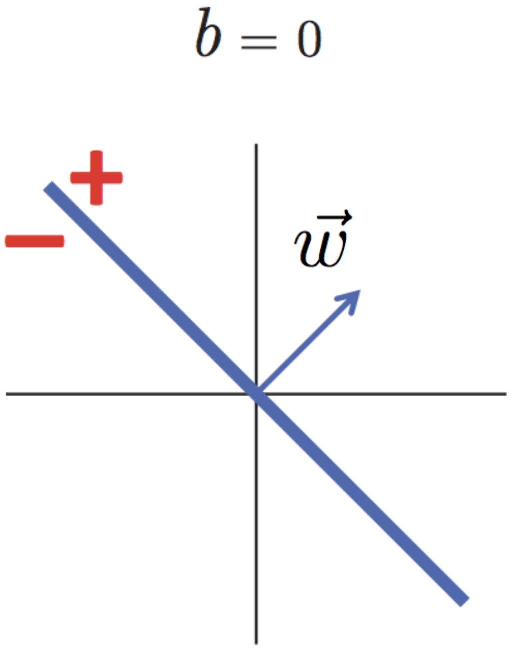
Repeat until training set is perfectly classified:

1. Randomly choose (x_i, y_i)
2. Predict $\widehat{y_i}$
3. If $\widehat{y_i} = y_i$:
    1. do nothing
4. Else:
    1. $\vec{w} \leftarrow \vec{w} + y_i \, \overrightarrow{x_i}$

The ***bias*** (*b*) and the *y*-intercept are different, but they both capture a “shift” away from the origin.

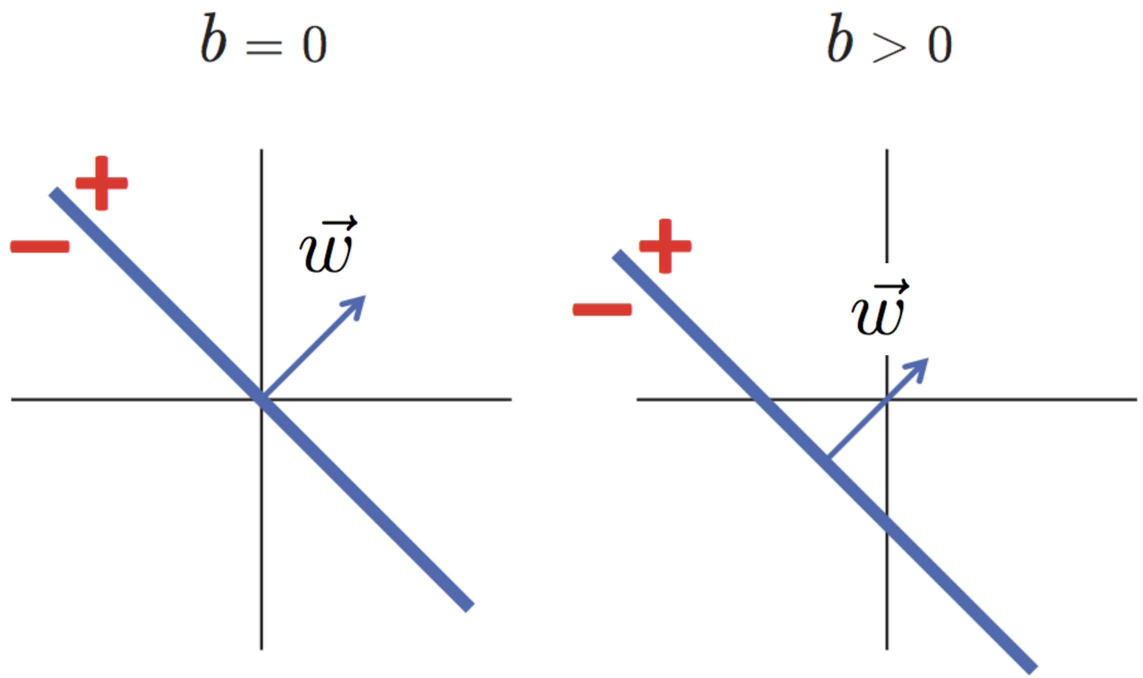
Example modified from Jessica Wu

The ***bias*** (*b*) and the *y*-intercept are different, but they both capture a “shift” away from the origin.



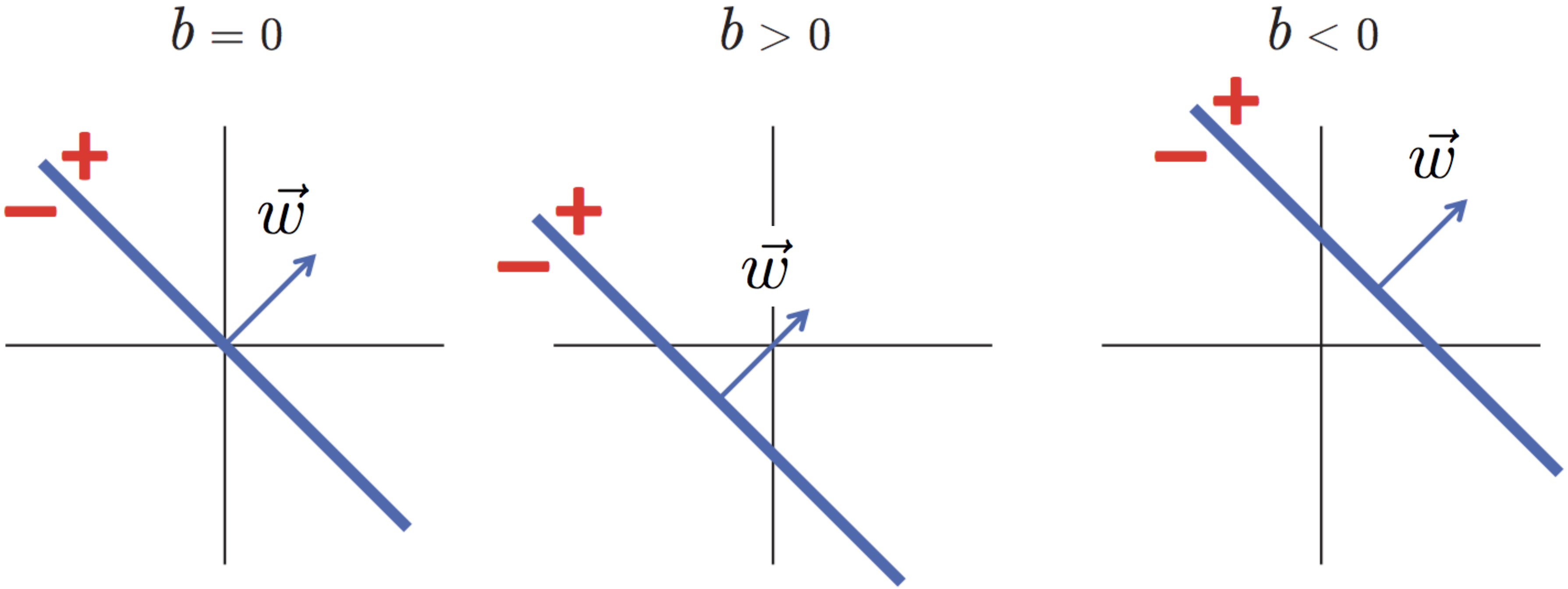
Example modified from Jessica Wu

The ***bias*** (b) and the y-intercept are different, but they both capture a "shift" away from the origin.



Example modified from Jessica Wu

The ***bias*** (*b*) and the *y*-intercept are different, but they both capture a "shift" away from the origin.



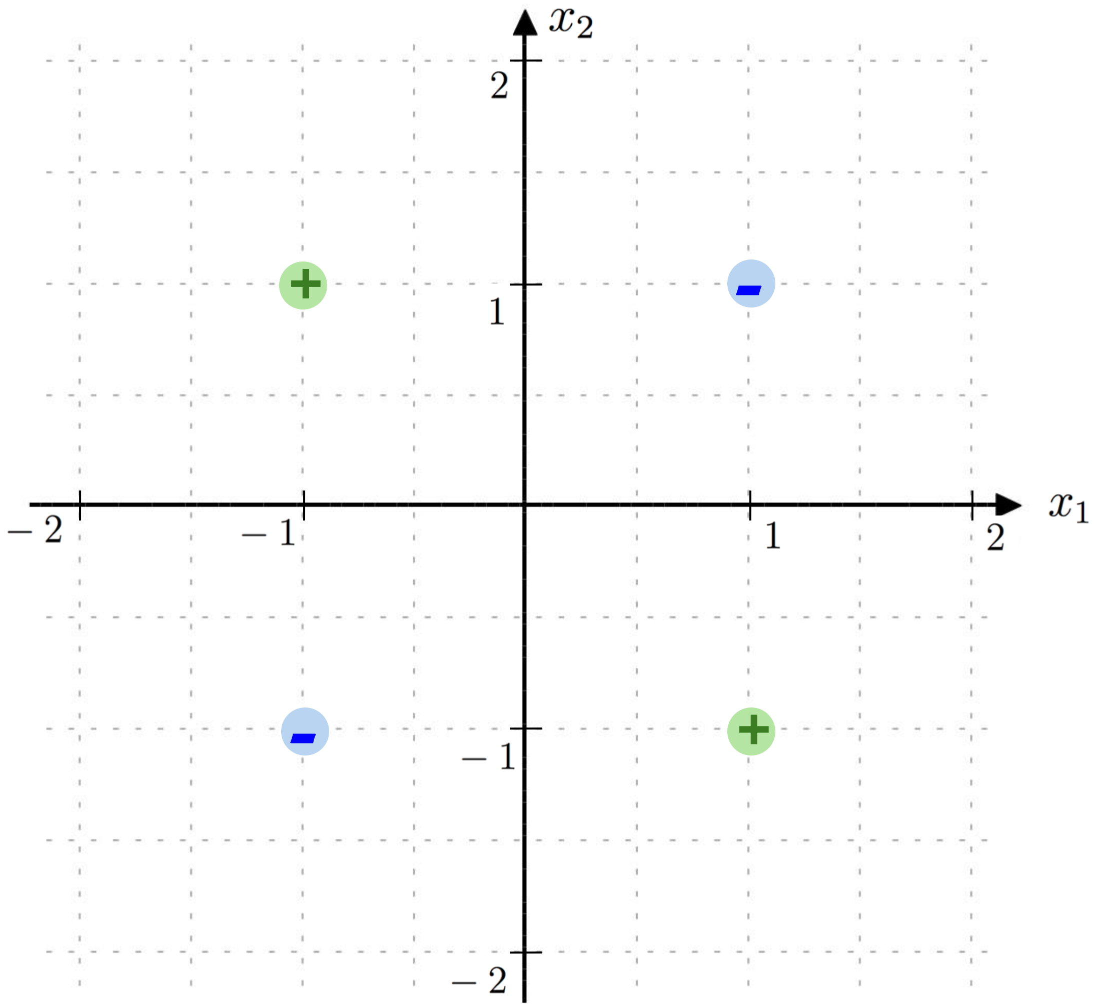
With p=2, if w_2 is positive, then the above example holds

Example modified from Jessica Wu

Perceptron cant learn XOR

(x_1 = 1 or x_2 = 1, but not both)

Why?

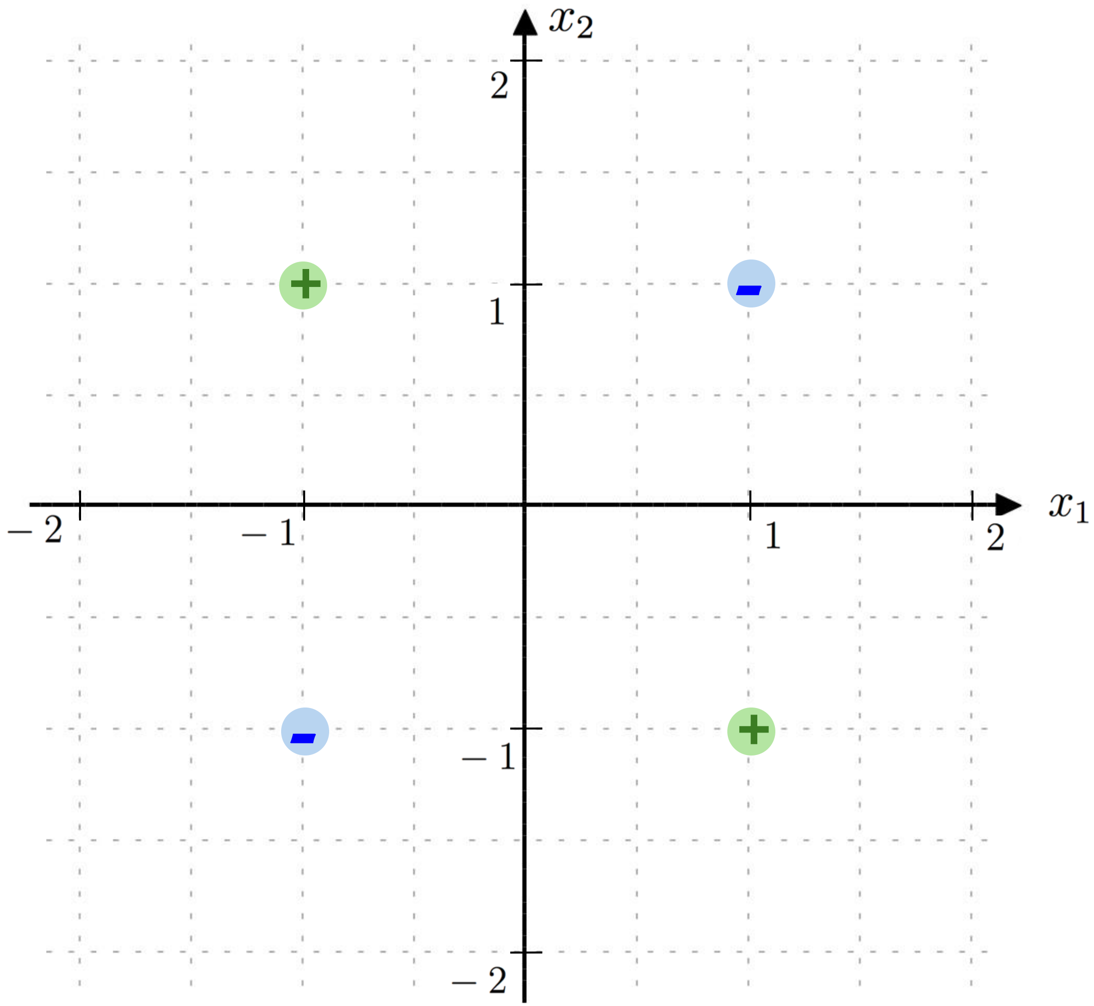


Perceptron cant learn XOR

(x_1 = 1 or x_2 = 1, but not both)

Why?

Not linearly separable!



Convergence Guarantee

- Perceptron is guaranteed to converge to a solution if a separating hyperplane exists
- Not guaranteed to converge to a “good” solution
- No guarantees about behavior if a separating hyperplane does not exist!

Cost function for Perceptron

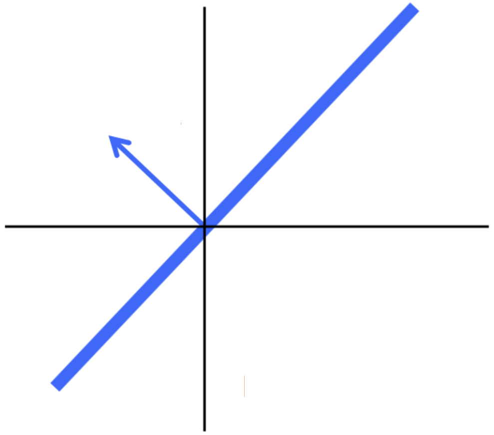
Hinge Loss

- $J(\vec{w}) = \sum_{i=1}^{n} max(0, -y_i \vec{w} x_i)$

1) What is the relationship between the weight vector ***w*** and the hyperplane?

1) What is the relationship between the weight vector ***w*** and the hyperplane?

They are perpendicular

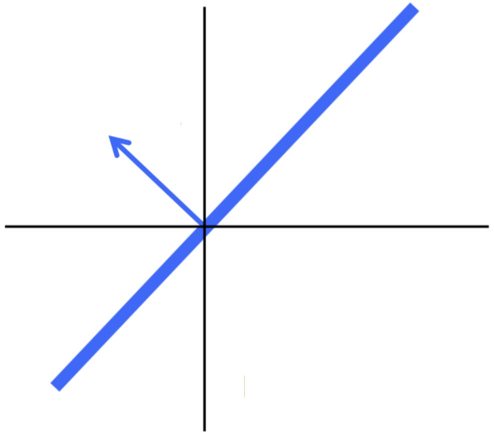


1) What is the relationship between the weight vector ***w*** and the hyperplane?

They are perpendicular

2) Why is the perceptron cost function intuitive?

$$J(\vec{w}) = \sum_{i=1}^{n} \max\left(0, -y_i(\vec{w}^T\vec{x}_i)\right)$$



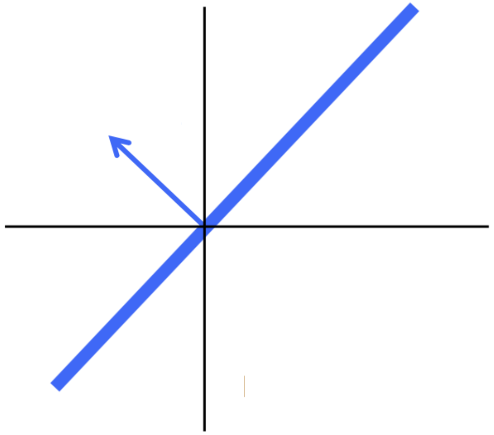
1) What is the relationship between the weight vector ***w*** and the hyperplane?

They are perpendicular

2) Why is the perceptron cost function intuitive?

Cost function is 0 when classification is correct, and positive when incorrect

$$J(\vec{w}) = \sum_{i=1}^{n} \max\left(0, -y_i(\vec{w}^T \vec{x}_i)\right)$$



1) What is the relationship between the weight vector ***w*** and the hyperplane?

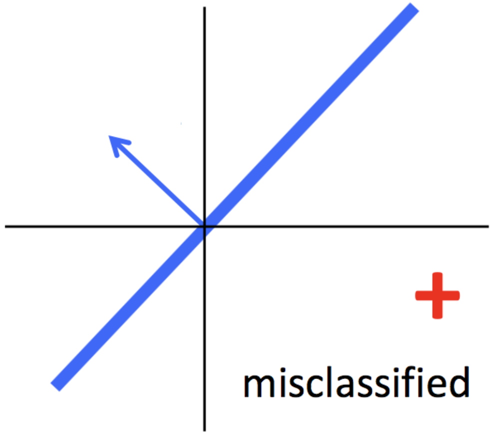
They are perpendicular

2) Why is the perceptron cost function intuitive?

Cost function is 0 when classification is correct, and positive when incorrect

$$J(\vec{w}) = \sum_{i=1}^{n} \max\left(0, -y_i(\vec{w}^T \vec{x}_i)\right)$$

3) In the example to the right, how will the slope of the hyperplane change?



1) What is the relationship between the weight vector ***w*** and the hyperplane?

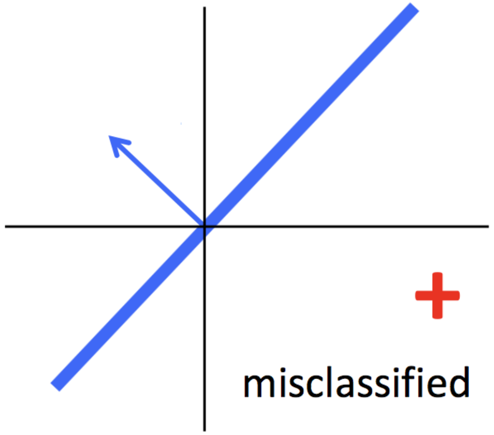
They are perpendicular

2) Why is the perceptron cost function intuitive?

Cost function is 0 when classification is correct, and positive when incorrect

$$J(\vec{w}) = \sum_{i=1}^{n} \max\left(0, -y_i(\vec{w}^T \vec{x}_i)\right)$$

3) In the example to the right, how will the slope of the hyperplane change?



Perceptron algorithm and intuition

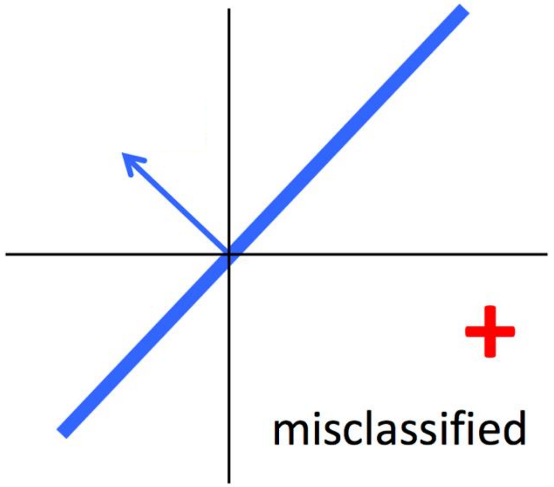


Image and Algorithm: modified from Eric Eaton

Perceptron algorithm and intuition

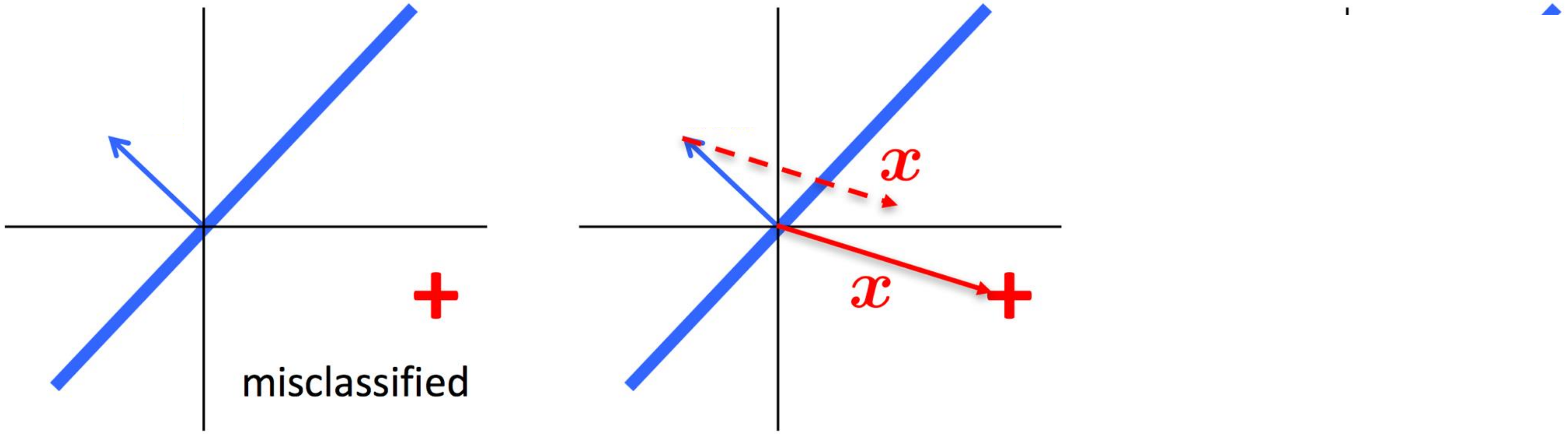


Image and Algorithm: modified from Eric Eaton

Perceptron algorithm and intuition

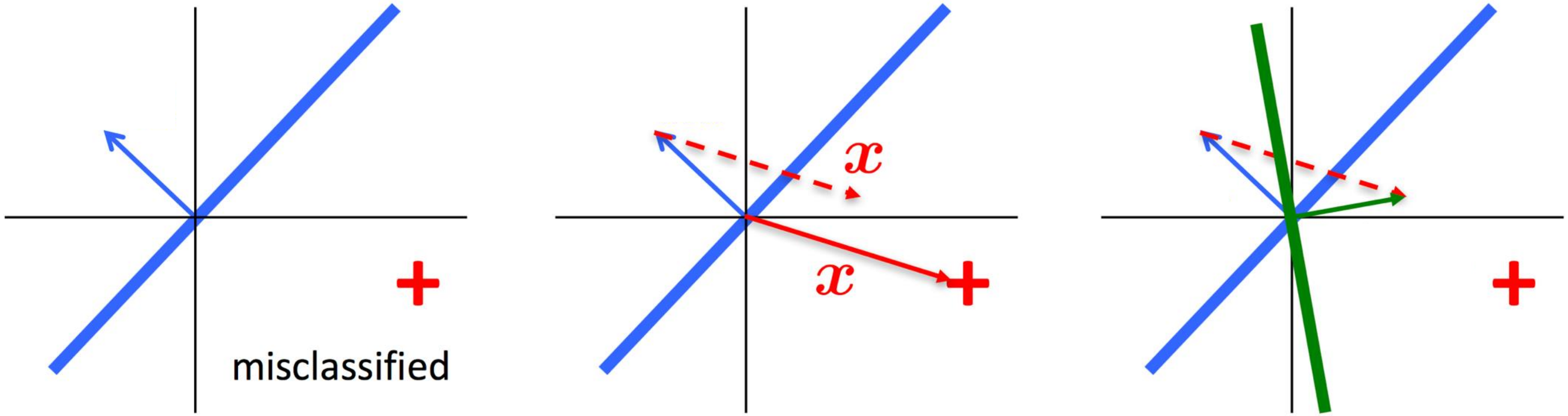
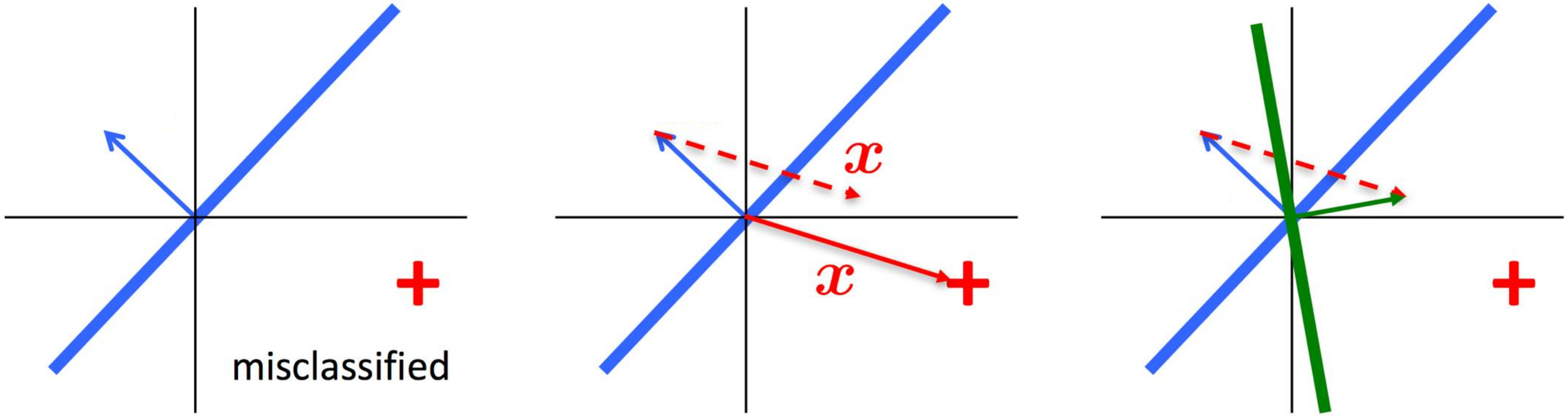


Image and Algorithm: modified from Eric Eaton

Perceptron algorithm and intuition



Let $\vec{w} = [0, 0, \cdots, 0]^T$
Repeat until convergence:
Receive training example $(\vec{x}_i, y_i)$
$y_i(\vec{w}^T \vec{x}_i) \leq 0$ (incorrectly classified)
$\vec{w} \leftarrow \vec{w} + \alpha y_i \vec{x}_i$

Convergence:
- All data points correctly classified
- Fixed number of iterations passed

Often: alpha = 1 (only changes magnitude of weight vector)

Image and Algorithm: modified from Eric Eaton

1) What is the relationship between the weight vector ***w*** and the hyperplane?

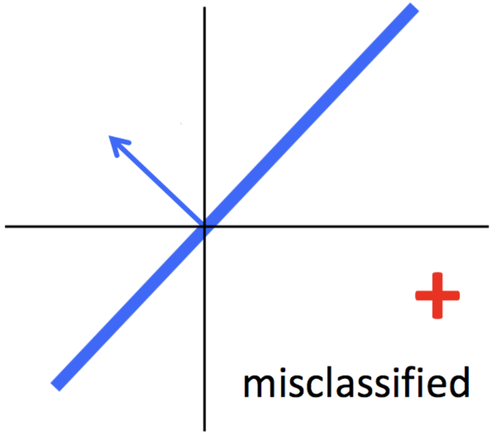
They are perpendicular

2) Why is the perceptron cost function intuitive?

Cost function is 0 when classification is correct, and positive when incorrect

$$J(\vec{w}) = \sum_{i=1}^{n} \max\left(0, -y_i(\vec{w}^T\vec{x}_i)\right)$$

3) In the example to the right, how will the slope of the hyperplane change?



4) What are the weaknesses of the perceptron? Create a binary classifier “wishlist”.

Wishlist

If data is linearly separable, want a “good” hyperplane (idea: far from points close to the boundary)

If data is not linearly separable, want something reasonable (not just give up or fail to converge)

Might not want to constrain ourselves to linear separators

Support Vector Machine (SVM)

- Will give us everything on our wishlist!
- Often considered the best "off the shelf" binary classifier
- Widely used in many fields

Brief history

- 1963: Initial idea by Vladimir Vapnik and Alexey Chervonenkis
- 1992: nonlinear SVMs by Bernhard Boser, Isabelle Guyon and Vladimir Vapnik
- 1993: "soft-margin" by Corinna Cortes and Vladimir Vapnik

Which is the best hyperplane?

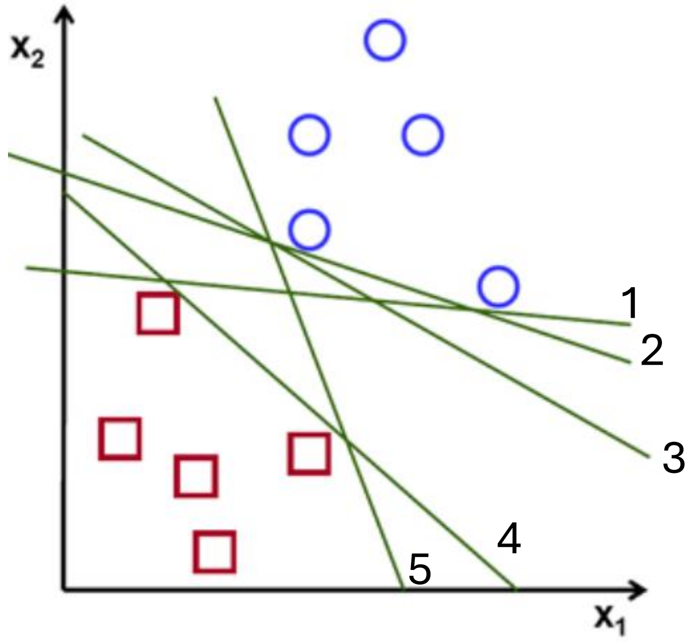


Image: "Towards Data Science"

The one with the highest margin

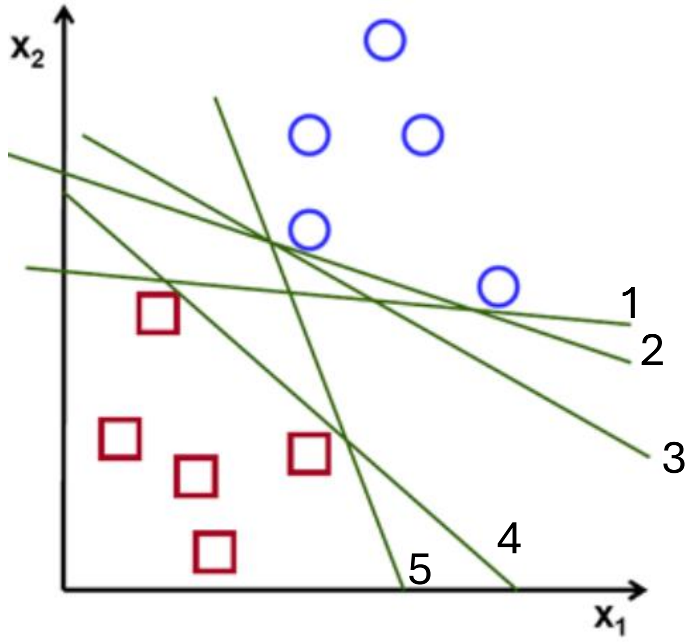


Image: “Towards Data Science”

The one with the highest margin

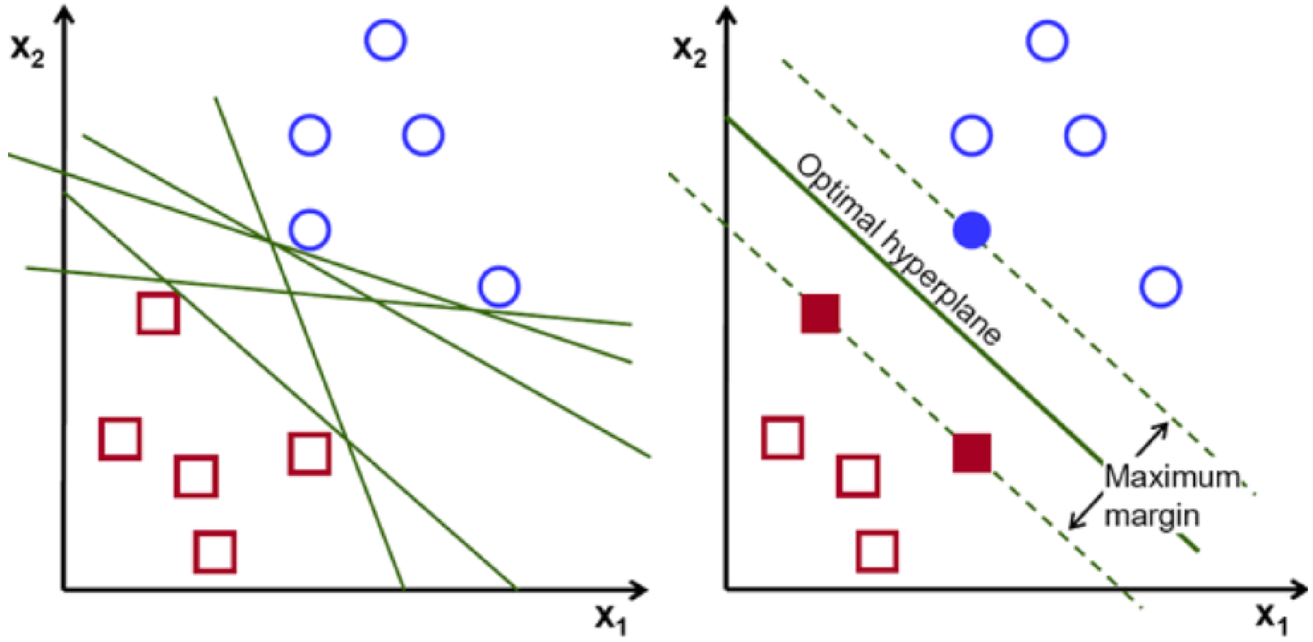
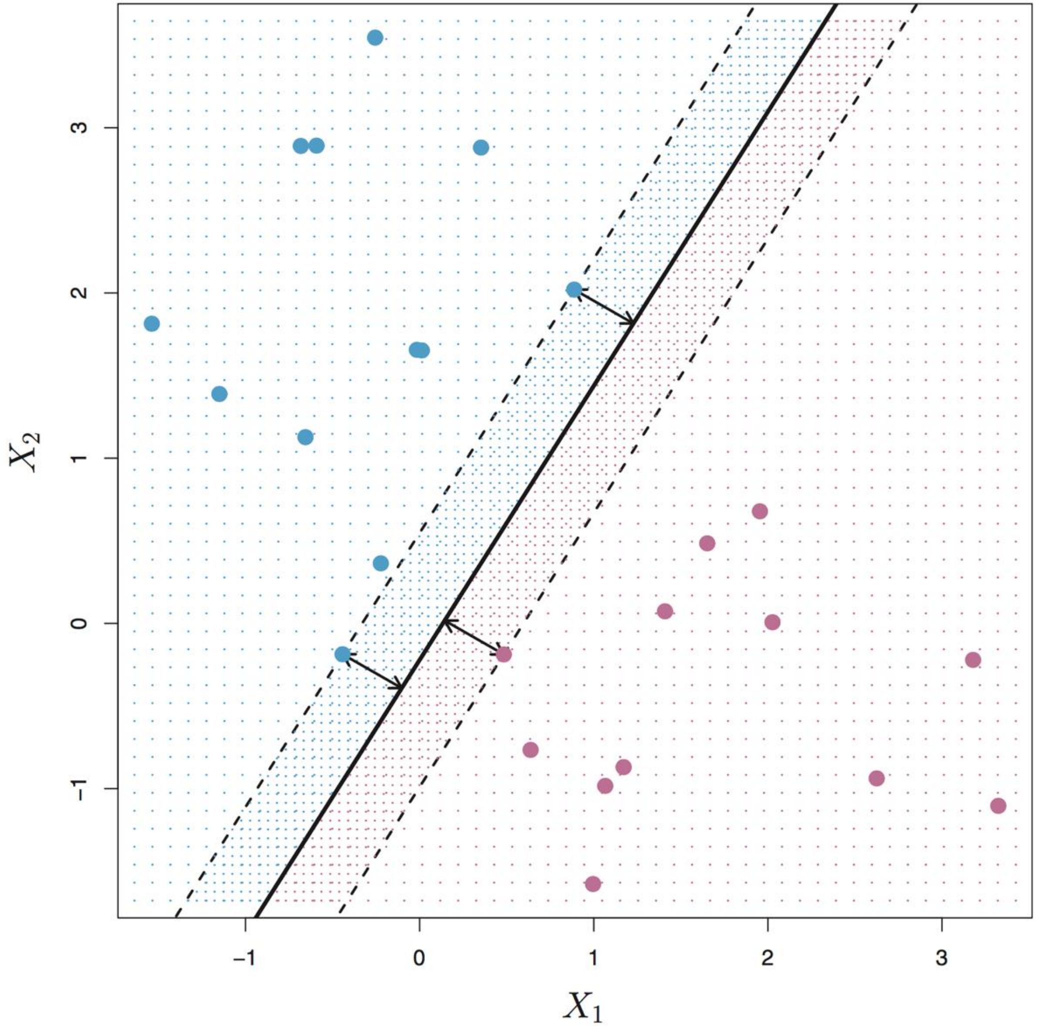


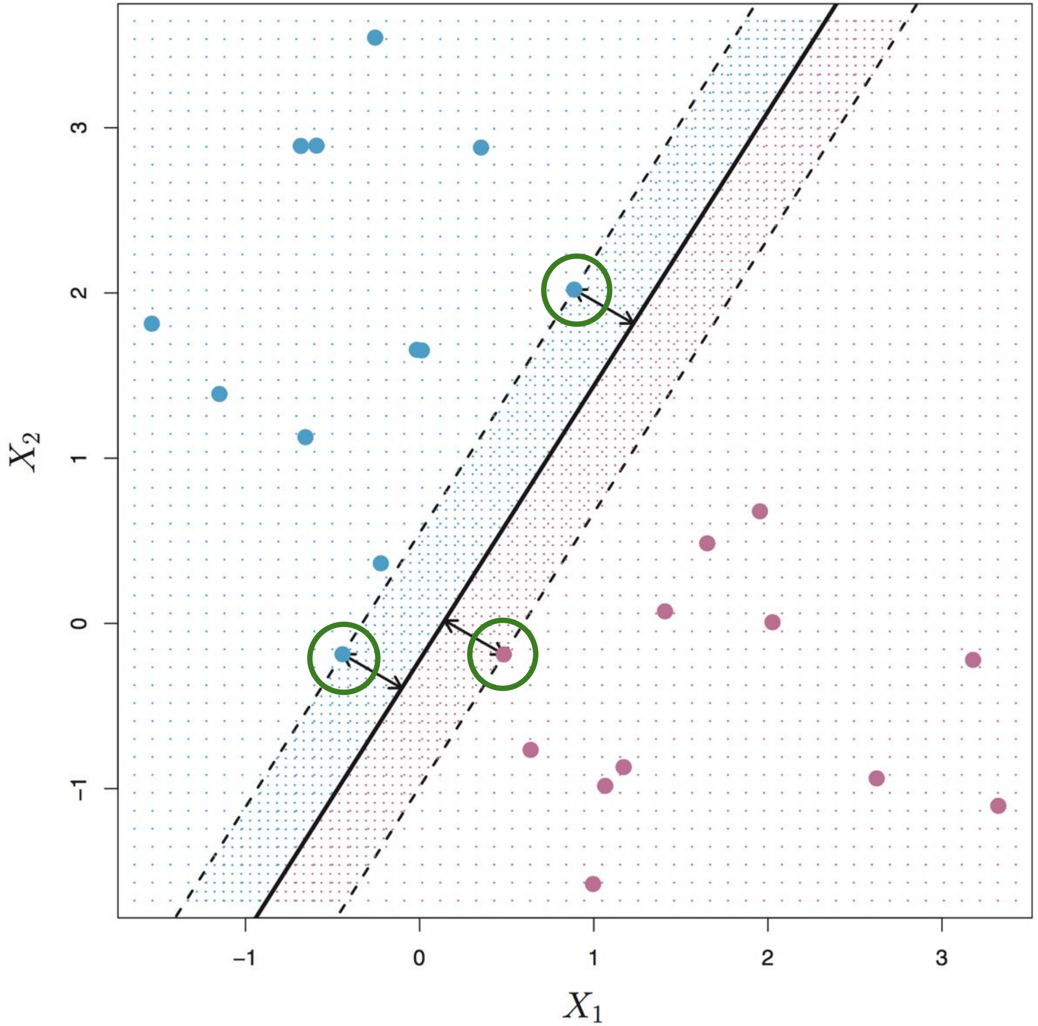
Image: “Towards Data Science”

Support vectors: data points on the margin



Modified from ISL: Figure 9.3

Support vectors: data points on the margin



Support vectors

Modified from ISL: Figure 9.3

Margins: Function vs Geometric

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\big(\vec{w} \cdot \vec{x} + b\big)$$

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\big(\vec{w} \cdot \vec{x} + b\big)$$

Functional Margin:

$$\hat{\gamma}_i = y_i(\vec{w} \cdot \vec{x}_i + b)$$

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\big(\vec{w} \cdot \vec{x} + b\big)$$

Functional Margin:

$$\hat{\gamma}_i = y_i(\vec{w} \cdot \vec{x}_i + b)$$

Geometric Margin:
(distance between example and hyperplane)

$$\gamma_i = y_i \left(\frac{\vec{w}}{||\vec{w}||} \cdot \vec{x}_i + \frac{b}{||\vec{w}||} \right)$$

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\left(\vec{w} \cdot \vec{x} + b\right)$$

Functional Margin:

$$\hat{\gamma}_i = y_i(\vec{w} \cdot \vec{x}_i + b)$$

Geometric Margin:
(distance between example and hyperplane)

$$\gamma_i = y_i \left(\frac{\vec{w}}{||\vec{w}||} \cdot \vec{x}_i + \frac{b}{||\vec{w}||} \right)$$

Note:

$$\boxed{\gamma_i = \frac{\hat{\gamma}_i}{||\vec{w}||}}$$

Optimization Problem: try 1

Goal: maximize the minimum distance between example and hyperplane

$$\gamma = \min_{i=1,\cdots,n} \gamma_i$$

Optimization Problem: try 1

Goal: maximize the minimum distance between example and hyperplane

$$\gamma = \min_{i=1,\cdots,n} \gamma_i$$

Formulation: optimize a function with respect to a constraint

$$\begin{aligned} \max_{\gamma,\vec{w},b} \quad & \gamma \\ \text{s.t.} \quad & y_i(\vec{w}\cdot\vec{x}_i + b) \geq \gamma, \quad i = 1,\cdots,n \\ \text{and} \quad & \|\vec{w}\| = 1 \end{aligned}$$

(force functional and geometric margin to be equal)

Optimization Problem: try 2

Idea: substitute functional margin divided by magnitude of weight vector

$$\max_{\hat{\gamma},\vec{w},b} \quad \frac{\hat{\gamma}}{||\vec{w}||}$$
$$\text{s.t.} \quad y_i(\vec{w}\cdot\vec{x}_i + b) \geq \hat{\gamma}, \quad i = 1,\cdots,n$$

(gets rid of non-convex constraint)

Optimization Problem: try 3

Idea: put arbitrary constraint on functional margin

$$\boxed{\hat{\gamma} = 1}$$

$$\min_{\vec{w},b} \quad \frac{1}{2}\|\vec{w}\|^2$$

$$\text{s.t.} \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad i = 1, \cdots, n$$

Optimization Problem: try 3

Idea: put arbitrary constraint on functional margin

$$\boxed{\hat{\gamma} = 1}$$

$$\min_{\vec{w},b} \quad \frac{1}{2}||\vec{w}||^2$$
$$\text{s.t.} \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad i = 1, \cdots, n$$

$$\boxed{\begin{aligned} &\min_{\vec{w},b} \quad \frac{1}{2}||\vec{w}||^2 \\ &\text{s.t.} \quad - y_i(\vec{w} \cdot \vec{x}_i + b) + 1 \leq 0, \quad i = 1, \cdots, n \end{aligned}}$$