Introduction to Machine Learning (CSCI-UA 473): Fall 2021

Lecture 6: Support Vector Machines - 1

Sumit Chopra
Courant Institute of Mathematical Sciences
Department of Radiology - Grossman School of Medicine
NYU

Slides derived from materials from Benjamin Peherstorfer, Kyunghyun Cho, Andrew Gordon Wilson

Lecture Outline

Multi-Class Classification

Support Vector Machines (SVMs)

Notion of Margins and Maximum Margin

Primal Formulation of SVMs

Multi-Class Classification

Logistic Regression: Binary Classification

$$P(Y|X) = \prod_{i=1}^{N} P(y^i|x^i) = \prod_{i=1}^{N} \sigma(y^i w^T x^i) \qquad y^i \in \{0,1\}$$

Likelihood is defined by

$$P(y^i|x^i) = \sigma(w^T x^i)^{y^i} \cdot (1 - \sigma(w^T x^i))^{(1-y^i)}$$

The loss function that is minimized is the negative log likelihood loss

$$\mathscr{L}_w = -\log\left[\prod_{i=1}^{N} P(y^i|x^i)\right]$$

$$= -\sum_{i=1}^{N}\left[y^i \cdot \log\sigma(w^T x^i) + (1 - y^i) \cdot \log(1 - \sigma(w^T x^i))\right]$$

Multi-Class Classification

$$P(Y|X) = \prod_{i=1}^{N} P(y^i|x^i) \qquad y^i \in \{1,2,\ldots,K\}$$

Softmax function

Likelihood for a single sample is defined by

$\mathbf{w} \in \Re^{P\times K}$

$$P(y^i = k\,|\,x^i) = \frac{e^{a_k}}{\sum_{j=1}^{K} e^{a_j}} \qquad \text{Where} \qquad a_k = \mathbf{w}_k^T \cdot x^i$$

Likelihood of the data set is defined as

$$P(Y|X) = \prod_{i=1}^{N} P(y^i|x^i) = \prod_{i=1}^{N}\prod_{j=1}^{K} P(y^i = j\,|\,x^i)^{y_j^i}$$

$$\mathscr{L}_w = -\log\left[\prod_{i=1}^{N} P(y^i|x^i)\right] = -\sum_{i=1}^{N}\sum_{j=1}^{K} y_j^i \cdot \log\frac{e^{a_k}}{\sum_{j=1}^{K} e^{a_j}}$$

Support Vector Machines

The Perceptron Model

$$h(x) = sign\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

$sign(s) = +1$ if $s > 0$ and $sign(s) = -1$ if $s < 0$

The Decision Algorithm

$h(x) = +1$ ==> approve credit

$h(x) = -1$ ==> reject credit

In other words

Approve credit ==> $\sum_{i=1}^{d} w_i x_i > -b$

Reject credit ==> $\sum_{i=1}^{d} w_i x_i < -b$

$$\sum_{i=1}^{d} w_i x_i > -b$$

Thus the bias b determines the *threshold*

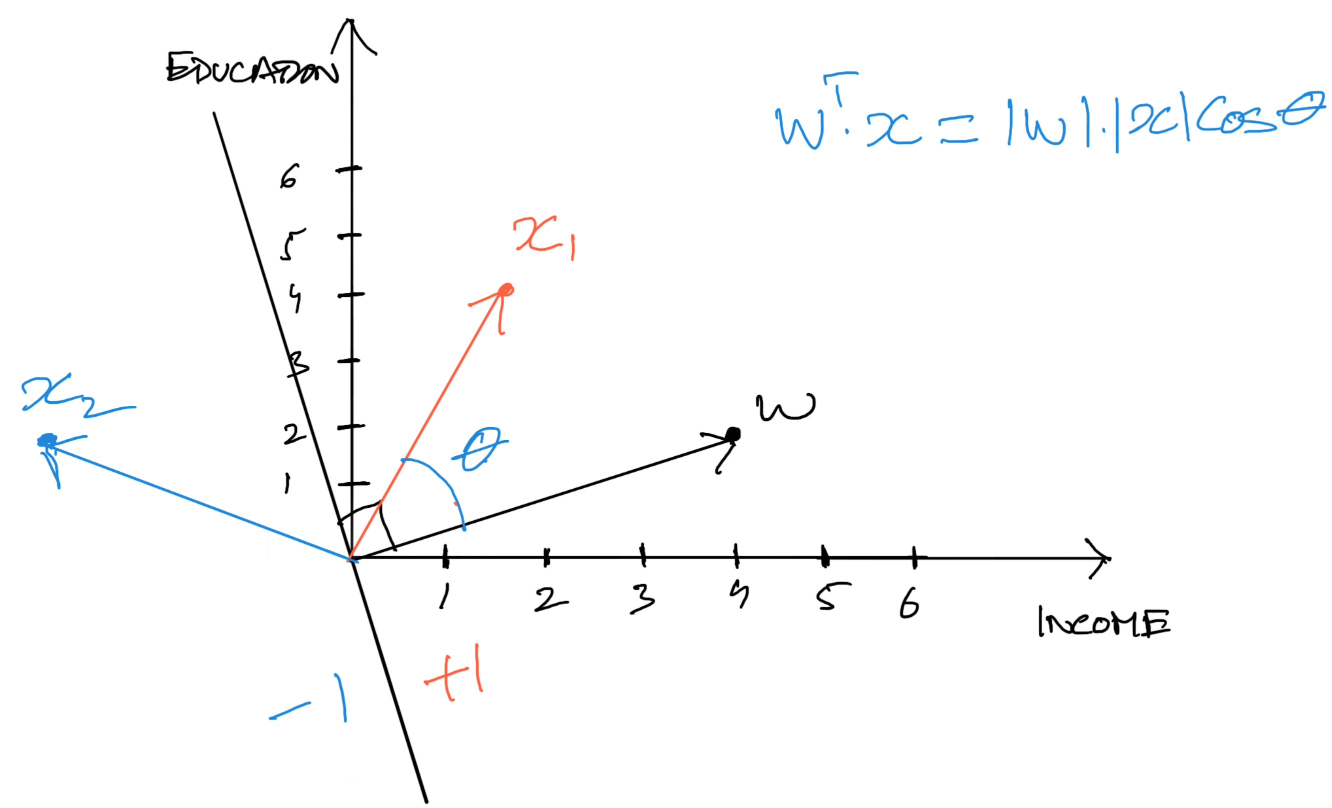
The Perceptron Model

$$h(x) = sign\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

In 2 dimensions

$$x = \begin{pmatrix} BIAS \\ income \\ education \end{pmatrix} = \begin{pmatrix} 1 \\ \cdot \\ \cdot \end{pmatrix}, \qquad w = \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix}$$

If we ignore the bias we have



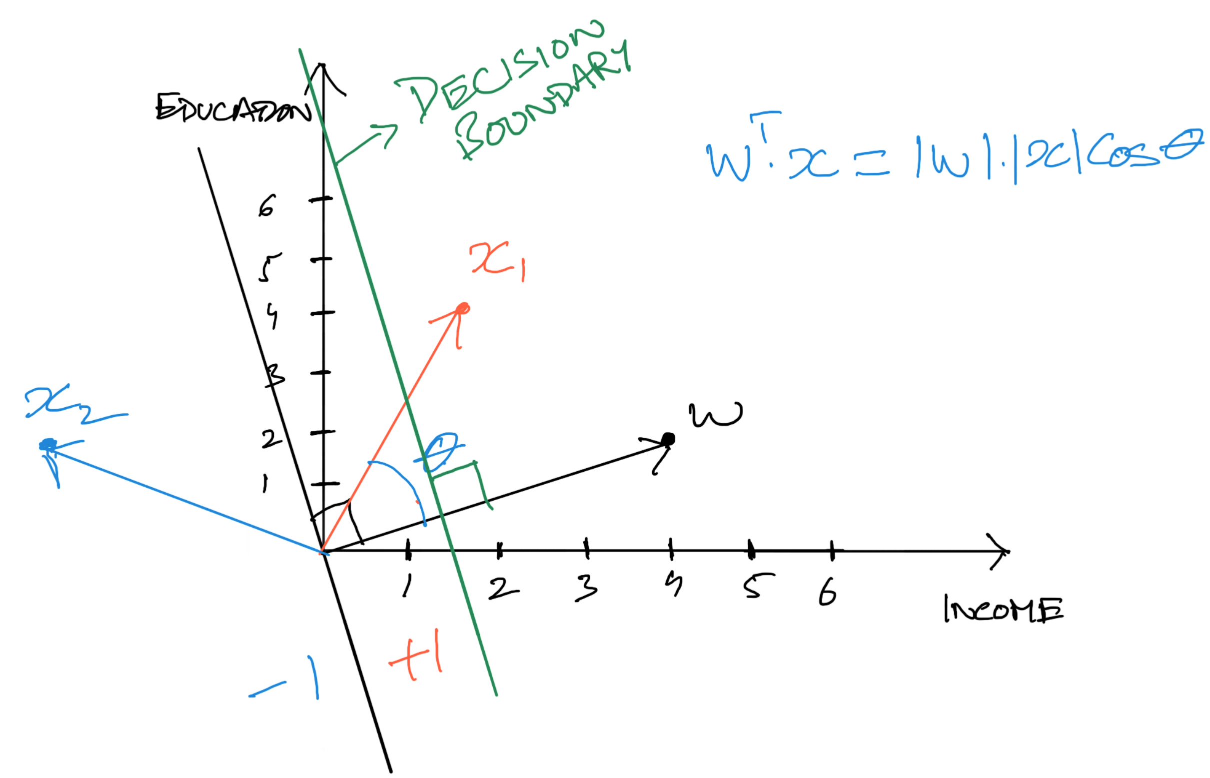
The Perceptron Model

$$h(x) = sign\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

In 2 dimensions

$$x = \begin{pmatrix} BIAS \\ income \\ education \end{pmatrix} = \begin{pmatrix} 1 \\ \cdot \\ \cdot \end{pmatrix}, \qquad w = \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix}$$

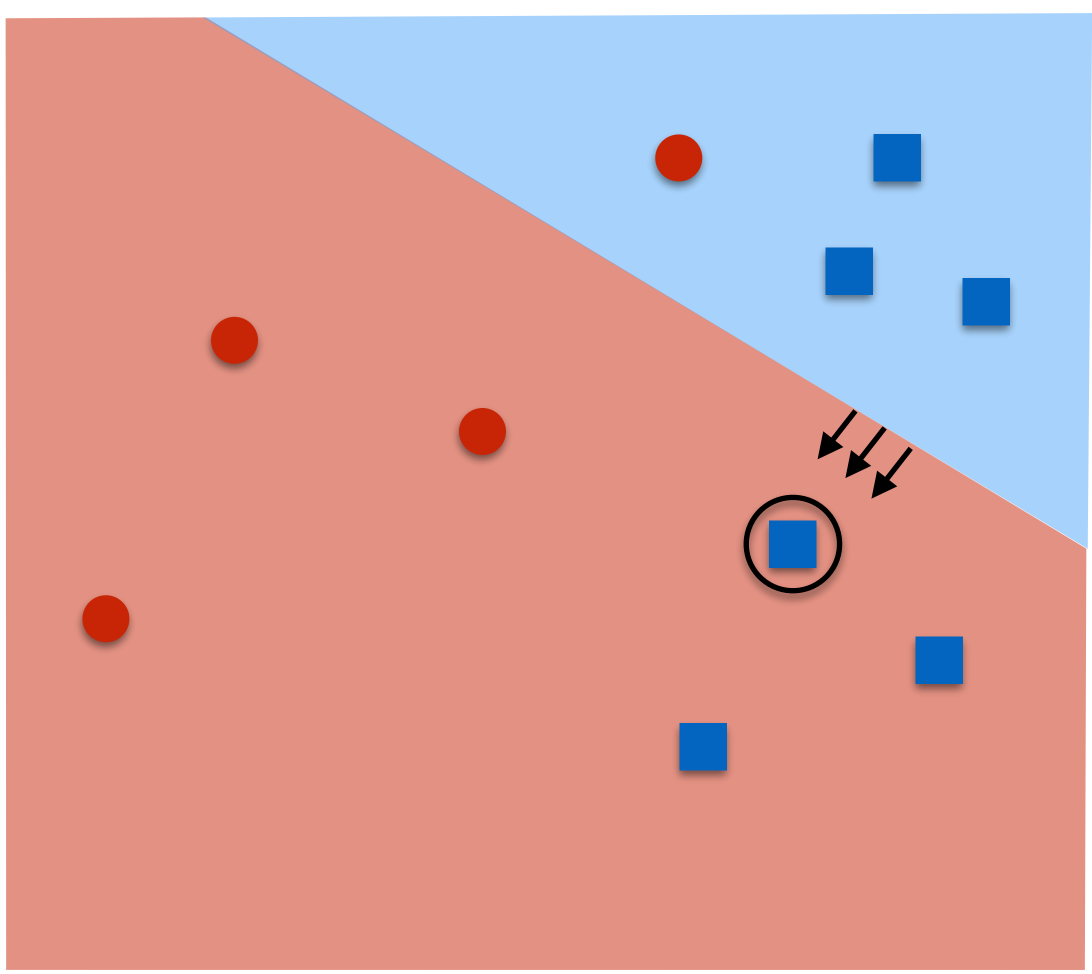
With the bias we have



Perceptron Learning Algorithm (PLA)

$$h(x) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + b\right) \longrightarrow h(x) = sign\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}\right)$$

$$\mathbf{w} = [b, w_1, w_2, \ldots, w_d] \;\; and \;\; \mathbf{x} = [1, x_1, x_2, \ldots, x_d]$$



Random Weight Vector

Let $w(t)$ be the weight vector at iteration t

for $t = 0$ until no example is misclassified

1. Pick a random sample $(x(t), y(t))$ from the set D which is misclassified
2. Update the weight vector with the following update rule

$$w(t+1) \leftarrow w(t) + y(t)x(t)$$

Note that since the example is misclassified

$$y(t) \neq sign(w(t)^T x(t))$$

One can prove that so long as the data is linearly separable the above algorithm will find a separating hyperplane

Linear Separators

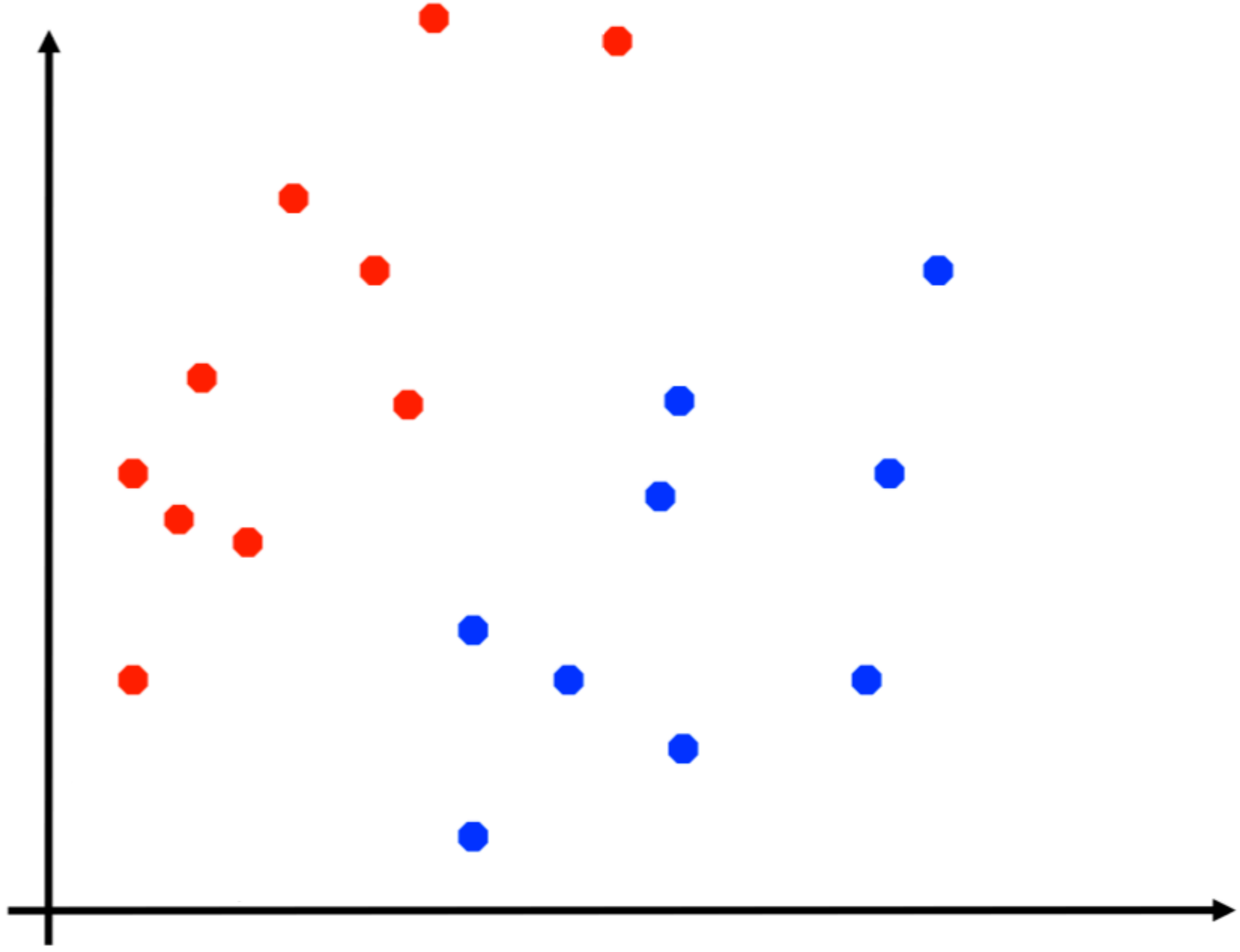


Figure credits David Sontag

Linear Separators

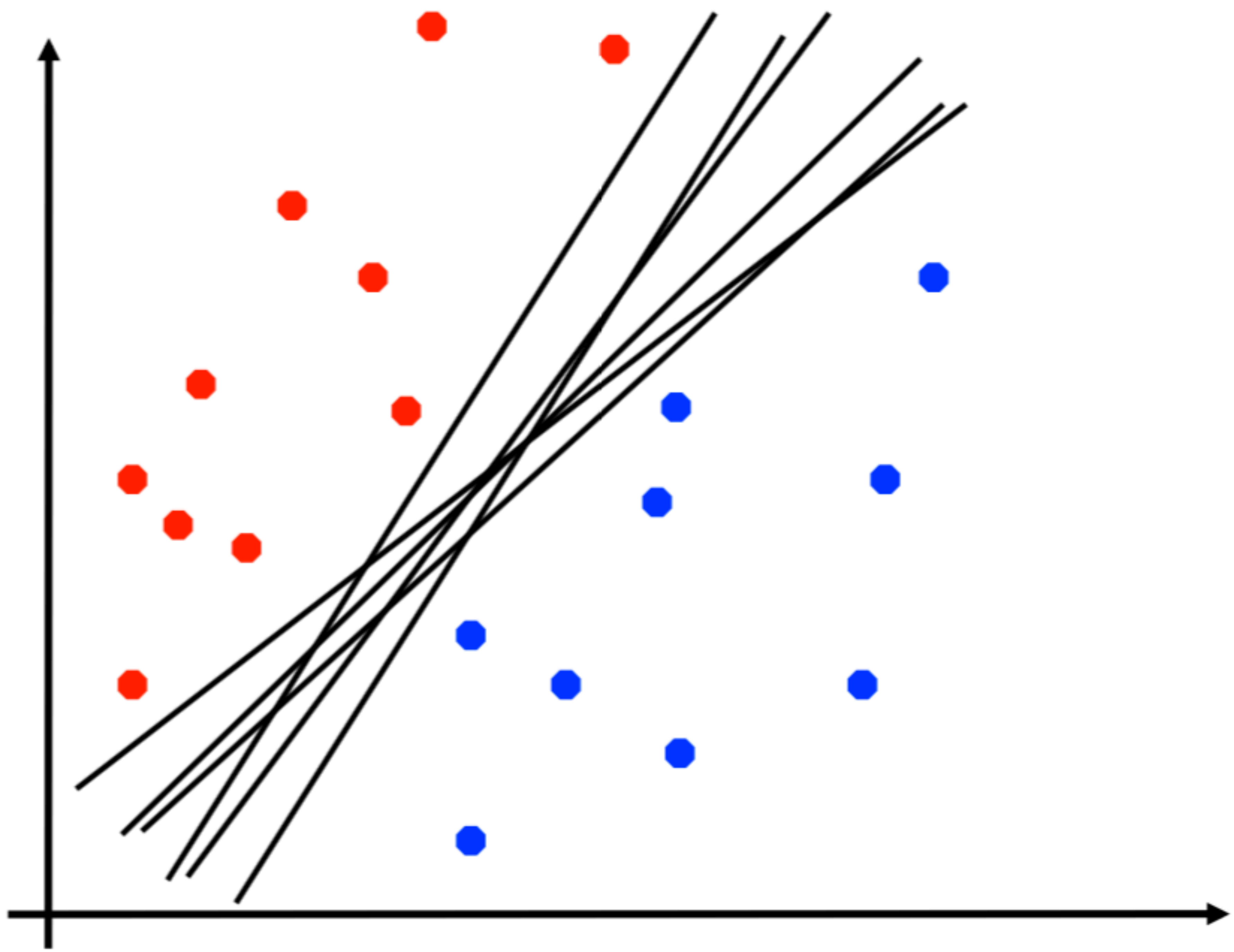
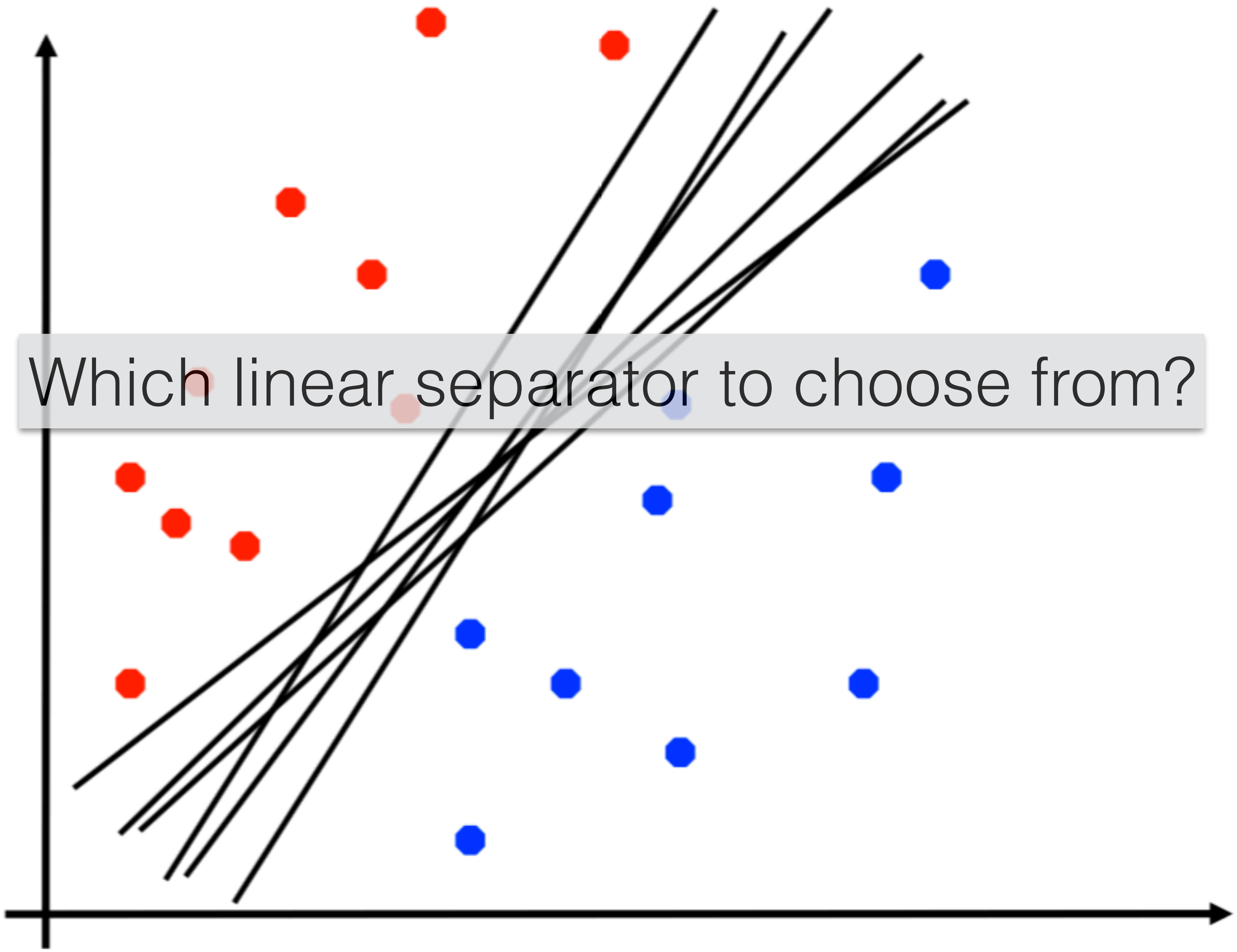


Figure credits David Sontag

Linear Separators



Which linear separator to choose from?

Figure credits David Sontag

Linear Separators

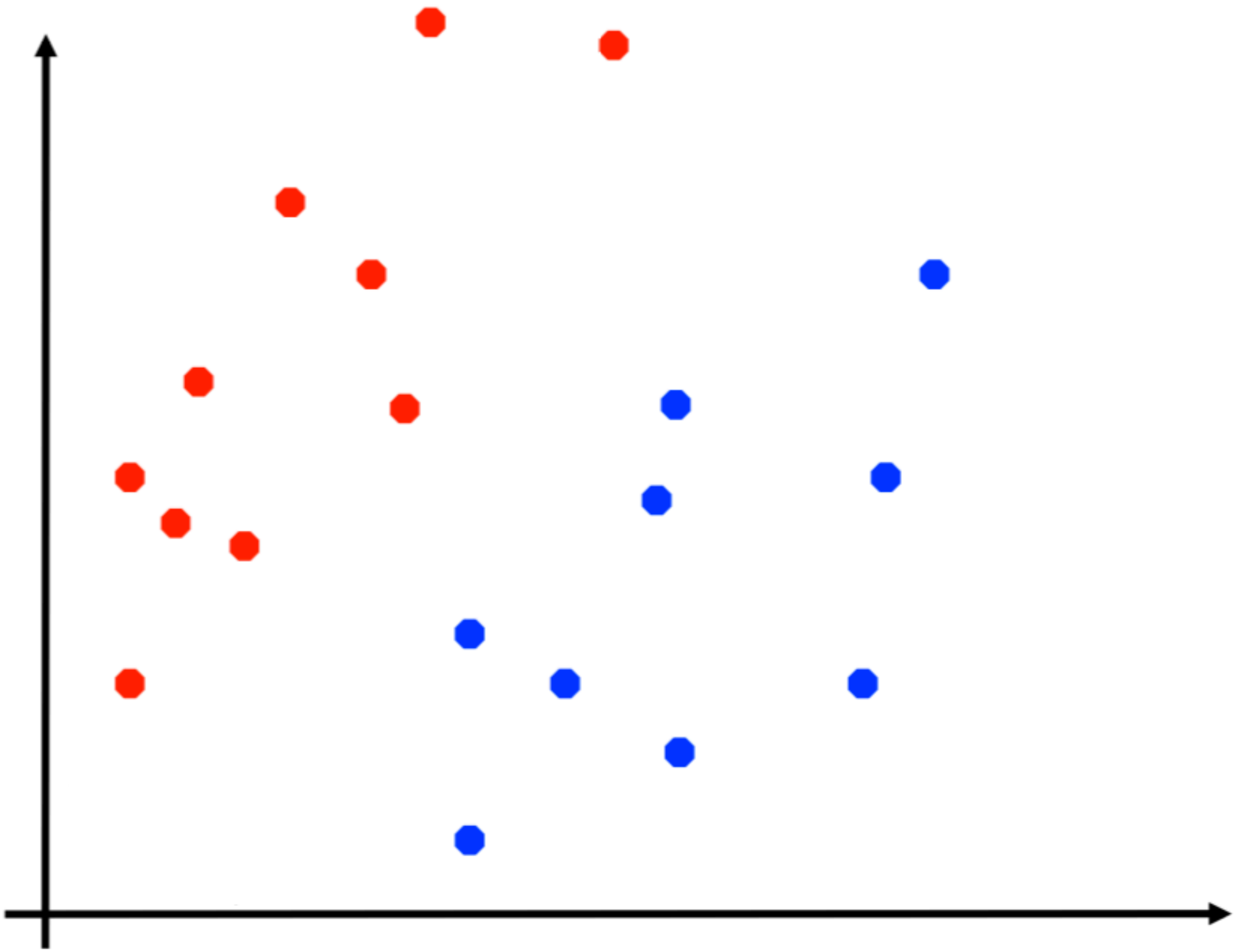


Figure credits David Sontag

Linear Separators

Intuition says that we should pick the one that is farthest from any point belonging to the two classes

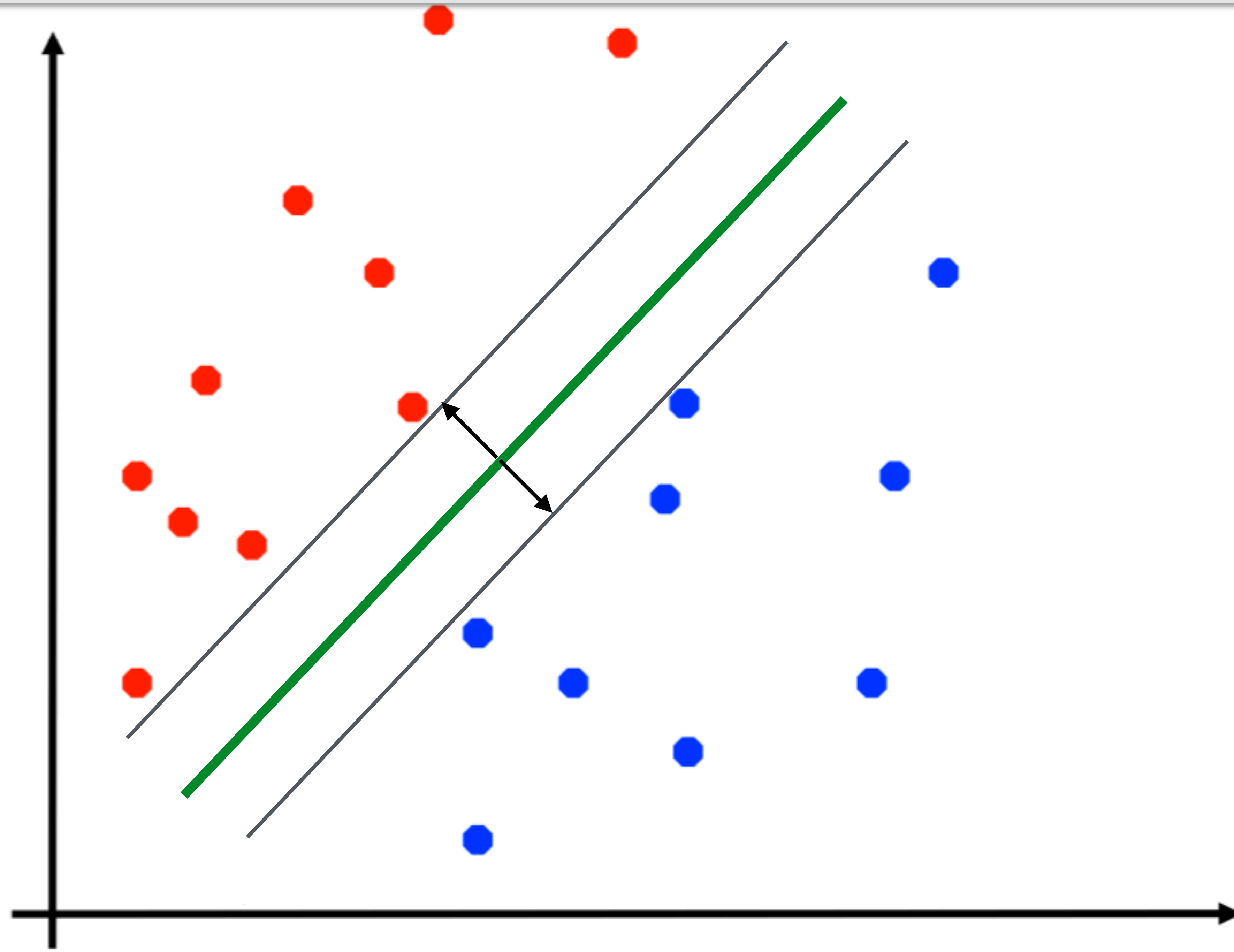
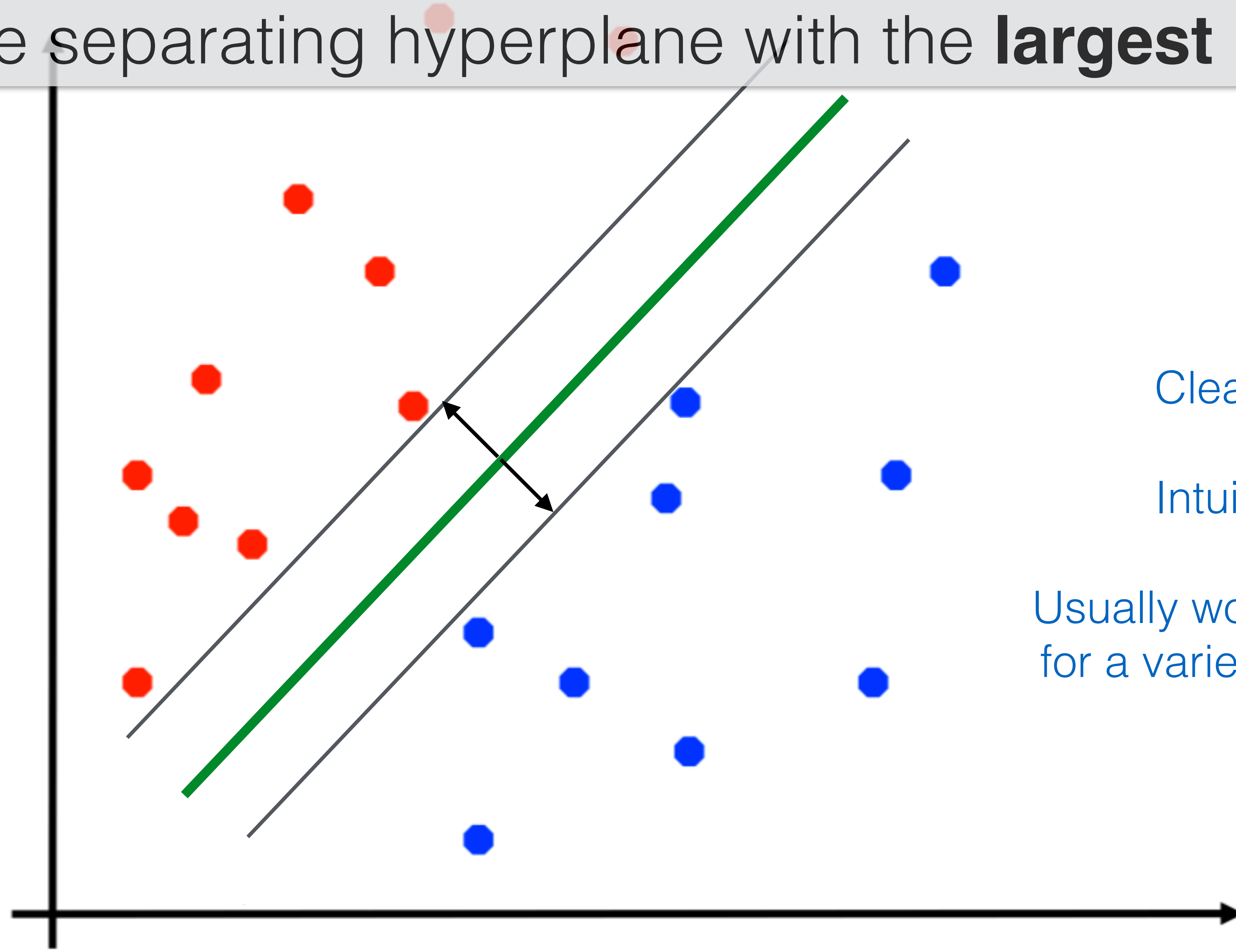


Figure credits David Sontag

Linear Separators

Support Vector Machines does exactly that!

Find the separating hyperplane with the **largest margin**



Clean theory

Intuitive idea

Usually works in practice
for a variety of problems

Figure credits David Sontag

Support Vector Machines

Based on three key ideas

Seeks large margin separator to improve generalization

Uses optimization theory to find efficient and optimal solutions (with few errors)

Uses kernel trick to make computations efficient specially in cases where the feature vectors is huge

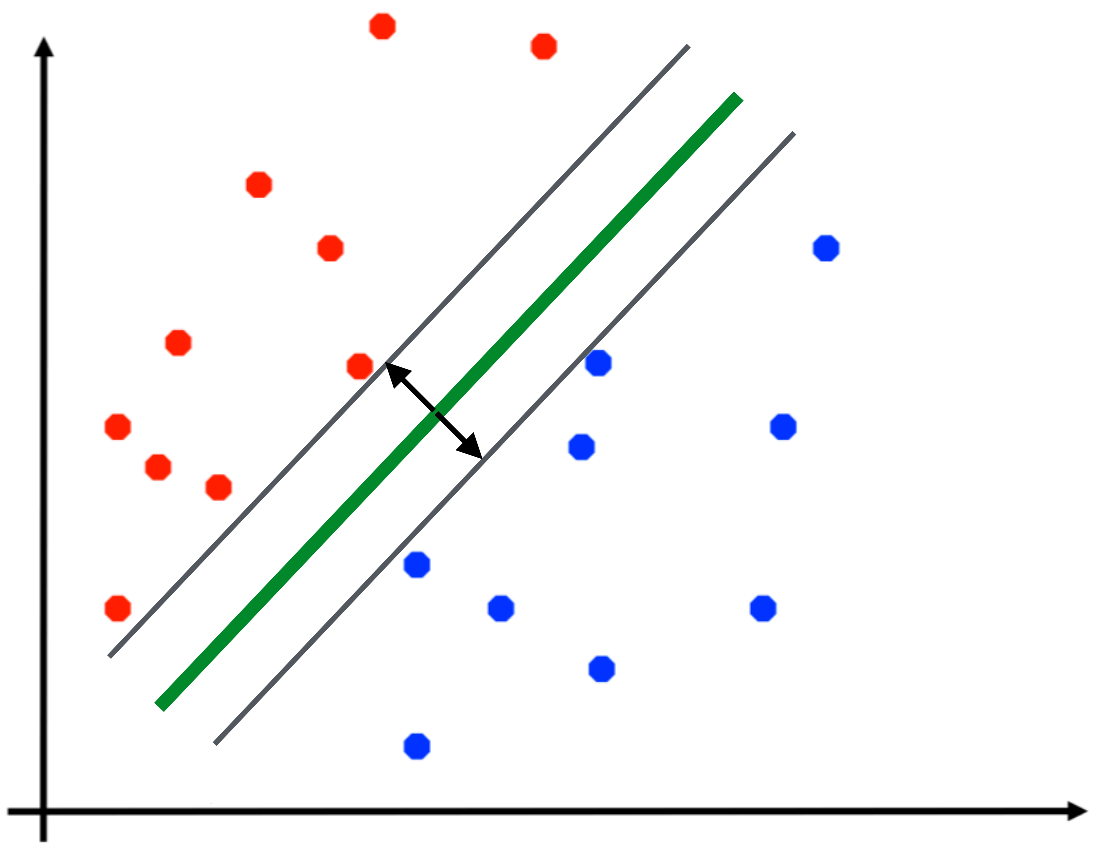


Figure credits David Sontag

Some Notations

Inputs: x

Outputs (class labels): $y \in \{-1, +1\}$

Parameters: w

Bias (Intercept): b

$$h_\theta(x) = g(w^T x + b) \qquad \theta = [w, b]$$

$$g(z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$

Functional Margin

Define functional margin of a single training sample (x^i, y^i) w.r.t. (w, b) as

$$\hat{\gamma}^i = y^i(w^T x^i + b)$$

If $y^i = 1$ then for $\hat{\gamma}^i$ to be large we need $w^T x^i + b \gg 0$

If $y^= - 1$ the for $\hat{\gamma}^i$ to be large we need $w^T x^i + b \ll 0$

Note that when the above is true, the prediction is also correct: $y^i(w^T x^i + b) > 0$

Large functional margin implies two things

Predictions are correct

You are confident about your predictions

Functional Margin

Define functional margin of a single training sample (x^i, y^i) w.r.t. (w, b) as

$$\hat{\gamma}^i = y^i(w^T x^i + b)$$

If $y^i = 1$ then for $\hat{\gamma}^i$ to be large we need $w^T x^i + b \gg 0$

If $y^= -1$ the for $\hat{\gamma}^i$ to be large we need $w^T x^i + b \ll 0$

Note that when the above is true, the prediction is also correct: $y^i(w^T x^i + b) > 0$

Large functional margin implies two things

Predictions are correct

You are confident about your predictions

$$h_\theta(x) = g(w^T x + b)$$

$$g(z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$

Scaling

$$h_\theta(x) = g(w^T x + b) = g(2w^T x + 2b)$$

$h_\theta()$ is invariant under scaling

$\hat{\gamma}^i$ is not invariant under scaling

This allows us to normalize the parameters to ensure their norm is 1: $||w|| = 1$
without affecting the solution

Functional Margin

Define functional margin of a single training sample (x^i, y^i) w.r.t. (w, b) as

$$\hat{\gamma}^i = y^i(w^T x^i + b)$$

If $y^i = 1$ then for $\hat{\gamma}^i$ to be large we need $w^T x^i + b \gg 0$

If $y^{=} - 1$ the for $\hat{\gamma}^i$ to be large we need $w^T x^i + b \ll 0$

Note that when the above is true, the prediction is also correct: $y^i(w^T x^i + b) > 0$

Think of it like a testing function telling you whether a particular example is properly classified.

Large functional margin implies two things

Predictions are correct

You are confident about your predictions

$h_\theta(x) = g(w^T x + b)$

$$g(z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$

Scaling

$$h_\theta(x) = g(w^T x + b) = g(2w^T x + 2b)$$

$h_\theta()$ is invariant under scaling

$\hat{\gamma}^i$ is not invariant under scaling

This allows us to normalize the parameters to ensure their norm is 1: $||w|| = 1$
without affecting the solution

Geometric Margin

For a point (x^i, y^i) Geometric Margin is defined as the distance of (x^i, y^i) from the hyperplane

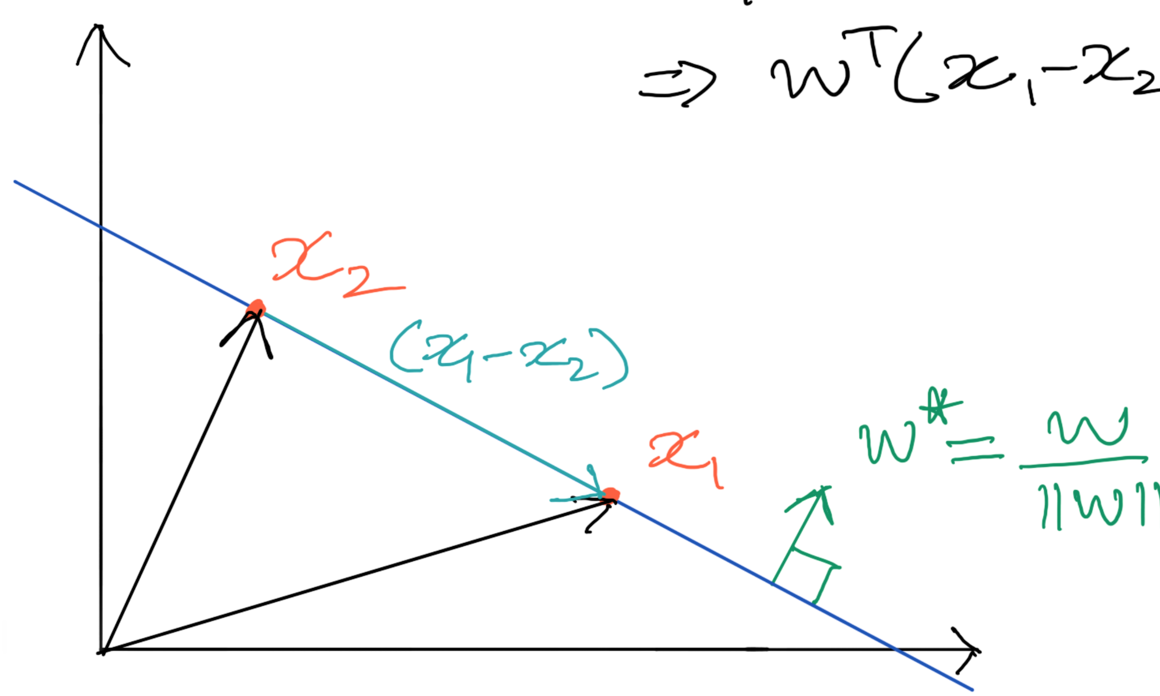
Property 1

For any two points x_1 and x_2 lying on the hyper-plane we have $w^t(x_1 - x_2) = 0$

Hence $w^* = \dfrac{w}{||w||}$ is a unit vector perpendicular to the hyper-plane

$w^T x_1 + b = 0 = w^T x_2 + b$

$\Rightarrow w^T(x_1 - x_2) = 0$



Geometric Margin

For a point (x^i, y^i) Geometric Margin is defined as the distance of (x^i, y^i) from the hyperplane

Property 2

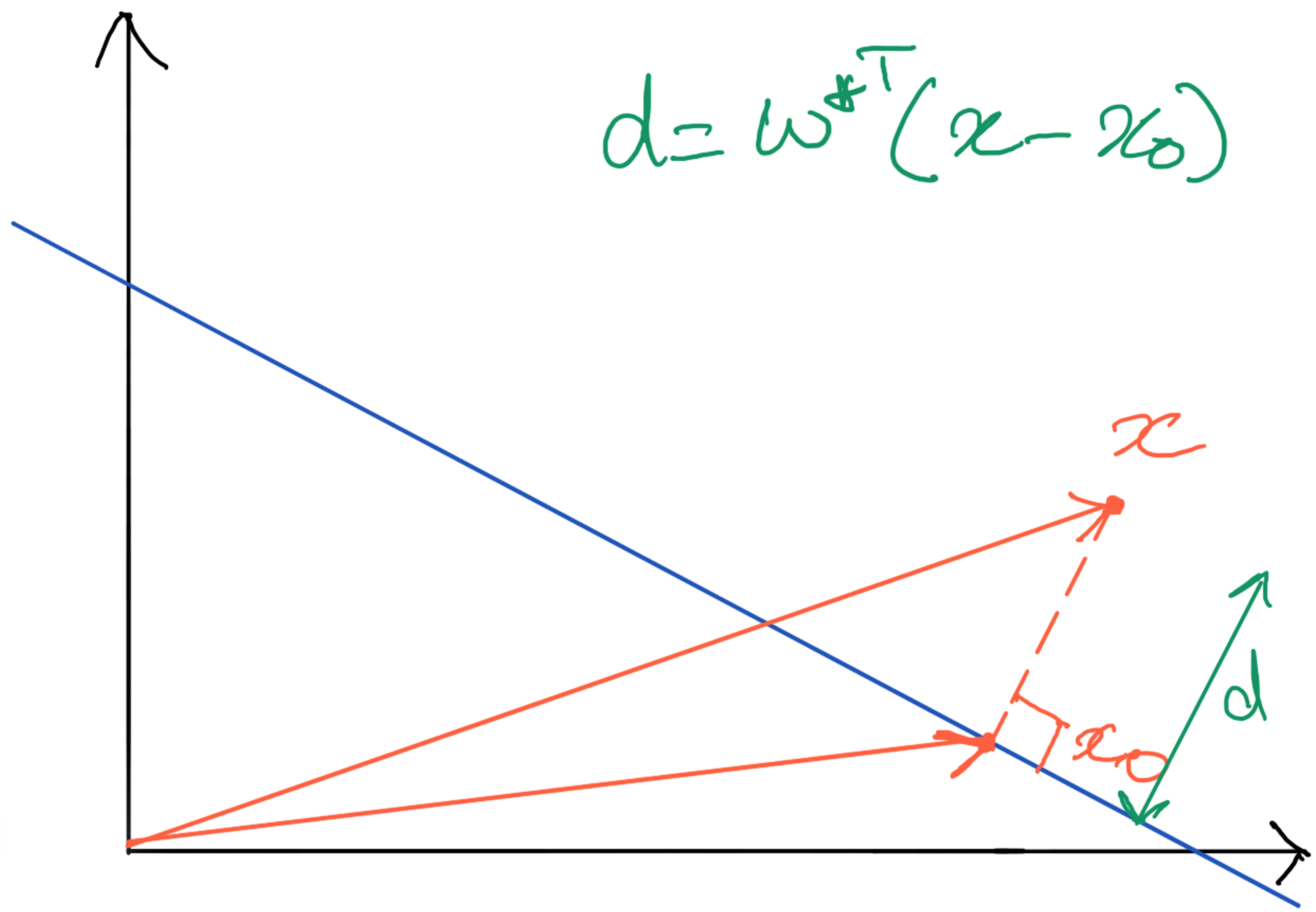
For any points x_0 on the hyper-plane we have $w^t x_0 = -b$

Geometric Margin

For a point (x^i, y^i) Geometric Margin is defined as the distance of (x^i, y^i) from the hyperplane

Property 3

Signed distance between any point x to the hyper-plane is given by



$$w^{*T}(x - x_0) = \frac{1}{||w||}(w^T x - w^T x_0)$$

$$= \frac{1}{||w||}(w^T x + b)$$

Geometric Margin

For a point (x^i, y^i) Geometric Margin is defined as the distance of (x^i, y^i) from the hyperplane

Thus for a particular class the Geometric Margin is given by

$$\gamma^i = \frac{1}{||w||}(w^T x^i + b)$$

We represent Geometric Margin for both the classes by multiplying by class label y^i

$$\gamma^i = \frac{y^i(w^T x^i + b)}{||w||} = y^i \left(\frac{w^T x^i}{||w||} + \frac{b}{||w||} \right)$$

Thus Geometric Margin is nothing but a scaled version of the Functional Margin

$$\gamma^i = \frac{\hat{\gamma}^i}{||w||}$$

Geometric Margin

For a point (x^i, y^i) Geometric Margin is defined as the distance of (x^i, y^i) from the hyperplane

Geometric Margin is invariant to scaling of parameters

$$\gamma^i = y^i \left(\frac{w^T x^i}{||w||} + \frac{b}{||w||} \right)$$

Replacing w and b by kw and kb will not change the value of γ^i

Geometric Margin with respect to the data set $\mathscr{D} = \{(x^1, y^1), \ldots, (x^n, y^n)\}$ is defined by

$$\gamma^* = \min_{i=1,\ldots,n} \gamma^i$$

Maximum Margin Classifiers

Given the training set $\mathscr{D} = \{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$ find the hyper-plane that maximizes the Geometric Margin for the data set $\mathscr{D}$

Assumption (strong!): the training set $\mathscr{D}$ is linearly separable

$$\rho = \max_{w,b\,:\,y^i(w^T x^i + b) \geq 0} \gamma^*$$

Find the parameters of the hyper-plane such that it maximizes the Geometric Margin for the data set

$$= \max_{w,b\,:\,y^i(w^T x^i + b) \geq 0} \left[\min_{i=1,\ldots,n} \gamma^i \right]$$

$$= \max_{w,b\,:\,y^i(w^T x^i + b) \geq 0} \left[\min_{i=1,\ldots,n} \frac{|w^T x^i + b|}{||w||} \right]$$

Maximum Margin Classifiers

Given the training set $\mathscr{D} = \{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$ find the hyper-plane that maximizes the Geometric Margin for the data set $\mathscr{D}$

Assumption (strong!): the training set $\mathscr{D}$ is linearly separable

$$\rho = \max_{w,b \,:\, y^i(w^T x^i + b) \geq 0} \left[\min_{i=1,\ldots,n} \frac{|w^T x^i + b|}{||w||} \right]$$

This can be posed as a constrained optimization problem

$$\begin{aligned} \max_{\gamma,w,b} \quad & \gamma \\ \mathbf{s.t.} \quad & y^i(w^T x^i + b) \geq \gamma, \qquad \forall i = 1,\ldots,n \\ & ||w|| = 1 \end{aligned}$$

The constraint $||w|| = 1$ is imposed to ensure that the Functional Margin $\left(\hat{\gamma}^i = y^i(w^T x^i + b)\right)$ is equal to the Geometric Margin

However $||w|| = 1$ induces non-convexities and is hard to optimize. A better constraint is $||w|| \leq 1$

Maximum Margin Classifiers

Note that the Geometric Margin is nothing but a scaled version of Functional Margin

$$\gamma = \frac{\hat{\gamma}}{||w||}$$

Thus the optimization problem

$$\begin{aligned} \max_{\gamma, w, b} \quad & \gamma \\ \textbf{s.t.} \quad & y^i(w^T x^i + b) \geq \gamma, \qquad \forall i = 1, \ldots, n \\ & ||w|| = 1 \end{aligned}$$

Can be written as

$$\begin{aligned} \max_{\hat{\gamma}, w, b} \quad & \frac{\hat{\gamma}}{||w||} \\ \textbf{s.t.} \quad & y^i(w^T x^i + b) \geq \hat{\gamma}, \qquad \forall i = 1, \ldots, n \end{aligned}$$

Got rid of the constraint $||w|| = 1$. However we modified the objective to $\dfrac{\hat{\gamma}}{||w||}$

Maximum Margin Classifiers

Remember that if we scale w, b the value of $\hat{\gamma}$ changes, however this scaling will not affect the final solution

Impose the constraint that $\hat{\gamma} = 1$ (which is a scaling constraint on w, b)

Thus the optimization problem

$$\max_{\hat{\gamma},w,b} \quad \frac{\hat{\gamma}}{||w||}$$

$$\mathbf{s.t.} \quad y^i(w^T x^i + b) \geq \hat{\gamma}, \qquad \forall i = 1,\ldots,n$$

Can be written as

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

$$\mathbf{s.t.} \quad y^i(w^T x^i + b) \geq 1, \qquad \forall i = 1,\ldots,n$$

Note that maximizing $\frac{\hat{\gamma}}{||w||}$ with $\hat{\gamma} = 1$ is the same as minimizing $\frac{1}{2}||w||^2$

Convex quadratic optimization with linear constraints

End of Lecture 06