Support Vector Machines

Machine Learning
Spring 2018

The slides are mainly from Vivek Srikumar

Big picture

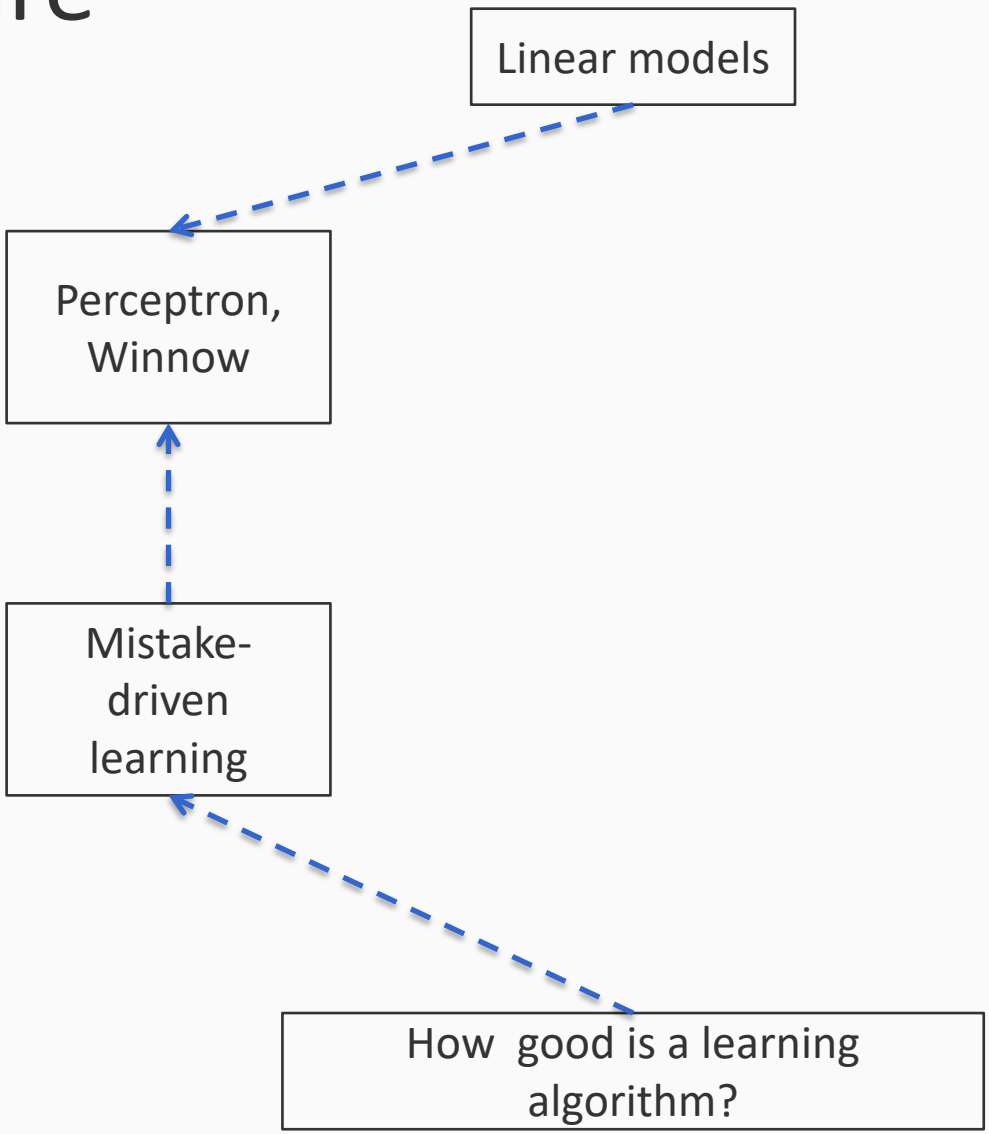
Linear models

Big picture

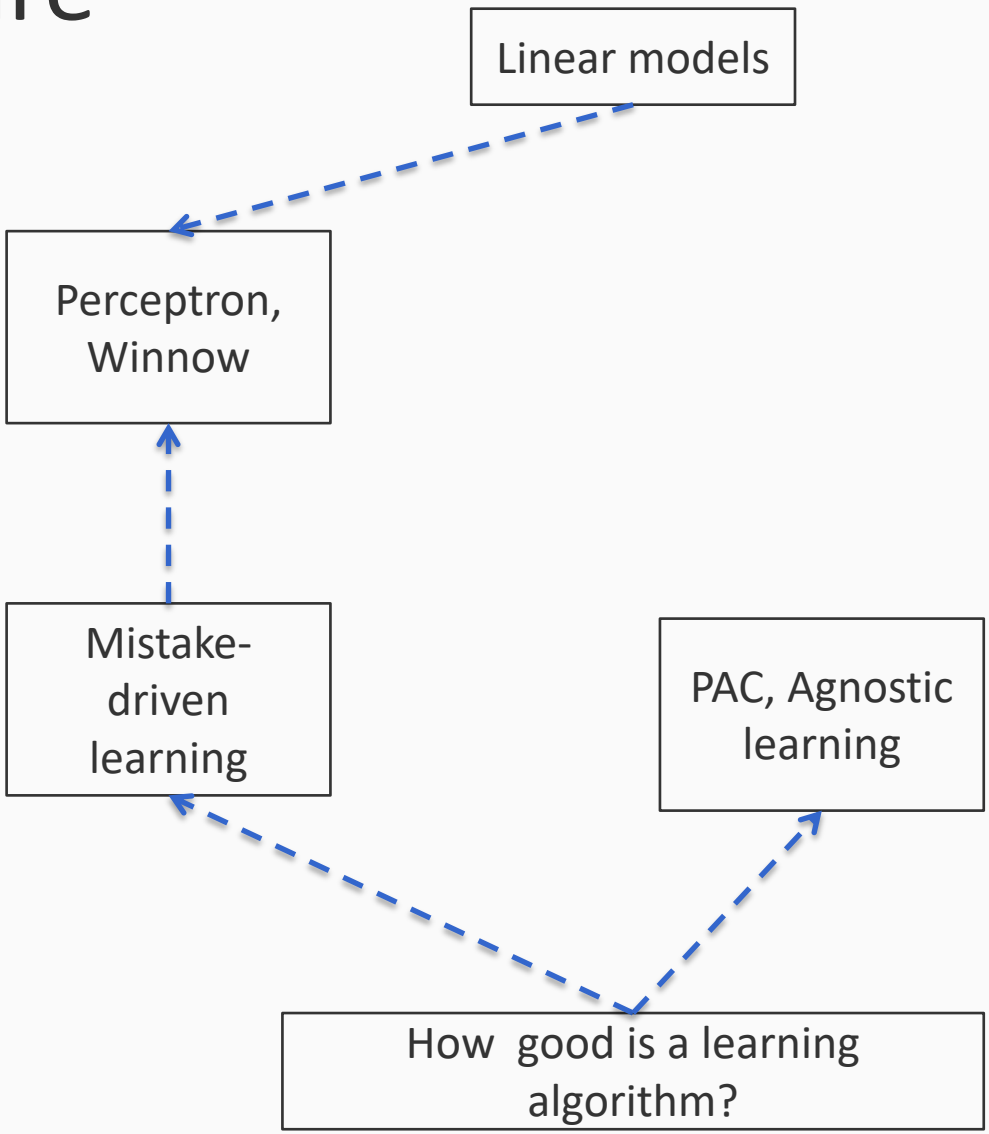
Linear models

How good is a learning algorithm?

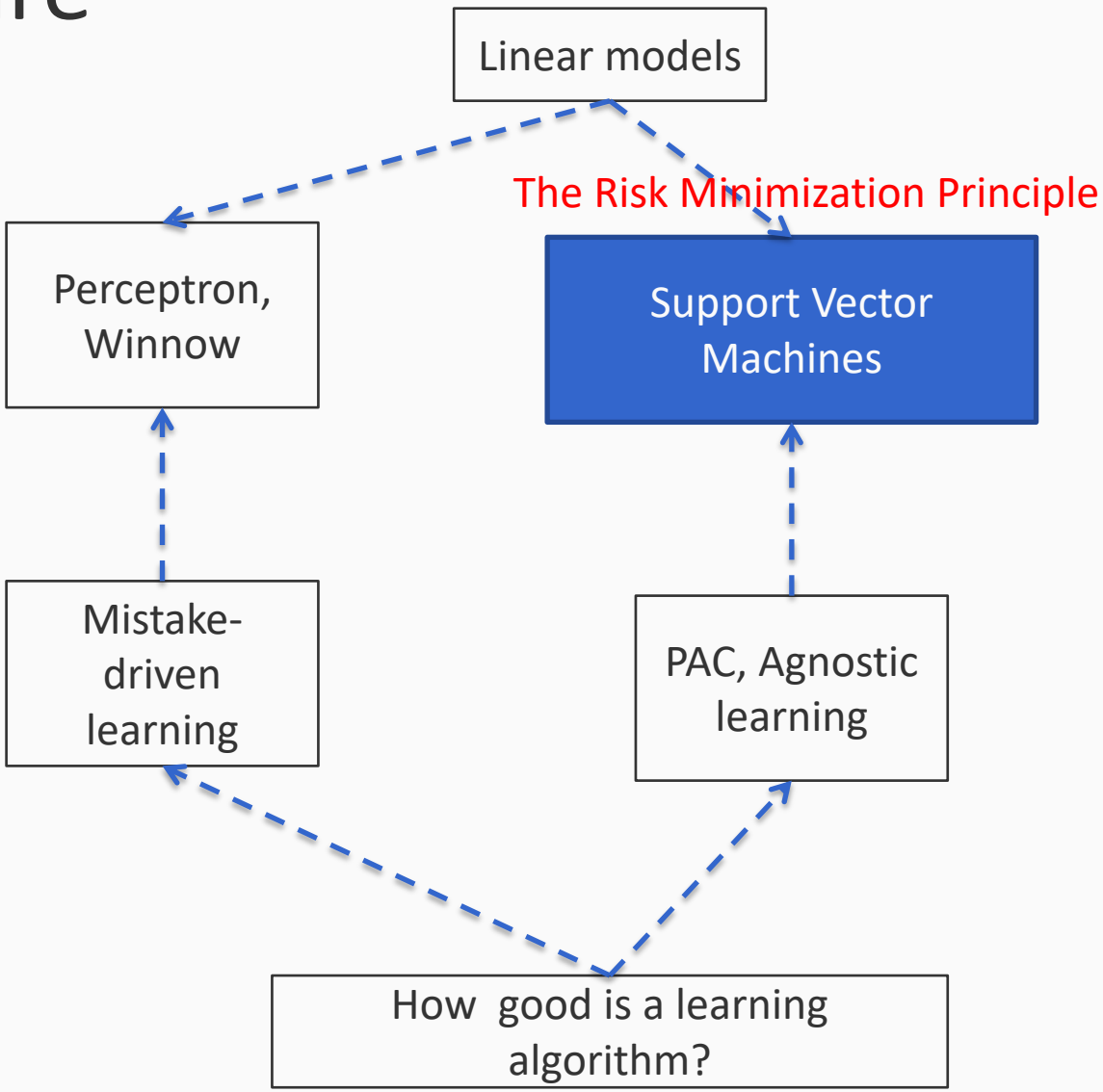
Big picture


Linear models
Perceptron, Winnow
Mistake-driven learning
How good is a learning algorithm?

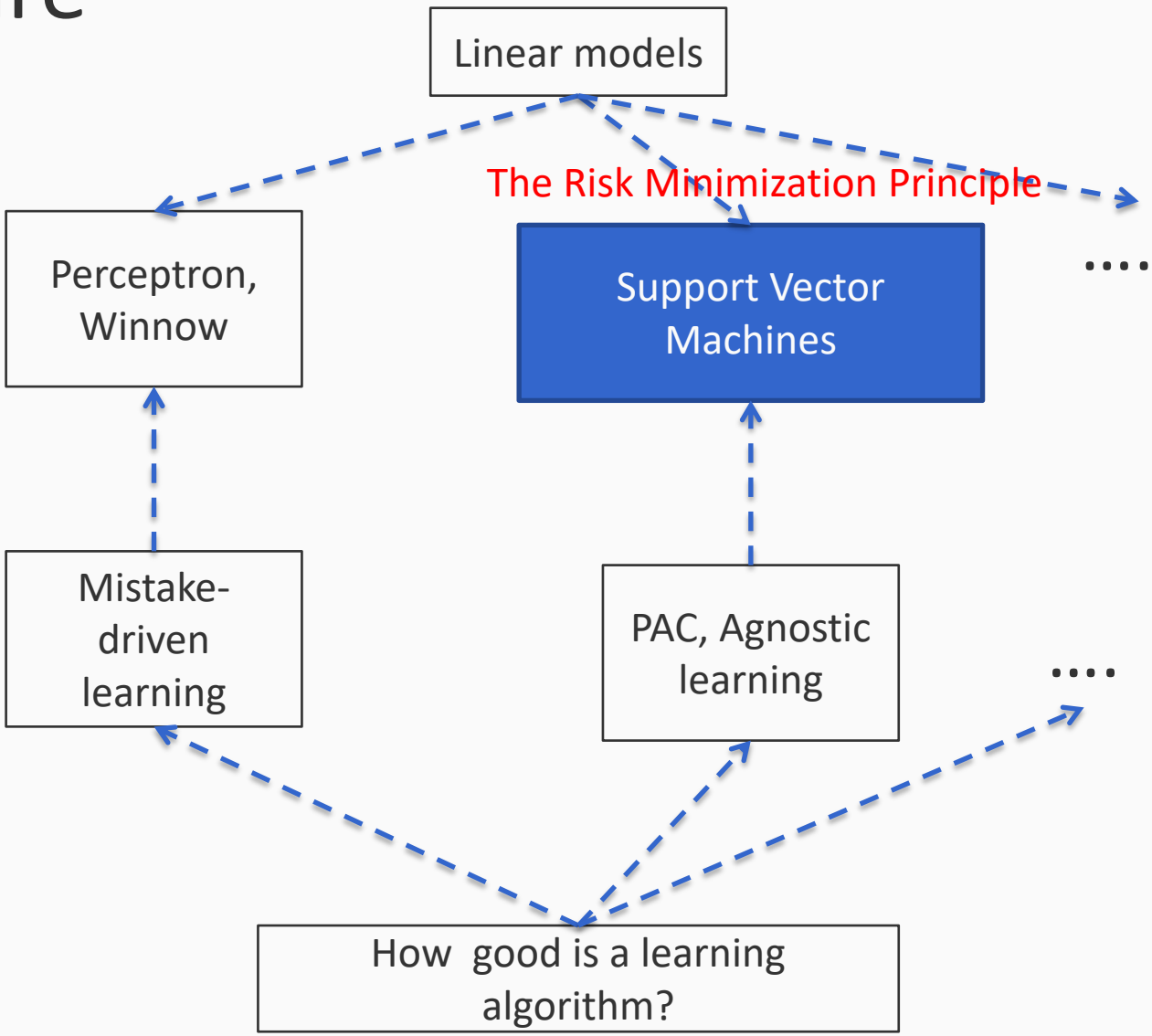
Big picture


Linear models
Perceptron, Winnow
Mistake-driven learning
PAC, Agnostic learning
How good is a learning algorithm?

Big picture


Linear models
The Risk Minimization Principle
Perceptron, Winnow
Support Vector Machines
Mistake-driven learning
PAC, Agnostic learning
How good is a learning algorithm?

Big picture


Linear models
The Risk Minimization Principle
Perceptron, Winnow
Support Vector Machines
....
Mistake-driven learning
PAC, Agnostic learning
....
How good is a learning algorithm?

This lecture: Support vector machines

- Training by maximizing margin
- The SVM objective
- Solving the SVM optimization problem
- Support vectors, duals and kernels

This lecture: Support vector machines

- Training by maximizing margin
- The SVM objective
- Solving the SVM optimization problem
- Support vectors, duals and kernels

VC dimensions and linear classifiers

What we know so far

1. If we have m examples, then with probability 1 - δ, the true error of a hypothesis h with training error $err_S(h)$ is bounded by

$$err_D(h) \leq err_S(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

Generalization error

Training error

A function of VC dimension.

Low VC dimension gives tighter bound

VC dimensions and linear classifiers

What we know so far

1. If we have m examples, then with probability 1 - δ, the true error of a hypothesis h with training error $err_S(h)$ is bounded by

$$err_D(h) \leq err_S(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

Generalization error — Training error — A function of VC dimension.

Low VC dimension gives tighter bound

2. VC dimension of a linear classifier in d dimensions = d + 1

VC dimensions and linear classifiers

What we know so far

1. If we have m examples, then with probability 1 - δ, the true error of a hypothesis h with training error $err_S(h)$ is bounded by

$$err_D(h) \leq err_S(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

Generalization error — Training error — A function of VC dimension.

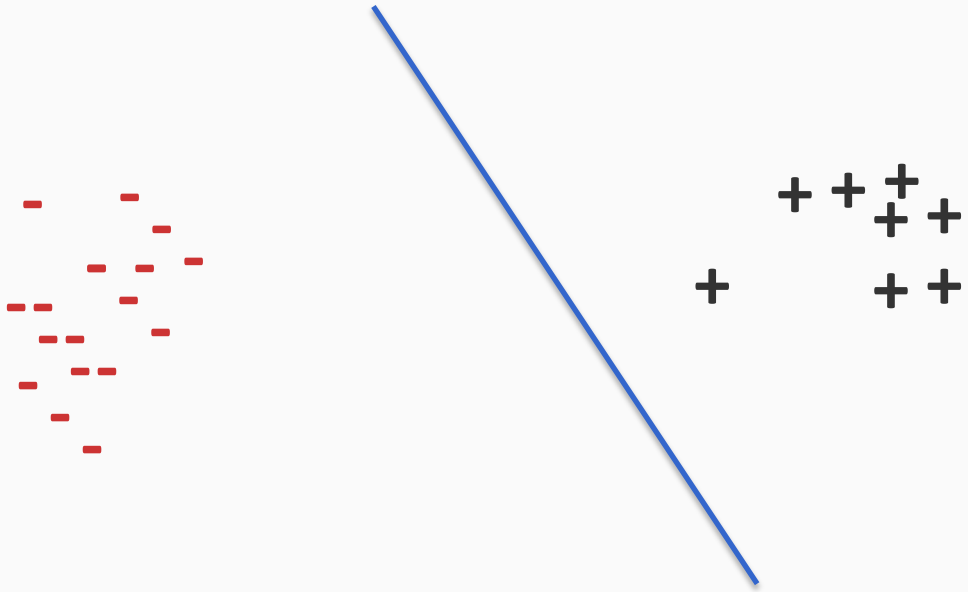
Low VC dimension gives tighter bound

2. VC dimension of a linear classifier in d dimensions = d + 1

But are all linear classifiers the same?

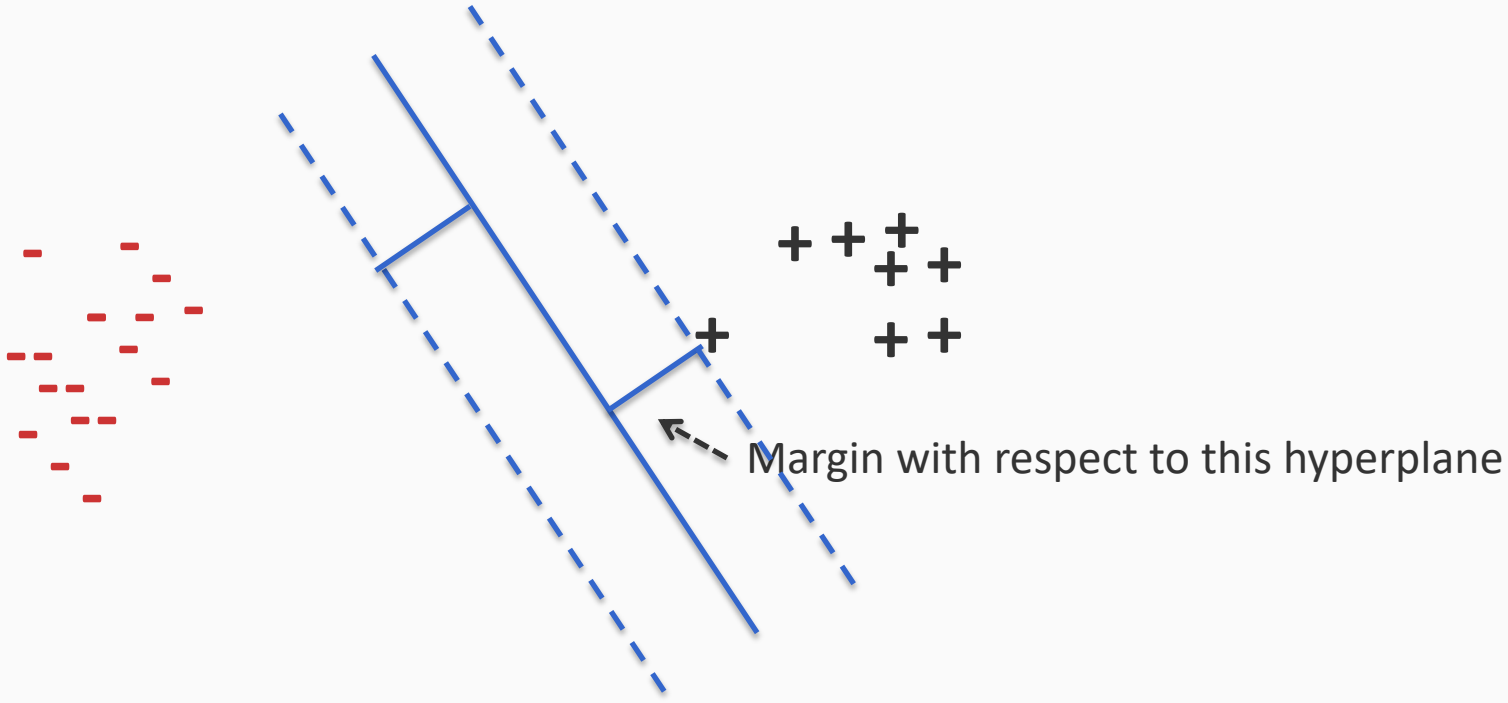
Recall: Margin

The margin of a hyperplane for a dataset is the distance between the hyperplane and the data point nearest to it.



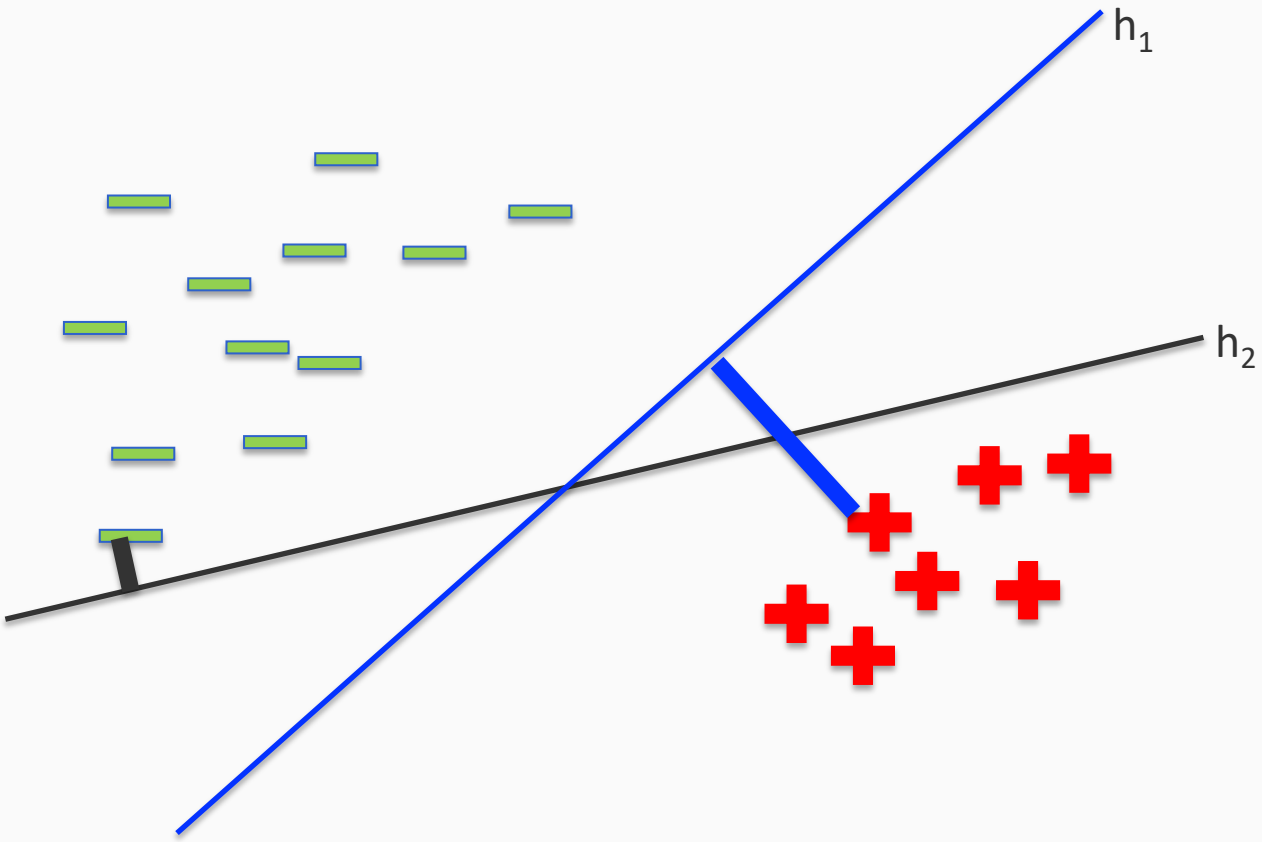
Recall: Margin

The margin of a hyperplane for a dataset is the distance between the hyperplane and the data point nearest to it.



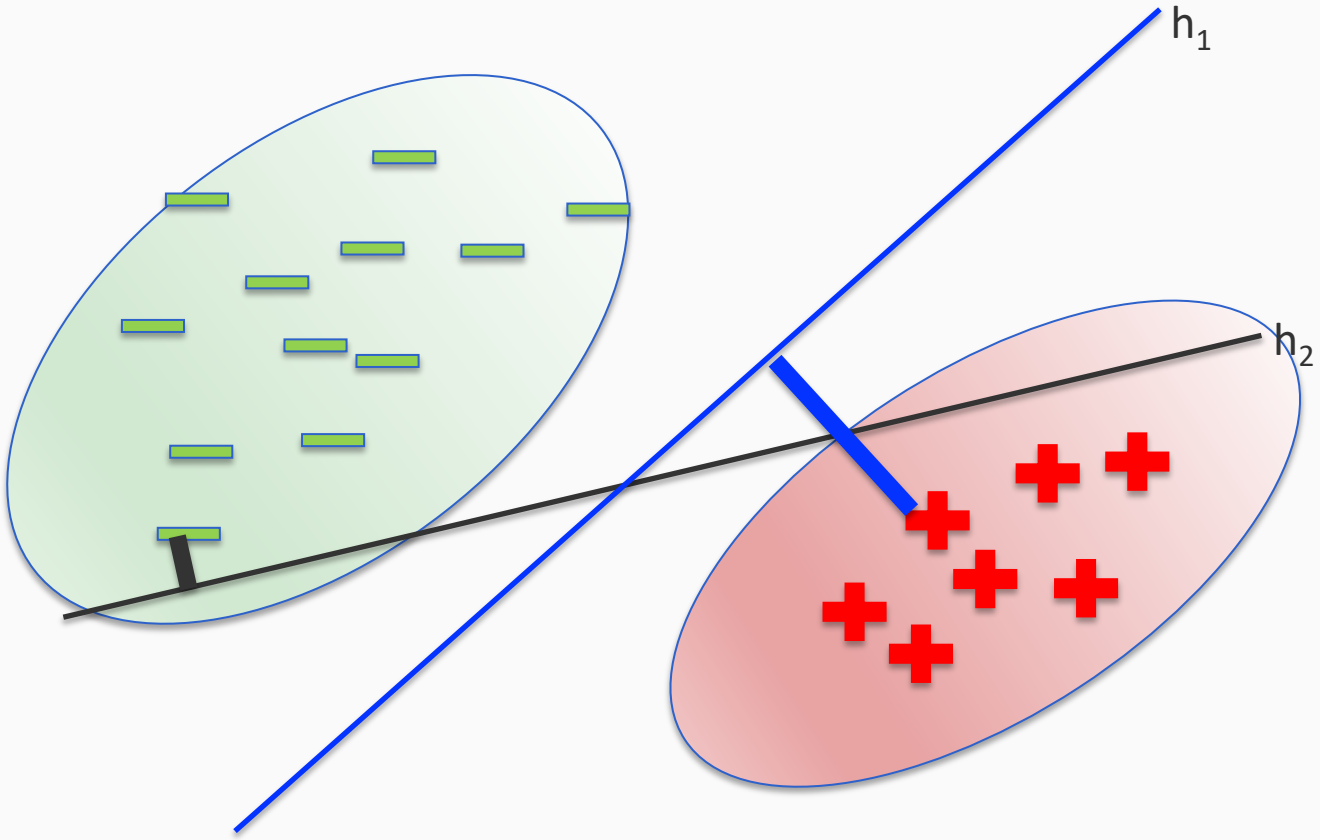
Recall: Margin

Which hyperplane is better?



Recall: Margin

Which hyperplane is better?



The farther from the data points, the less chance to make wrong prediction

Data dependent VC dimension

- Intuitively, larger margins are better

- Suppose we only consider linear classifiers with margins γ_1 and γ_2
  - H_1 = linear classifiers that have a margin at least γ_1
  - H_2 = linear classifiers that have a margin at least γ_2
  - And $\gamma_1 > \gamma_2$

- The entire set of functions H_1 is "better"

Data dependent VC dimension

Theorem (Vapnik):

- Let H be the set of linear classifiers that separate the training set by a margin at least γ
- Then

$$VC(H) \leq \min\left(\frac{R^2}{\gamma^2}, d\right) + 1$$

- R is the radius of the smallest sphere containing the data

Data dependent VC dimension

Theorem (Vapnik):

- Let H be the set of linear classifiers that separate the training set by a margin at least γ
- Then

$$VC(H) \leq \min\left(\frac{R^2}{\gamma^2}, d\right) + 1$$

- R is the radius of the smallest sphere containing the data

Larger margin $\Longrightarrow$ Lower VC dimension

Lower VC dimension $\Longrightarrow$ Better generalization bound

Learning strategy

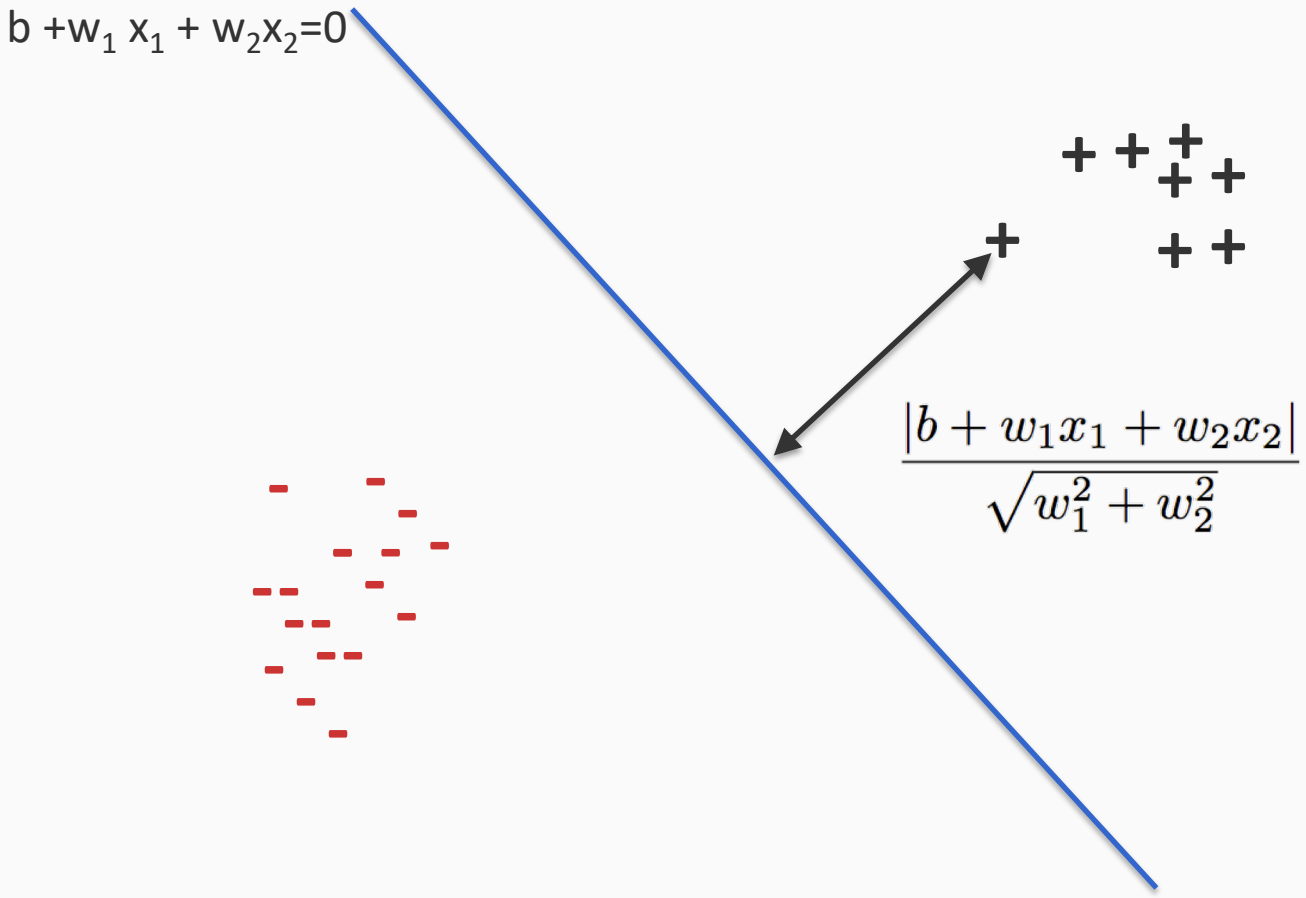
Find the linear classifier that maximizes the margin

This lecture: Support vector machines

- Training by maximizing margin
- The SVM objective
- Solving the SVM optimization problem
- Support vectors, duals and kernels

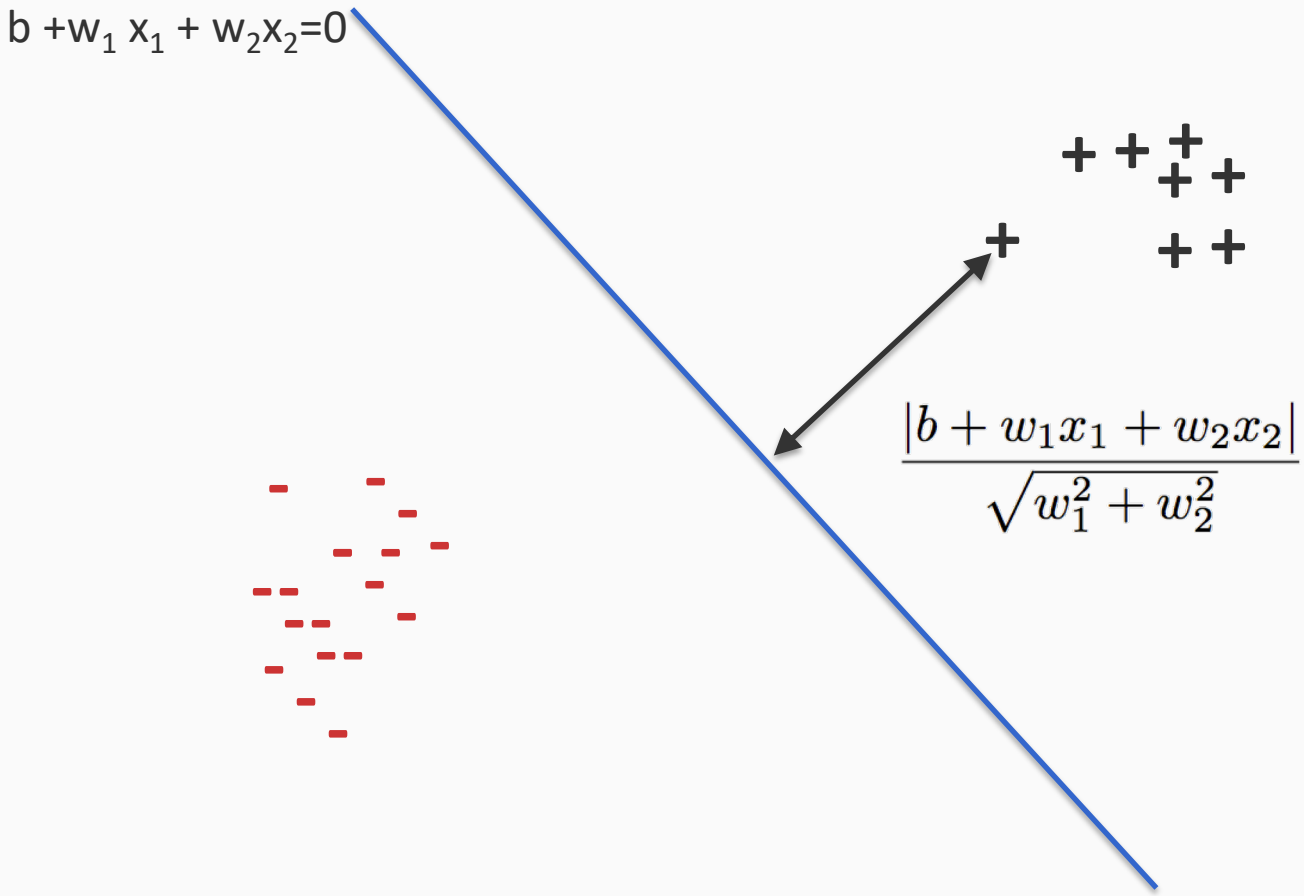
Recall: The geometry of a linear classifier

Prediction = sgn(b +w_1 x_1 + w_2x_2)



Recall: The geometry of a linear classifier

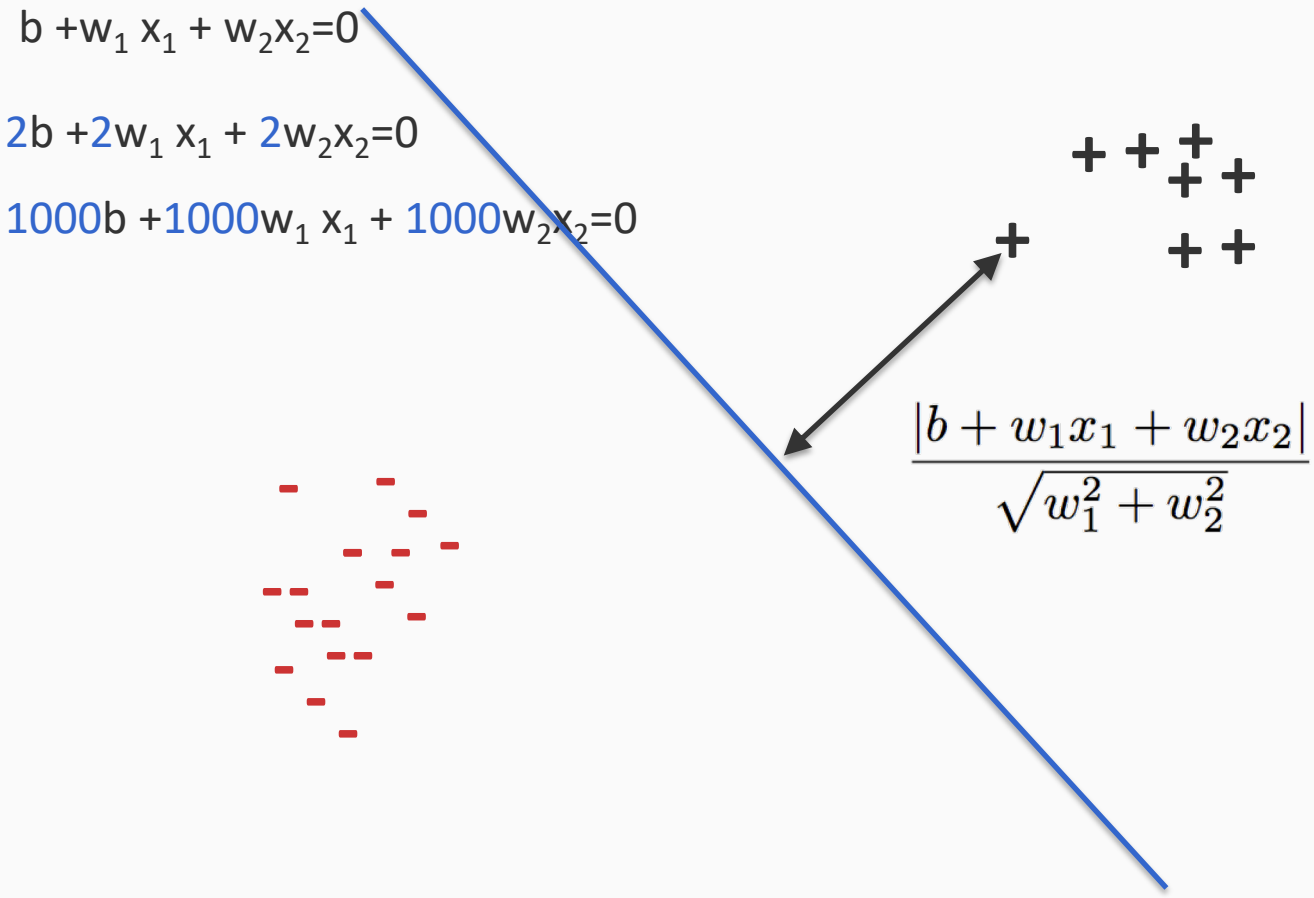
Prediction = *sgn*$(b + w_1 x_1 + w_2 x_2)$



We only care about the sign, not the magnitude

Recall: The geometry of a linear classifier

Prediction = sgn(b +w_1 x_1 + w_2x_2)



We only care about the sign, not the magnitude

Maximizing margin

- Margin = distance of the closest point from the hyperplane

$$\gamma = \min_{\mathbf{x}_i, y_i} \frac{y_i \left(\mathbf{w}^T \mathbf{x}_i + b \right)}{||\mathbf{w}||}$$

Maximizing margin

- Margin = distance of the closest point from the hyperplane

$$\gamma = \min_{\mathbf{x}_i, y_i} \frac{y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right)}{||\mathbf{w}||}$$

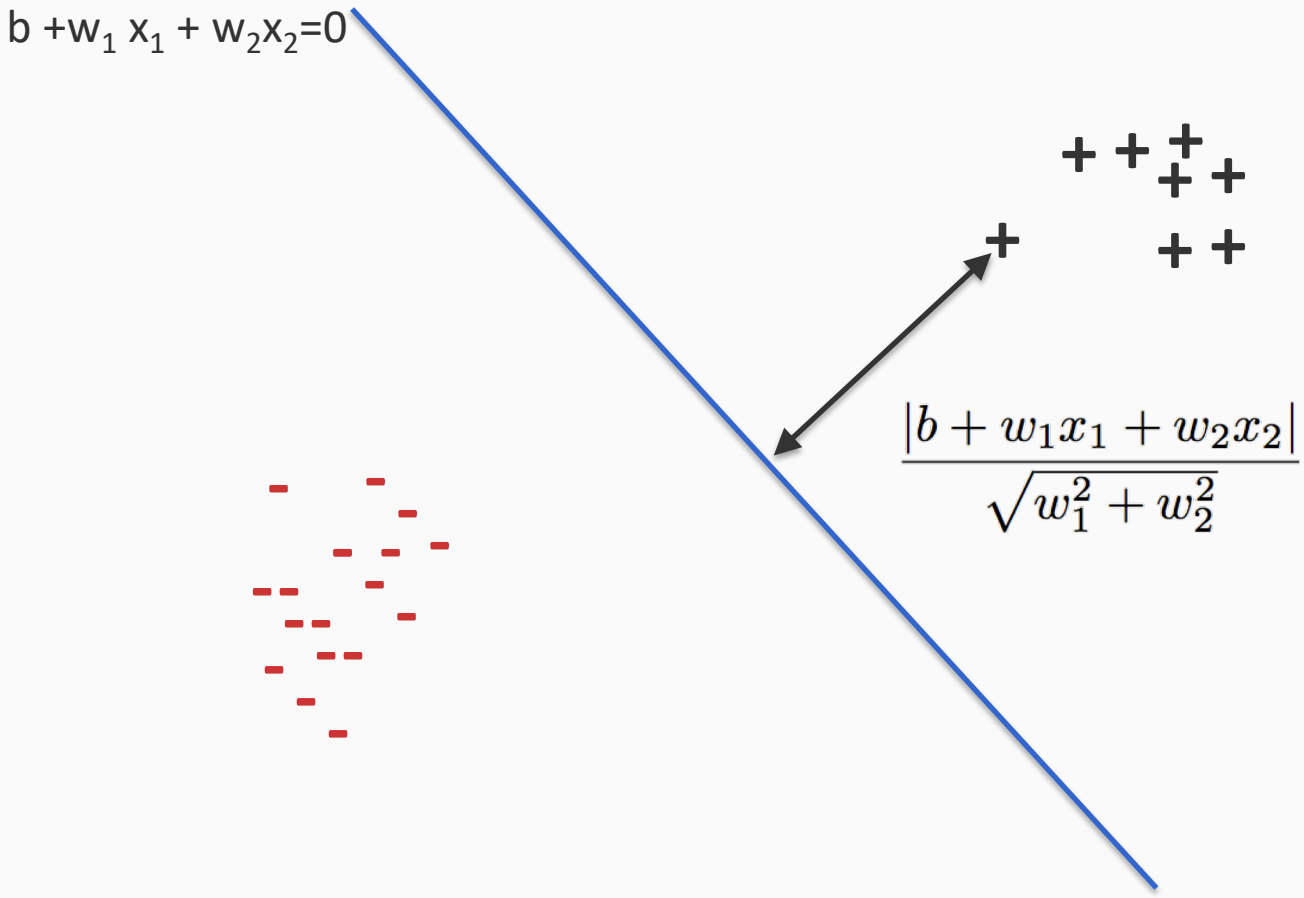
- We want $\max_{\mathbf{w}} \gamma$

Some people call this the *geometric margin*

The numerator alone is called the *functional margin*

Recall: The geometry of a linear classifier

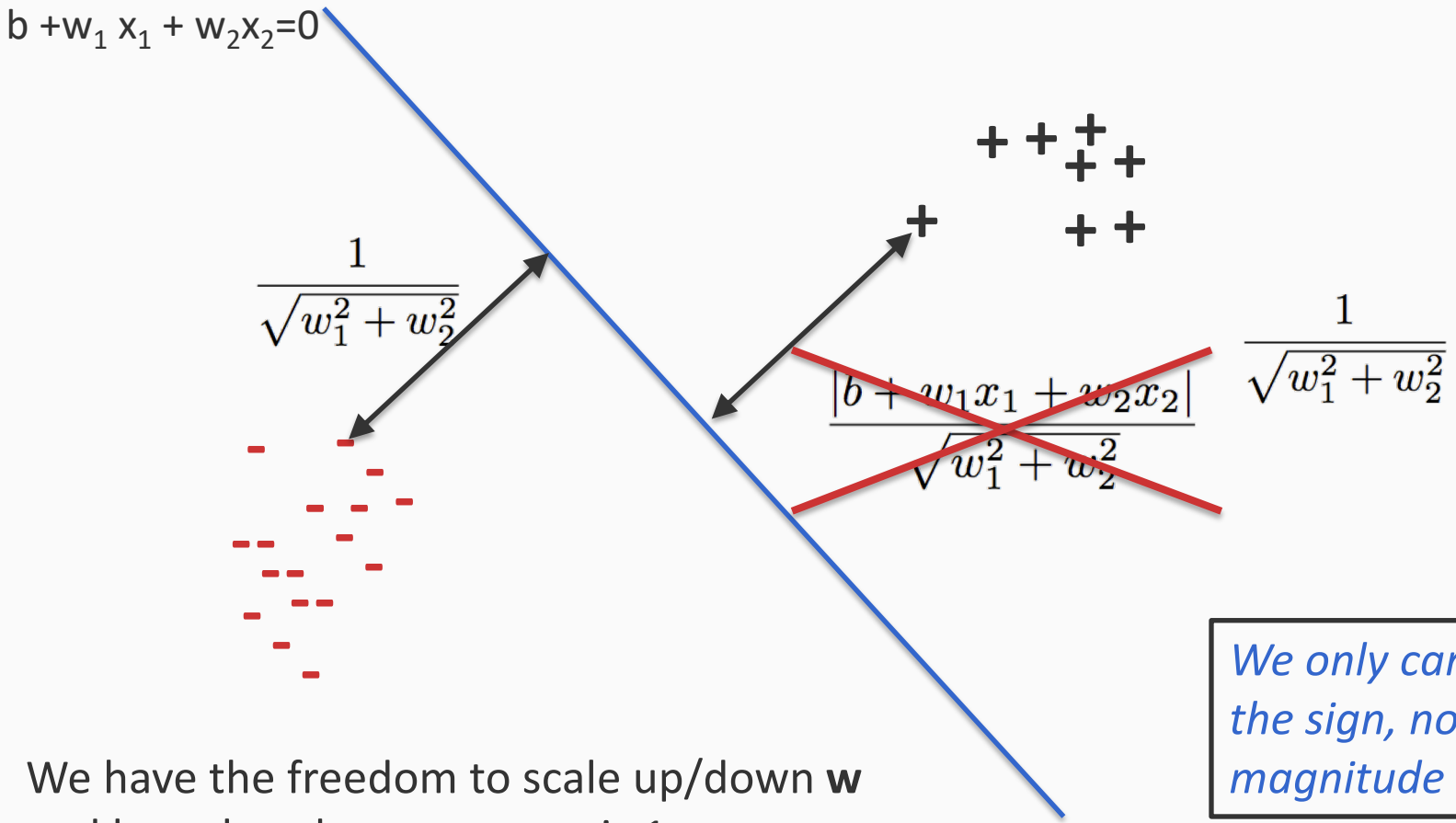
Prediction = sgn(b +w_1 x_1 + w_2x_2)



We only care about the sign, not the magnitude

Recall: The geometry of a linear classifier

Prediction = sgn(b +w_1 x_1 + w_2x_2)



We have the freedom to scale up/down **w** and b so that the numerator is 1.

We only care about the sign, not the magnitude

Maximizing margin

- Margin = distance of the closest point from the hyperplane

$$\gamma = \min_{\mathbf{x}_i, y_i} \frac{y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right)}{||\mathbf{w}||}$$

- We want $\max_{\mathbf{w}} \gamma$
- We only care about the sign of $\mathbf{w}^T\mathbf{x}_i$+b in the end and not the magnitude
  - Set the absolute score (functional margin) of the closest point to be 1 and allow **w** to adjust itself

$\max_{\mathbf{w}} \gamma$ is equivalent to $\max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|}$ in this setting

Max-margin classifiers

$$\gamma = \min_{\mathbf{x}_i, y_i} \frac{y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right)}{||\mathbf{w}||}$$

- Learning problem:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$
$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Max-margin classifiers

$$\gamma = \min_{\mathbf{x}_i, y_i} \frac{y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right)}{||\mathbf{w}||}$$

- Learning problem:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$
$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Minimizing gives us $\max_{\mathbf{w}} \frac{1}{||\mathbf{w}||}$

Max-margin classifiers

$$\gamma = \min_{\mathbf{x}_i, y_i} \frac{y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right)}{||\mathbf{w}||}$$

- Learning problem:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Minimizing gives us $\max_{\mathbf{w}} \frac{1}{||\mathbf{w}||}$

This condition is true for every example, specifically, for the example closest to the separator

Max-margin classifiers

$$\gamma = \min_{\mathbf{x}_i, y_i} \frac{y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right)}{||\mathbf{w}||}$$

- Learning problem:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Minimizing gives us $\max_{\mathbf{w}} \frac{1}{||\mathbf{w}||}$

This condition is true for every example, specifically, for the example closest to the separator

- This is called the "hard" Support Vector Machine

We will look at how to solve this optimization problem later

What if the data is not separable?

Hard SVM

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$$

Maximize margin

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$$

Every example has an functional margin of at least 1

What if the data is not separable?

Hard SVM

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$$

Maximize margin

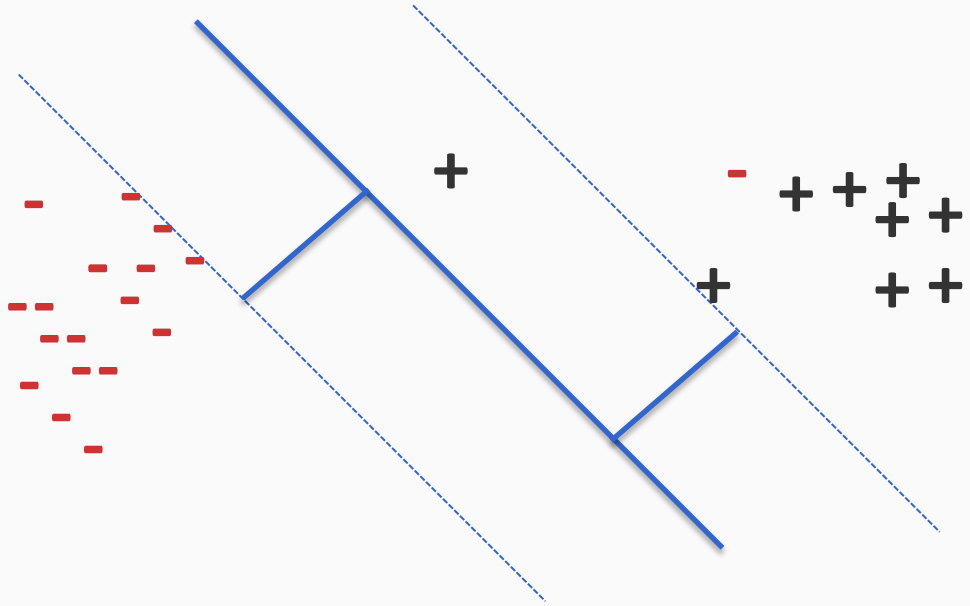
$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$$

Every example has an functional margin of at least 1

- This is a constrained optimization problem
- If the data is not separable, there is no **w** that will classify the data
- Infeasible problem, no solution!

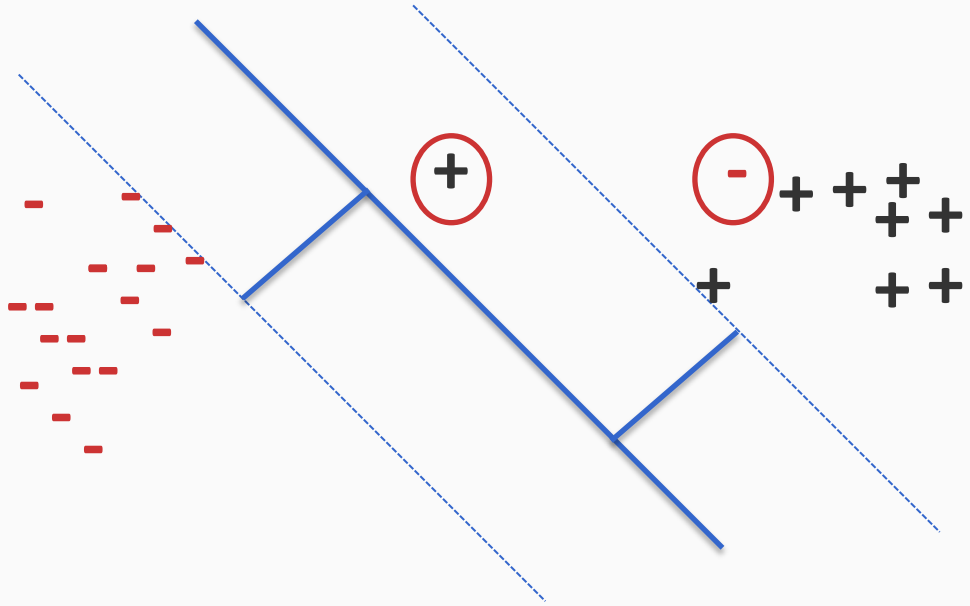
Dealing with non-separable data

Key idea: Allow some examples to “break into the margin”



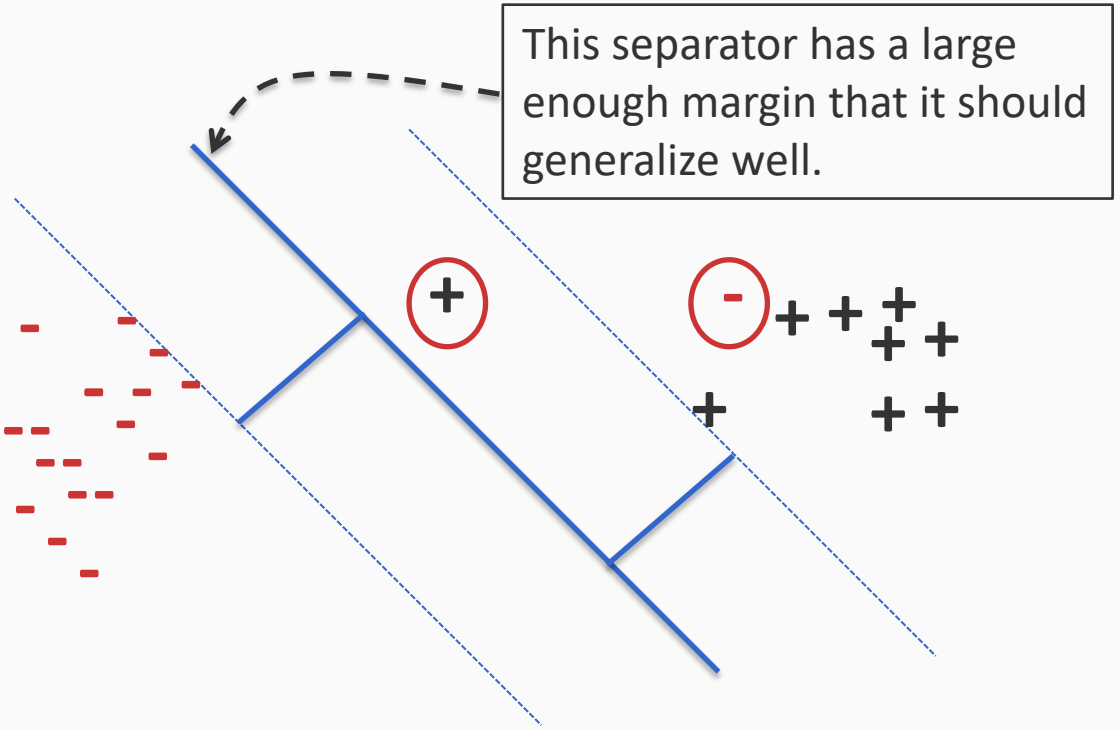
Dealing with non-separable data

Key idea: Allow some examples to “break into the margin”



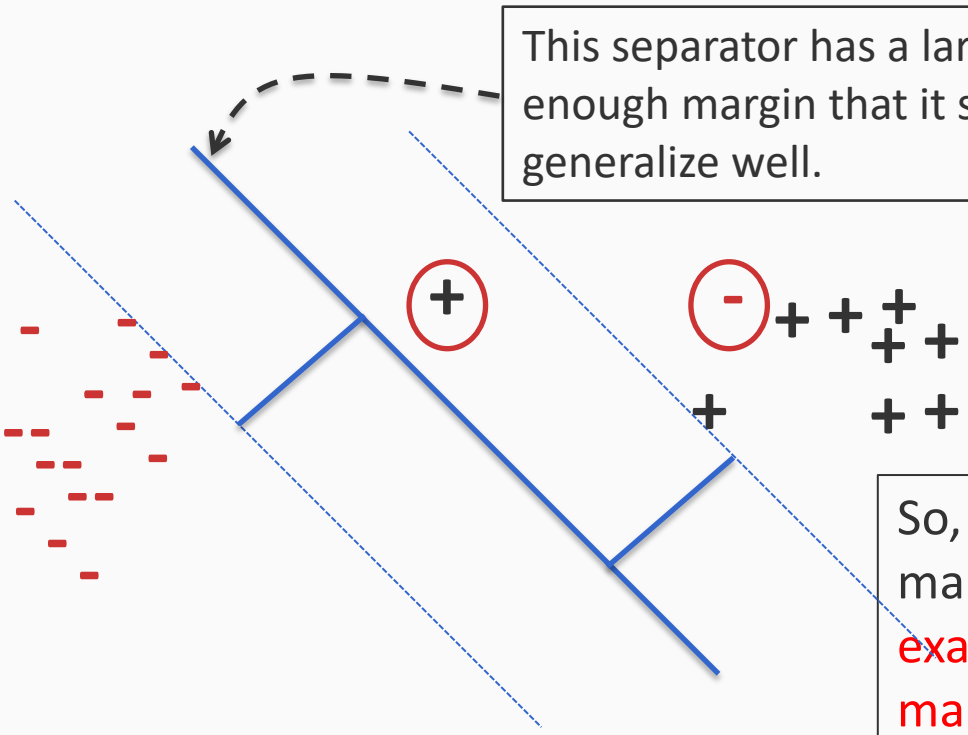
Dealing with non-separable data

Key idea: Allow some examples to "break into the margin"



Dealing with non-separable data

Key idea: Allow some examples to "break into the margin"



Soft SVM

- Hard SVM:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$

Maximize margin

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Every example has a functional margin of at least 1

Soft SVM

- Hard SVM: $\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$ Maximize margin

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$$ Every example has a functional margin of at least 1

- Introduce one *slack variable* ξ_i per example
  - And require $y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

Soft SVM

- Hard SVM: $\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$ Maximize margin

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$$ Every example has a functional margin of at least 1

- Introduce one *slack variable* ξ_i per example
  - And require $y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$

Intuition: The slack variable allows examples to "break" into the margin

If the slack value is zero, then the example is either on or outside the margin

Soft SVM

- Hard SVM: $\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$ Maximize margin

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$$ Every example has a functional margin of at least 1

- Introduce one *slack variable* ξ_i per example
  - And require $y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$
- New optimization problem for learning

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Soft SVM

- Hard SVM:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$$
$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1$$

Maximize margin

Every example has a functional margin of at least 1

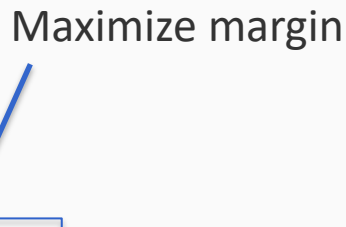
- Introduce one *slack variable* ξ_i per example
  - And require $y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$
- New optimization problem for learning

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$
$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

Soft SVM

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_i \xi_i \\ \text{s.t.} \forall i, \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

Soft SVM

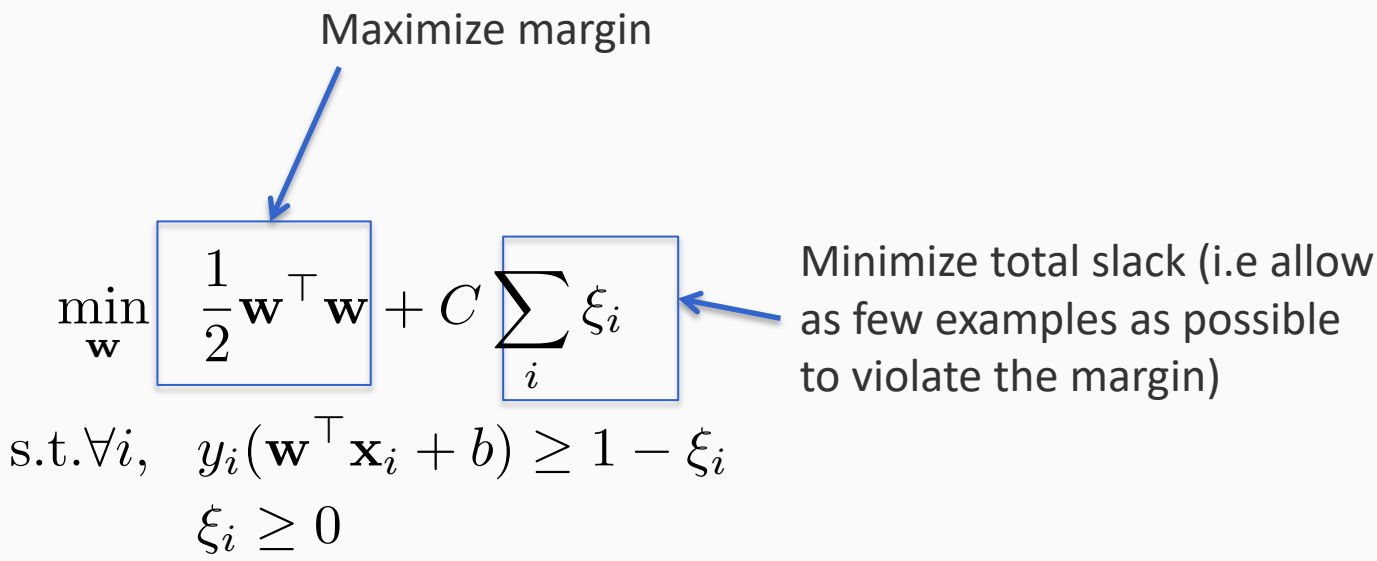


$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

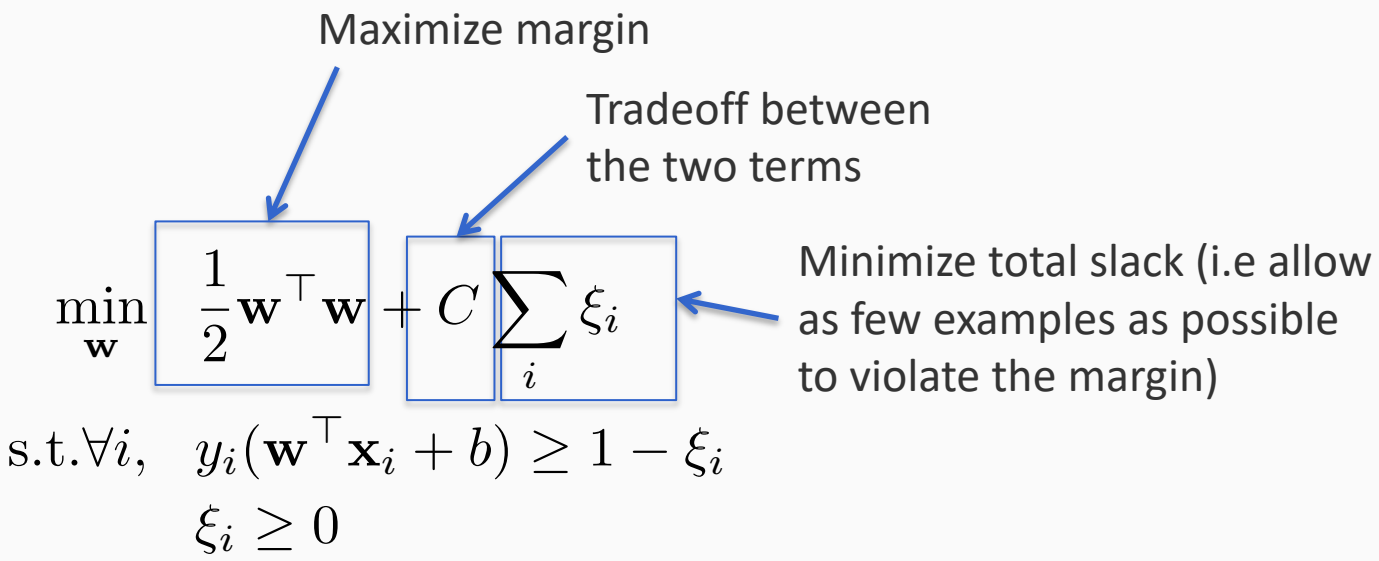
$$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

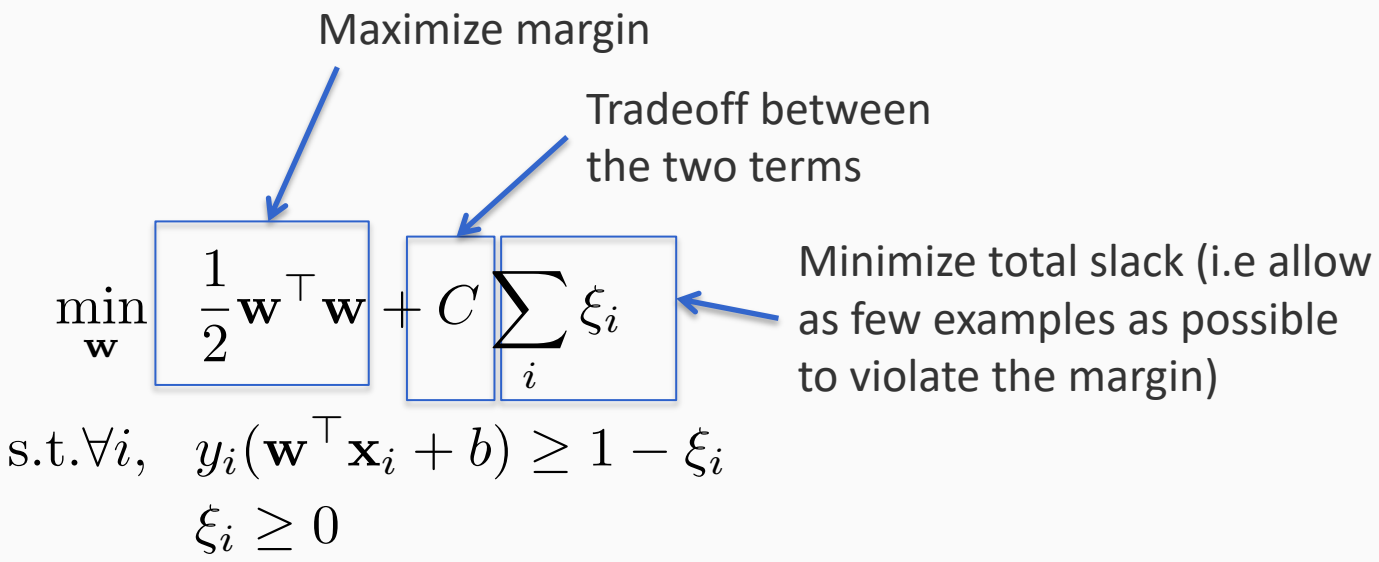
Soft SVM


Maximize margin
$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$
Minimize total slack (i.e allow as few examples as possible to violate the margin)
$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$
$\xi_i \geq 0$

Soft SVM

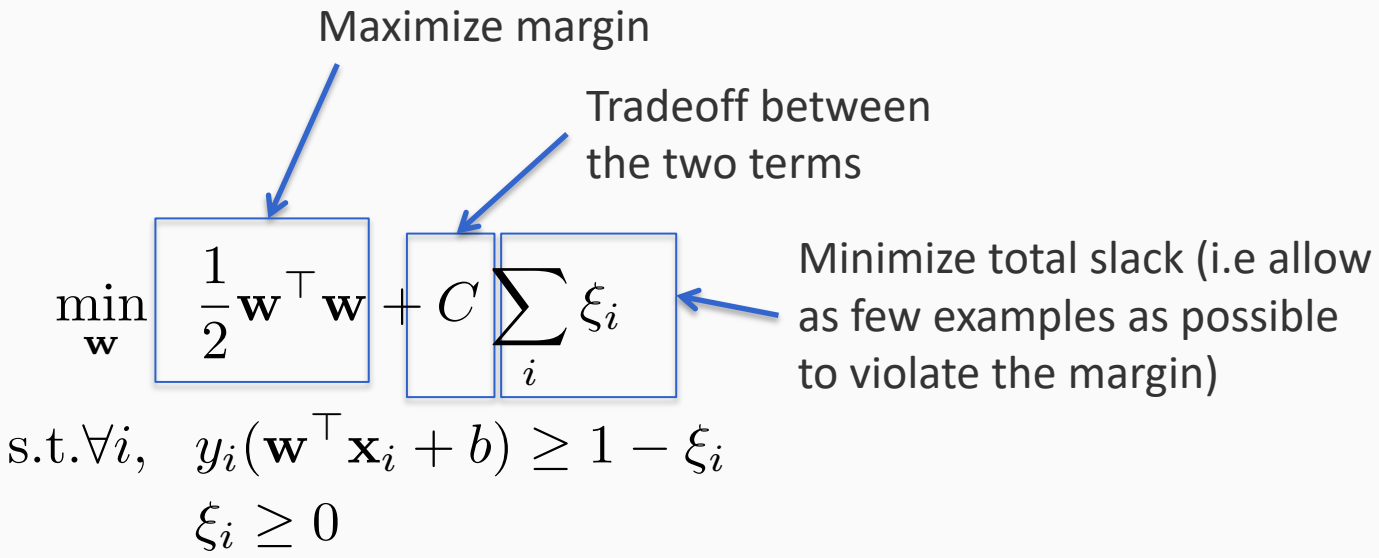

Maximize margin
Tradeoff between the two terms
Minimize total slack (i.e allow as few examples as possible to violate the margin)
$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$
$\text{s.t.} \forall i, \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i$
$\xi_i \geq 0$

Soft SVM



Eliminate the slack variables to rewrite this

Soft SVM



Eliminate the slack variables to rewrite this

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

This form is more interpretable

Maximizing margin and minimizing loss

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Maximize margin

Penalty for the prediction

Maximizing margin and minimizing loss

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Maximize margin

Penalty for the prediction

We can consider three cases

Maximizing margin and minimizing loss

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Maximize margin

Penalty for the prediction

We can consider three cases

- Example is correctly classified and is *outside* the margin: penalty = 0

Maximizing margin and minimizing loss

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Maximize margin

Penalty for the prediction

We can consider three cases

- Example is correctly classified and is *outside* the margin: penalty = 0
- Example is incorrectly classified: penalty = $1 - y_i\,(\mathbf{w}^\top\mathbf{x}_i + b)$

Maximizing margin and minimizing loss

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Maximize margin

Penalty for the prediction

We can consider three cases

- Example is correctly classified and is *outside* the margin: penalty = 0
- Example is incorrectly classified: penalty = $1 - y_i\,(\mathbf{w}^\top\mathbf{x}_i + b)$
- Example is correctly classified but within the margin: penalty = $1 - y_i\,(\mathbf{w}^\top\mathbf{x}_i + b)$

Maximizing margin and minimizing loss

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Maximize margin

Penalty for the prediction

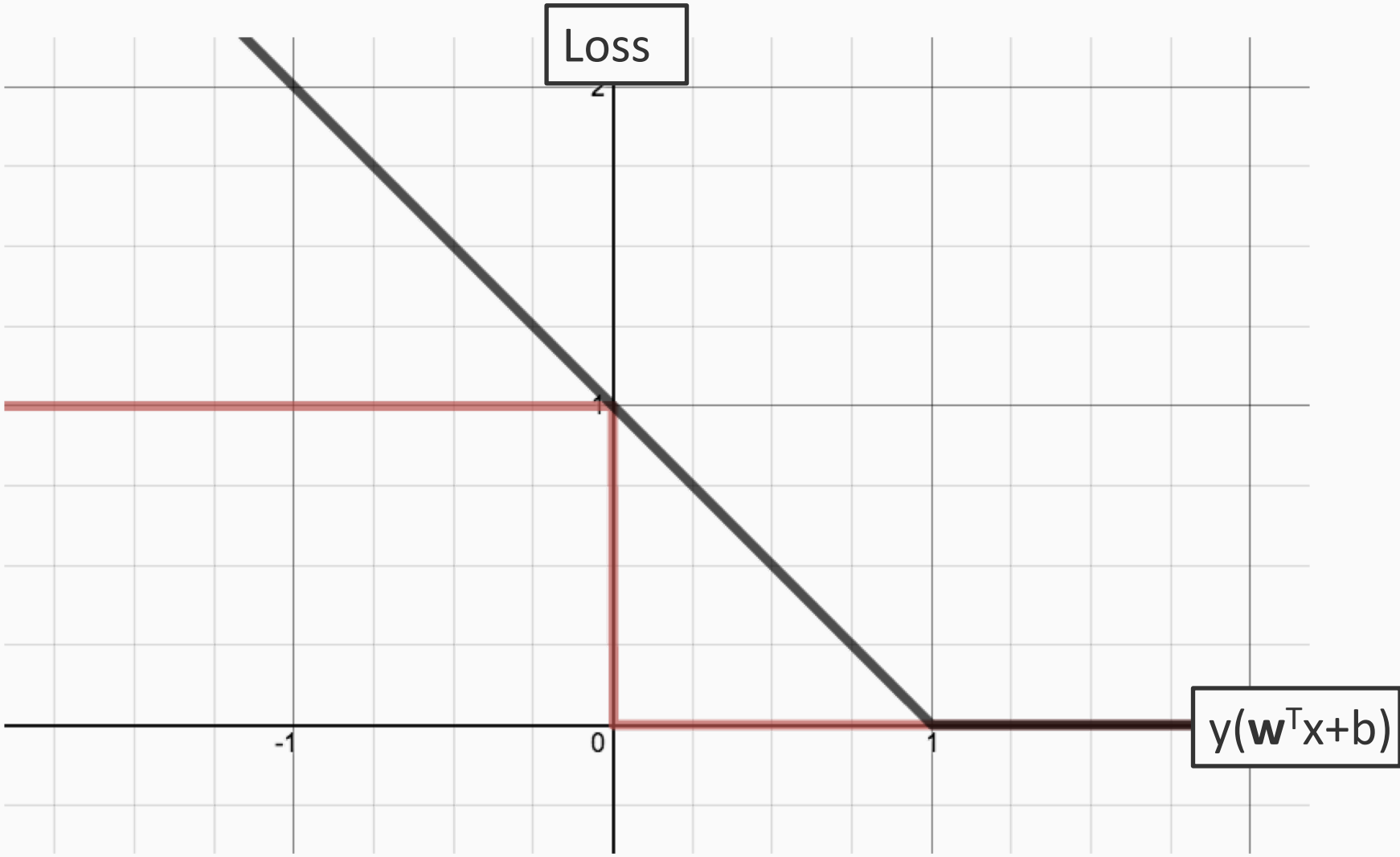
We can consider three cases

- Example is correctly classified and is *outside* the margin: penalty = 0
- Example is incorrectly classified: penalty = $1 - y_i\,(\mathbf{w}^\top\mathbf{x}_i + b)$
- Example is correctly classified but within the margin: penalty = $1 - y_i\,(\mathbf{w}^\top\mathbf{x}_i + b)$

This gives us the hinge loss function

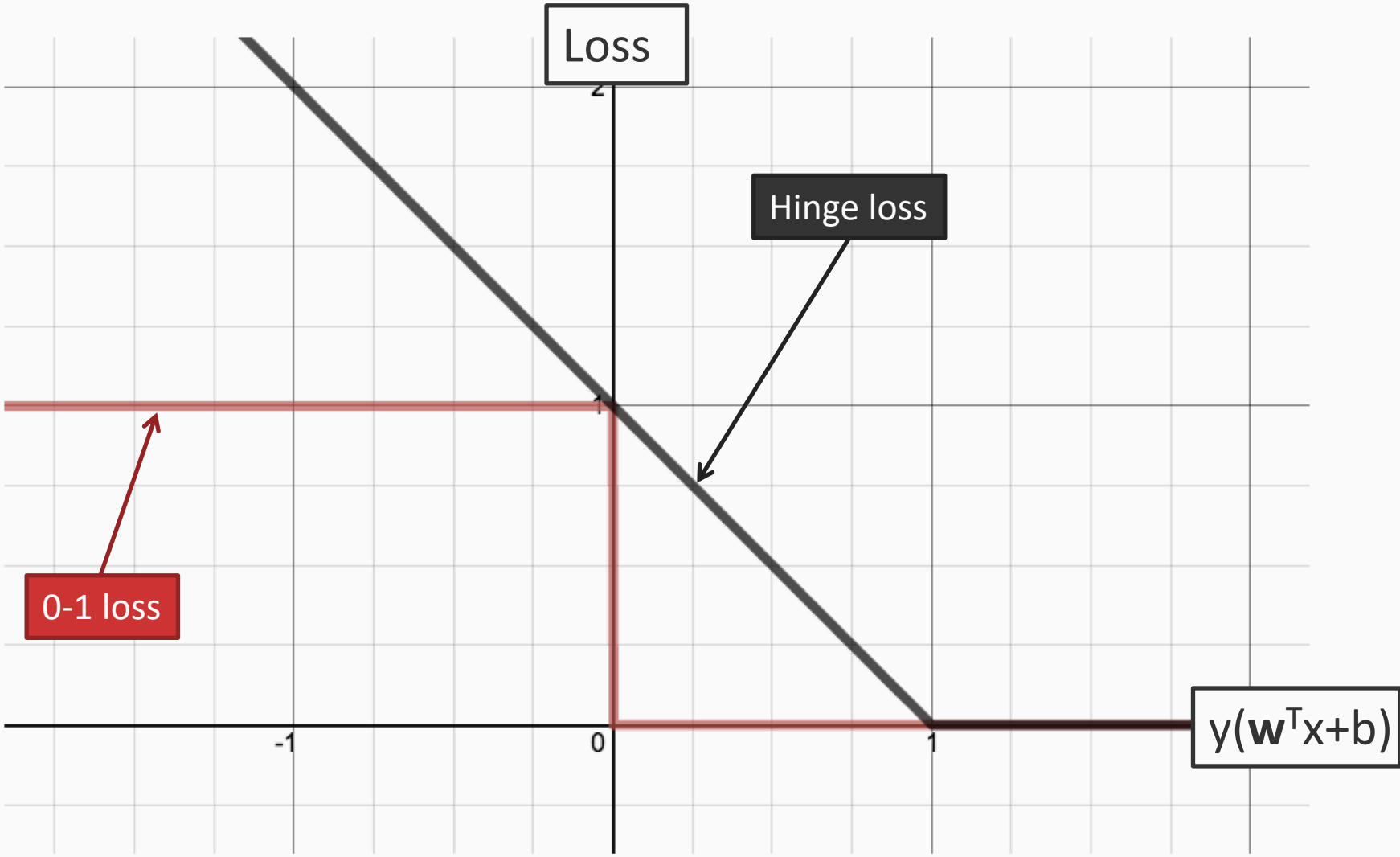
$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}, b) = \max\left(0, 1 - y(\mathbf{w}^\top\mathbf{x} + b)\right)$$

The Hinge Loss



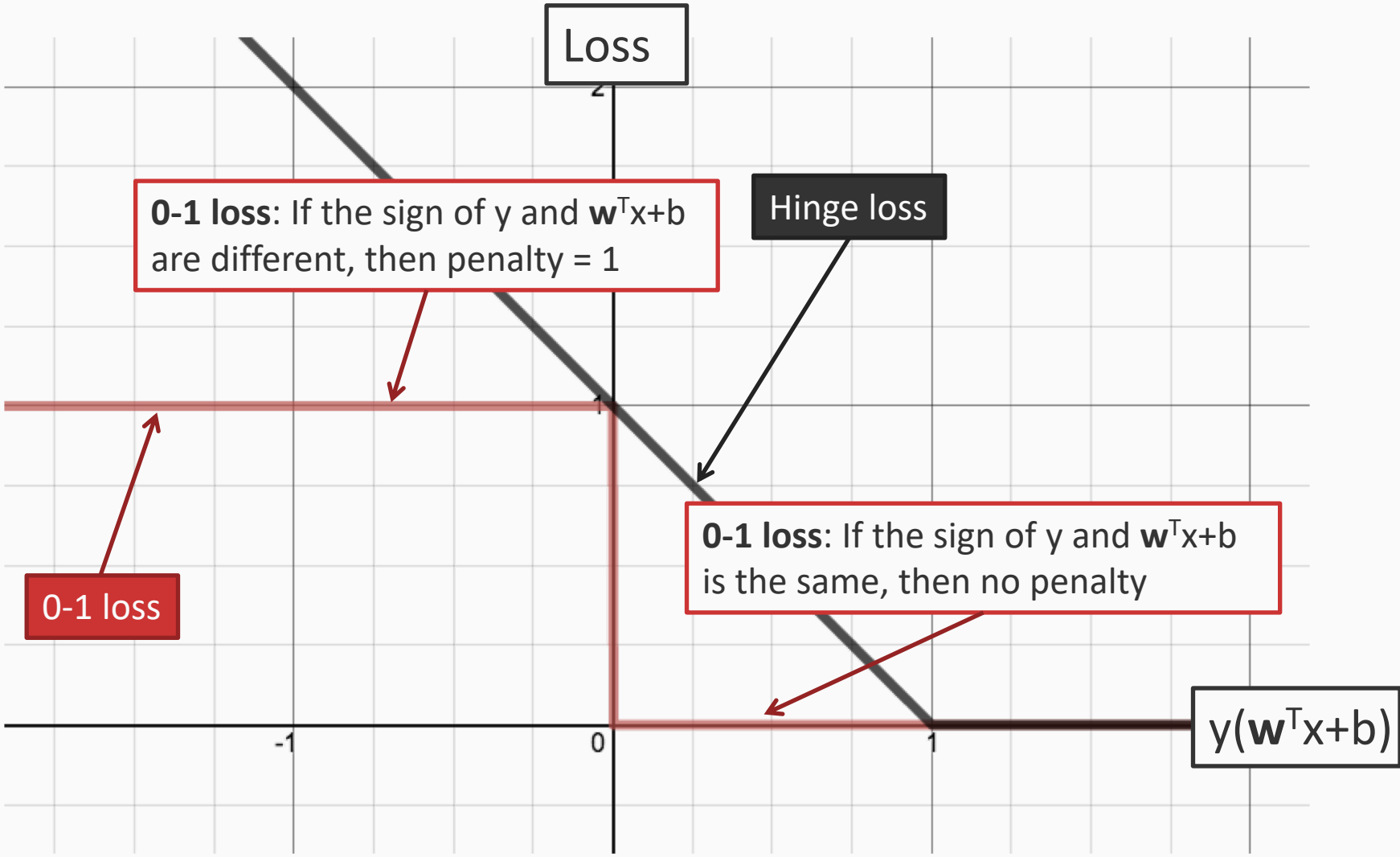
$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}, b) = \max\left(0, 1 - y(\mathbf{w}^\top \mathbf{x} + b)\right)$$

The Hinge Loss



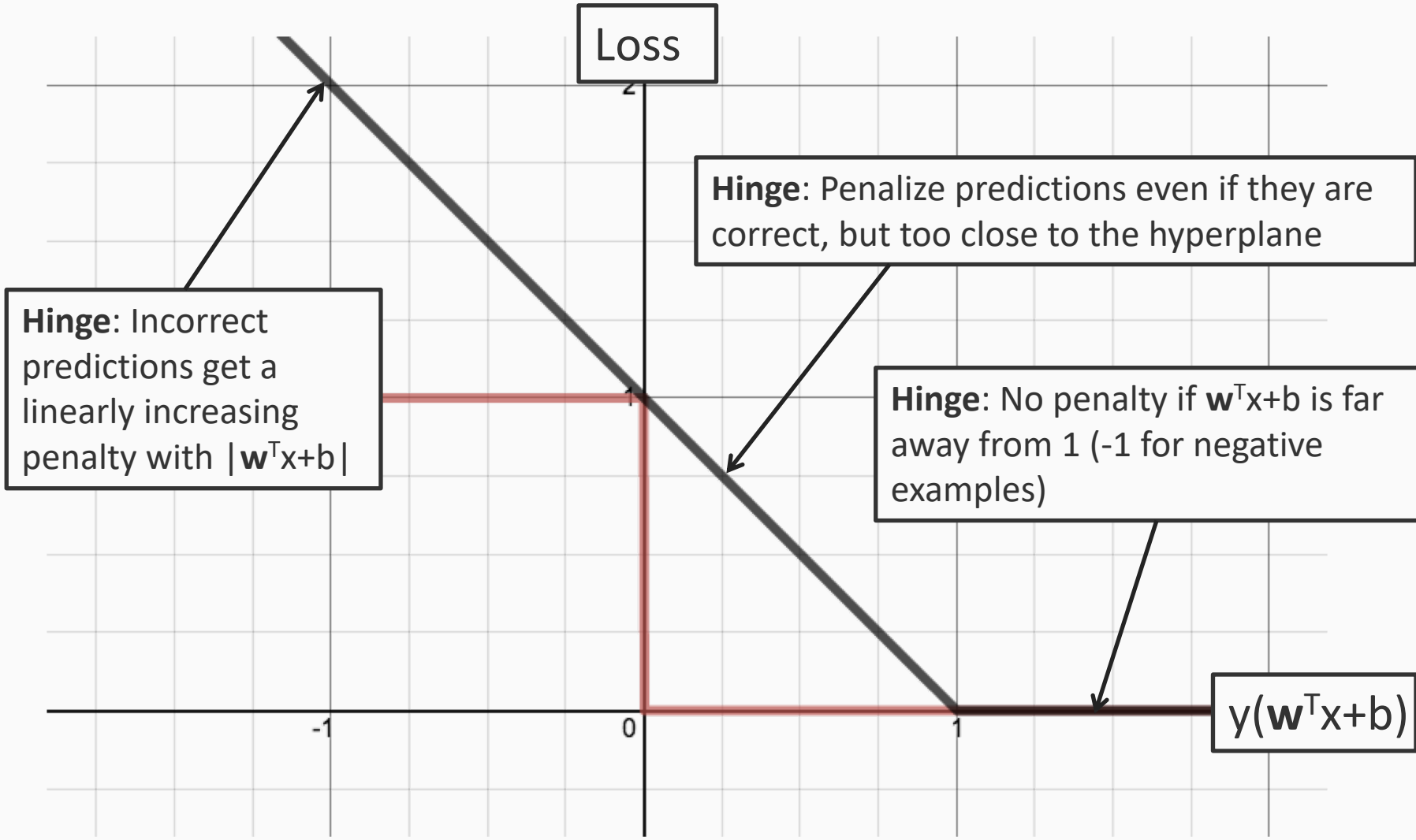
$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}, b) = \max\left(0, 1 - y(\mathbf{w}^\top \mathbf{x} + b)\right)$$

The Hinge Loss



$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}, b) = \max\left(0, 1 - y(\mathbf{w}^\top \mathbf{x} + b)\right)$$

The Hinge Loss



Maximizing margin and minimizing loss

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Maximize margin

Penalty for the prediction

Three cases

- Example is correctly classified and is outside the margin: penalty = 0
- Example is incorrectly classified: penalty = $1 - y_i\,\mathbf{w}^\top\mathbf{x}_i$
- Example is correctly classified but within the margin: penalty = $1 - y_i\,\mathbf{w}^\top\mathbf{x}_i$

General learning principle

Risk minimization

Define the notion of “loss” over the training data as a function of a hypothesis

Learning = find the hypothesis that has lowest loss on the training data

General learning principle

Regularized risk minimization

Define a regularization function that penalizes over-complex hypothesis.

Capacity control gives better generalization

Define the notion of "loss" over the training data as a function of a hypothesis

Learning =
find the hypothesis that has lowest
[Regularizer + loss on the training data]

SVM objective function

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Regularization term:

- Maximize the margin
- Imposes a preference over the hypothesis space and pushes for better generalization
- Can be replaced with other regularization terms which impose other preferences

Empirical Loss:

- Hinge loss
- Penalizes weight vectors that make mistakes
- Can be replaced with other loss functions which impose other preferences

SVM objective function

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \max\left(0, 1 - y_i(\mathbf{w}^\top\mathbf{x}_i + b)\right)$$

Regularization term:

- Maximize the margin
- Imposes a preference over the hypothesis space and pushes for better generalization
- Can be replaced with other regularization terms which impose other preferences

Empirical Loss:

- Hinge loss
- Penalizes weight vectors that make mistakes
- Can be replaced with other loss functions which impose other preferences

A hyper-parameter that controls the tradeoff between a large margin and a small hinge-loss