Lecture 14:
Support Vector Machines
Fall 2022

Kai-Wei Chang
CS @ UCLA
kw+cm146@kwchang.net

The instructor gratefully acknowledges Dan Roth, Vivek Srikuar, Sriram Sankararaman, Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

Announcement

- Quiz – Due on Wed 11:59pm
- Midterm / Hw1 grades will be released soon

- The Final will be an in-person closed-book exam cover all lectures

- No TA session this week (Veterans Day)

Feedback

- 200 responses. Thanks!
- More exercises; practice questions
- Typos in slides; math notations; handwriting
- Review in TA session

- Too much math/ Too little math
- Too much homework/ Too little homework

Max-margin classifiers

- Learning problem:

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

$$s.t. \ \ \forall i, \quad y_i (w^T x_i + b) \geq 1$$

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

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

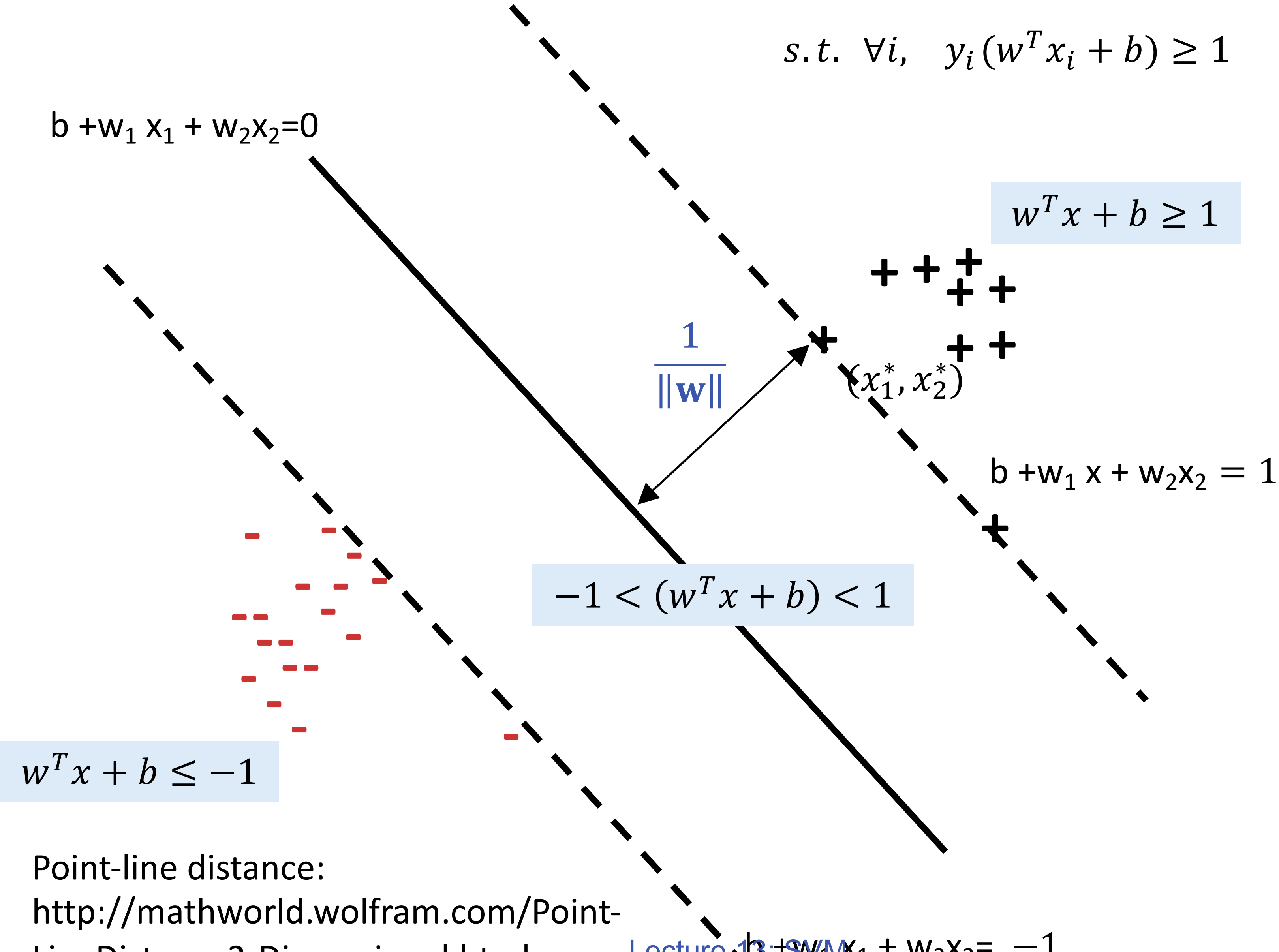
- This is called the “hard” Support Vector Machine

We will look at how to solve this optimization problem later

Hard SVM

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

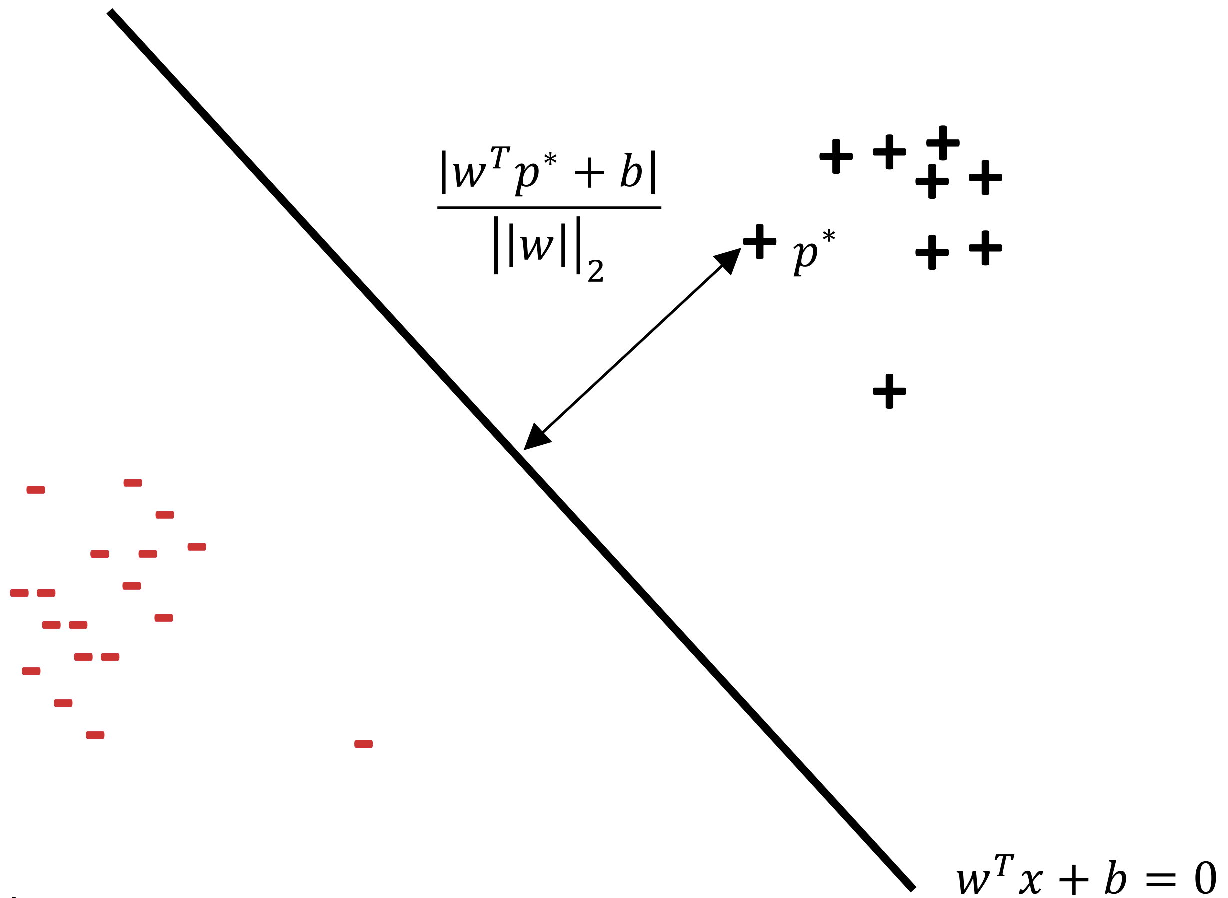
$$s.t.\ \ \forall i, \quad y_i (w^T x_i + b) \geq 1$$



Point-line distance:
http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html

Recall: The geometry of a linear classifier

Prediction = sgn($w^T x + b$)



Point-line distance:
http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html

Margin

What is the distance between
$w^T x + b = 1$ and $w^T x + b = 0$

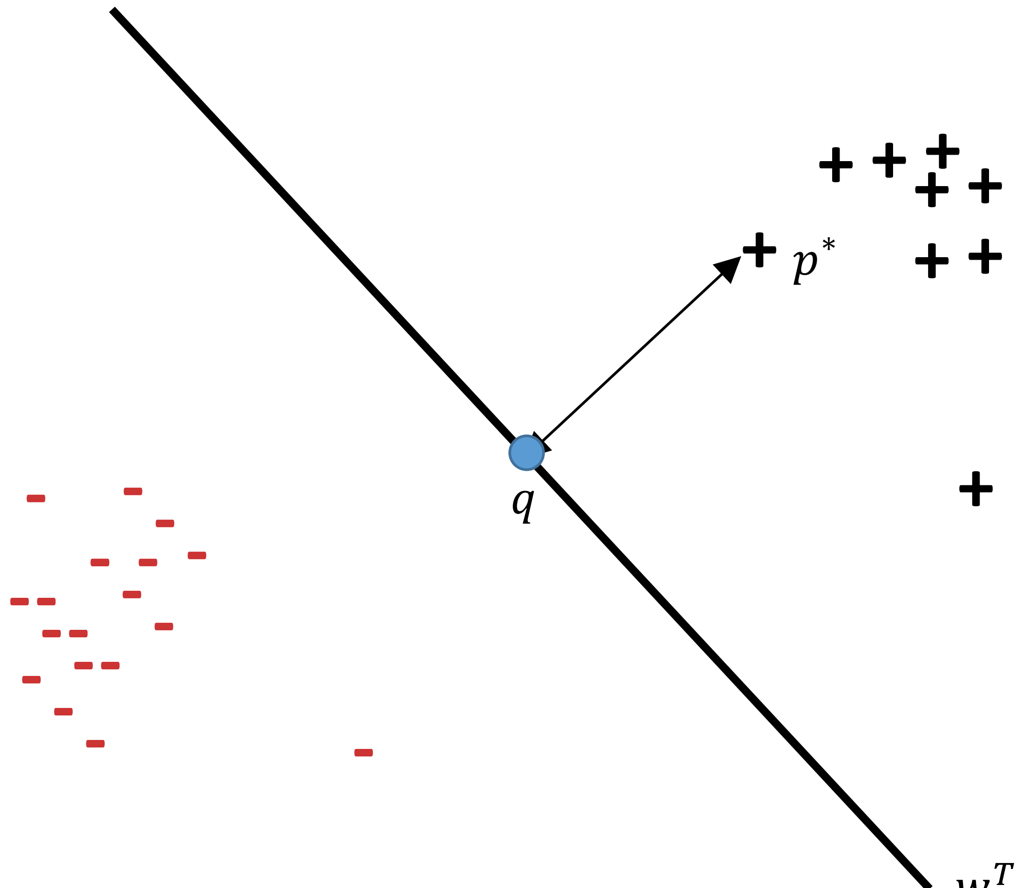
$$q = p^* - dw/||w||$$

q is in $w^T x + b = 0$

$$w^T\left(p^* - \frac{dw}{||w||}\right) + b = 0$$

$$w^T p^* - d||w|| + b = 0$$

$$d = \frac{w^T p^* + b}{||w||}$$

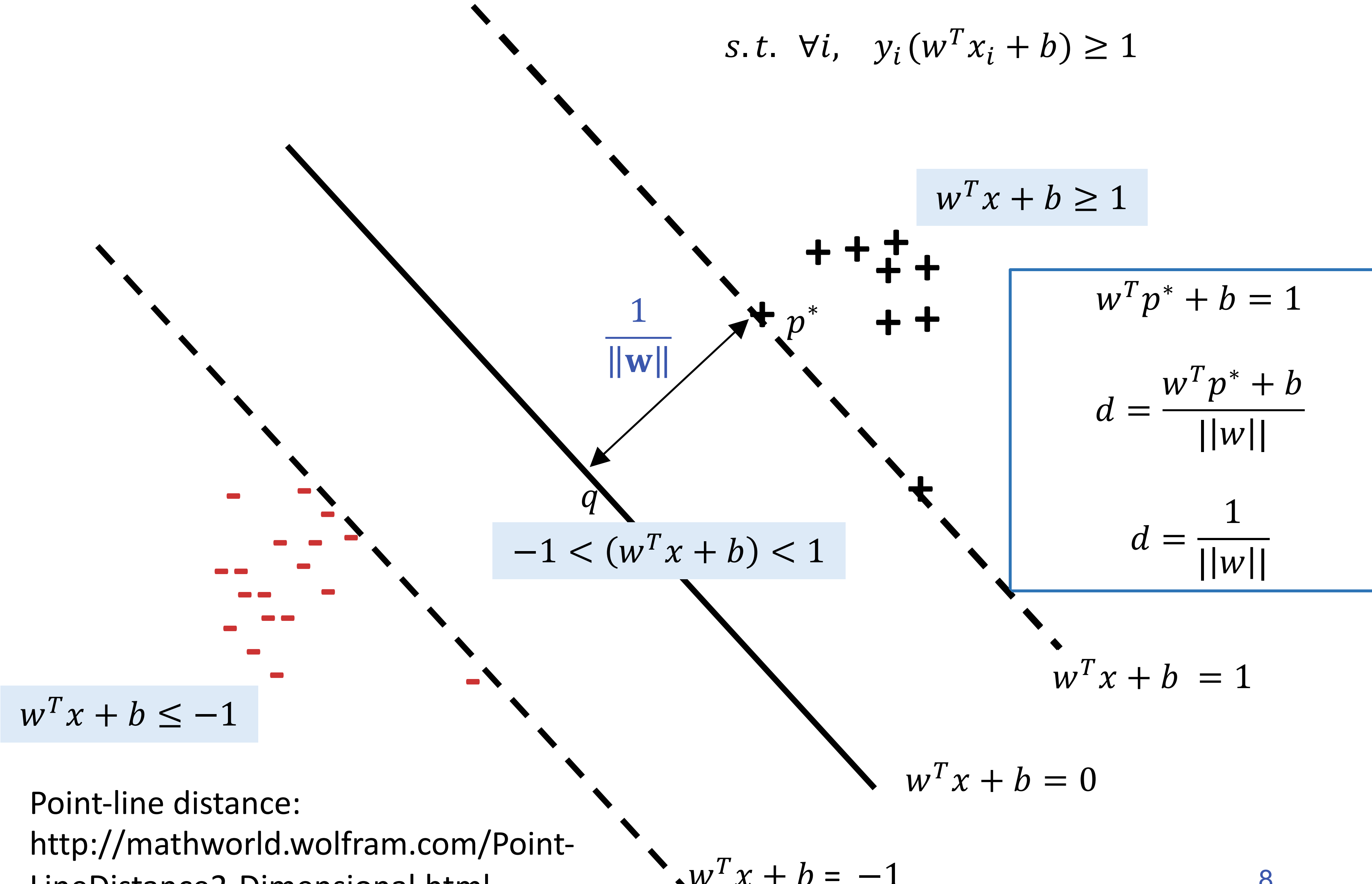


Point-line distance:
http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html

Hard SVM

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

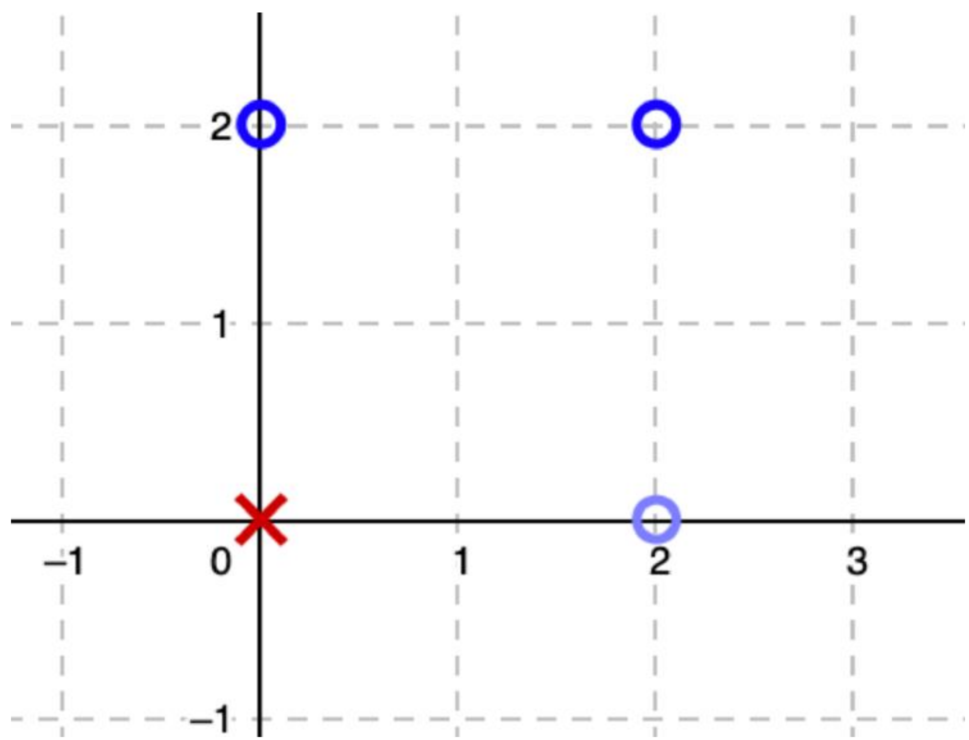
$$s.t. \ \forall i, \quad y_i (w^T x_i + b) \geq 1$$



Point-line distance: http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html

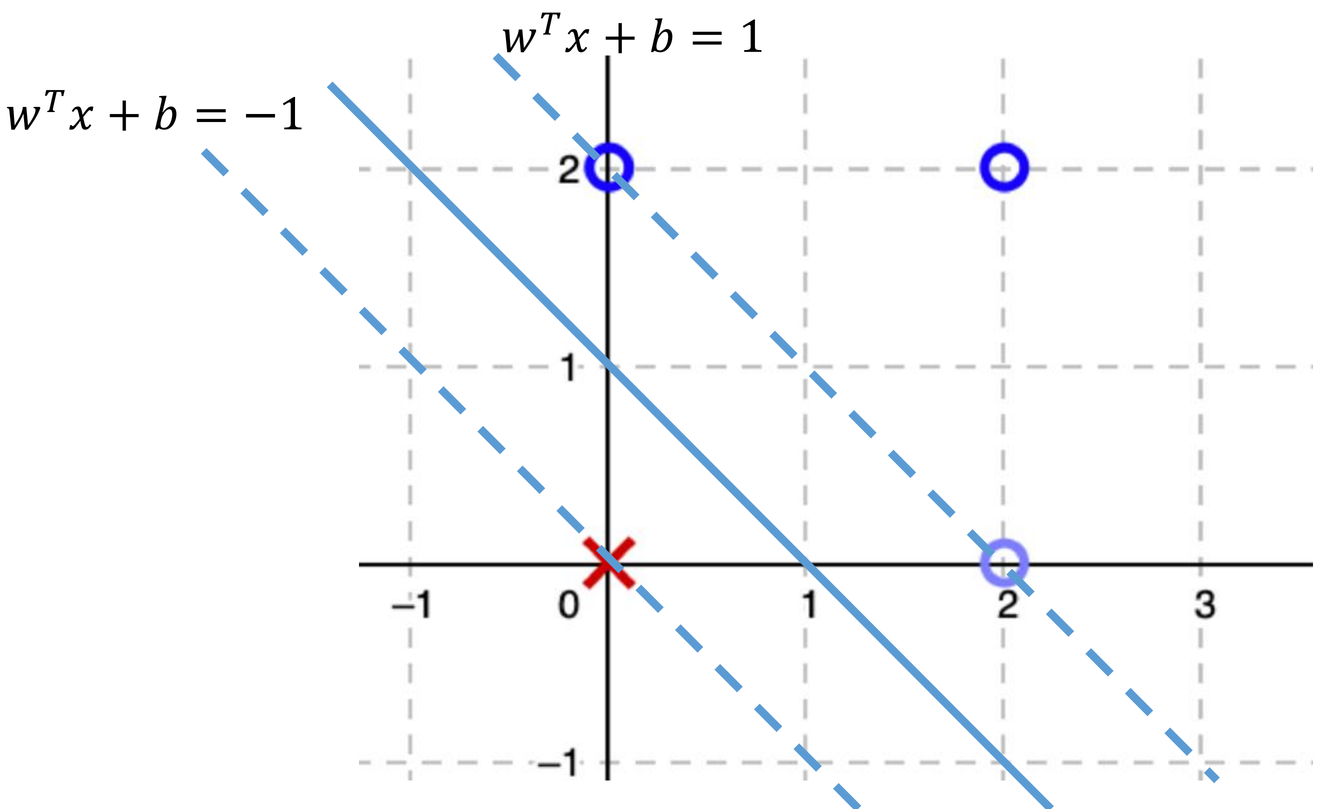
Exercise

- Given the following training data, what is the w and b for the SVM model?
- What is the margin?



Exercise

- Given the following training data, what is the w and b for the SVM model?
- What is the margin γ?



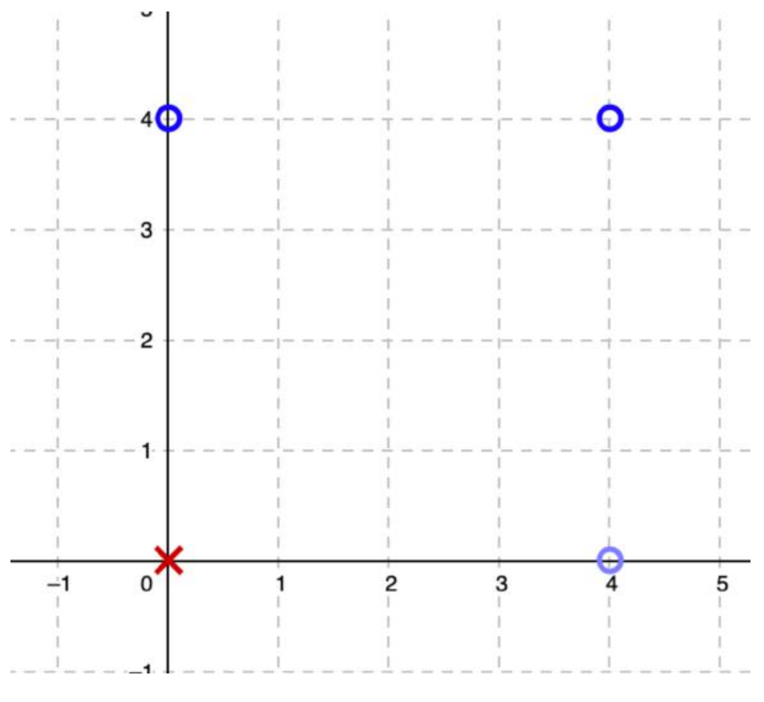
$b = -1$

$w = [1,1]$

$\gamma = \frac{\sqrt{2}}{2}$

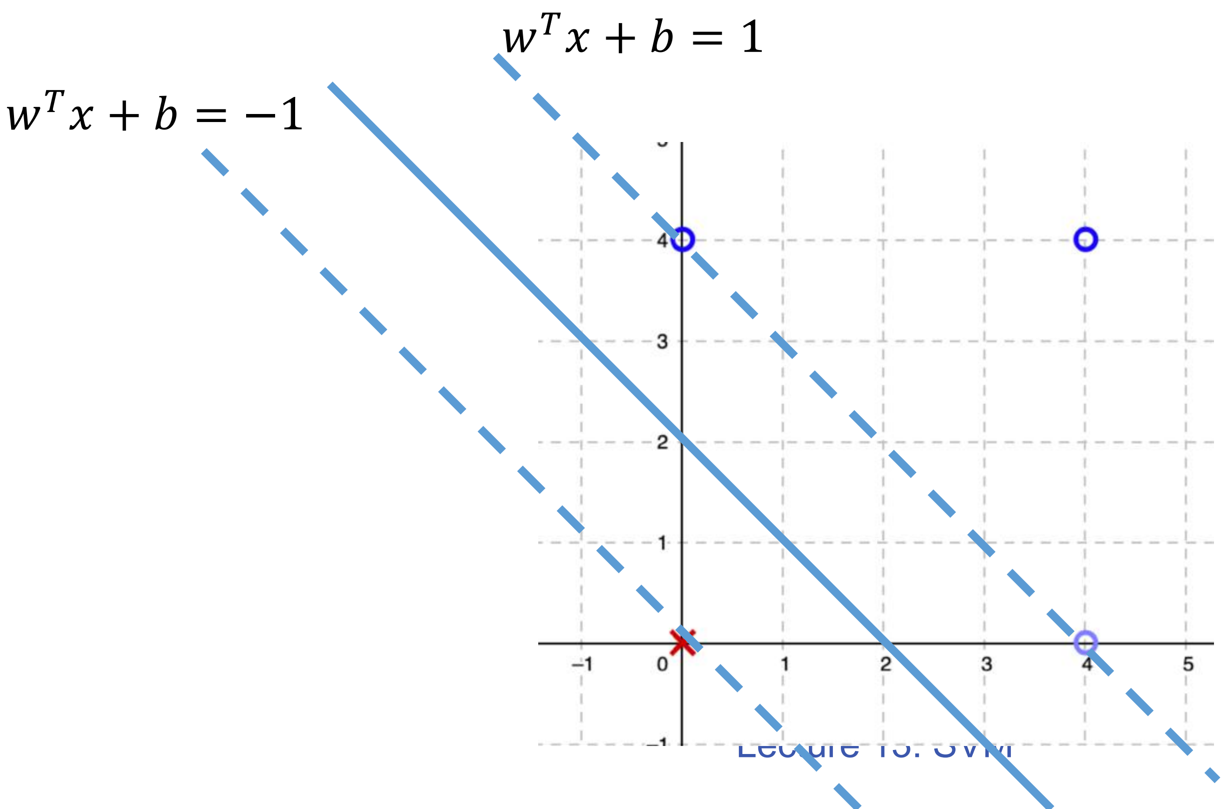
Exercise

- If we make all points two times larger, what is the w and b for the SVM model
- What is the margin?



Exercise

- If we make all points two times larger, what is the w and b for the SVM model
- What is the margin?



$b = -1$

$w = \left[\frac{1}{2}, \frac{1}{2}\right]$

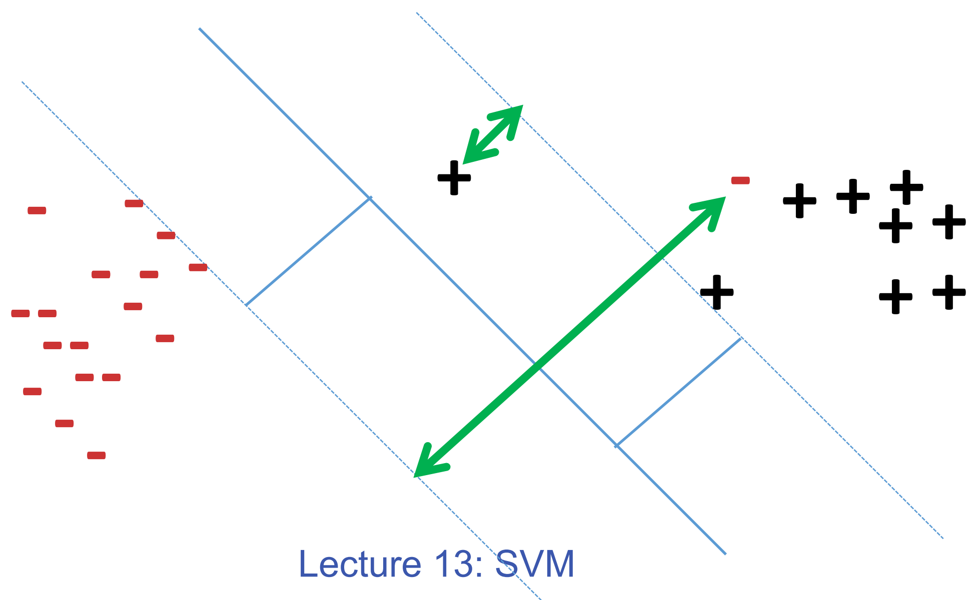
$\gamma = \sqrt{2}$

Soft SVM

$$\min_{w,b,\xi_i} \ \frac{1}{2} w^T w + C \sum_i \xi_i$$

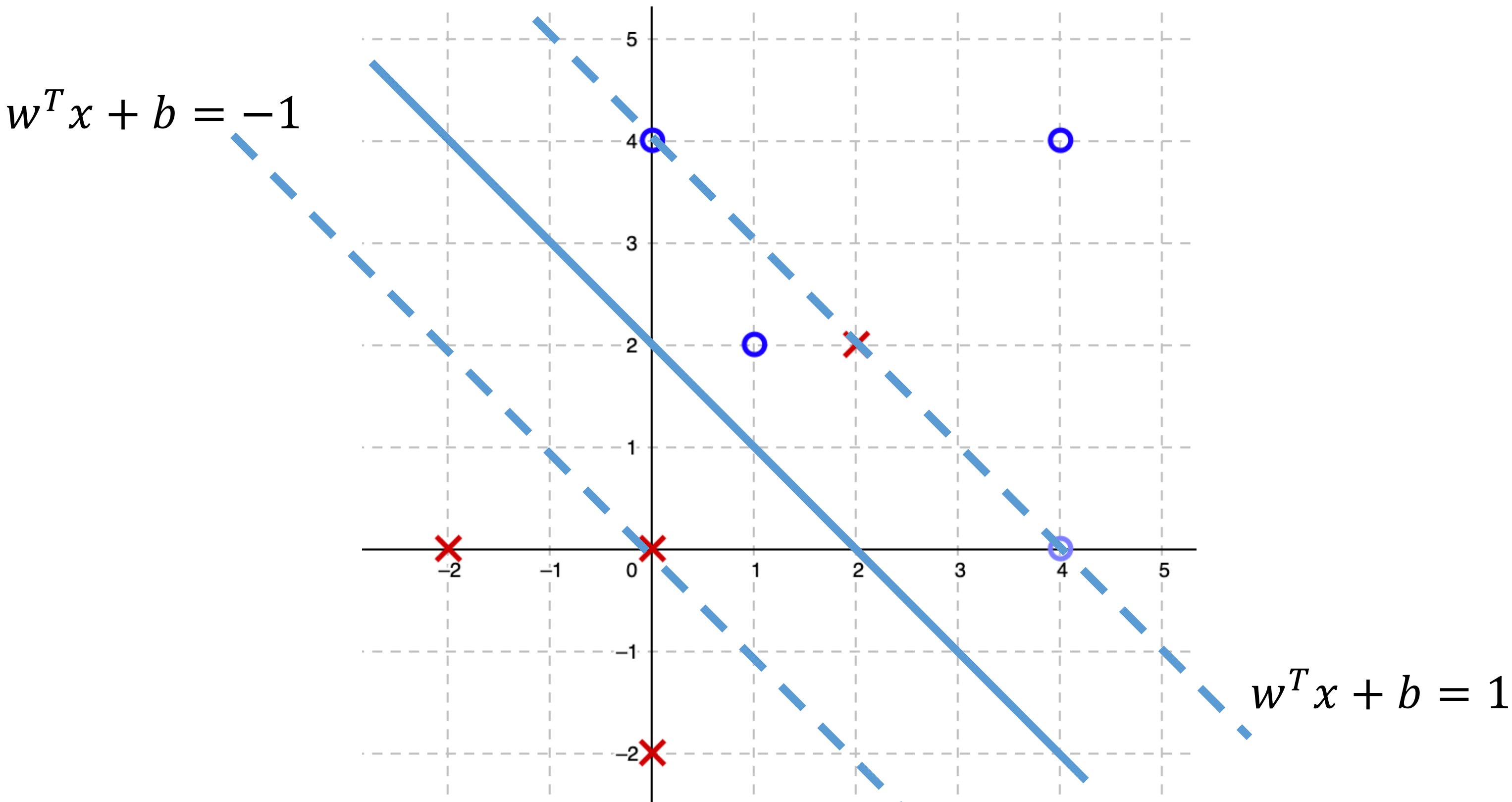
$$s.t. \ \ \forall i, \quad y_i (w^T x_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

- Introduce one *slack variable* ξ_i per example
  - And require $y_i(w^T x_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ instead



Exercise

- Given the following data and an SVM model with $w = \left[\frac{1}{2}, \frac{1}{2}\right]$, $b = -1$, what is the slack value for each data point, such that $y_i(w^T x_i + b) \geq 1 - \xi_i$?



Maximizing margin and minimizing loss

$$\min_{w,b,} \ \frac{1}{2} w^T w + C \sum_i \boxed{\max(0, 1 - y_i(w^T x_i + b))}$$

Maximize margin

Penalty for the prediction

SVM objective function

$$\min_{w,b,} \frac{1}{2} w^T w + C \sum_i \max(0, 1 - y_i(w^T x_i + b))$$

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

$$\min_{w,b,} \ \frac{1}{2} w^T w + C \sum_i \max(0, 1 - y_i(w^T x_i + b))$$

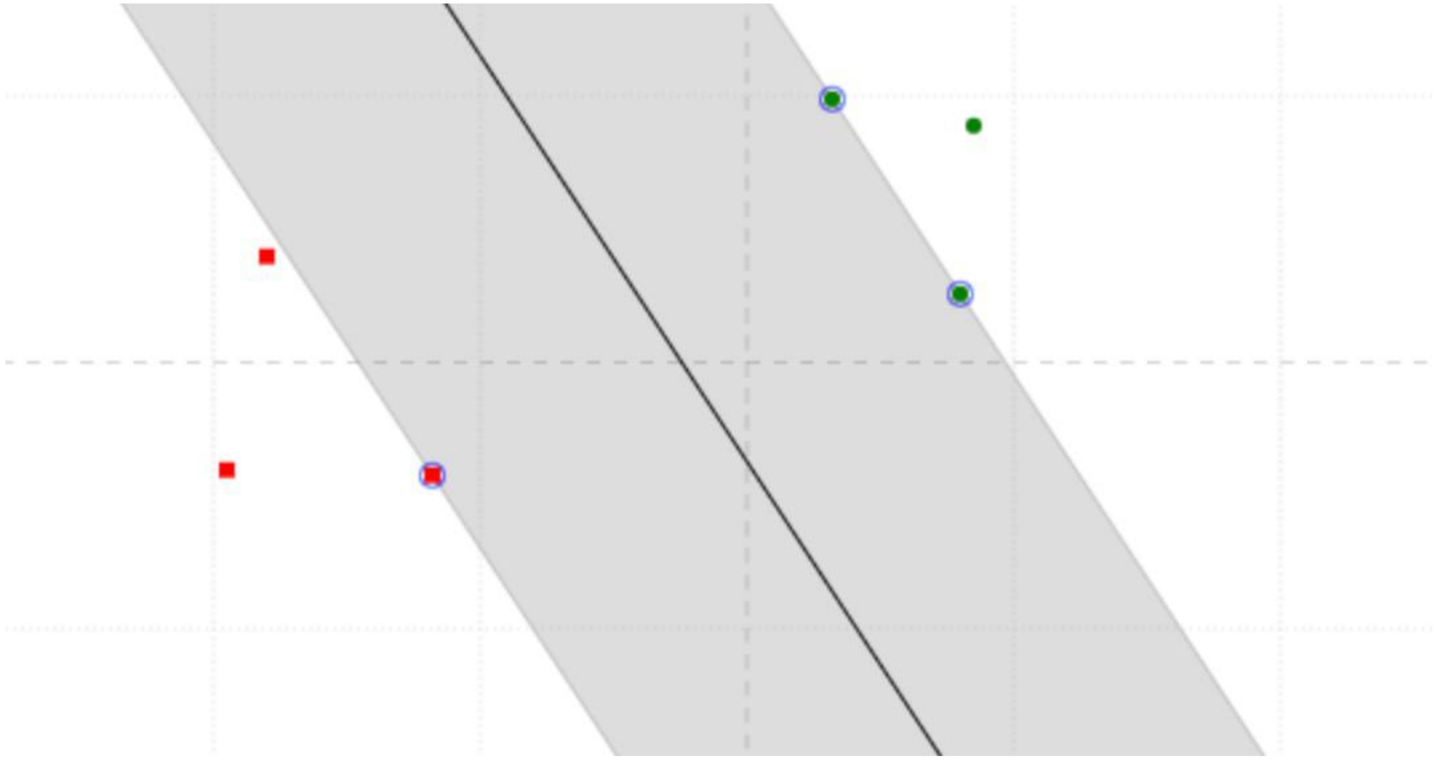
Regularization term:
- Maximize the margin
- Imposes a preference over the hypothesis space and pushes for better generalization
- Can be replaced with other regularization terms which impose other preferences

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

SVM Demo

- https://greitemann.dev/svm-demo



For simplicity, I ignore b in the following slides

SVM objective function

$$\boxed{\min_{w \in R^d} R(w) + C \sum_{(x,y)\in \hat{D}} [L(x,w,y)]}$$

$$\min_{w,b,} \frac{1}{2} w^T w + C \sum_i \max(0, 1 - y_i(w^T x_i + b))$$

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

General learning principle

Risk minimization

$$\min_{w \in R^d} \; [L(x, w, y)]$$

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]

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

$$\min_{w \in R^d} R(w) \; + \; C \sum_{(x,y)\in\widehat{D}} [L(x, w, y)]$$

General learning principle

Regularized risk minimization

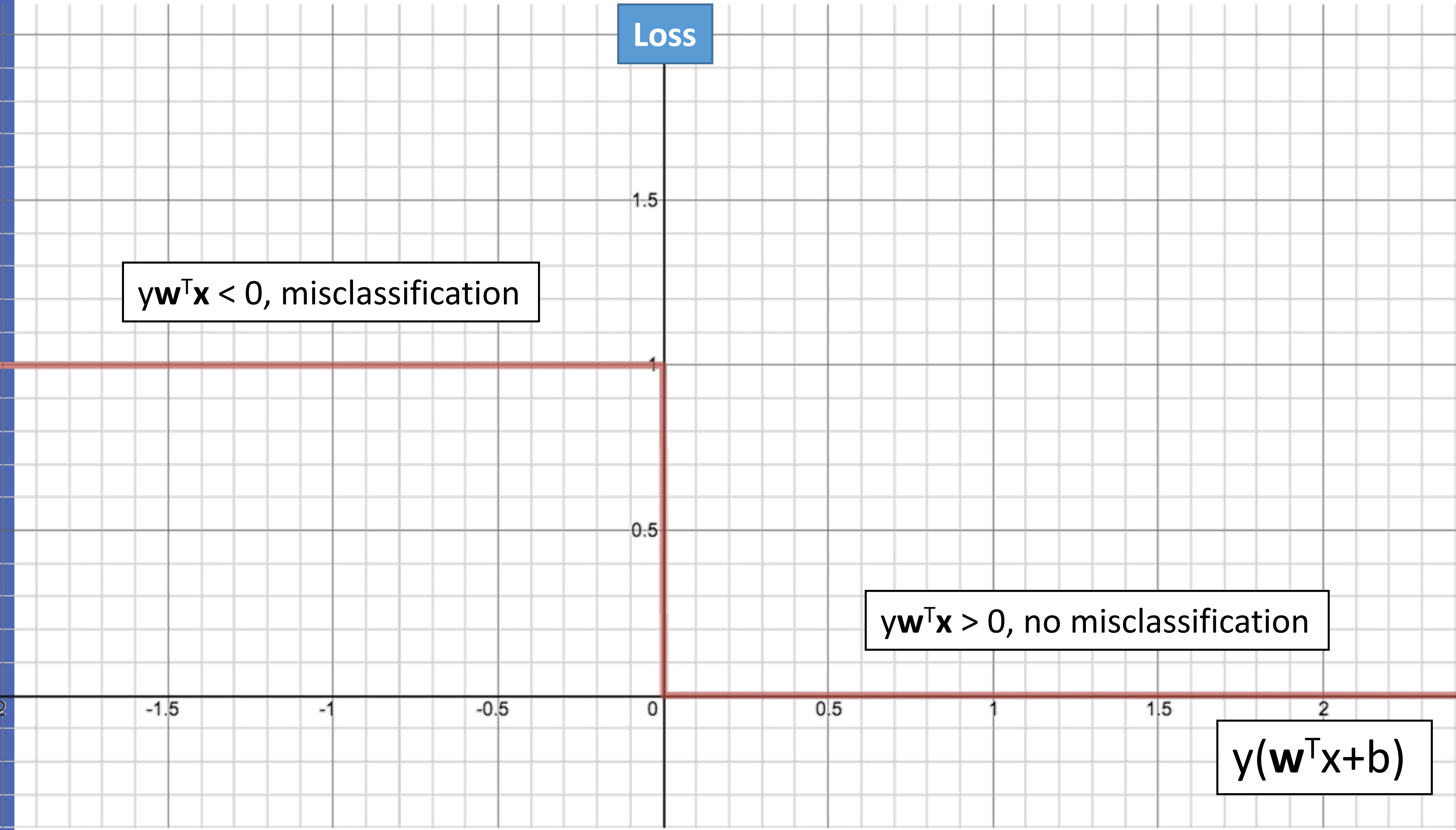
Define a regularization function that penalizes over-complex hypothesis.

Capacity control gives better generalization

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

$$\min_{w \in R^d} R(w) \; + \; C \sum_{(x,y) \in \hat{D}} [L(x, w, y)]$$

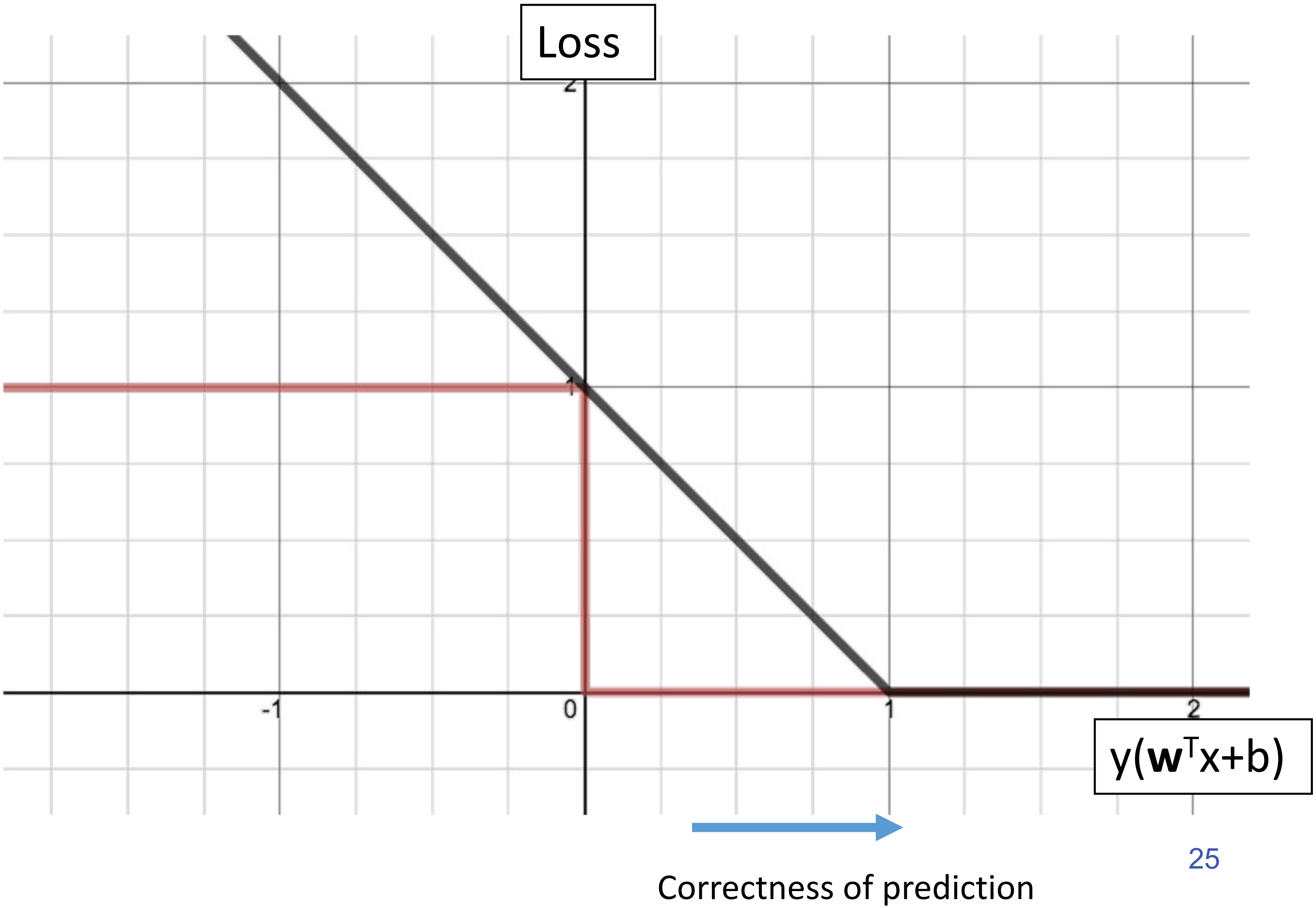
The 0-1 loss



Loss
1.5
1
0.5
-1.5
-1
-0.5
0
0.5
1
1.5
2
ywᵀx < 0, misclassification
ywᵀx > 0, no misclassification
y(wᵀx+b)

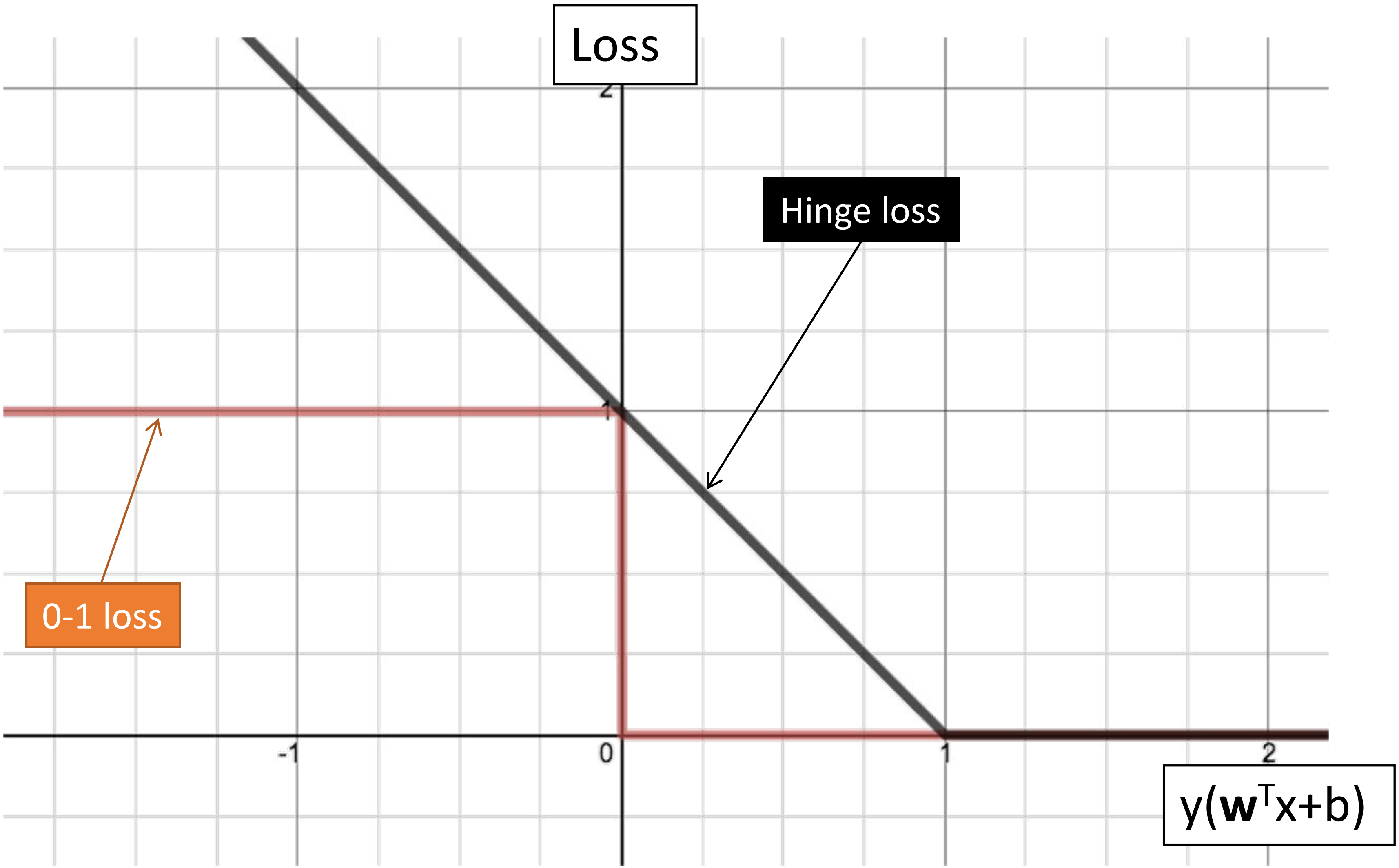
The Hinge Loss

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



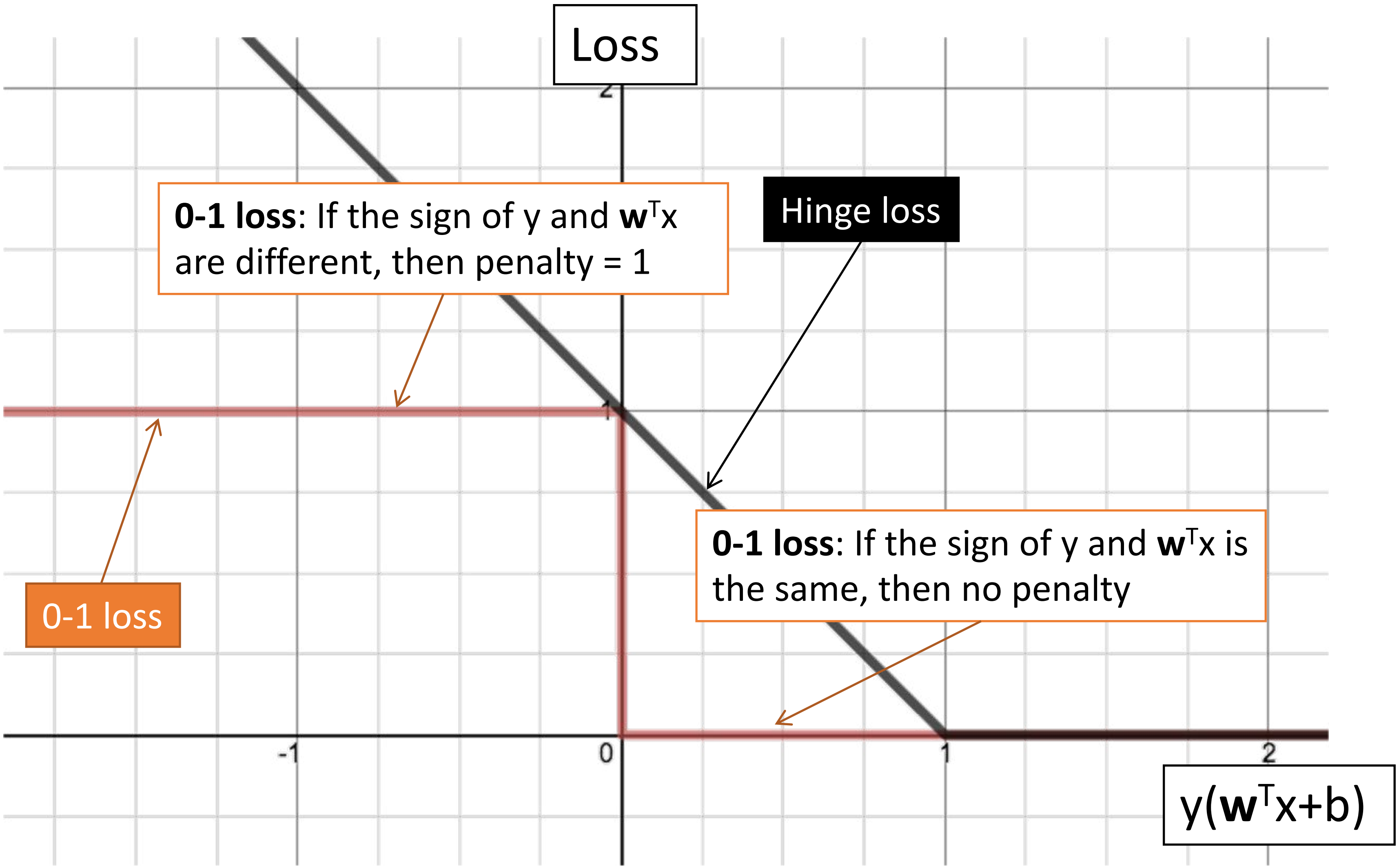
The Hinge Loss

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



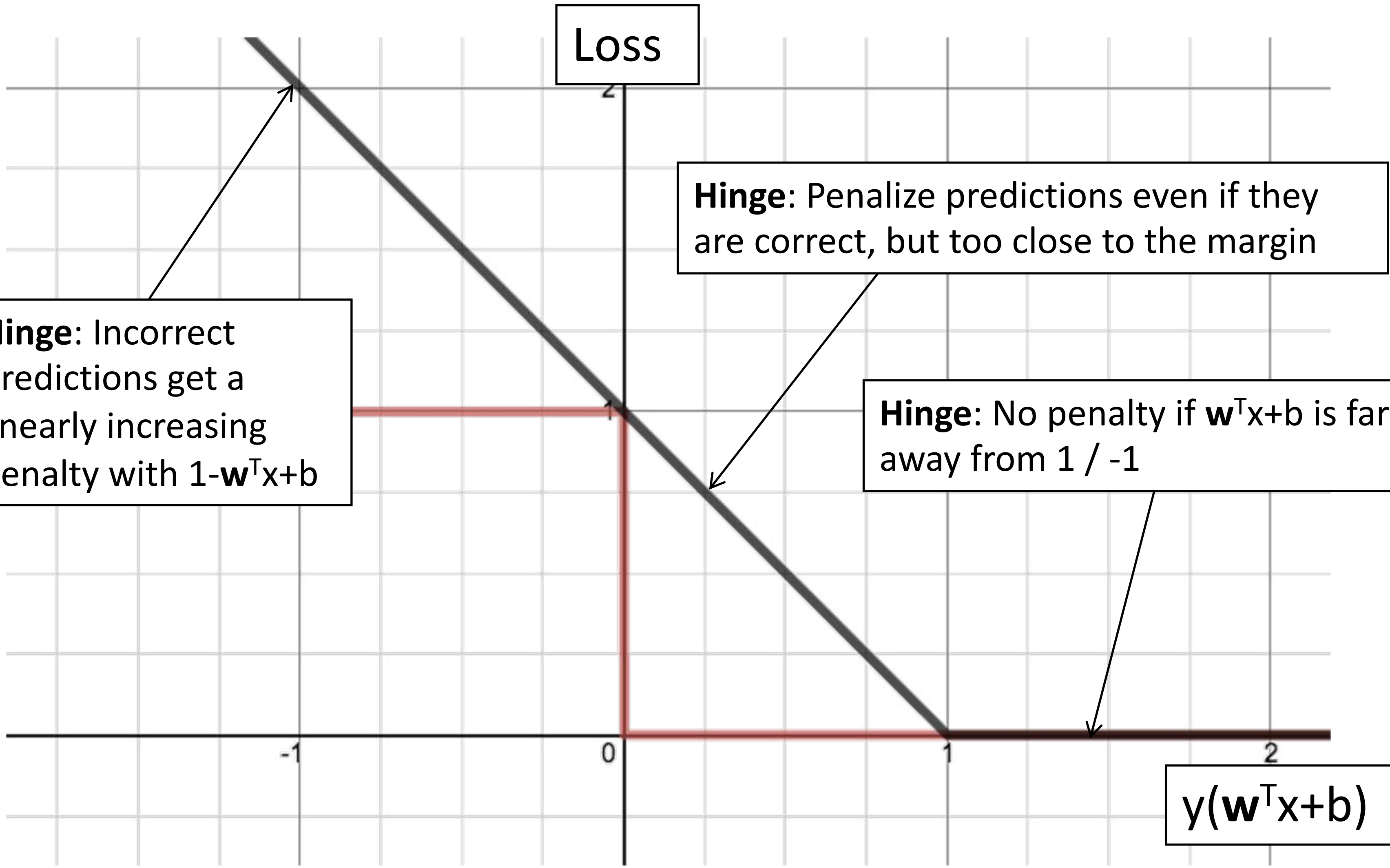
The Hinge Loss

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



The Hinge Loss

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



Hinge: Incorrect predictions get a linearly increasing penalty with 1-$\mathbf{w}^T$x+b

Hinge: Penalize predictions even if they are correct, but too close to the margin

Hinge: No penalty if $\mathbf{w}^T$x+b is far away from 1 / -1

The loss function zoo

Many loss functions

- Perceptron loss

$$L_{Perceptron}(y, \mathbf{x}, \mathbf{w}) = \max(0, -y(w^T x + b))$$

- Logistic loss (logistic regression)

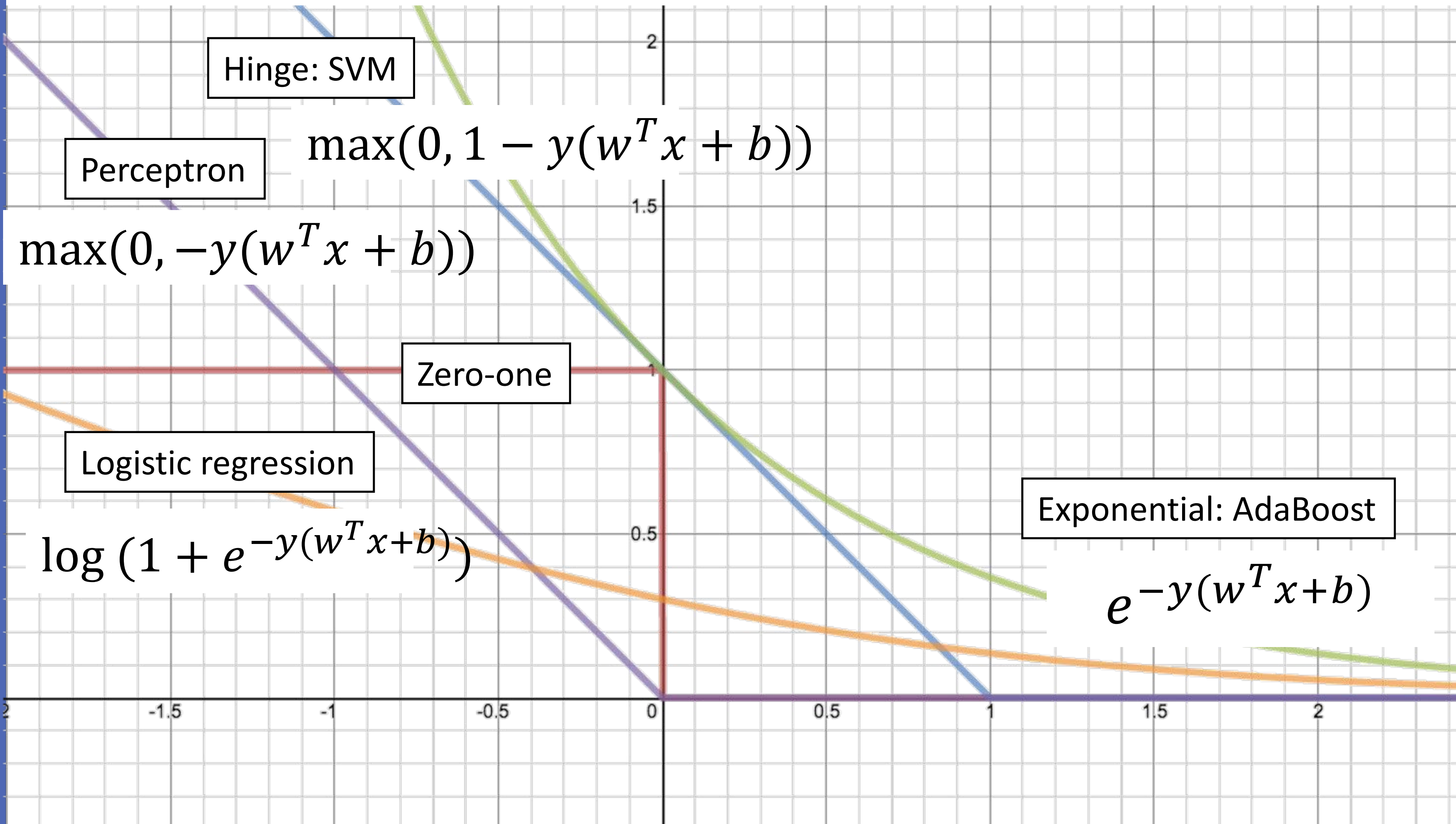
$$L_{Logistic}(y, \mathbf{x}, \mathbf{w}) = \log\left(1 + e^{-y(w^T x + b)}\right)$$

- Hinge loss (SVM)

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

The loss function zoo

The loss function zoo



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

$$\min_{w \in R^d} R(w) + C \sum_{(x,y) \in \widehat{D}} [L(x, w, y)]$$

Many choices of regularization function

- Minimizing the empirical loss with linear function

$$\min_{w \in R^d} R(w) + C \sum_{(x,y)\in\hat{D}} [L(x, w, y)]$$

- Prefer simpler model: (how?)
  - Sparse:
    $R(w) =$ #non-zero elements in w (L0 regularizer)
    $R(w) = \sum_i |w_i|$ (L1 regularizer)
  - Gaussian prior (large margin w/ hinge loss):
    $R(w) = \sum_i w_i^2 = w^T w$ (L2 regularizer)

This lecture: Support vector machines

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

Outline: Training SVM by optimization

1. Stochastic gradient descent
2. Sub-derivatives of the hinge loss
3. Stochastic sub-gradient descent for SVM
4. Comparison to perceptron

Stochastic gradient Descent

Given a training set $\mathcal{D} = \{(\boldsymbol{x}, y)\}$

1. Initialize $\boldsymbol{w} \leftarrow \mathbf{0} \in \mathbb{R}^n$
2. For epoch $1 \ldots T$:
3. For $(\boldsymbol{x}, y)$ in $\mathcal{D}$:
4. Update $w \leftarrow w - \eta \nabla_{\mathrm{w}} f(x, y)$
5. Return $\boldsymbol{w}$

$$\min \sum_{(x,y) \in D} f(x, y)$$

We will see more example later in this lecture

Hinge loss is not differentiable!

What is the derivative of the hinge loss with respect to w?

$$\frac{1}{2}w^T w + C \max(0, 1 - y_i(w^T x_i + b))$$

Sub-gradient of the SVM objective

$$J^t(w) = \frac{1}{2} w^T w + C \max(0, 1 - y_i(w^T x_i + b))$$

General strategy: First solve the max and compute the gradient for each case

$$\nabla J^t = \begin{cases} \mathbf{w} & \text{if } max(0, 1 - y_i(w^T x_i + b)) = 0 \\ \mathbf{w} - C y_i \mathbf{x}_i & \text{otherwise} \end{cases}$$

Outline: Training SVM by optimization

1. ~~Stochastic gradient descent~~
2. ~~Sub-derivatives of the hinge loss~~
3. Stochastic sub-gradient descent for SVM
4. Comparison to perceptron

Stochastic gradient Descent

Given a training set $\mathcal{D} = \{(\boldsymbol{x}, y)\}$

Initialize $\boldsymbol{w} \leftarrow \boldsymbol{0} \in \mathbb{R}^n$

For epoch $1 \dots T$:

For $(\boldsymbol{x}, y)$ in $\mathcal{D}$:

if $y(w^T x + b) < 1$

$\mathrm{w} \leftarrow w - \eta(w - Cyx)$

$b \leftarrow b + \eta C y$

else

$\mathrm{w} \leftarrow w - \eta \mathrm{w}$

Return $\boldsymbol{w}$

$\boxed{\mathrm{w} \leftarrow (1 - \eta) w + \eta C y x}$

$$\nabla J^t = \begin{cases} \mathbf{w} & \text{if} \quad max(0, 1 - y_i(w^T x_i + b)) = 0 \\ \mathbf{w} - C y_i \mathbf{x}_i & \text{otherwise} \end{cases}$$

Recap: The Perceptron Algorithm

Given a training set $\mathcal{D} = \{(\boldsymbol{x}, y)\}$

1. Initialize $\boldsymbol{w} \leftarrow \boldsymbol{0} \in \mathbb{R}^n$
2. For $(\boldsymbol{x}, y)$ in $\mathcal{D}$:
3. if $y(\boldsymbol{w}^\top \boldsymbol{x} + b) \leq 0$
4. $\boldsymbol{w} \leftarrow \boldsymbol{w} + y\boldsymbol{x}$
5. $b \leftarrow b + y$
6. Return $\boldsymbol{w}$

SVM:

if $y(w^T x + b) < 1$

$\mathrm{w} \leftarrow (1 - \eta)w + \eta C y x$

$b \leftarrow b + \eta C y$

else

$\mathrm{w} \leftarrow w - \eta \mathrm{w}$

Prediction: $y^{\text{test}} \leftarrow sgn(\boldsymbol{w}^\top \boldsymbol{x}^{\text{test}})$

Footnote: For some algorithms it is mathematically easier to represent False as -1, and at other times, as 0. For the Perceptron algorithm, treat -1 as false and +1 as true.

Perceptron vs. SVM

- Perceptron: Stochastic sub-gradient descent for a different loss
  - No regularization though

$$\boxed{L_{perceptron}(y, x, w) = \max(0, -y_i(w^T x_i + b))}$$

- SVM optimizes the hinge loss
  - With regularization

$$\boxed{L_{Hinge}(y, x, w) = \max(0, 1 - y_i(w^T x_i + b))}$$

This lecture: Support vector machines

- Training by maximizing margin

- The SVM objective

- Solving the SVM optimization problem

- Support vectors, duals and kernels

Maximizing margin and minimizing loss

$$\min_{w,b} \; \frac{1}{2}w^T w + C\sum_i \boxed{\max(0, 1 - y_i(w^T x_i + b))}$$

Maximize margin

Penalty for the prediction

There are 3 cases

- Example is correctly classified and is <u>*outside*</u> the margin: penalty = 0
- Example is incorrectly classified: penalty = $1 - y_i(w^T x_i + b)$
- Example is correctly classified but within the margin: penalty = $1 - y_i(w^T x_i + b)$

$$\boxed{L_{Hinge}(y, x, w) = \max(0, 1 - y_i(w^T x_i + b))}$$

This is the hinge loss function

SVM: Primal and dual

The SVM objective

$$\min_{w,b,\xi_i} \frac{1}{2} w^T w + C \sum_i \xi_i$$

$$s.t. \;\; \forall i, \quad y_i (w^T x_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

This is called the *primal form* of the objective

This can be converted to its *dual form*, which will let us prove a very useful property

Support vector machines

Let **w** be the minimizer of the SVM problem for some dataset with m examples: $\{(\mathbf{x}_i, y_i)\}$

Then, for i = 1…m, there exist $\alpha_i \geq 0$ such that the optimum w can be written as

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$
$$C \geq \alpha_i \geq 0$$

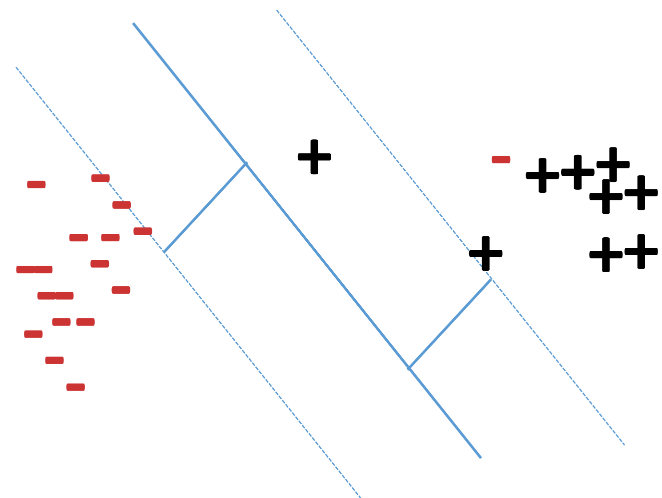
See the proof at https://www.csie.ntu.edu.tw/~cjlin/papers/bottou_lin.pdf

Support vector machines

Let **w** be the minimizer of the SVM problem for some dataset with m examples: $\{(\mathbf{x}_i, y_i)\}$

Then, for i = 1…m, there exist $\alpha_i \geq 0$ such that the optimum w can be written as

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$



Furthermore,

$$y_i(w^T x_i + b) > 1 \Rightarrow \alpha_i = 0$$

$$y_i(w^T x_i + b) < 1 \Rightarrow \alpha_i = C$$

$$y_i(w^T x_i + b) = 1 \Rightarrow 0 \leq \alpha_i \leq C$$

Support vector machines

Let **w** be the minimizer of the SVM problem for some dataset with m examples: {($\mathbf{x}_i$, y_i)}

Then, for i = 1…m, there exist $\alpha_i \geq 0$ such that the optimum w can be written as

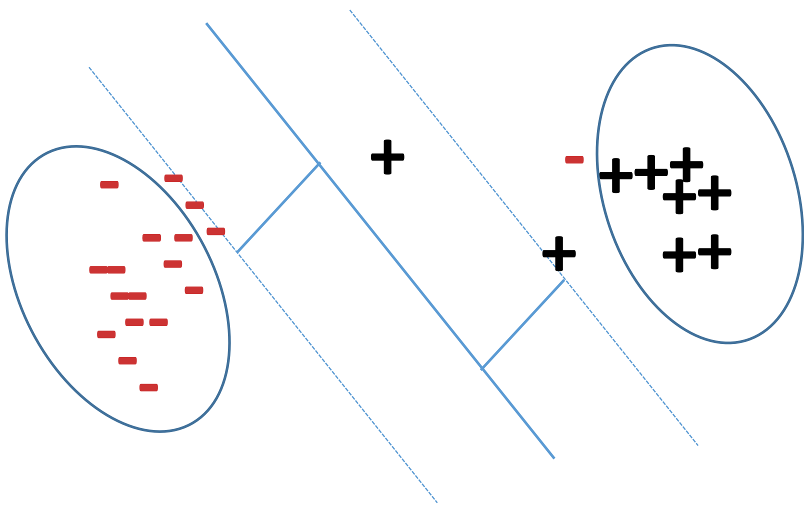
$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

Furthermore,

$$\boxed{y_i(w^T x_i + b) > 1 \Rightarrow \alpha_i = 0}$$

$$y_i(w^T x_i + b) < 1 \Rightarrow \alpha_i = C$$

$$y_i(w^T x_i + b) = 1 \Rightarrow 0 \leq \alpha_i \leq C$$



Support vector machines

Let **w** be the minimizer of the SVM problem for some dataset with m examples: $\{(\mathbf{x}_i, y_i)\}$

Then, for i = 1…m, there exist $\alpha_i \geq 0$ such that the optimum w can be written as

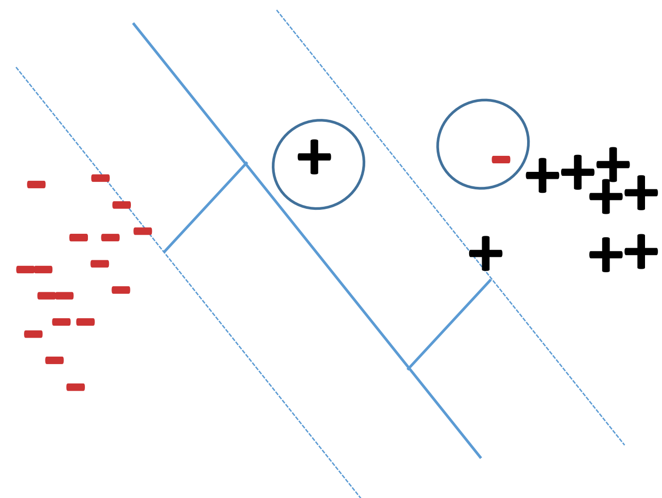
$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

Furthermore,

$$y_i(w^T x_i + b) > 1 \Rightarrow \alpha_i = 0$$

$$\boxed{y_i(w^T x_i + b) < 1 \Rightarrow \alpha_i = C}$$

$$y_i(w^T x_i + b) = 1 \Rightarrow 0 \leq \alpha_i \leq C$$



Support vector machines

Let **w** be the minimizer of the SVM problem for some dataset with m examples: $\{(\mathbf{x}_i, y_i)\}$

Then, for i = 1…m, there exist $\alpha_i \geq 0$ such that the optimum w can be written as

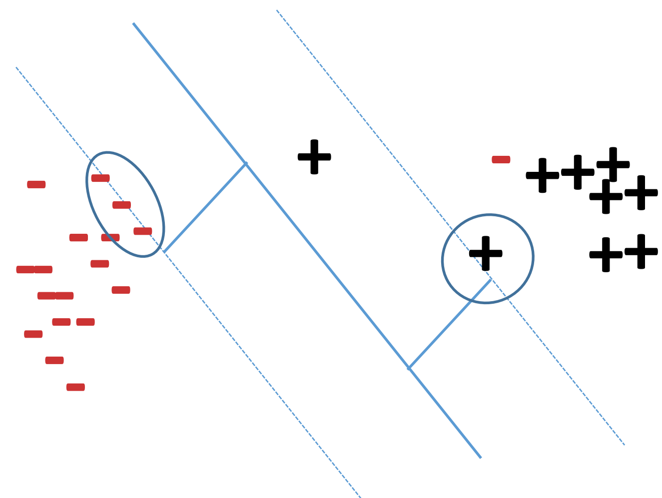
$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

Furthermore,

$$y_i(w^T x_i + b) > 1 \Rightarrow \alpha_i = 0$$

$$y_i(w^T x_i + b) < 1 \Rightarrow \alpha_i = C$$

$$\boxed{y_i(w^T x_i + b) = 1 \Rightarrow 0 \leq \alpha_i \leq C}$$



Support vectors

The weight vector is completely defined by training examples whose α_is are not zero

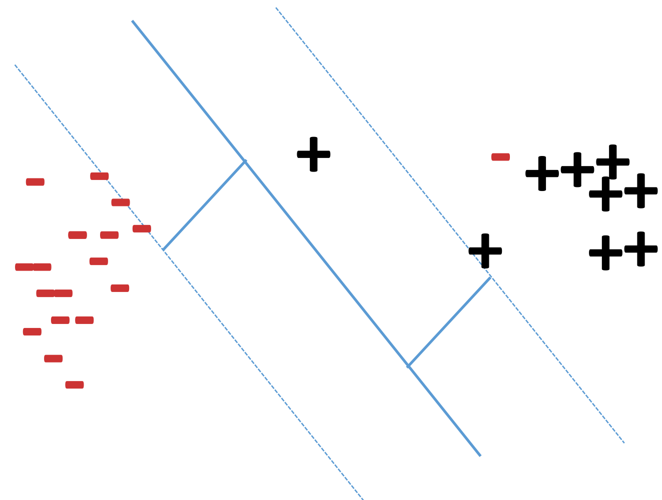
$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

These examples are called the *support vectors*

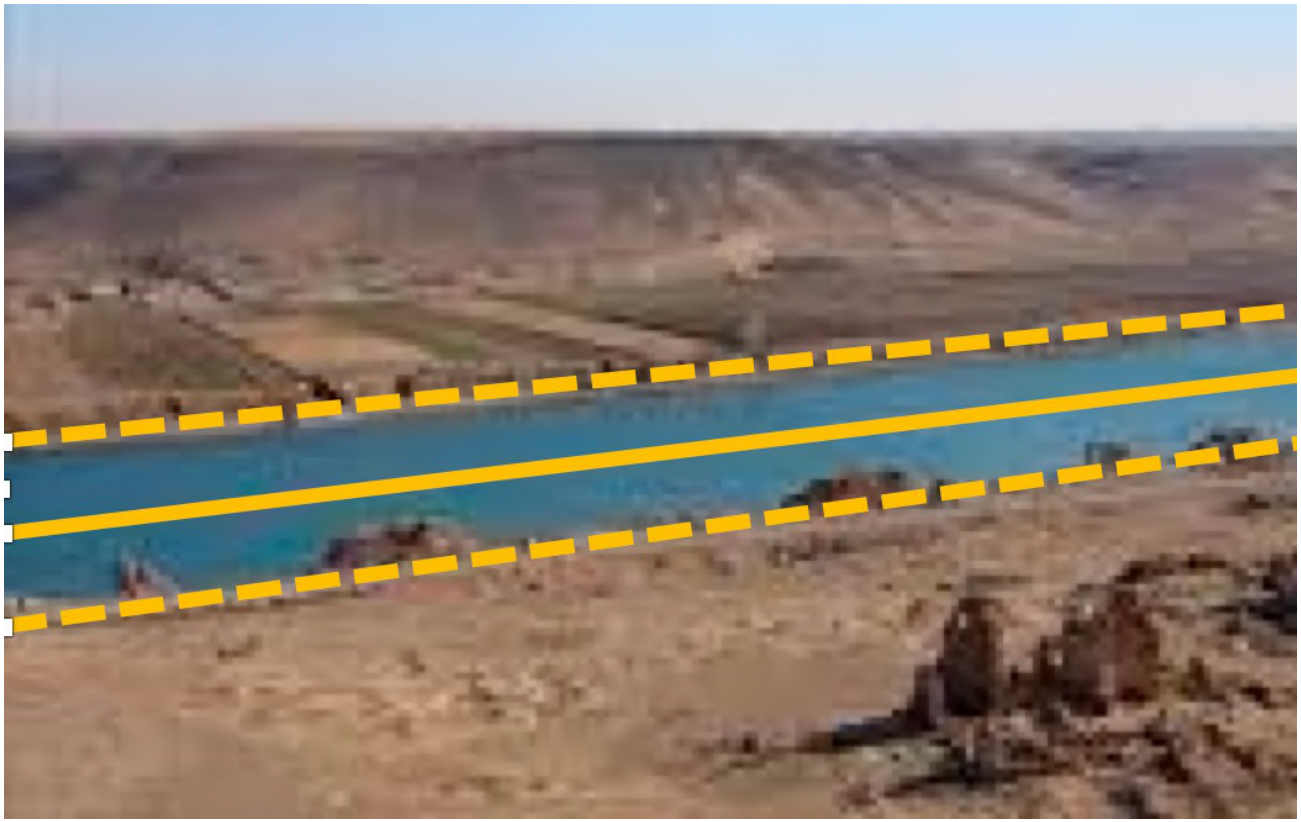
$$y_i(w^T x_i + b) > 1 \Rightarrow \alpha_i = 0$$

$$y_i(w^T x_i + b) < 1 \Rightarrow \alpha_i = C$$

$$y_i(w^T x_i + b) = 1 \Rightarrow 0 \leq \alpha_i \leq C$$



Why it called support vector machines?



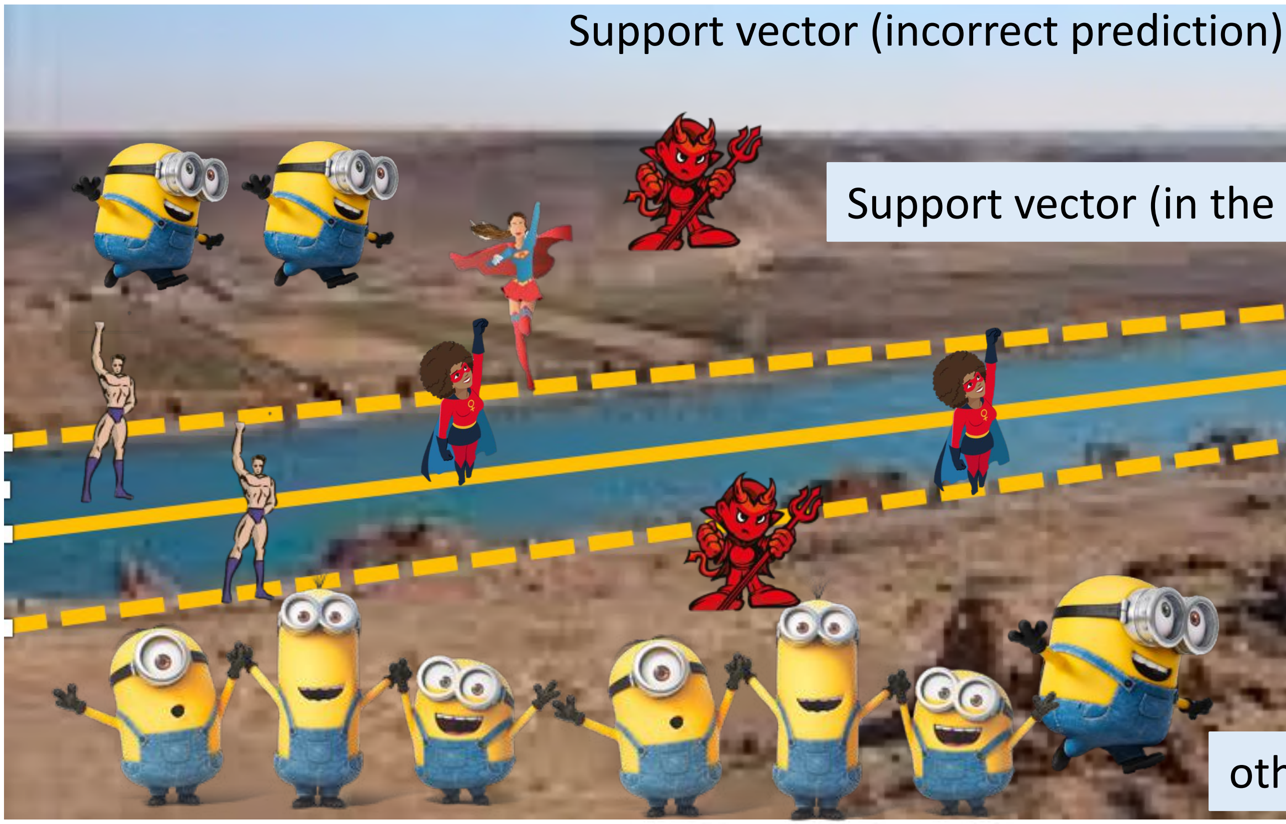
margin (upper)

Decision boundary

margin (lower)

Why it called support vector machines?

ohter vector (correct samples outside margin)



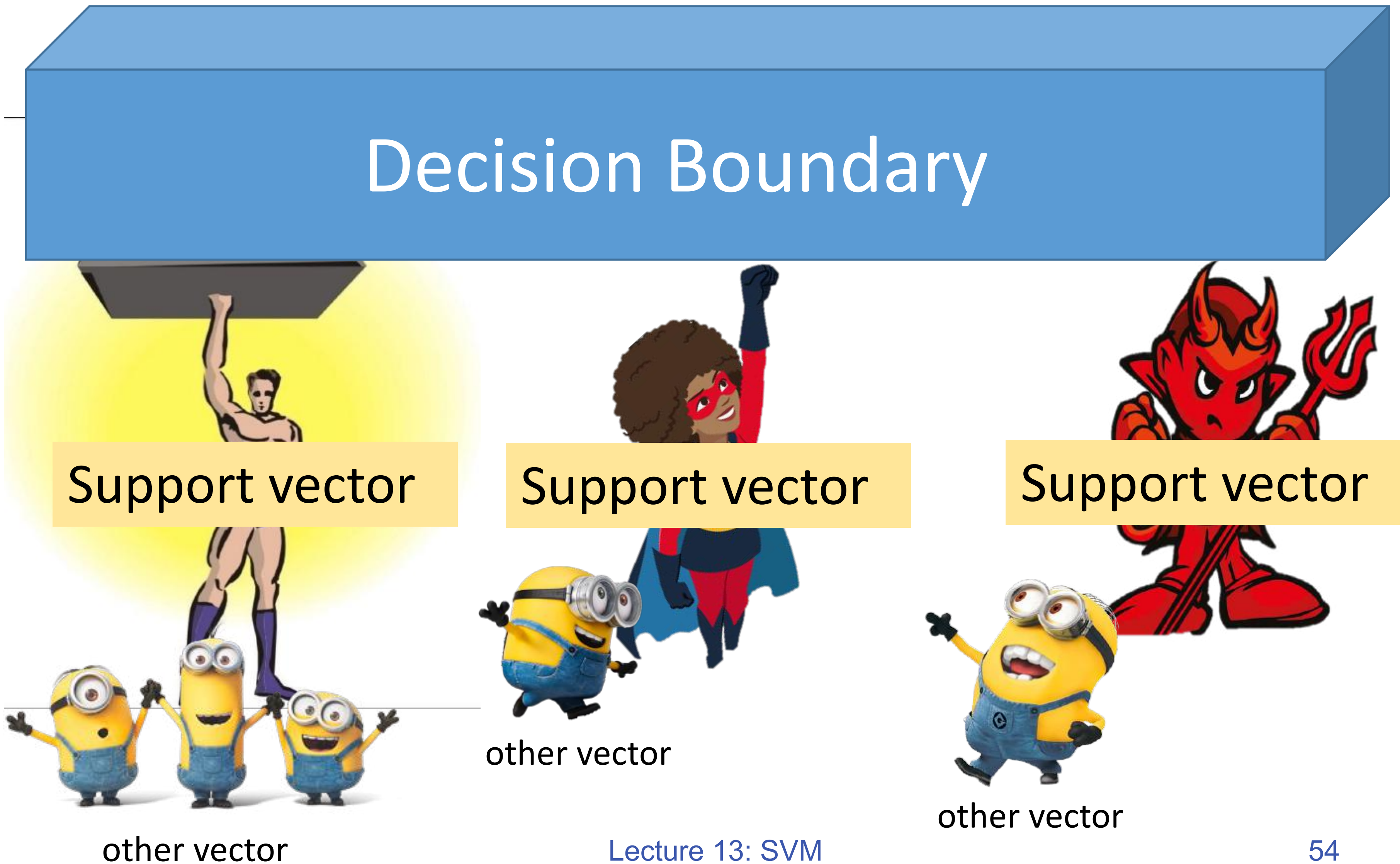
margin (upper)

Decision boundary

margin (lower)

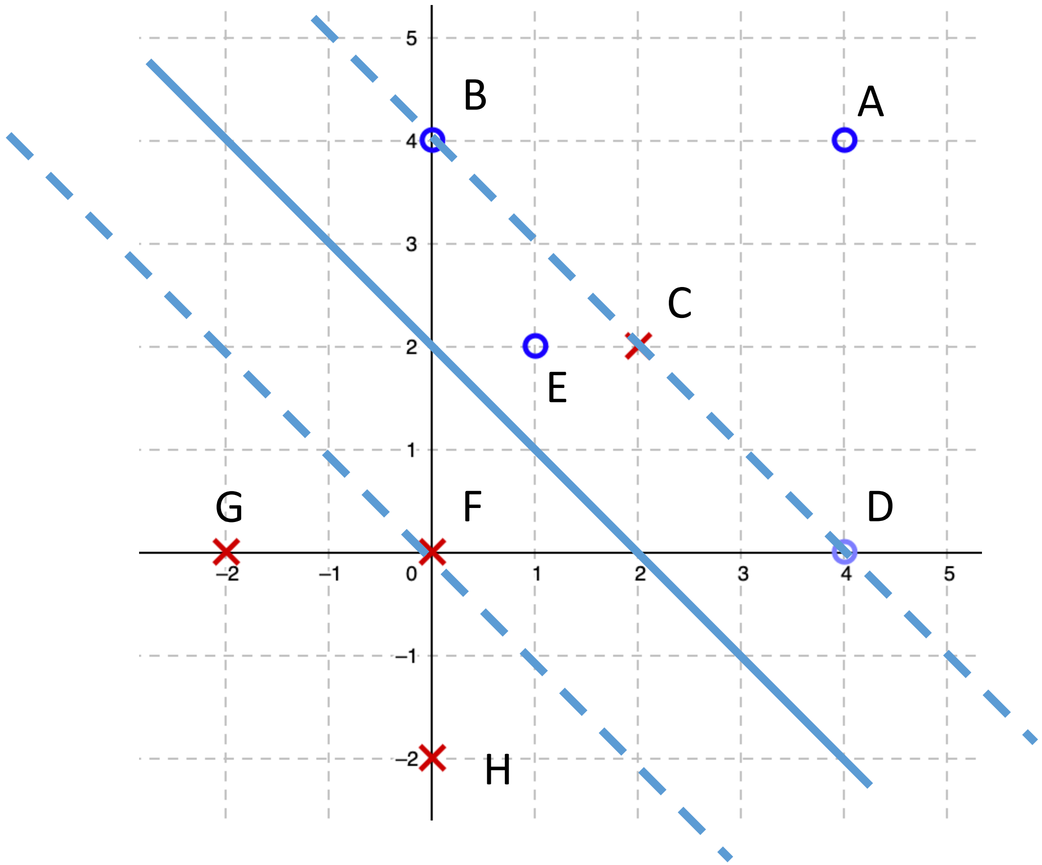
Support vector (in the margin)

Support vector (incorrect prediction)


Decision Boundary
Support vector
Support vector
Support vector
other vector
other vector
other vector

Exercise

- Which points are support vectors?



Recap

- SVM formulation, large margin

$$\min_{w,b,\xi_i} \ \frac{1}{2}w^T w + C\sum_i \xi_i$$

$$s.t.\ \ \forall i, \quad y_i(w^T x_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

- Regularization and loss
- Subgradient and dual problem