Machine Learning Basics
Support Vector Machine

HKUST MSBD 6000B

Instructor: Yu Zhang

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \boldsymbol{\mathcal{H}}$ that minimizes $\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f, x_i, y_i)$
- s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

Machine learning

- Collect data and extract features
- Build model: choose hypothesis class $\mathcal{H}$ and loss function l
- Optimization: minimize the empirical loss

Loss function

- l_2 loss: linear regression
- Cross-entropy: logistic regression
- Hinge loss: Perceptron

- General principle: maximum likelihood estimation (MLE)
  - l_2 loss: corresponds to Normal distribution
  - logistic regression: corresponds to sigmoid conditional distribution

Optimization

- Linear regression: closed form solution
- Logistic regression: gradient descent
- Perceptron: stochastic gradient descent

- General principle: local improvement
  - SGD: Perceptron; can also be applied to linear regression/logistic regression

Principle for hypothesis class?

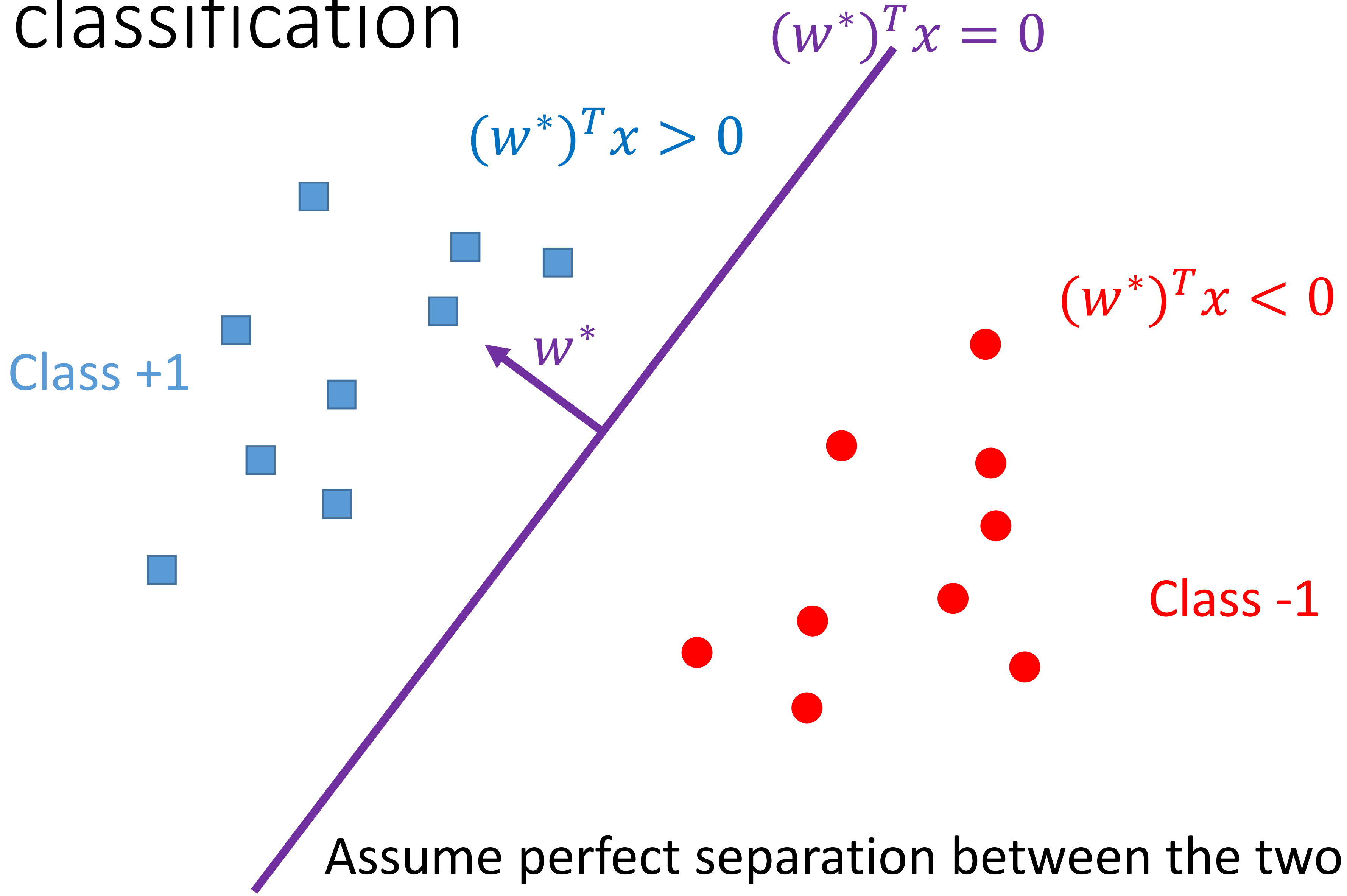
- Yes, there exists a general principle (at least philosophically)

- Different names/faces/connections
  - Occam's razor
  - VC dimension theory
  - Minimum description length
  - Tradeoff between Bias and variance; uniform convergence
  - The curse of dimensionality

- Running example: Support Vector Machine (SVM)

Motivation

Linear classification



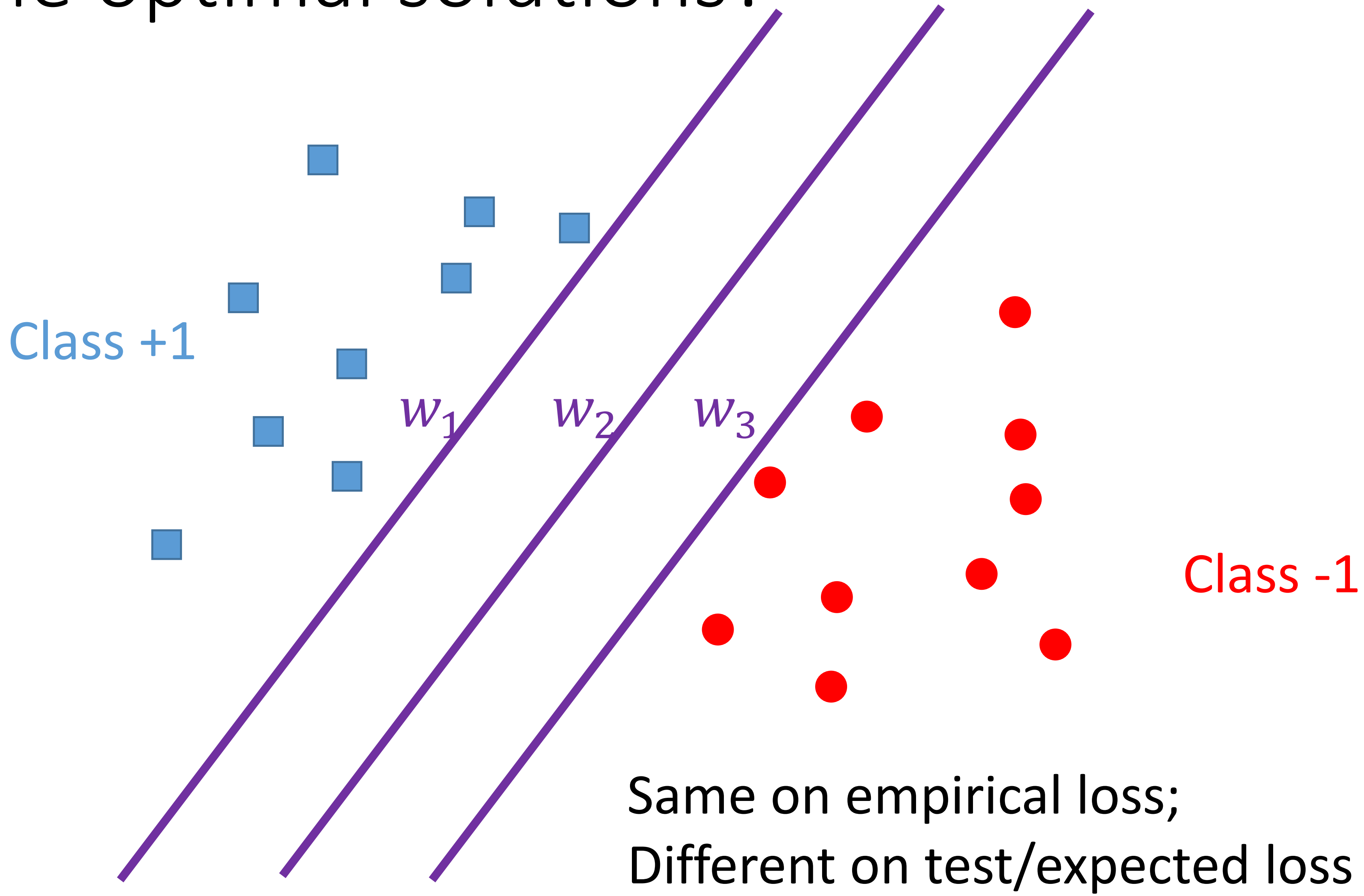
Assume perfect separation between the two classes

Attempt

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Hypothesis $y = \text{sign}(f_w(x)) = \text{sign}(w^T x)$
  - $y = +1$ if $w^T x > 0$
  - $y = -1$ if $w^T x < 0$

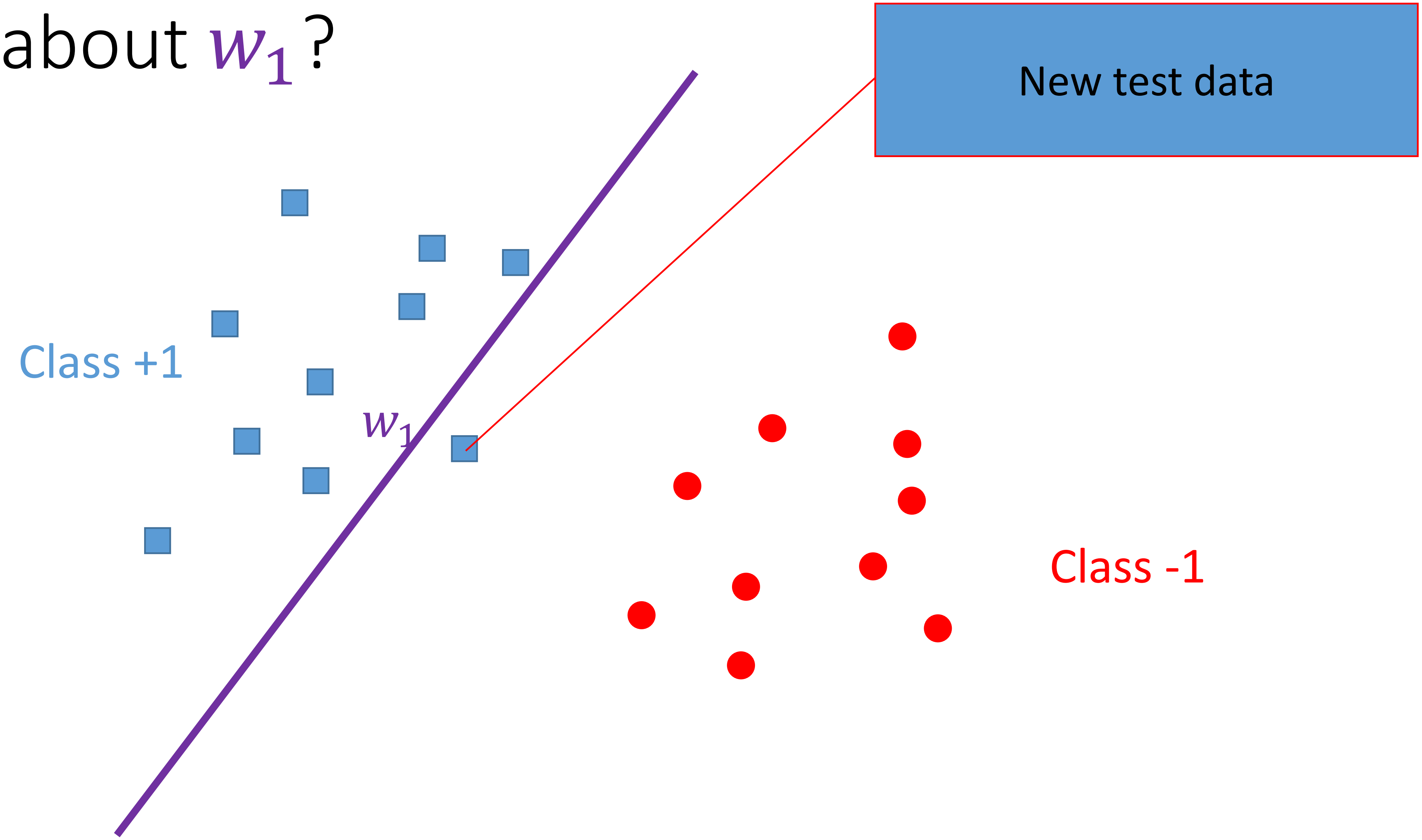
- Let's assume that we can optimize to find w

Multiple optimal solutions?

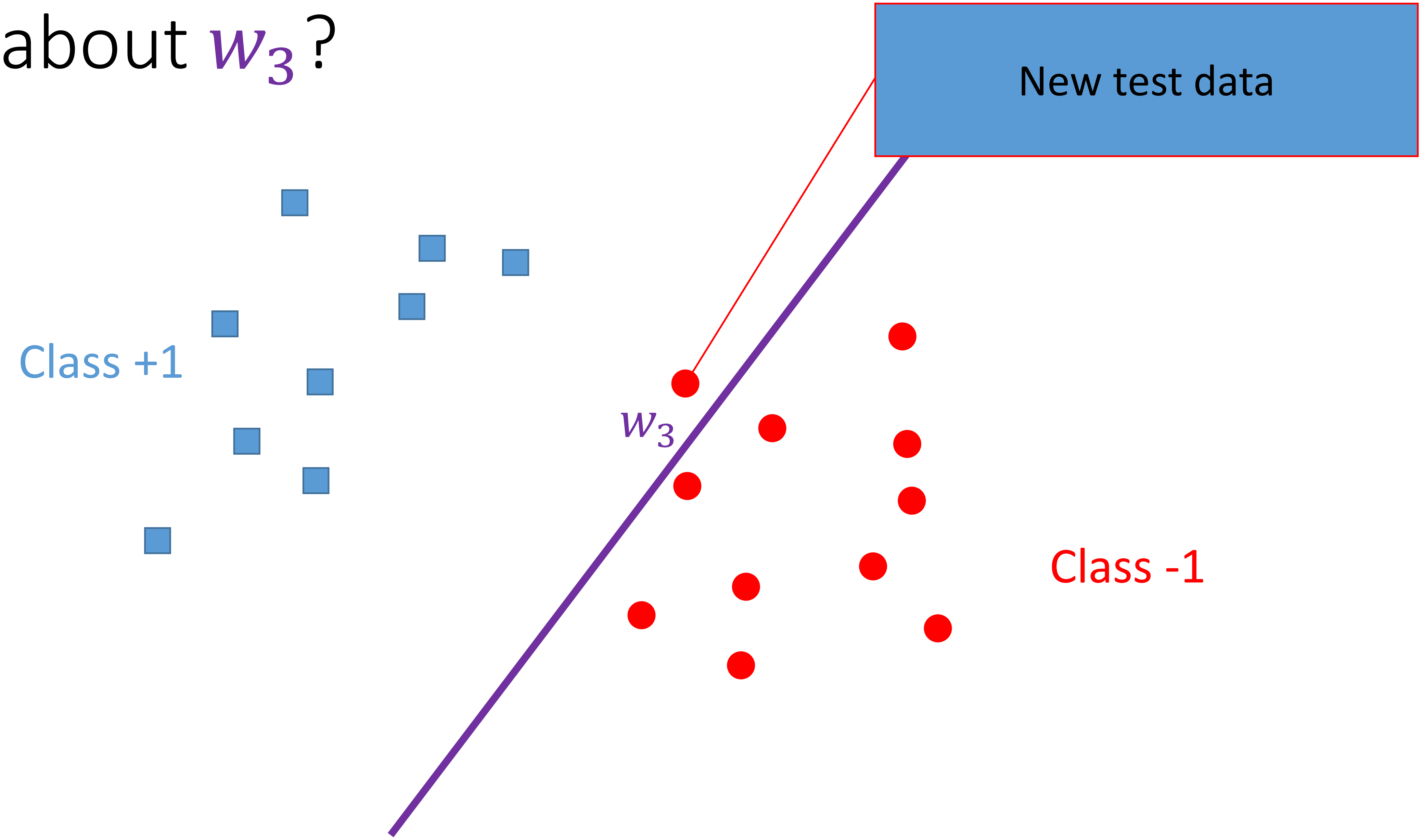


Same on empirical loss;
Different on test/expected loss

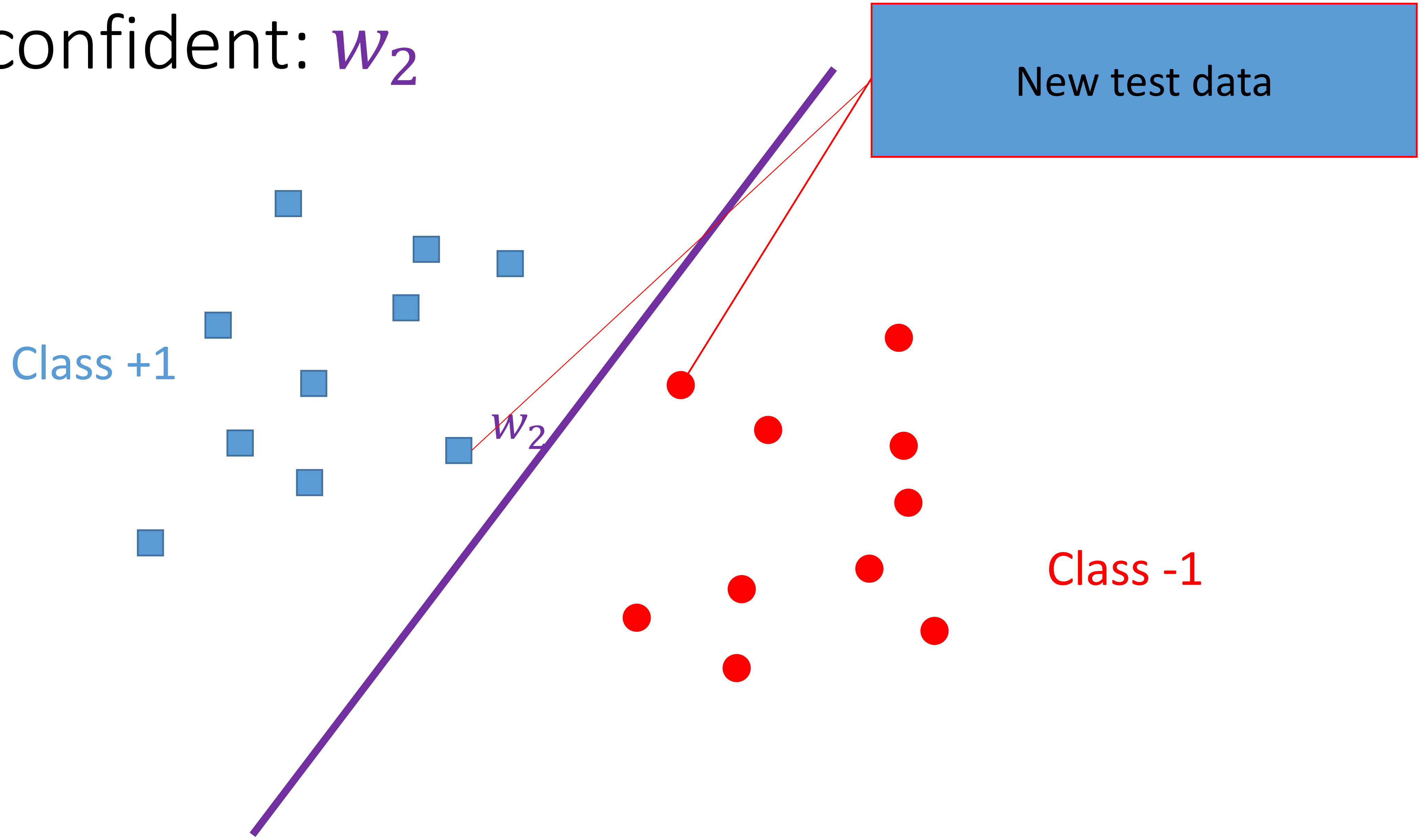
What about w_1?



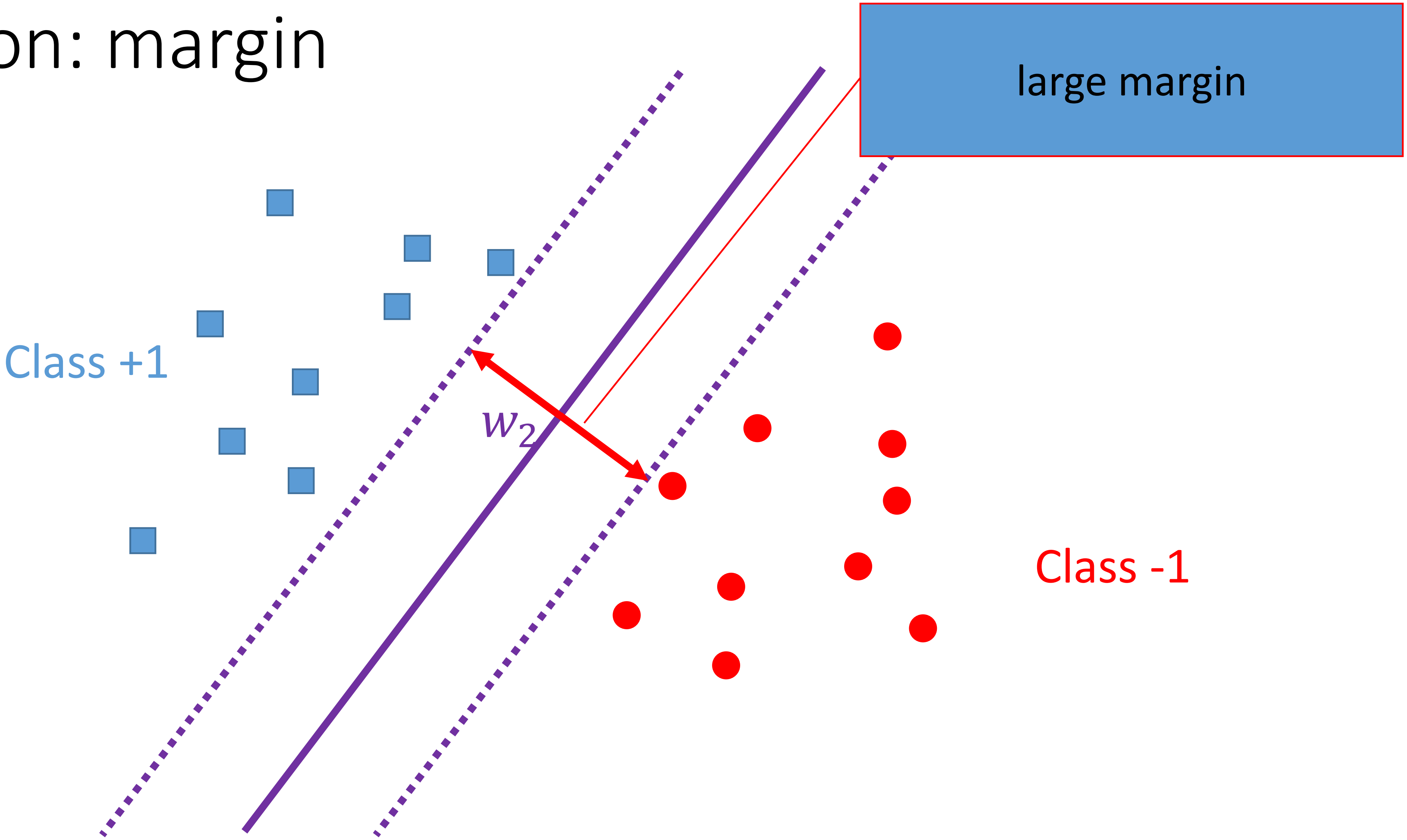
What about w_3?



Most confident: w_2



Intuition: margin



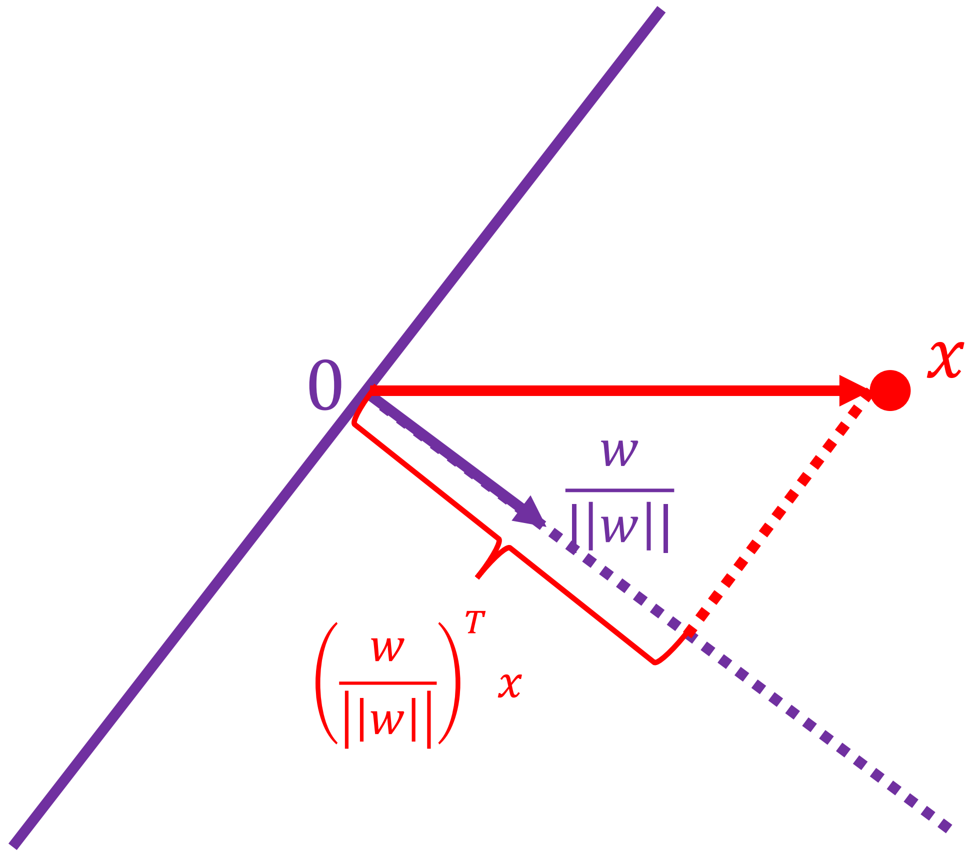
Margin

Margin

- Lemma 1: x has distance $\frac{|f_w(x)|}{||w||}$ to the hyperplane $f_w(x) = w^T x = 0$

Proof:

- w is orthogonal to the hyperplane
- The unit direction is $\frac{w}{||w||}$
- The projection of x is $\left(\frac{w}{||w||}\right)^T x = \frac{f_w(x)}{||w||}$



Margin: with bias

- Claim 1: w is orthogonal to the hyperplane $f_{w,b}(x) = w^T x + b = 0$

Proof:

- pick any x_1 and x_2 on the hyperplane
- $w^T x_1 + b = 0$
- $w^T x_2 + b = 0$

- So $w^T (x_1 - x_2) = 0$

Margin: with bias

- Claim 2: 0 has distance $\frac{-b}{||w||}$ to the hyperplane $w^T x + b = 0$

Proof:

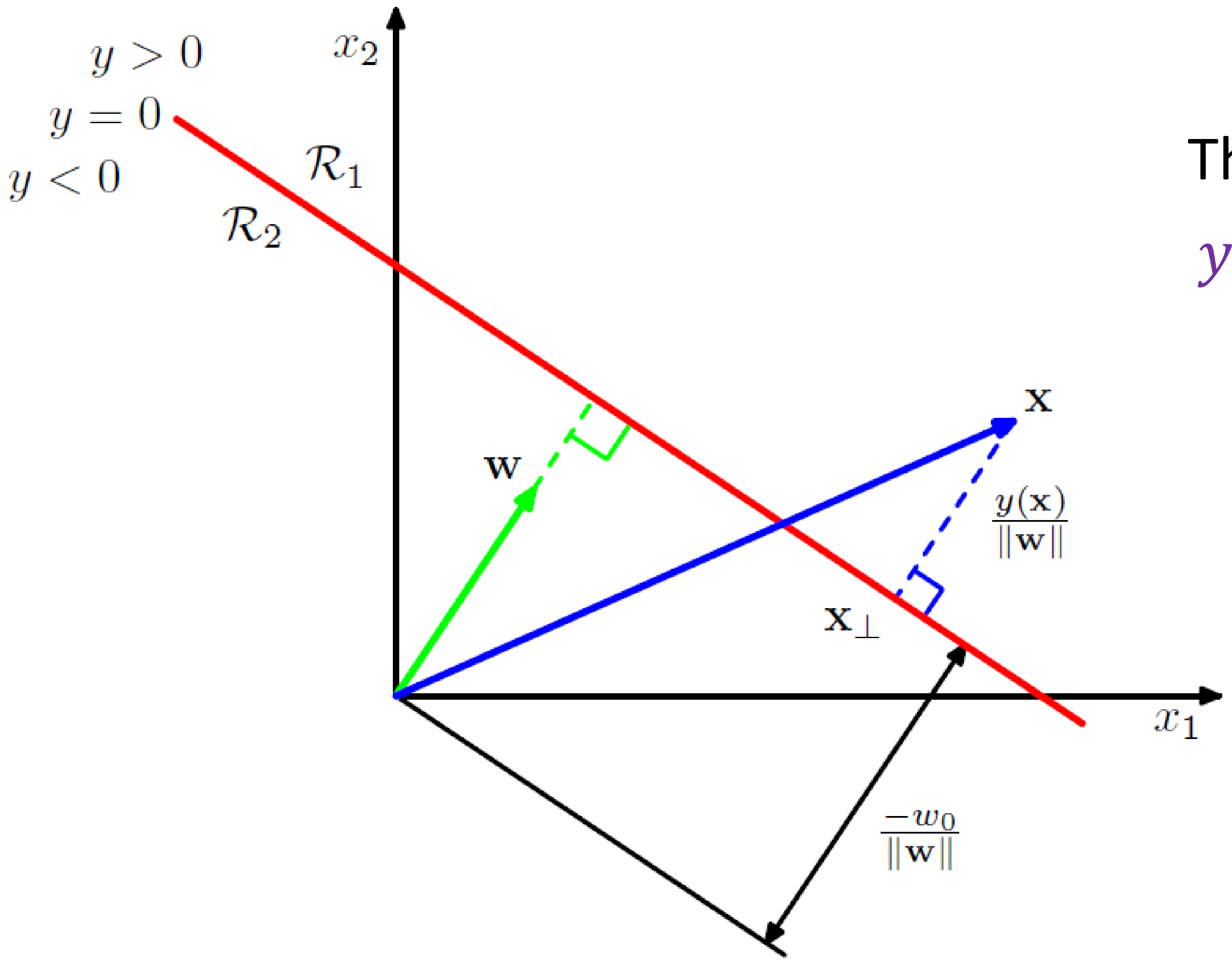
- pick any x_1 the hyperplane
- Project x_1 to the unit direction $\frac{w}{||w||}$ to get the distance
- $\left(\frac{w}{||w||}\right)^T x_1 = \frac{-b}{||w||}$ since $w^T x_1 + b = 0$

Margin: with bias

- Lemma 2: x has distance $\frac{|f_{w,b}(x)|}{||w||}$ to the hyperplane $f_{w,b}(x) = w^T x + b = 0$

Proof:

- Let $x = x_\perp + r\frac{w}{||w||}$, then $|r|$ is the distance
- Multiply both sides by w^T and add b
- Left hand side: $w^T x + b = \ f_{w,b}(x)$
- Right hand side: $w^T x_\perp + r\frac{w^T w}{||w||} + b = 0 + r||w||$



The notation here is:

$$y(x) = w^T x + w_0$$

Figure from *Pattern Recognition and Machine Learning*, Bishop

Support Vector Machine (SVM)

SVM: objective

- Margin over all training data points:

$$\gamma = \min_i \frac{|f_{w,b}(x_i)|}{||w||}$$

- Since only want correct $f_{w,b}$, and recall $y_i \in \{+1, -1\}$, we have

$$\gamma = \min_i \frac{y_i f_{w,b}(x_i)}{||w||}$$

- If $f_{w,b}$ incorrect on some x_i, the margin is negative

SVM: objective

- Maximize margin over all training data points:

$$\max_{w,b} \gamma = \max_{w,b} \min_i \frac{y_i f_{w,b}(x_i)}{||w||} = \max_{w,b} \min_i \frac{y_i(w^T x_i + b)}{||w||}$$

- A bit complicated …

SVM: simplified objective

- Observation: when (w, b) scaled by a factor c, the margin unchanged

$$\frac{y_i(cw^T x_i + cb)}{||cw||} = \frac{y_i(w^T x_i + b)}{||w||}$$

- Let's consider a fixed scale such that

$$y_{i^*}(w^T x_{i^*} + b) = 1$$

where x_{i^*} is the point closest to the hyperplane

SVM: simplified objective

- Let's consider a fixed scale such that

$$y_{i^*}(w^T x_{i^*} + b) = 1$$

  where x_{i^*} is the point closet to the hyperplane

- Now we have for all data

$$y_i(w^T x_i + b) \geq 1$$

  and at least for one i the equality holds

- Then the margin is $\frac{1}{||w||}$

SVM: simplified objective

- Optimization simplified to

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

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

- How to find the optimum $\hat{w}^*$?

SVM: principle for hypothesis class

Thought experiment

- Suppose pick an R, and suppose can decide if exists w satisfying

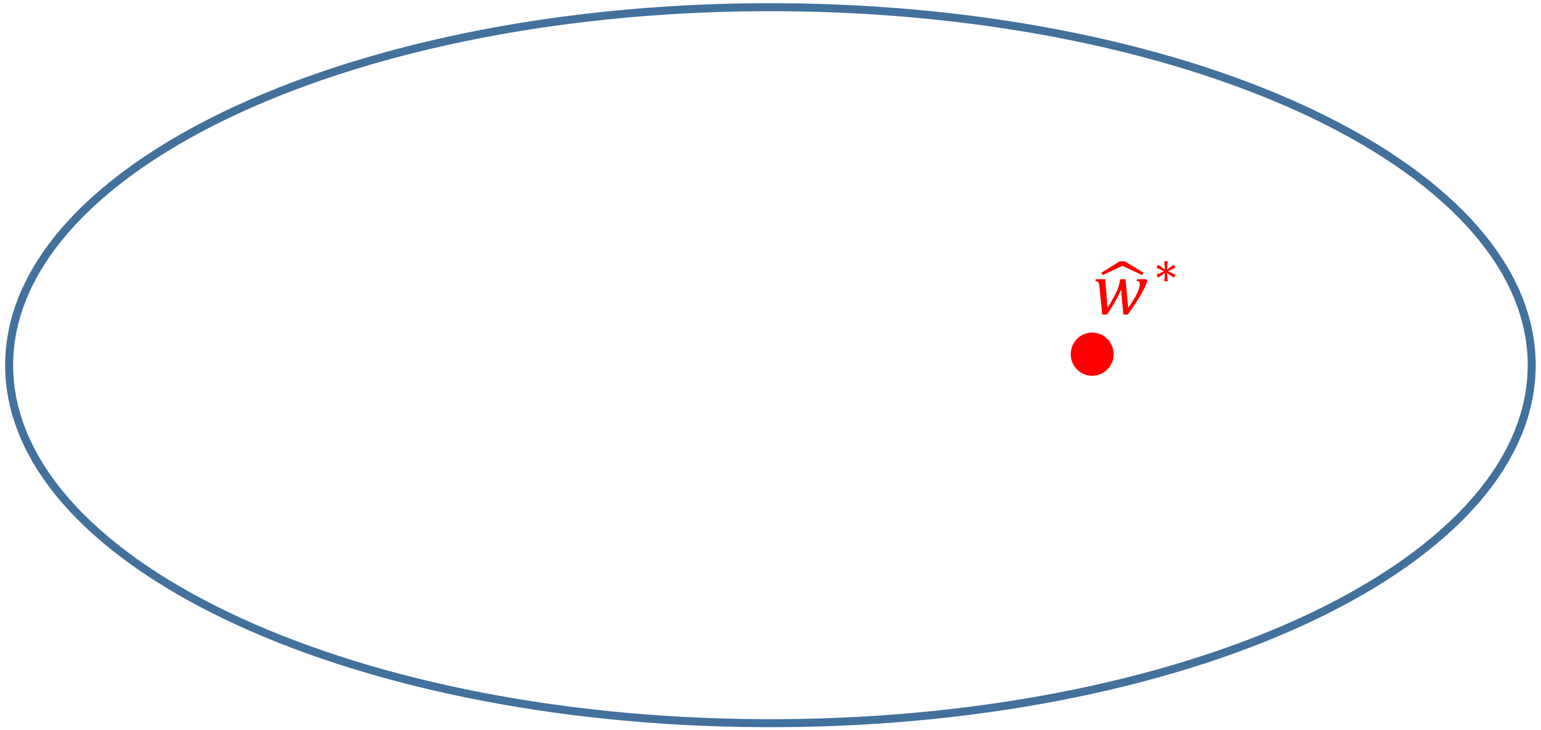
$$\frac{1}{2}||w||^2 \leq R$$

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

- Decrease R until cannot find w satisfying the inequalities

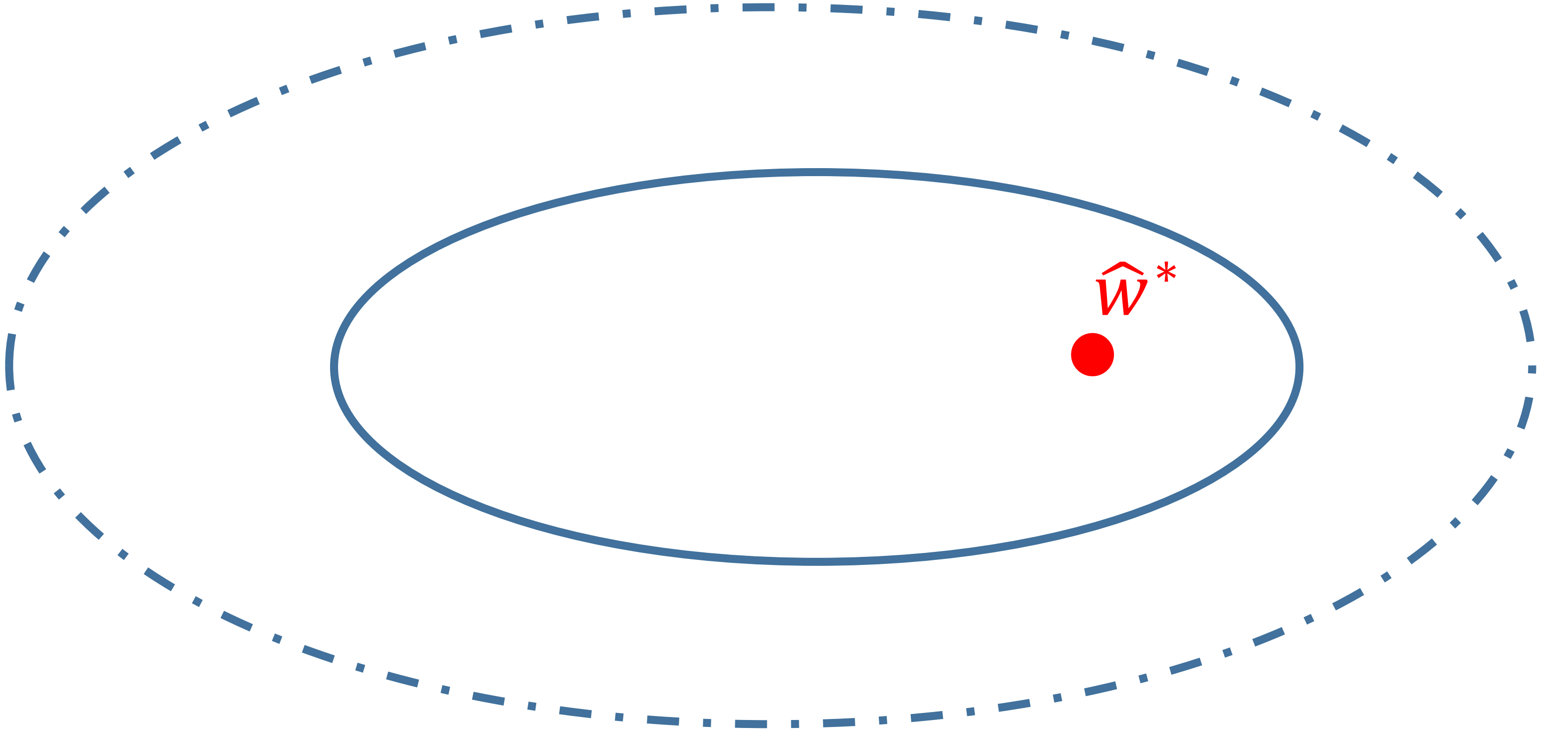
Thought experiment

- $\hat{w}^*$ is the best weight (i.e., satisfying the smallest R)



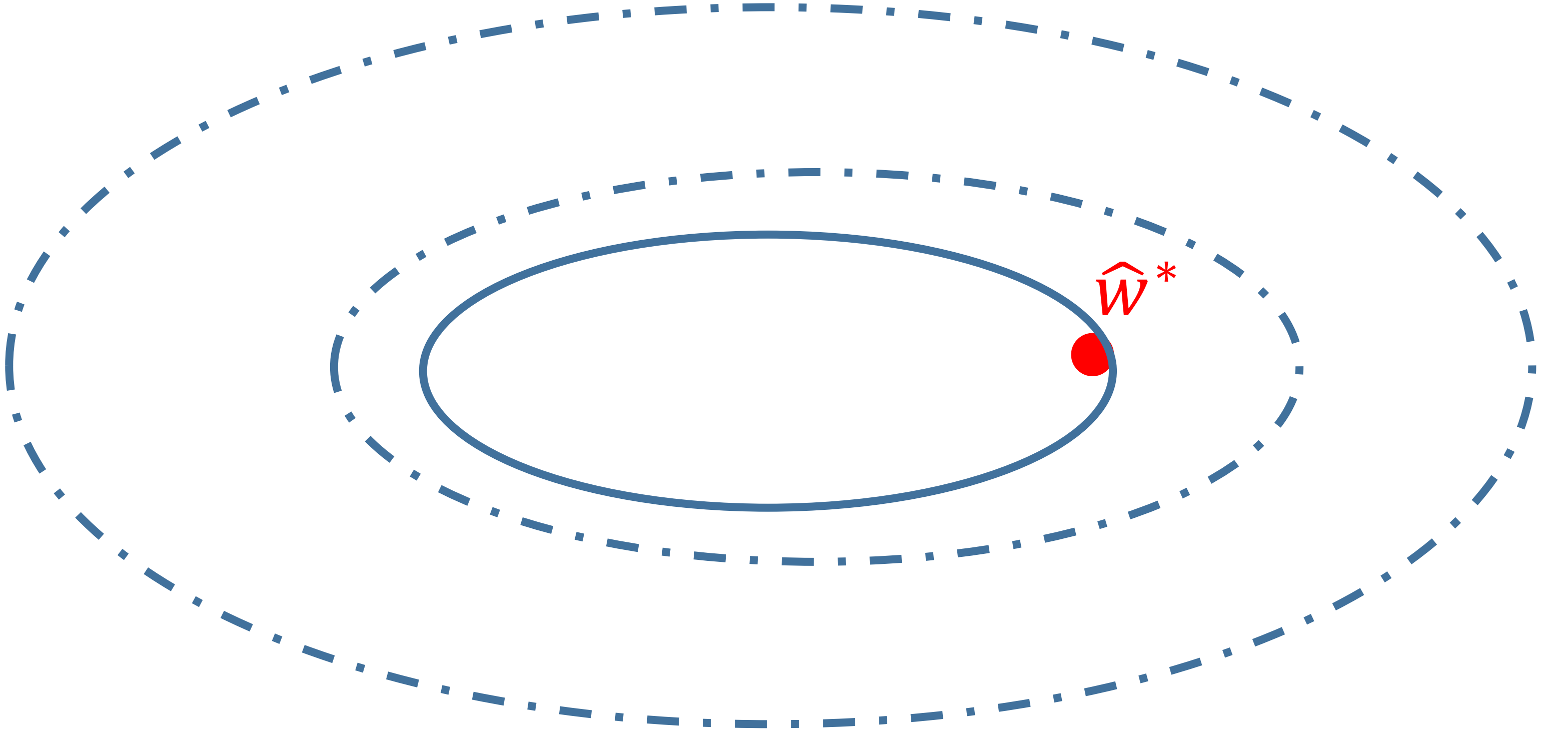
Thought experiment

- $\hat{w}^*$ is the best weight (i.e., satisfying the smallest R)



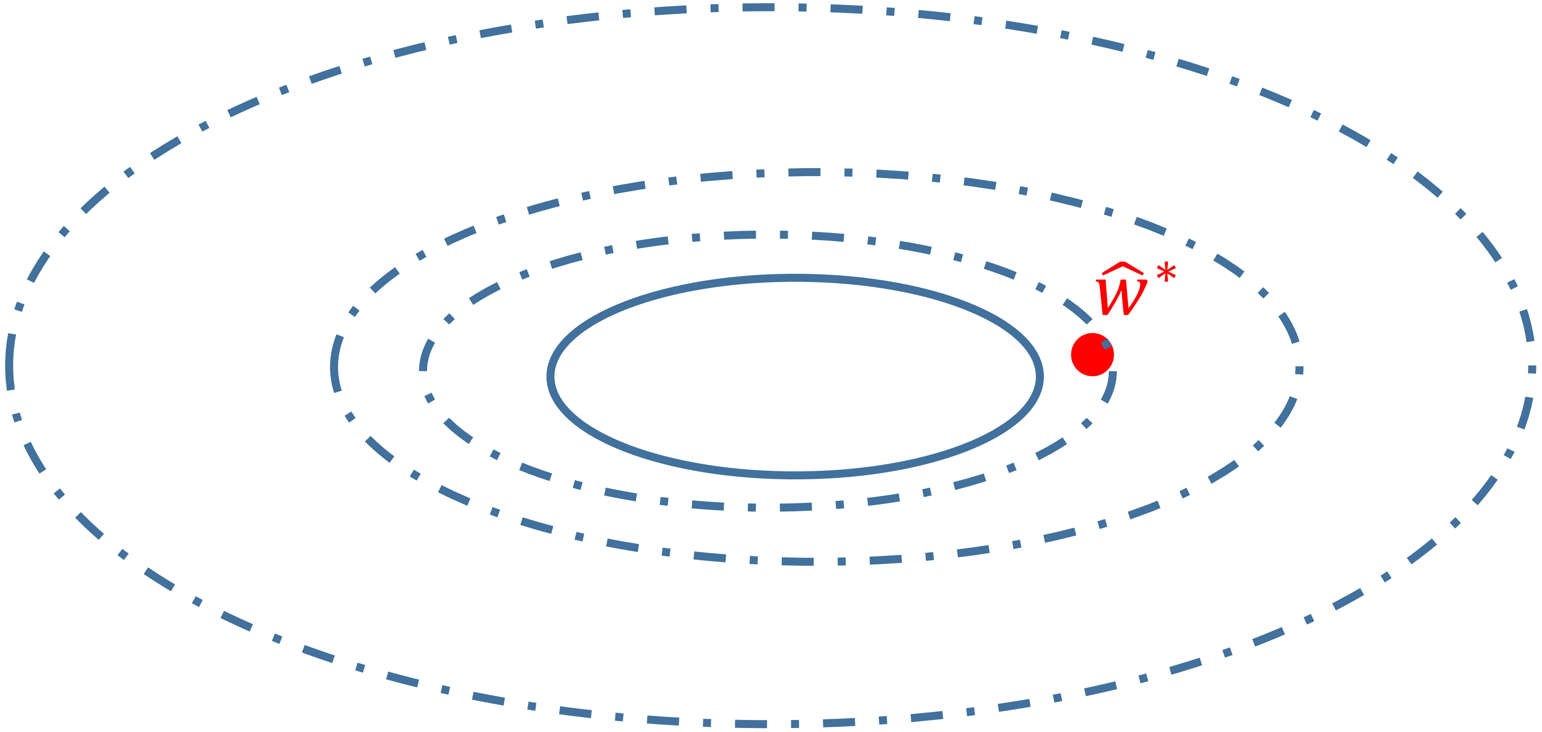
Thought experiment

- $\hat{w}^*$ is the best weight (i.e., satisfying the smallest R)



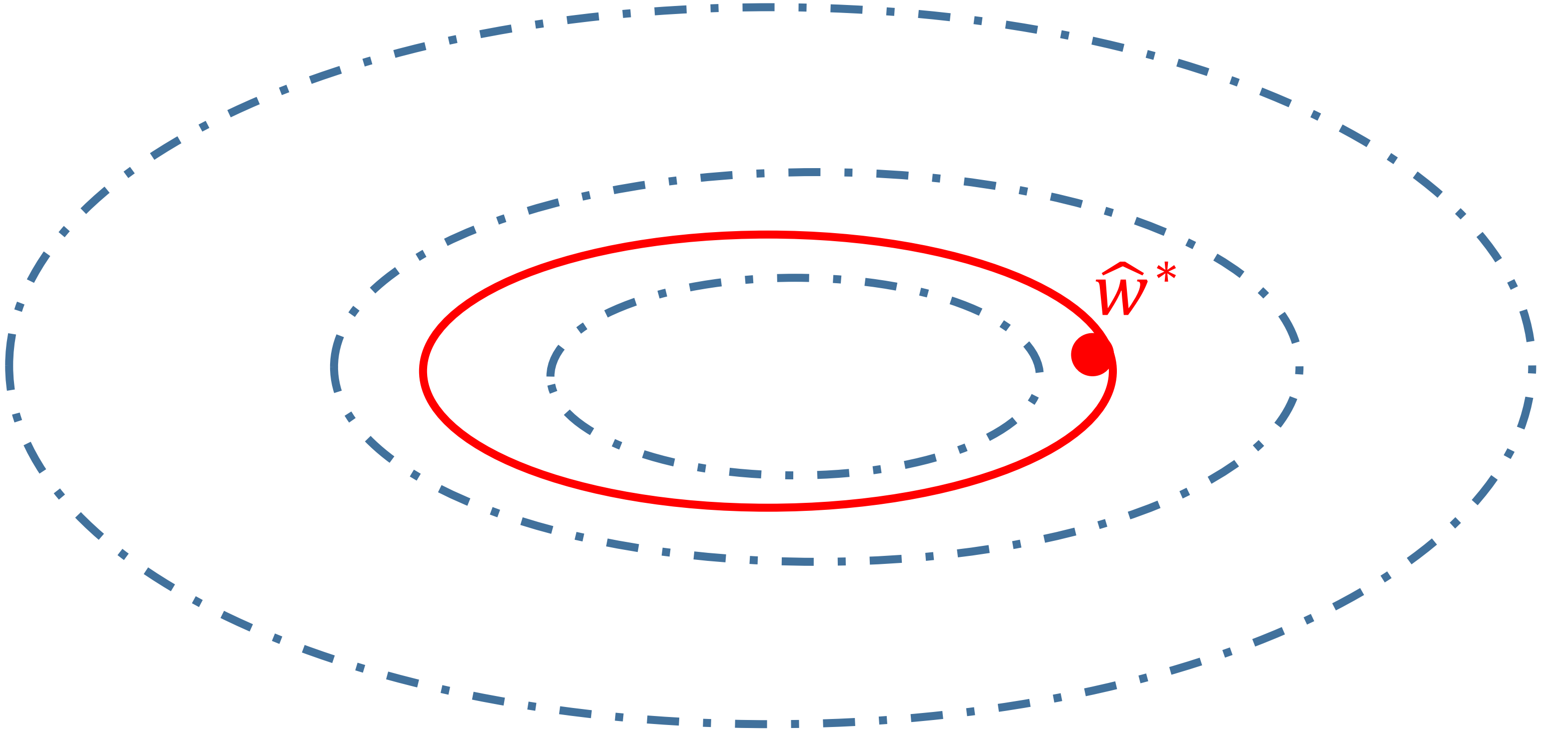
Thought experiment

- $\hat{w}^*$ is the best weight (i.e., satisfying the smallest R)



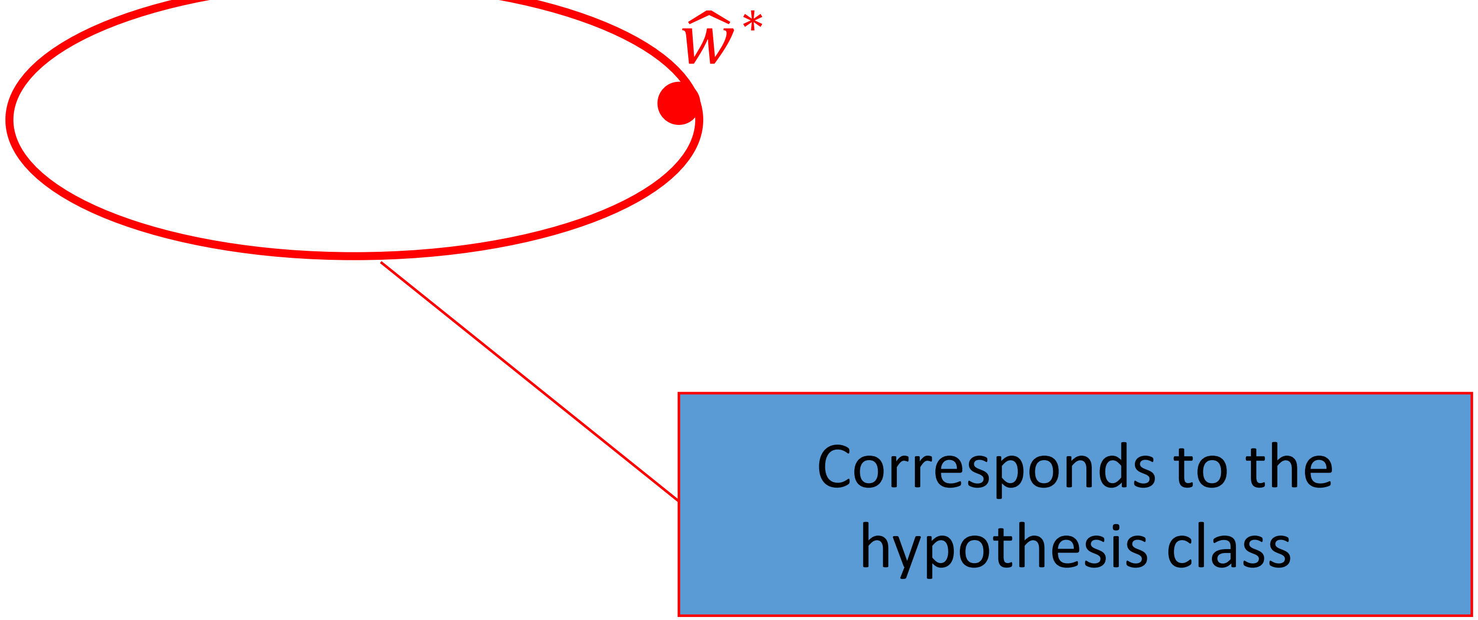
Thought experiment

- $\hat{w}^*$ is the best weight (i.e., satisfying the smallest R)



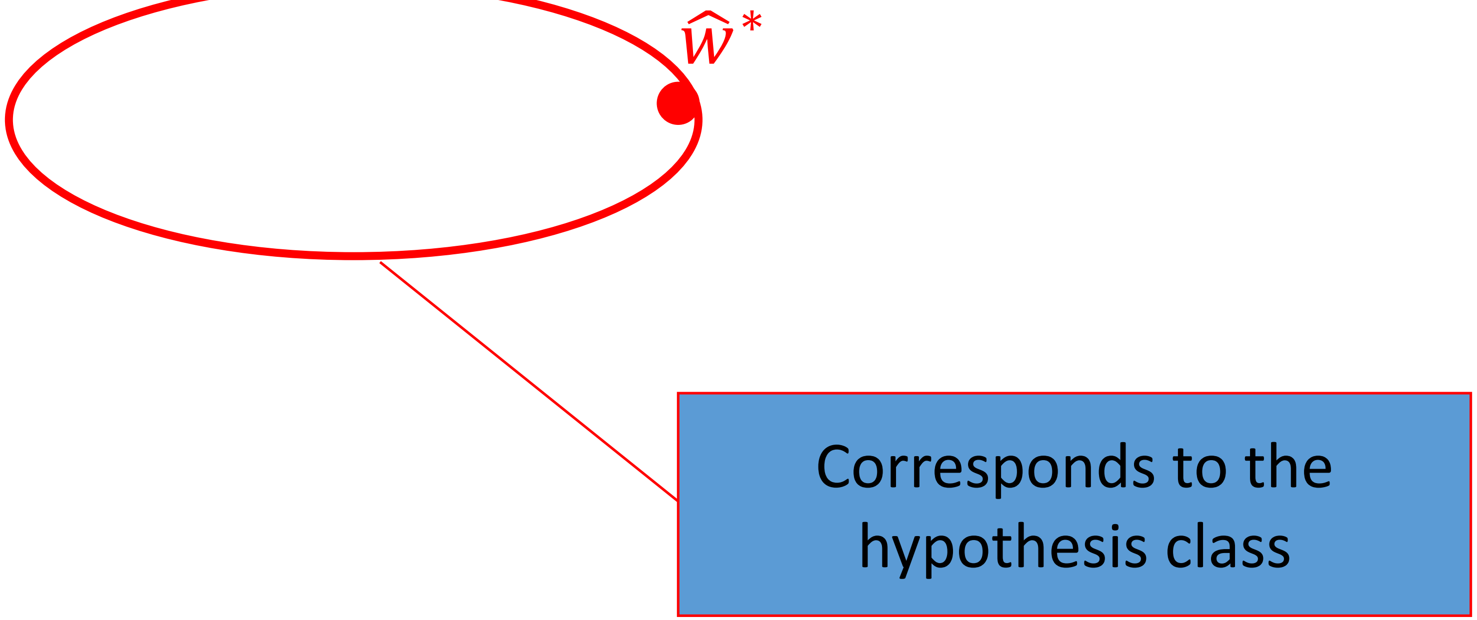
Thought experiment

- To handle the difference between empirical and expected losses →
- Choose large margin hypothesis (high confidence) →
- Choose a small hypothesis class



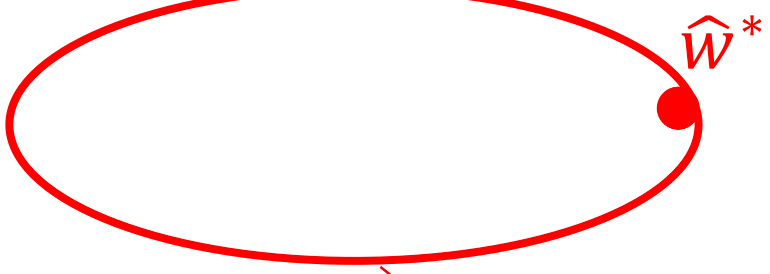
Thought experiment

- Principle: use smallest hypothesis class still with a correct/good one
  - Also true beyond SVM
  - Also true for the case without perfect separation between the two classes
  - Math formulation: VC-dim theory, etc.



Thought experiment

- Principle: use smallest hypothesis class still with a correct/good one
  - Whatever you know about the ground truth, add it as constraint/regularizer



Corresponds to the hypothesis class

SVM: optimization

- Optimization (Quadratic Programming):

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

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

- Solved by Lagrange multiplier method:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \frac{1}{2} ||w||^2 - \sum_i \alpha_i [y_i (w^T x_i + b) - 1]$$

where $\boldsymbol{\alpha}$ is the Lagrange multiplier

Lagrange multiplier

Lagrangian

- Consider optimization problem:

$$\min_{w} \ f(w)$$

$$\text{s.t. } \ h_i(w) = 0, \forall 1 \leq i \leq l$$

- Lagrangian:

$$\mathcal{L}(w, \boldsymbol{\beta}) = \ f(w) + \sum_{i} \beta_i h_i(w)$$

where β_i's are called Lagrange multipliers

Lagrangian

- Consider optimization problem:

$$\min_{w} \; f(w)$$

$$\text{s.t. } h_i(w) = 0, \forall 1 \leq i \leq l$$

- Solved by setting derivatives of Lagrangian to 0

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0; \quad \frac{\partial \mathcal{L}}{\partial \beta_i} = 0$$

Generalized Lagrangian

- Consider optimization problem:

$$\min_{w} \ f(w)$$

$$\text{s.t.} \ \ g_i(w) \leq 0, \forall 1 \leq i \leq k$$

$$h_j(w) = 0, \forall 1 \leq j \leq l$$

- Generalized Lagrangian:

$$\mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(w) + \sum_i \alpha_i g_i(w) + \sum_j \beta_j h_j(w)$$

where α_i, β_j's are called Lagrange multipliers

Generalized Lagrangian

- Consider the quantity:

$$\theta_P(w) := \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- Why?

$$\theta_P(w) = \begin{cases} f(w), & \text{if } w \text{ satisfies all the constraints} \\ +\infty, & \text{if } w \text{ does not satisfy the constraints} \end{cases}$$

- So minimizing $f(w)$ is the same as minimizing $\theta_P(w)$

$$\min_w f(w) = \min_w \theta_P(w) = \min_w \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Lagrange duality

- The primal problem

$$p^* \coloneqq \min_w f(w) = \min_w \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- The dual problem

$$d^* \coloneqq \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \min_w \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- Always true:

$$d^* \leq p^*$$

Lagrange duality

- The primal problem

$$p^* := \min_{w} f(w) = \min_{w} \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- The dual problem

$$d^* := \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\alpha_i \geq 0} \min_{w} \mathcal{L}(w, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- Interesting case: when do we have

$$d^* = p^*?$$

Lagrange duality

- Theorem: under proper conditions, there exists $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ such that

$$d^* = \mathcal{L}(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = p^*$$

Moreover, $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ satisfy Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0, \qquad \alpha_i g_i(w) = 0$$

$$g_i(w) \leq 0, \;\; h_j(w) = 0, \qquad \alpha_i \geq 0$$

Lagrange duality

- Theorem: under proper conditions, there exists $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ such that

$$d^* = \mathcal{L}(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = p^*$$

Moreover, $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ satisfy Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0, \qquad \alpha_i g_i(w) = 0$$

$$g_i(w) \leq 0, \ \ h_j(w) = 0, \qquad \alpha_i \geq 0$$

Lagrange duality

- Theorem: under proper conditions, there exists $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ such that

$$d^* = \mathcal{L}(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = p^*$$

- Moreover, $(w^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ satisfy Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0, \qquad \alpha_i g_i(w) = 0$$

$$g_i(w) \leq 0, \; h_j(w) = 0, \qquad \alpha_i \geq 0$$

Lagrange duality

- What are the proper conditions?
- A set of conditions (Slater conditions):
  - f, g_i convex, h_j affine
  - Exists w satisfying all $g_i(w) < 0$

- There exist other sets of conditions
  - Search Karush–Kuhn–Tucker conditions on Wikipedia

SVM: optimization

SVM: optimization

- Optimization (Quadratic Programming):

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

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

- Generalized Lagrangian:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \frac{1}{2}||w||^2 - \sum_i \alpha_i [y_i(w^T x_i + b) - 1]$$

where $\boldsymbol{\alpha}$ is the Lagrange multiplier

SVM: optimization

- KKT conditions:

$$\frac{\partial \mathcal{L}}{\partial w} = 0, \rightarrow w = \sum_i \alpha_i y_i x_i \quad (1)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0, \rightarrow 0 = \sum_i \alpha_i y_i \quad (2)$$

- Plug into $\mathcal{L}$:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (3)$$

combined with $\sum_i \alpha_i y_i = 0$, $\alpha_i \geq 0$

SVM: optimization

- Reduces to dual problem:

$$\mathcal{L}(w, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

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

- Since $w = \sum_i \alpha_i y_i x_i$, we have $w^T x + b = \sum_i \alpha_i y_i x_i^T x + b$

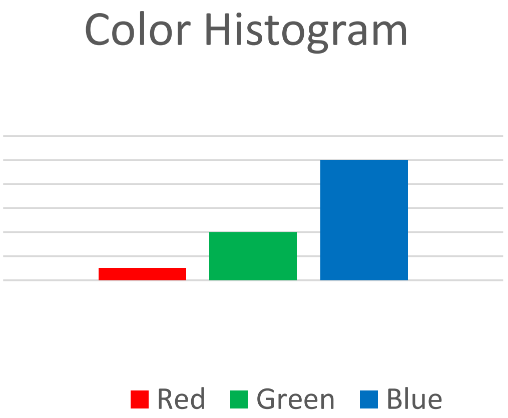
Kernel methods

Features

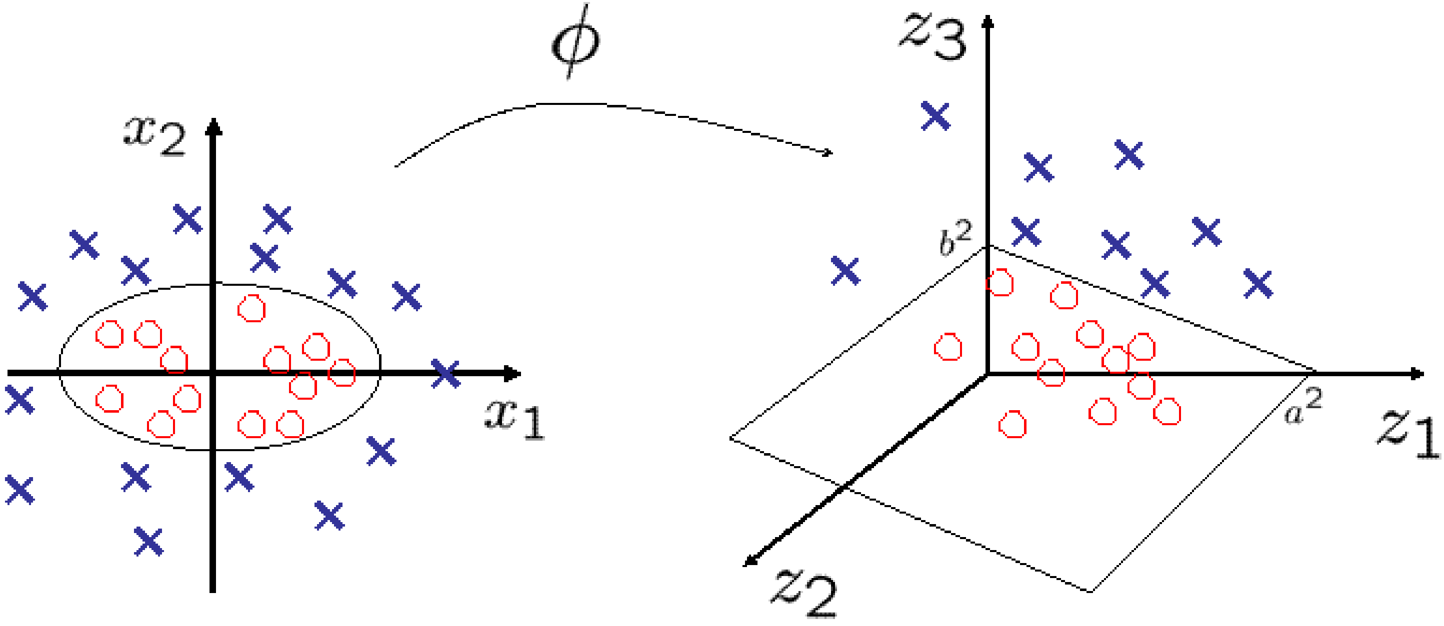
x

Extract features

$\phi(x)$



Features



$$\phi : (x_1, x_2) \longrightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

$$\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 = 1 \longrightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = 1$$

Features

- Proper feature mapping can make non-linear to linear
- Using SVM on the feature space $\{\phi(x_i)\}$: only need $\phi(x_i)^T\phi(x_j)$

- Conclusion: no need to design $\phi(\cdot)$, only need to design

$$k(x_i, x_j) = \phi(x_i)^T\phi(x_j)$$

Polynomial kernels

- Fix degree d and constant c:

$$k(x, x') = (x^T x' + c)^d$$

- What are $\phi(x)$?
- Expand the expression to get $\phi(x)$

Polynomial kernels

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^2, \quad K(\mathbf{x}, \mathbf{x}') = (x_1 x_1' + x_2 x_2' + c)^2 = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} {x'}_1^2 \\ {x'}_2^2 \\ \sqrt{2}\, x_1' x_2' \\ \sqrt{2c}\, x_1' \\ \sqrt{2c}\, x_2' \\ c \end{bmatrix}$$

Figure from Foundations of Machine Learning, by M. Mohri, A. Rostamizadeh, and A. Talwalkar

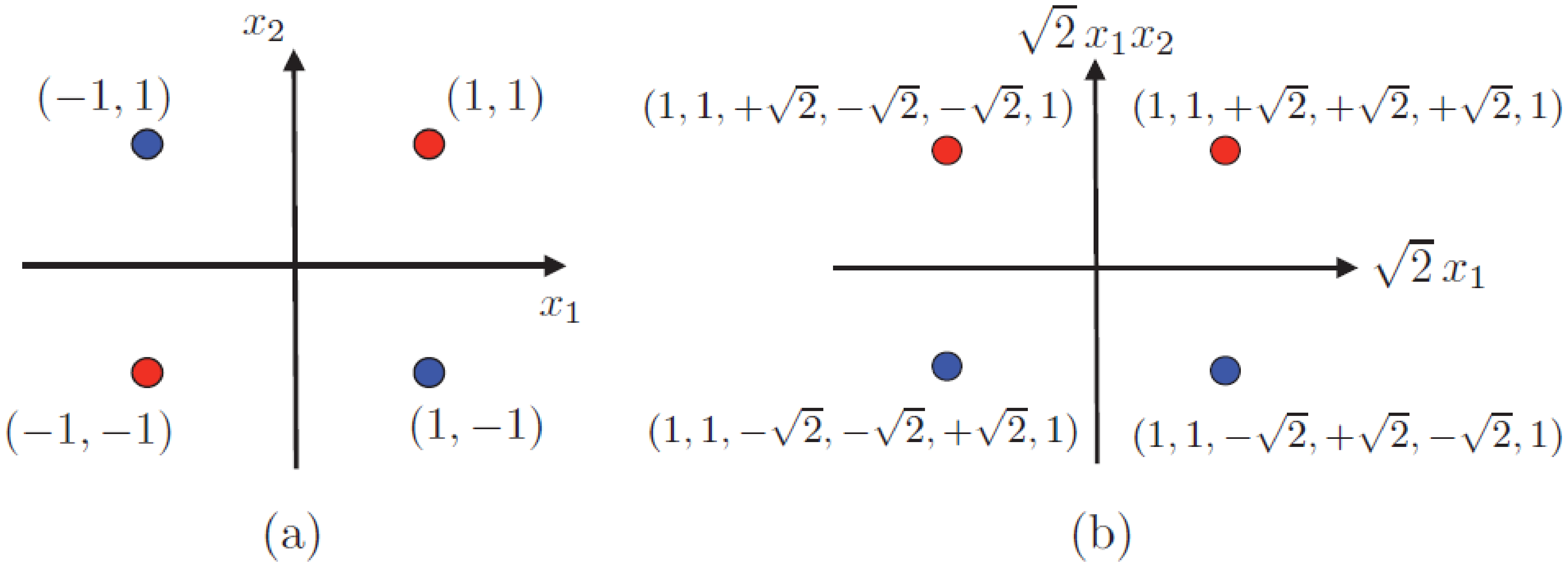


Figure 5.2 Illustration of the XOR classification problem and the use of polynomial kernels. (a) XOR problem linearly non-separable in the input space. (b) Linearly separable using second-degree polynomial kernel.

Figure from Foundations of Machine Learning, by M. Mohri, A. Rostamizadeh, and A. Talwalkar

Gaussian kernels

- Fix bandwidth σ:

$$k(x, x') = \exp(-||x - x'||^2 / 2\sigma^2)$$

- Also called radial basis function (RBF) kernels

- What are $\phi(x)$? Consider the un-normalized version

$$k'(x, x') = \exp(x^T x' / \sigma^2)$$

- Power series expansion:

$$k'(x, x') = \sum_{i}^{+\infty} \frac{(x^T x')^i}{\sigma^i i!}$$

Mercer’s condition for kenerls

- Theorem: $k(x, x')$ has expansion

$$k(x, x') = \sum_{i}^{+\infty} a_i \phi_i(x) \phi_i(x')$$

if and only if for any function $c(x)$,

$$\int \int c(x) c(x') k(x, x') dx dx' \geq 0$$

(Omit some math conditions for k and c)

Constructing new kernels

- Kernels are closed under positive scaling, sum, product, pointwise limit, and composition with a power series $\sum_i^{+\infty} a_i k^i(x, x')$

- Example: $k_1(x, x'), k_2(x, x')$ are kernels, then also is

$$k(x, x') = 2k_1(x, x') + 3k_2(x, x')$$

- Example: $k_1(x, x')$ is kernel, then also is

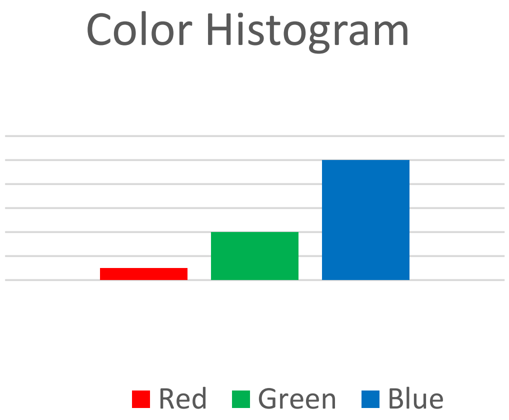
$$k(x, x') = \exp(k_1(x, x'))$$

Kernels v.s. Neural networks

Features

x

Extract features

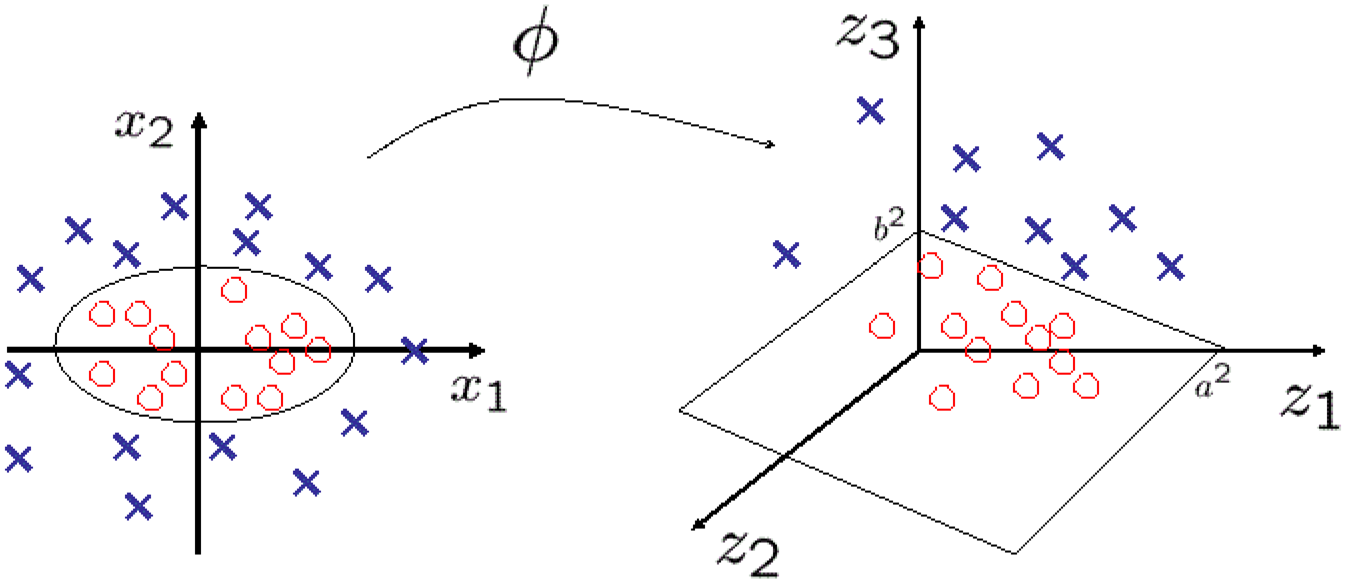


build hypothesis

$$y = w^T \phi(x)$$

Features: part of the model

Nonlinear model



build hypothesis $y = w^T\phi(x)$

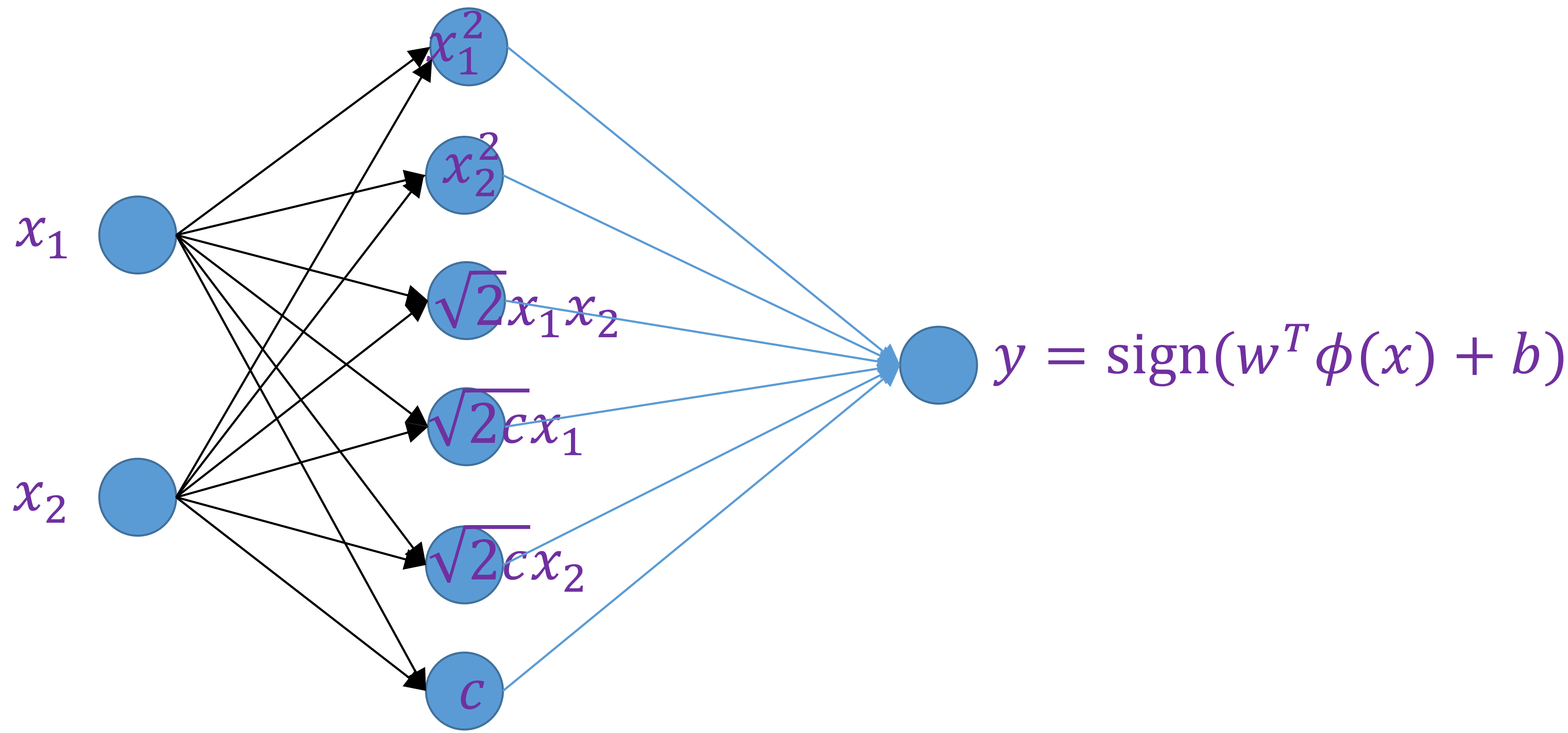
Linear model

Polynomial kernels

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^2, \quad K(\mathbf{x}, \mathbf{x}') = (x_1 x_1' + x_2 x_2' + c)^2 = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} {x'}_1^2 \\ {x'}_2^2 \\ \sqrt{2}\, x_1' x_2' \\ \sqrt{2c}\, x_1' \\ \sqrt{2c}\, x_2' \\ c \end{bmatrix}$$

Figure from Foundations of Machine Learning, by M. Mohri, A. Rostamizadeh, and A. Talwalkar

Polynomial kernel SVM as two layer neural network



First layer is fixed. If also learn first layer, it becomes two layer neural network