香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

COMP 5212
Machine Learning
Lecture 5

Kernel Methods,
Support Vector Machine

Junxian He
Feb 16, 2024

Attendance Recording

Please download HKUST iLearn in your devices, we are going to use iPRS for quizzes

HKUST iLearn 4+

HKUST Learning

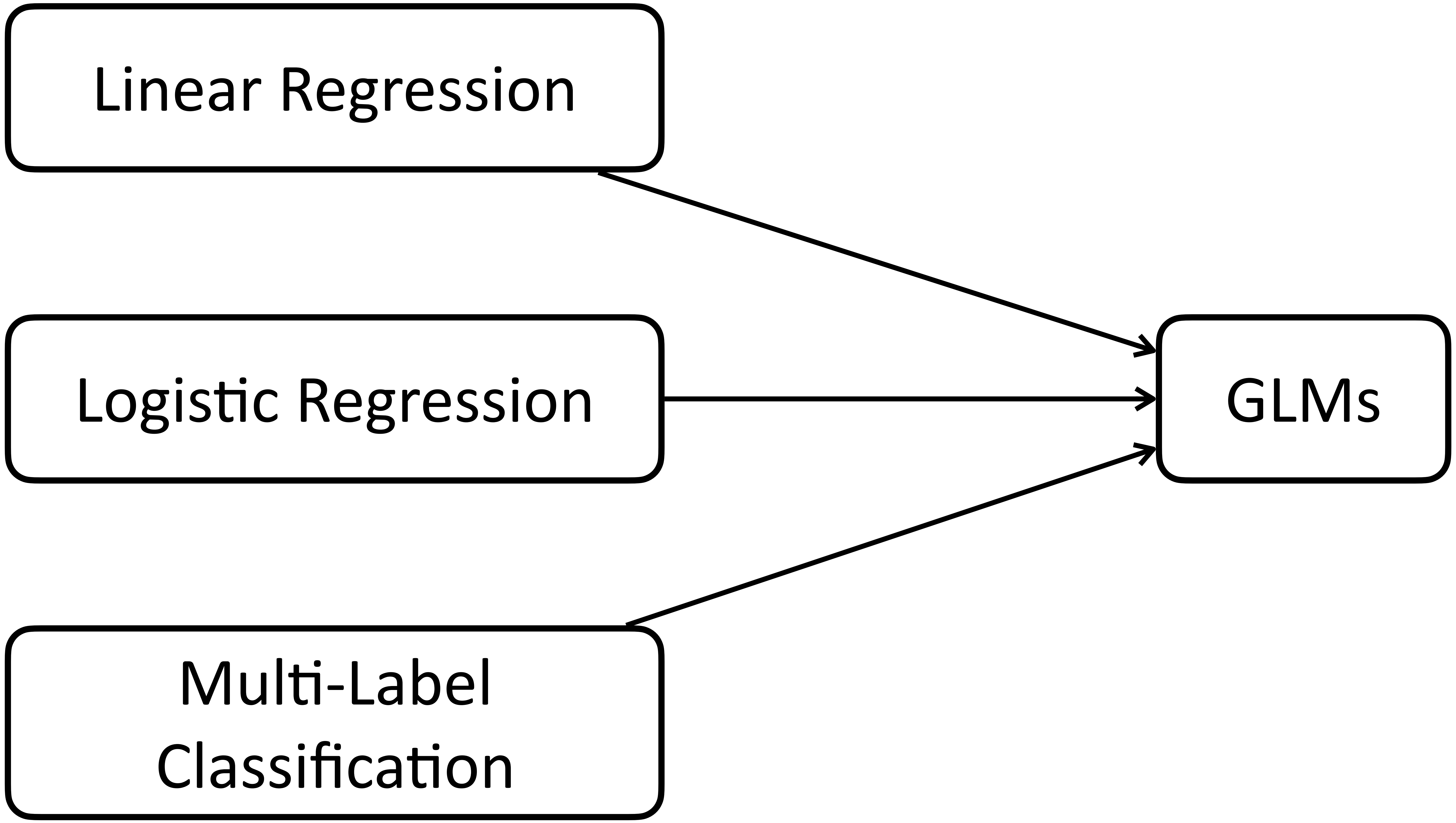
The Hong Kong University of Science and Technology

Designed for iPad

Free

View in Mac App Store ↗

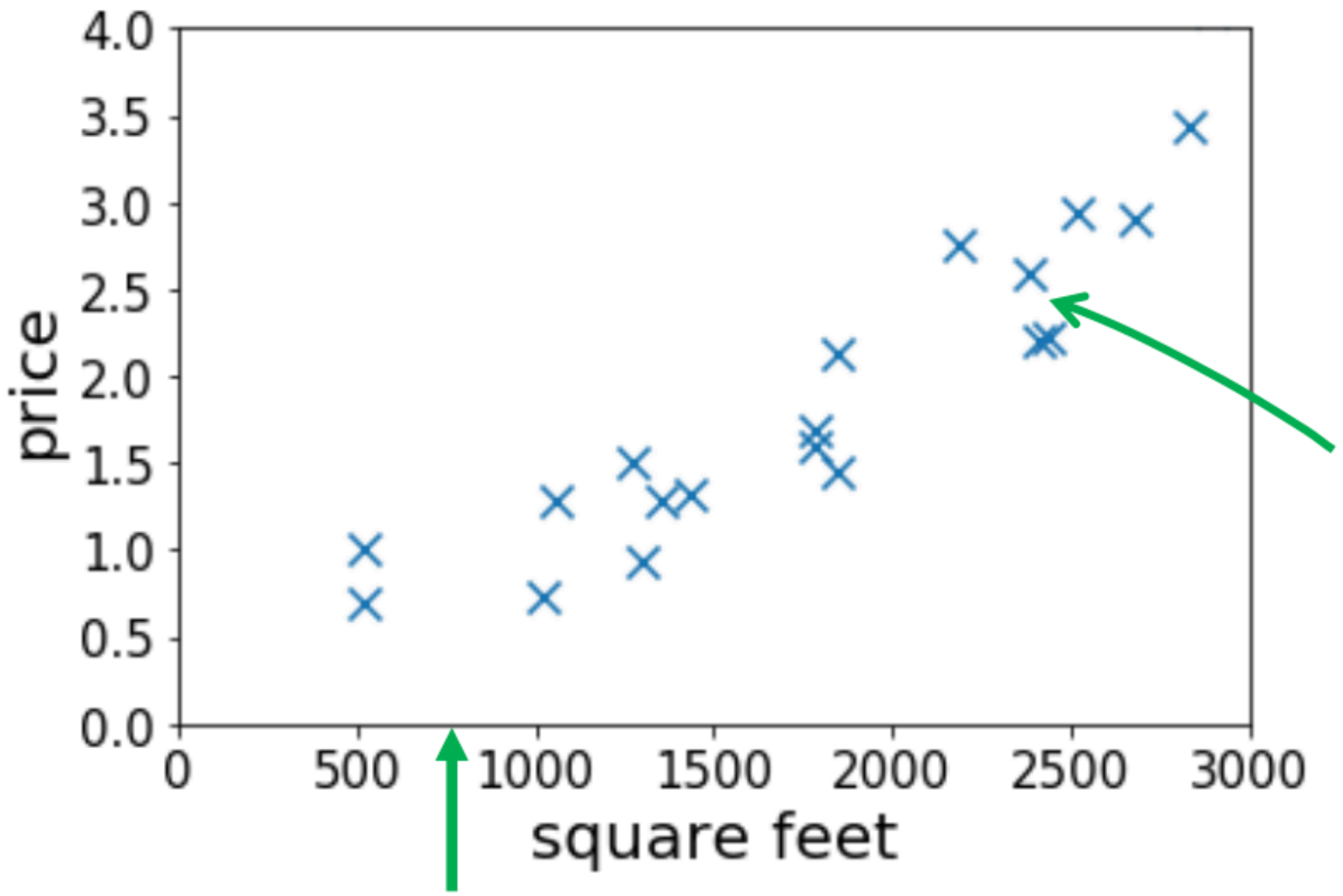
Recap: Generalized Linear Models



Recap: Generalized Linear Models

inference	$h_\theta(x) = \mathbb{E}[y \mid x; \theta]$ is the **output**.
learn	$\max_\theta \log p(y \mid x; \theta)$ by maximum likelihood.
algorithm: SGD	$\theta^{(t+1)} = \theta^{(t)} + \alpha \left(y^{(i)} - h_{\theta^{(t)}}(x^{(i)}) \right) x^{(i)}$

Recap: Kernel Methods (Feature Map)



15th sample
$(x^{(15)}, y^{(15)})$

$x = 800$
$y = ?$

$$y = \theta x$$

$$y = \theta_3 x^3 + \theta_2 x^2 + \theta_1 x + \theta_0$$

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \in \mathbb{R}^4.$$

Feature map
$\phi : R^d \rightarrow R^p$

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

LMS Update Rule with Features

$$\theta := \theta + \alpha \sum_{i=1}^{n} \left(y^{(i)} - \theta^T \phi(x^{(i)})\right) \phi(x^{(i)})$$

Recap: Kernel Trick

$$\theta = \sum_{i=1}^{n} \beta_i \phi(x^{(i)}) \qquad \beta_i \in R$$

$$\boxed{\beta_i := \beta_i + \alpha \left(y^{(i)} - \sum_{j=1}^{n} \beta_j \phi(x^{(j)})^T \phi(x^{(i)}) \right)}$$

Kernel $K(x, z)$ $\quad \mathcal{X} \times \mathcal{X} \to \mathbb{R} \quad$ $\mathcal{X}$ is the space of the input

$$K(x, z) \triangleq \langle \phi(x), \phi(z) \rangle$$

Recap: Kernel Trick

- Compute $K(\phi(x^{(i)}), \phi(x^{(j)})) = <\phi(x^{(i)}), \phi(x^{(j)})>$ for all i, j

- Loop $$\beta_i := \beta_i + \alpha \left(y^{(i)} - \sum_{j=1}^{n} \beta_j K(x^{(i)}, x^{(j)}) \right) \quad \forall i \in \{1, \ldots, n\}$$

Recall that n is the number of data samples

- Inference: $$\theta^T \phi(x) = \sum_{i=1}^{n} \beta_i \phi(x^{(i)})^T \phi(x) = \sum_{i=1}^{n} \beta_i K(x^{(i)}, x)$$

The Kernel function is all we need for training and inference!

Recap: Implicit Feature Map

- Explicit Feature Map: first define feature map $\phi(x)$, then compute the Kernel according to $\phi(x)$

- Implicit Feature Map: first define the Kernel Function K(), without knowing what the feature map is

Recap: Implicit Feature Map (Example)

$$K(x,z) = (x^T z)^2 \qquad x, z \in \mathbb{R}^d$$

What is the feature map to make K a valid kernel function?

$$\begin{aligned} K(x,z) &= \left(\sum_{i=1}^{d} x_i z_i\right)\left(\sum_{j=1}^{d} x_j z_j\right) \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j z_i z_j \\ &= \sum_{i,j=1}^{d} (x_i x_j)(z_i z_j) \end{aligned}$$

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}$$

Requires O(d^2) compute for feature mapping

Requires O(d) compute for Kernel function

Kernel as Similarity Metrics

- Generally $K(x, z) = \phi(x)^T\phi(z)$ is large when $\phi(x)$ and $\phi(z)$ are close to each other

- We can think of $K(x, z)$ as some measurement of how similar are $\phi(x)$ and $\phi(z)$, or of how similar are x and z

Example: Gaussian Kernel

$$K(x, z) = \exp\left(-\frac{||x - z||^2}{2\sigma^2}\right)$$

Corresponds to infinite dimensional feature mapping

What Makes a Valid Kernel Function: Necessary Condition

- Kernel Matrix $K_{ij} = K(x^{(i)}, x^{(j)}) = \phi(x^{(i)})^T \phi(x^{(j)})$

- K is symmetric

- K is positive semidefinite

$$
\begin{aligned}
z^T K z &= \sum_i \sum_j z_i K_{ij} z_j \\
&= \sum_i \sum_j z_i \phi(x^{(i)})^T \phi(x^{(j)}) z_j \\
&= \sum_i \sum_j z_i \sum_k \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\
&= \sum_k \sum_i \sum_j z_i \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\
&= \sum_k \left(\sum_i z_i \phi_k(x^{(i)}) \right)^2 \\
&\geq 0.
\end{aligned}
$$

What Makes a Valid Kernel Function: Necessary and Sufficient Condition

Theorem (Mercer). Let $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ be given. Then for K to be a valid (Mercer) kernel, it is necessary and sufficient that for any $\{x^{(1)}, \ldots, x^{(n)}\}$, $(n < \infty)$, the corresponding kernel matrix is symmetric positive semi-definite.

Application of Kernel Methods

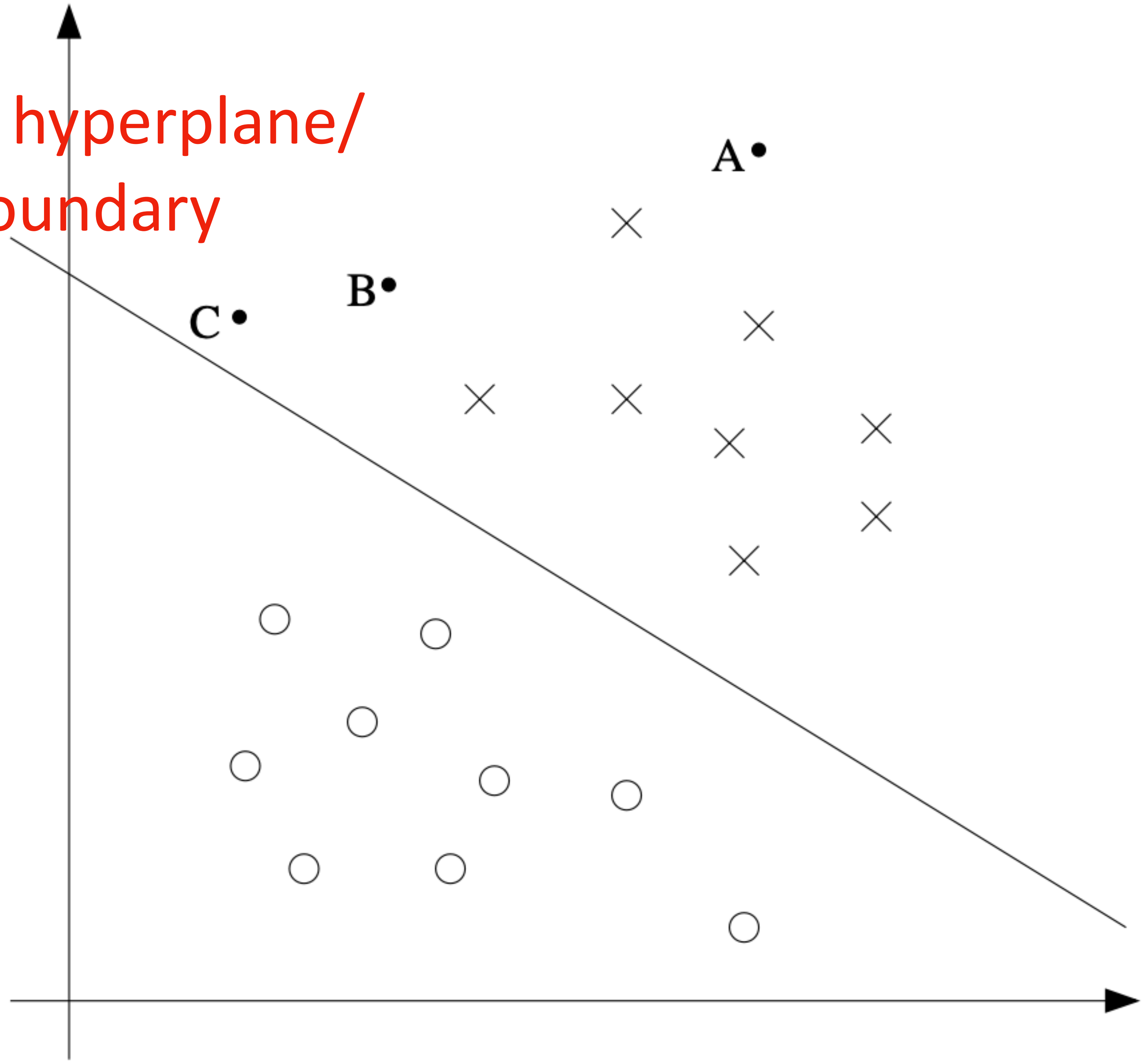
- In generalized linear models (which we have shown)
- In support vector machines (which we will show next)
- Any learning algorithm that you can write in terms of only <x, z>

Just replace <x, z> with K(x, z), you magically transform the algorithm to work efficiently in the *implicit* high dimensional feature space

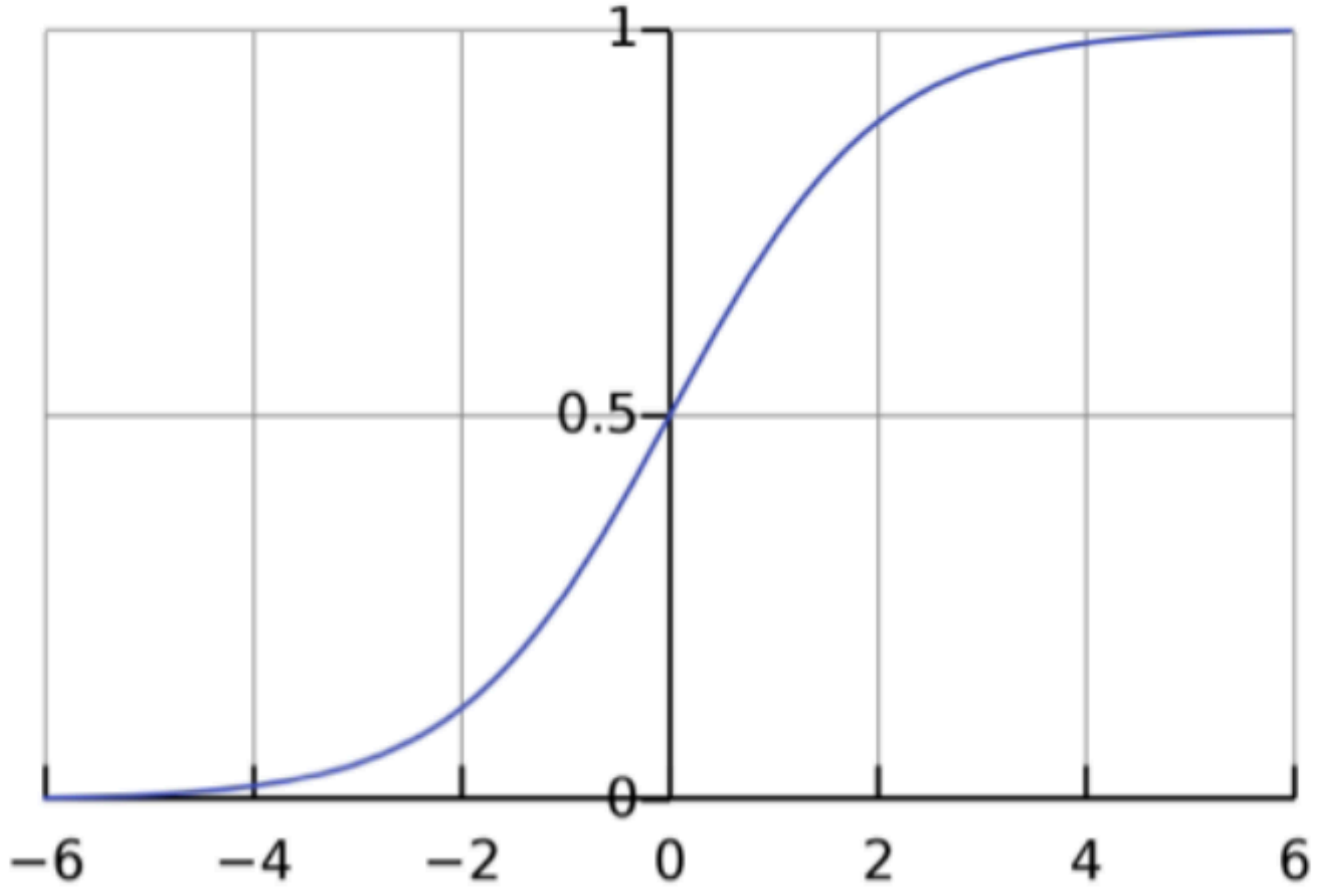
Support Vector Machines

Confidence in Logistic Regression

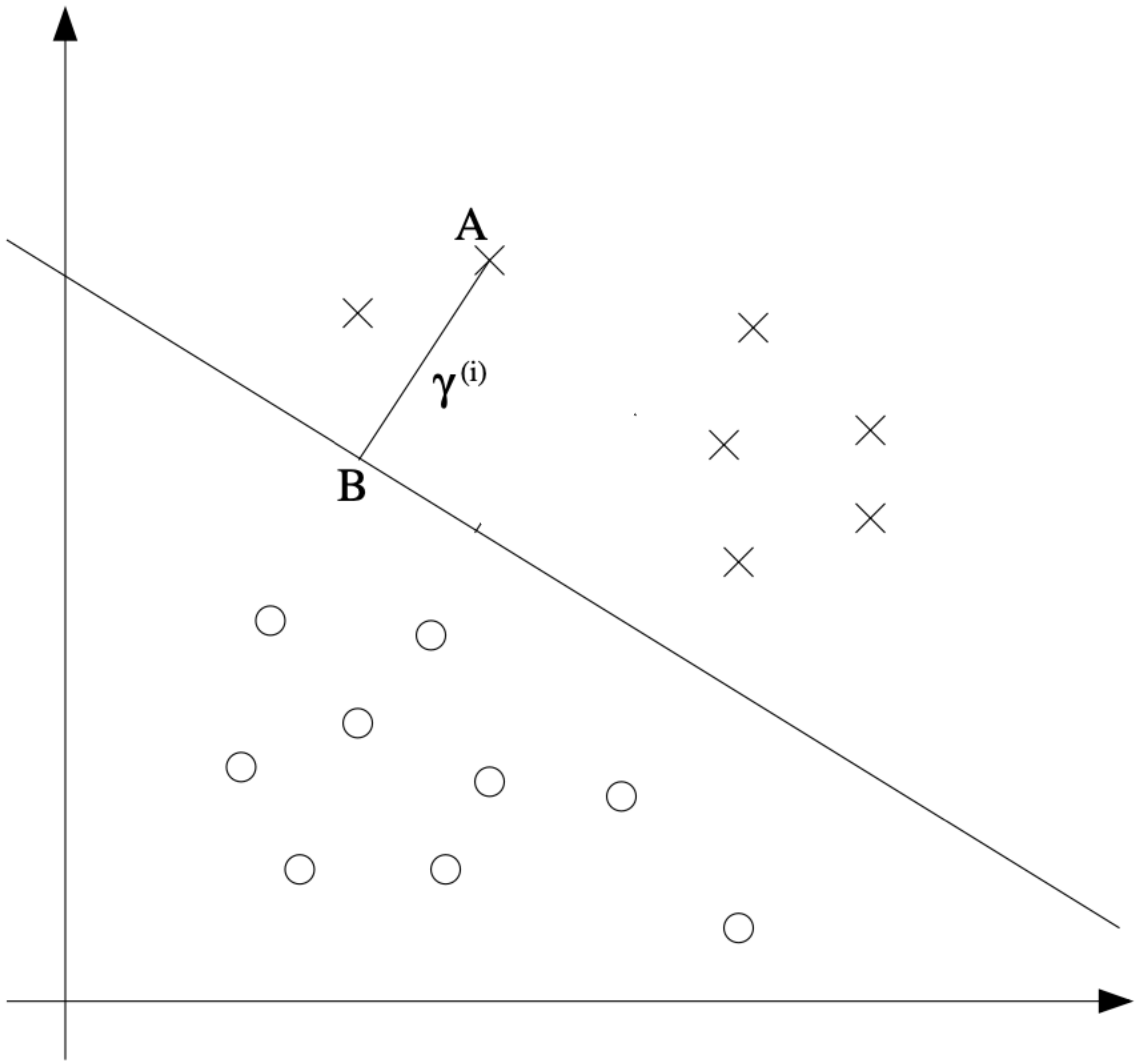
Separating hyperplane/
decision boundary



$$p(y) = \frac{1}{1 + e^{-\theta^T x}}$$



Margin



New Notations

Consider a binary classification problem, with the input feature x and $y \in \{-1,1\}$ (instead of $\{0,1\}$), the classifier is:

$$h_{w,b}(x) = g(w^T x + b).$$

$g(z) = 1$ if $z \geq 0$, and $g(z) = -1$

Functional Margin

Given a training example $(x^{(i)}, y^{(i)})$

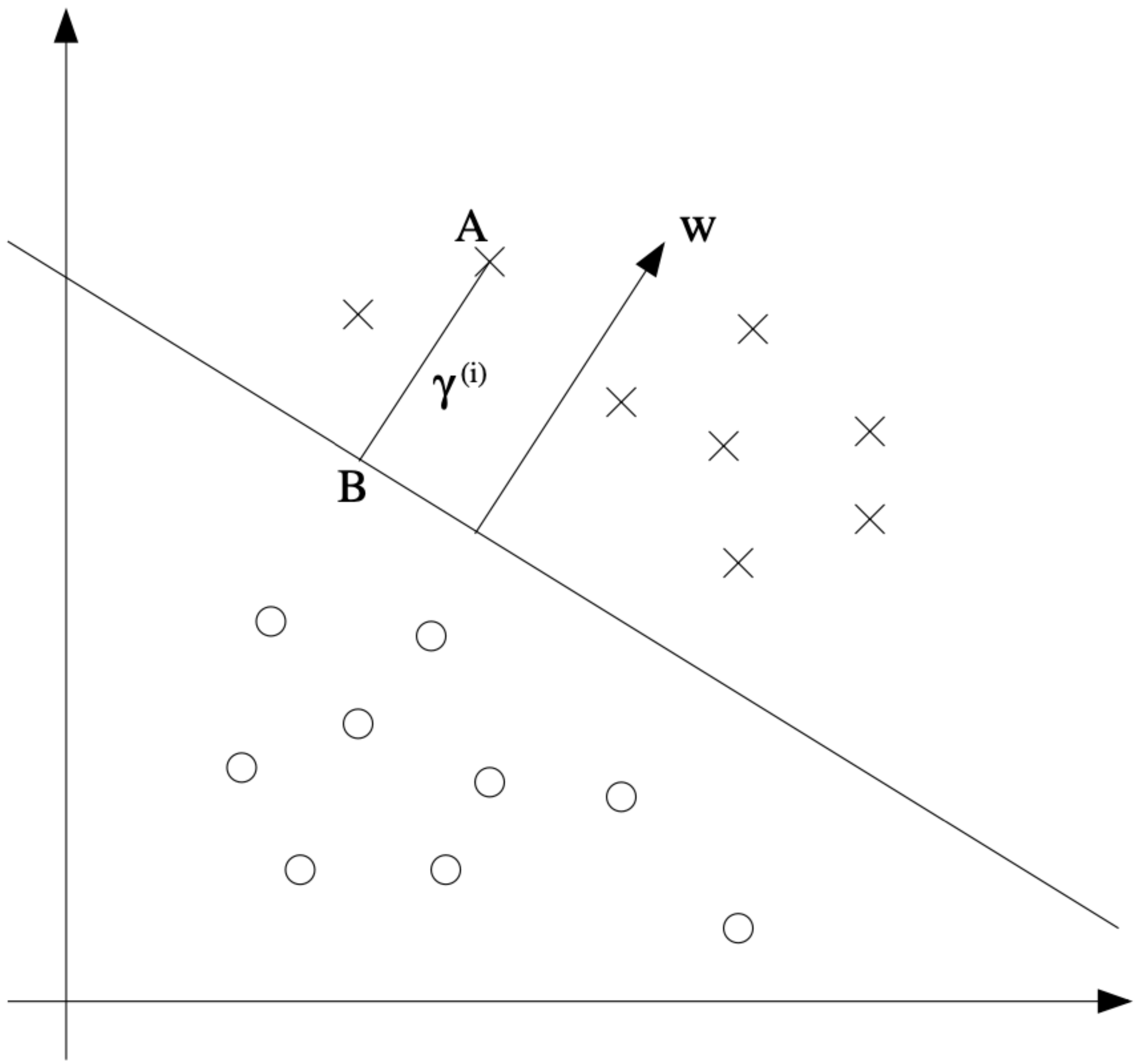
$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b).$$

Given a training set $S = \{(x^{(i)}, y^{(i)}); i = 1,\ldots,n\}$

$$\hat{\gamma} = \min_{i=1,\ldots,n} \hat{\gamma}^{(i)}$$

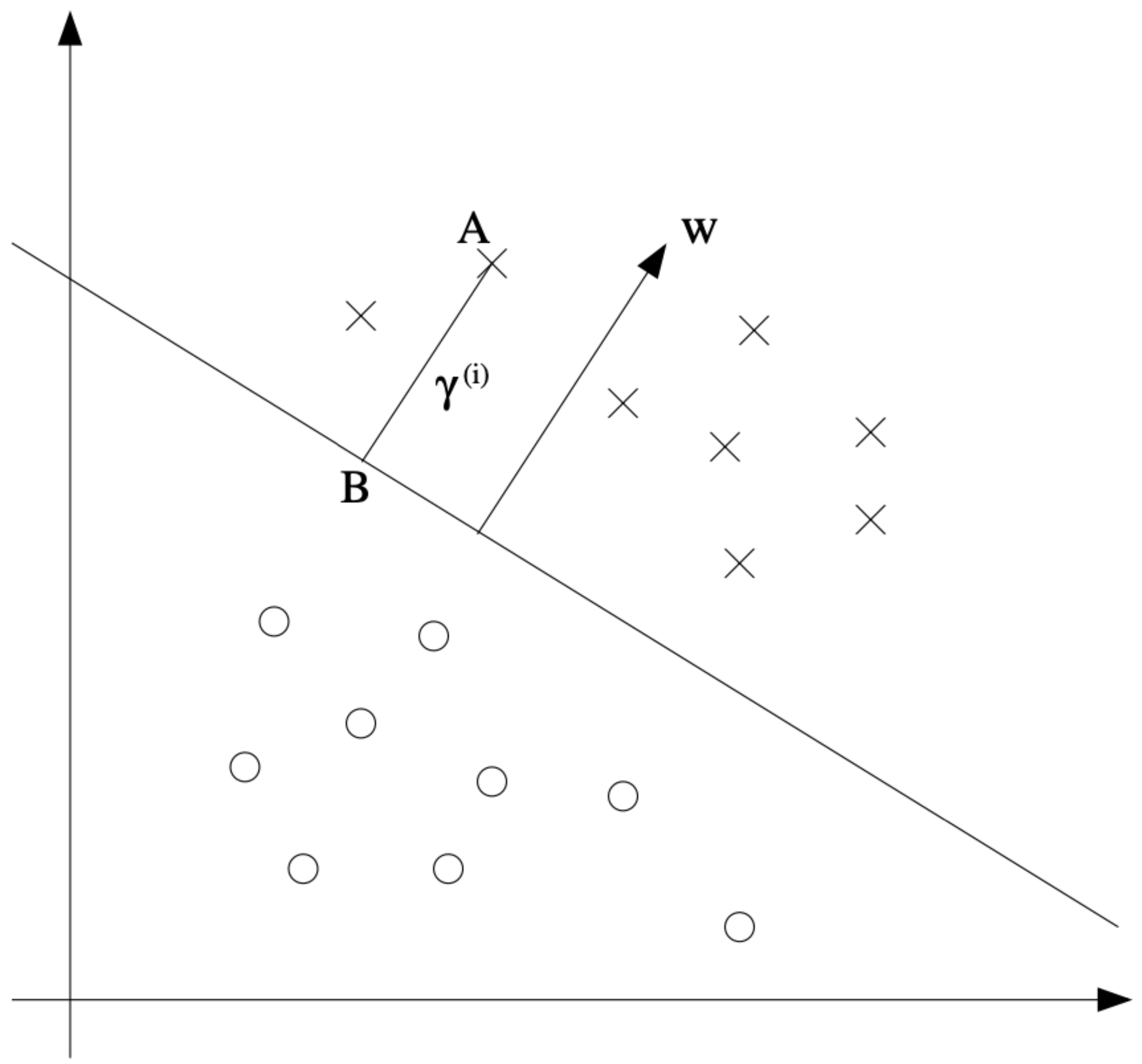
Functional margin changes rescaling parameters, making it a bad objective, e.g. when w->2w, b->2b, the functional margin changes while the separating plane does not really change

Geometric Margin



What is the geometric margin?

Geometric Margin



$$w^T\left(x^{(i)} - \gamma^{(i)}\frac{w}{||w||}\right) + b = 0.$$

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

Generally

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

Geometric Margin

Given a training set $S = \{(x^{(i)}, y^{(i)}); i = 1,\ldots,n\}$

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

The Optimization Problem

$$\max_{w,b} \min_{i=1,\ldots,n} \gamma^{(i)}$$

$$\Downarrow$$

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

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

Infinite solutions, as $\hat{\gamma}$ can be at any scale without changing the classifier

||w|| is not easy to deal with, non-convex objective

The Optimization Problem

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

$$\Downarrow$$

Add constraint $\hat{\gamma} = 1$

$$
\begin{aligned}
\min_{w,b} \quad & \frac{1}{2}||w||^2 \\
\text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, n
\end{aligned}
$$

Assumption: the training dataset is linearly separable

Lagrange Duality — Lagrange Multiplier

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & h_i(w) = 0, \quad i = 1, \ldots, l. \end{aligned}$$

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

Solve w, β

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

Thank You!

Q & A