Lecture 8: Machine Learning VI

Linear Models

http://cs.nju.edu.cn/yuy/course_dm13ms.ashx

Linear model

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$$

Linear model

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$$

$$w_1, w_2, \ldots, w_n \quad b$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_n \cdot x_n + b$$

Linear model

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$$

$$\boldsymbol{w} = \ w_1, w_2, \ldots, w_n \quad b$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_n \cdot x_n + b$$

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

Linear model

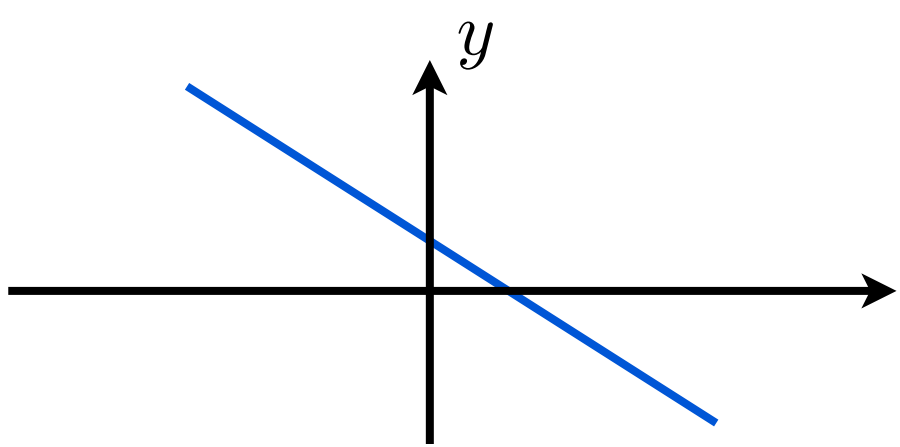
$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$$

$$\boldsymbol{w} = \ w_1, w_2, \ldots, w_n \quad b$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_n \cdot x_n + b$$

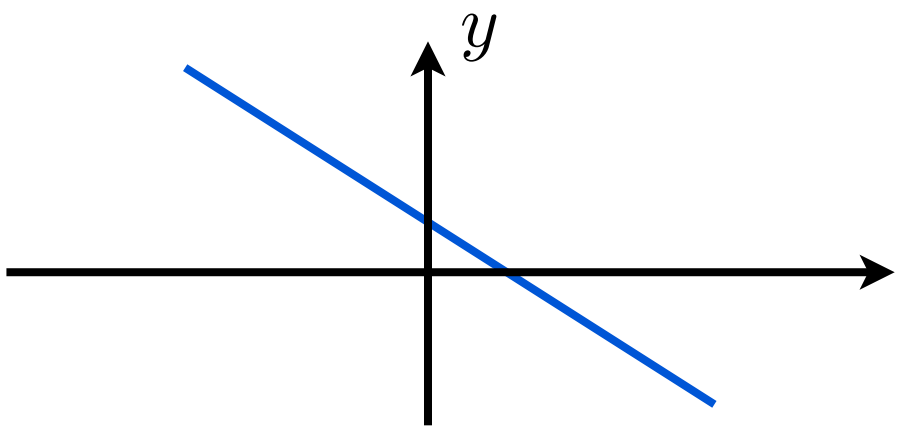
$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

$y = ax + b$

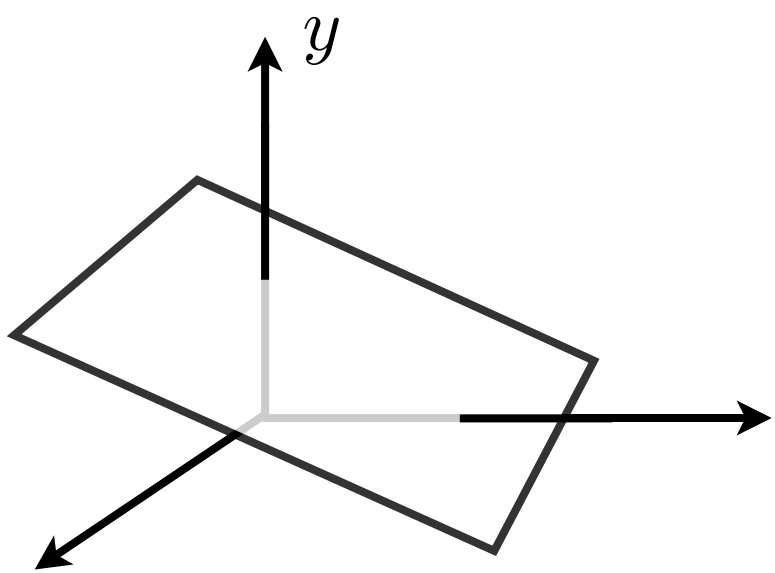


Linear model

$$y = ax + b$$

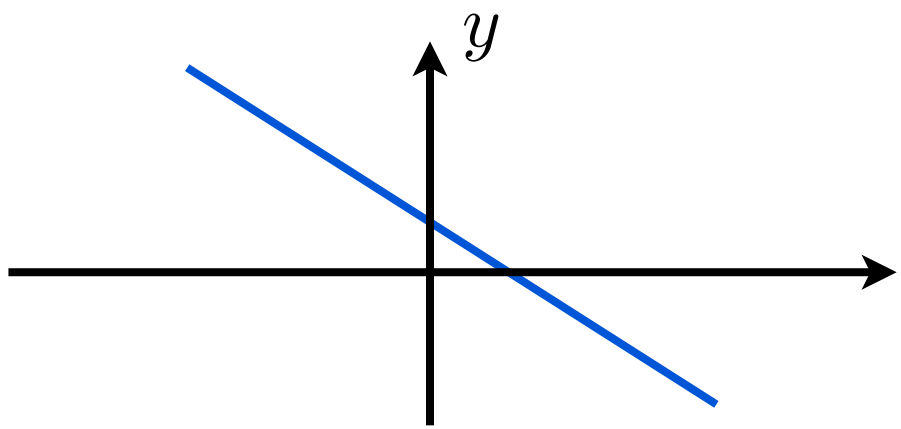


$$y = w_1 \cdot x_1 + w_2 \cdot x_2 + b$$

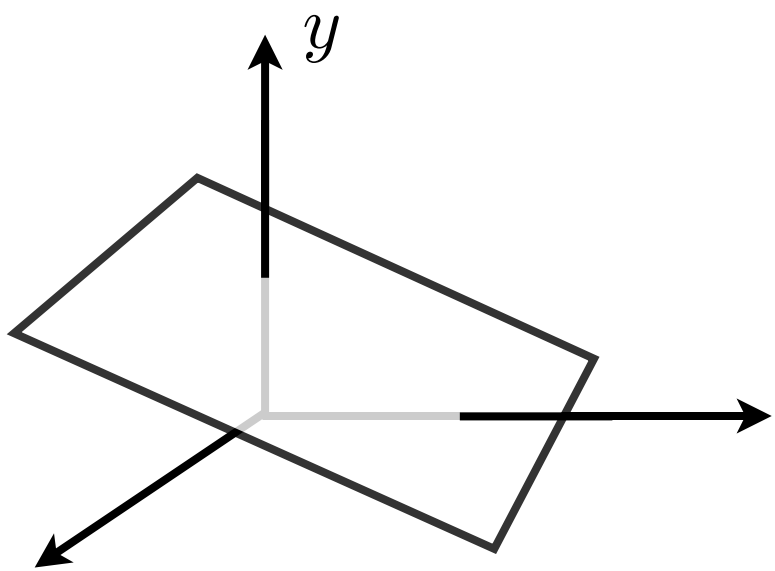


Linear model

$y = ax + b$



$y = w_1 \cdot x_1 + w_2 \cdot x_2 + b$



is the following a linear model?

$$y = w_1 \cdot x + w_2 \cdot x^2 + b$$

Linear model

$$y \qquad x_1 \quad x_2 \quad \dots \quad x_n$$

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

output/response variable

linear relationship independent parameters

basis

model space: $\mathbb{R}^{n+1}$

we sometimes omit the bias

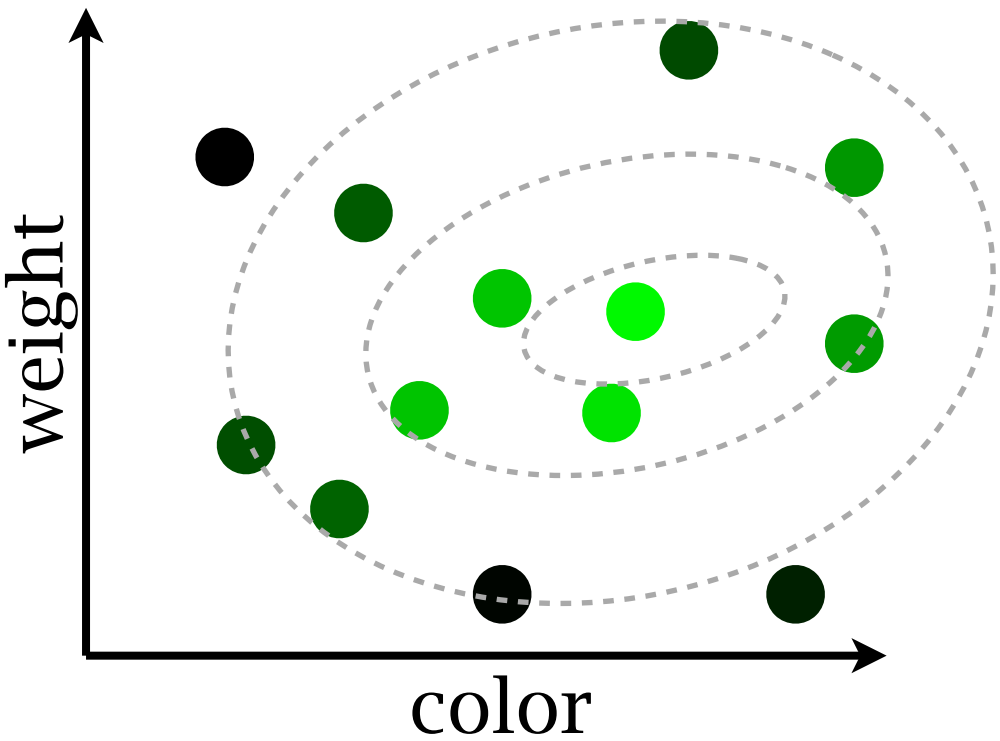
$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$$

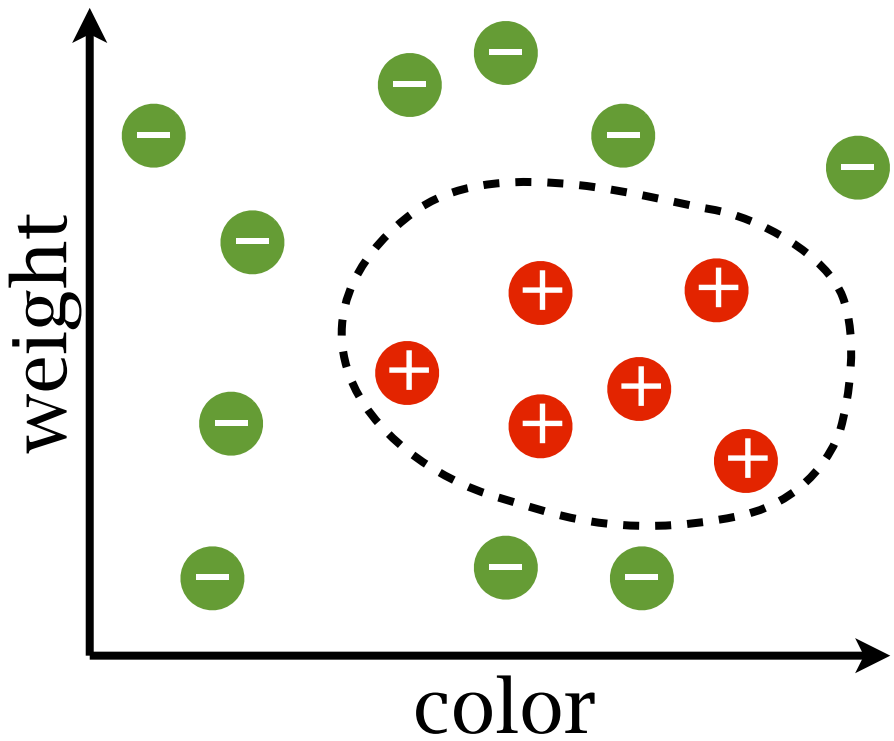
1. $\boldsymbol{x}$ is with a constant element
2. practically as good as with bias (centered data)

Learning with linear models

linear models can be used for both regression and classification tasks

suitable for continuous features





Least square regression

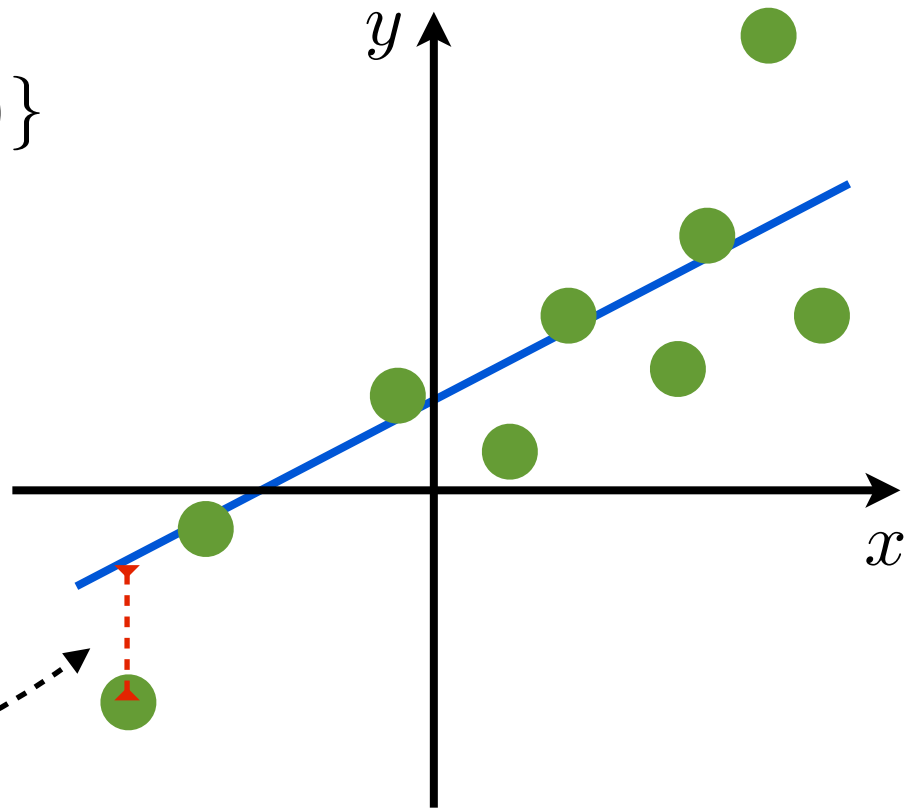
Regression: $y \in \mathbb{R}$

Training data:

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_m, y_m)\}$$

Least square loss:

$$\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$



Least square regression

$$L(\boldsymbol{w}, b) = \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$

Least square regression

$$L(\boldsymbol{w}, b) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i) = 0$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)\boldsymbol{x}_i = 0$$

Least square regression

$$L(\boldsymbol{w}, b) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i) = 0$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)\boldsymbol{x}_i = 0$$

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) = \bar{y} - \boldsymbol{w}^\top \bar{\boldsymbol{x}}$$

Least square regression

$$L(\boldsymbol{w}, b) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i) = 0$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)\boldsymbol{x}_i = 0$$

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) = \bar{y} - \boldsymbol{w}^\top \bar{\boldsymbol{x}}$$

$$\boldsymbol{w} = \left(\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top\right)^{-1}\left(\frac{1}{m}\sum_{i=1}^{m}(y_i\boldsymbol{x}_i) - \bar{y}\bar{\boldsymbol{x}}\right)$$

$$= var(\boldsymbol{x})^{-1} cov(\boldsymbol{x}, y) = (X^\top X)^{-1} X^\top Y$$

Least square regression

$$L(\boldsymbol{w}, b) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i) = 0$$

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = \frac{1}{m}\sum_{i=1}^{m} 2(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)\boldsymbol{x}_i = 0$$

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) = \bar{y} - \boldsymbol{w}^\top \bar{\boldsymbol{x}}$$

$$\boldsymbol{w} = \left(\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top\right)^{-1}\left(\frac{1}{m}\sum_{i=1}^{m}(y_i\boldsymbol{x}_i) - \bar{y}\bar{\boldsymbol{x}}\right)$$

$$= var(\boldsymbol{x})^{-1}cov(\boldsymbol{x}, y) = (X^\top X)^{-1}X^\top Y$$

closed form solution

Least absolute deviation regression

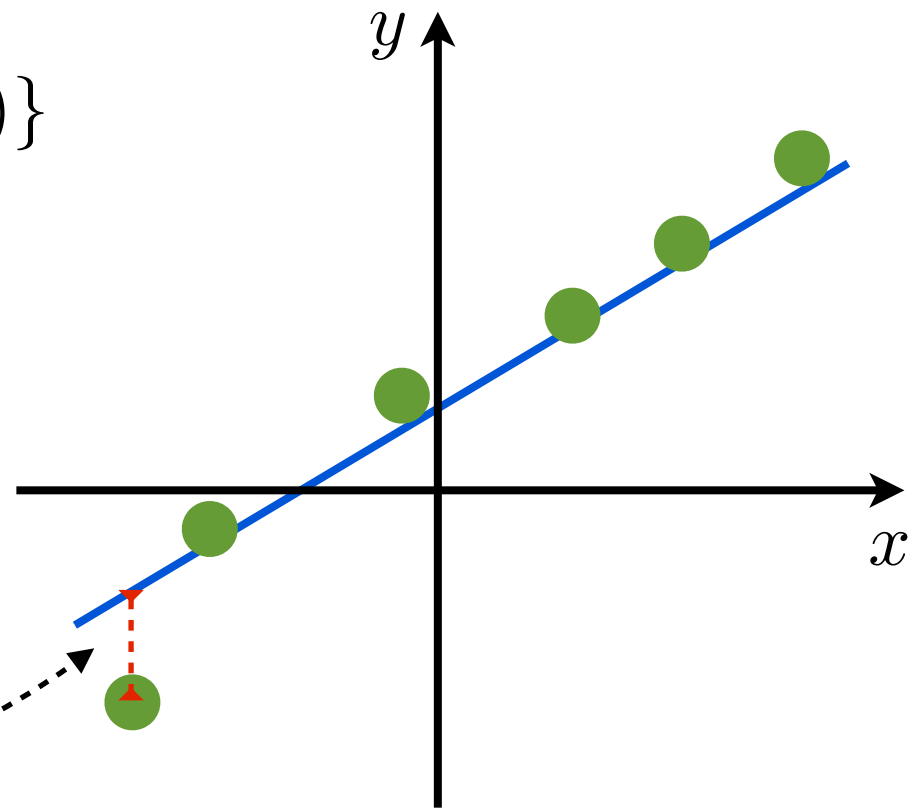
Regression: $y \in \mathbb{R}$

Training data:

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_m, y_m)\}$$

LAD loss:

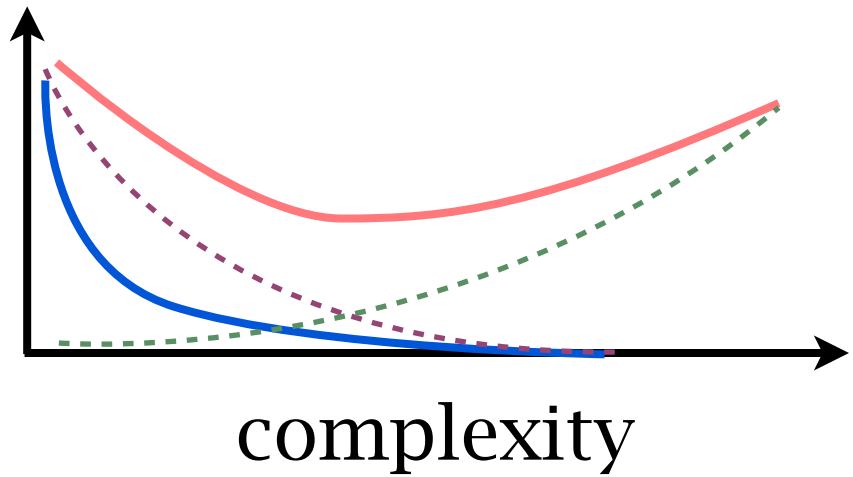
$$\frac{1}{m} \sum_{i=1}^{m} |\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i|$$



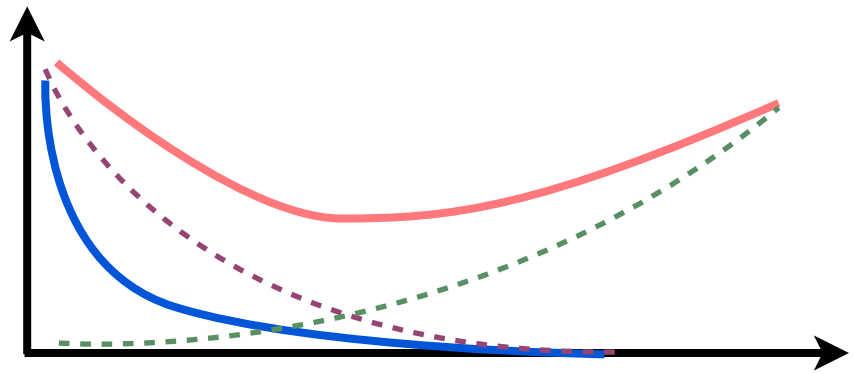
compare with least square regression:

- robust to noise
- unstable solution

Complexity of linear models



Complexity of linear models



complexity

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$$

possibility of $\boldsymbol{w}$

Regularization

make hypothesis space small

→ better generalization ability

make numerical analysis stable

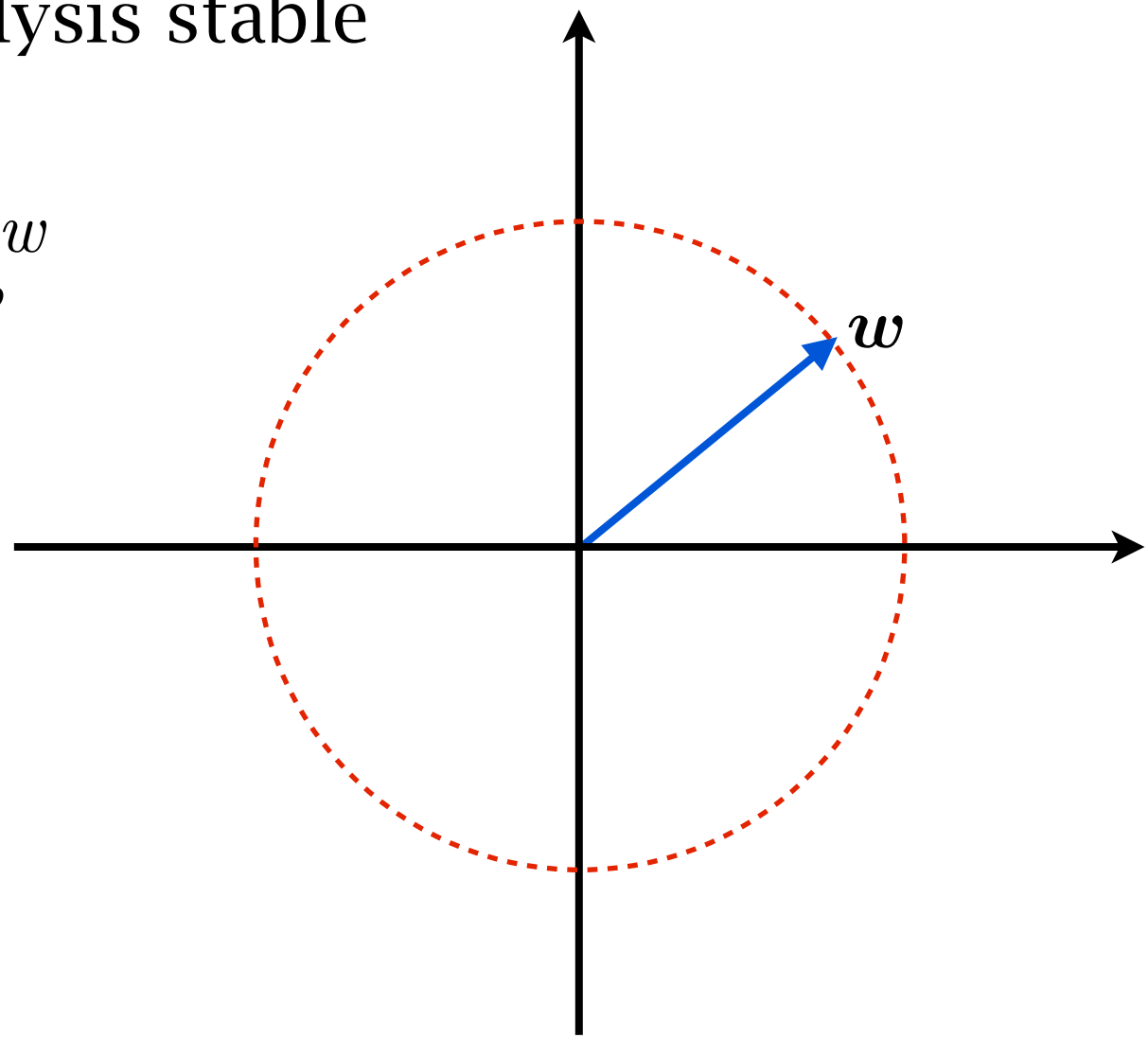
restrict the norm of w

$$\|\boldsymbol{w}\|_p = \left(\sum_{i=1}^{n} |w_i|^p \right)^{1/p}$$

$$\|\boldsymbol{w}\|_2 = \sqrt{\sum_{i=1}^{n} w_i^2}$$

$$\|\boldsymbol{w}\|_1 = \sum_{i=1}^{n} |w_i|$$

$$\|\boldsymbol{w}\|_\infty = \max_{i=1,\ldots,n} |w_i|$$



Ridge regression

Regression: $y \in \mathbb{R}$
Training data:

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_m, y_m)\}$$

objective:

$$\arg\min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$

$$s.t. \qquad \|\boldsymbol{w}\|_2 \leq \theta$$

or:

$$\arg\min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 + \lambda\|\boldsymbol{w}\|_2$$

Ridge regression

centered data, no bias:

$$\arg\min_{\boldsymbol{w}} \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda\|\boldsymbol{w}\|_2$$

closed form solution:

$$\boldsymbol{w} = \left(\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top + \lambda\boldsymbol{I}\right)^{-1}\left(\frac{1}{m}\sum_{i=1}^{m}(y_i\boldsymbol{x}_i) - \bar{y}\bar{\boldsymbol{x}}\right)$$

$$= (var(\boldsymbol{x}) + \lambda\boldsymbol{I})^{-1} cov(\boldsymbol{x}, y)$$

$$= (X^\top X + \lambda I)^{-1} X^\top Y$$

0

$\boldsymbol{I}$ is the identity matrix

Least absolute shrinkage and selection operator (LASSO)

Regression: $y \in \mathbb{R}$

Training data:

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_m, y_m)\}$$

objective:

$$\arg\min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2$$

$$s.t. \qquad \|\boldsymbol{w}\|_1 \leq \theta$$

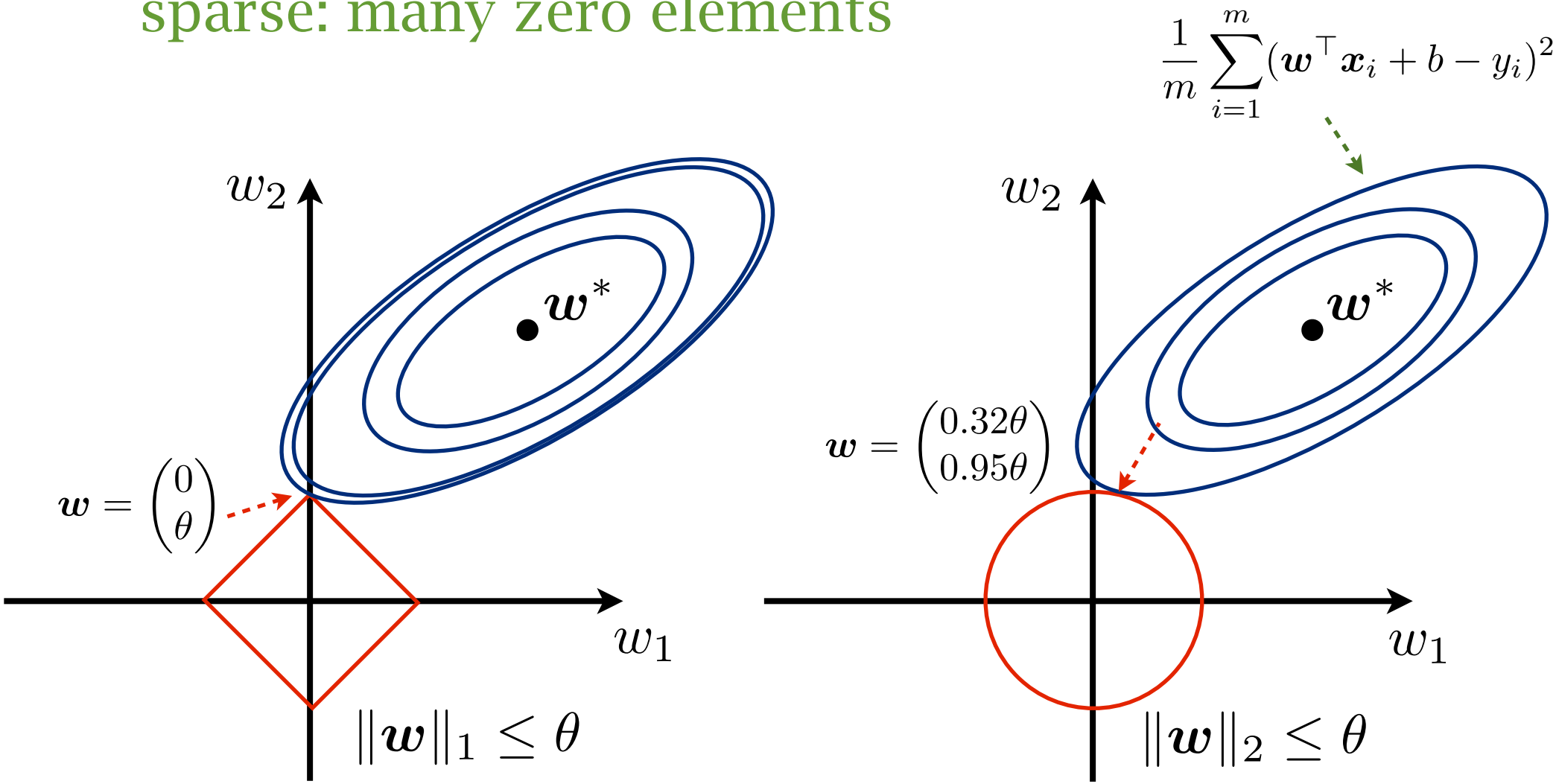
or:

$$\arg\min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 + \lambda\|\boldsymbol{w}\|_1$$

Comparing ridge regression with lasso

L1-norm leads to sparer solution, but worse empirical loss

sparse: many zero elements



A general framework

objective function:

$$\arg\min_{\boldsymbol{w},b} L(\boldsymbol{w},b) + \|\boldsymbol{w}\|_p$$

general optimization: gradient descent

$$(\boldsymbol{w},b)- = \eta \frac{\partial(L(\boldsymbol{w},b) + \|\boldsymbol{w}\|_p)}{\partial(\boldsymbol{w},b)}$$

good for convex objective functions

$$f(\alpha\boldsymbol{w}_1 + (1-\alpha)\boldsymbol{w}_2)) \geq \alpha f(\boldsymbol{w}_1) + (1-\alpha) f(\boldsymbol{w}_2)$$

linear, quadratic

convex + convex $\rightarrow$ convex

Linear classifier

model space: $\mathbb{R}^{n+1}$

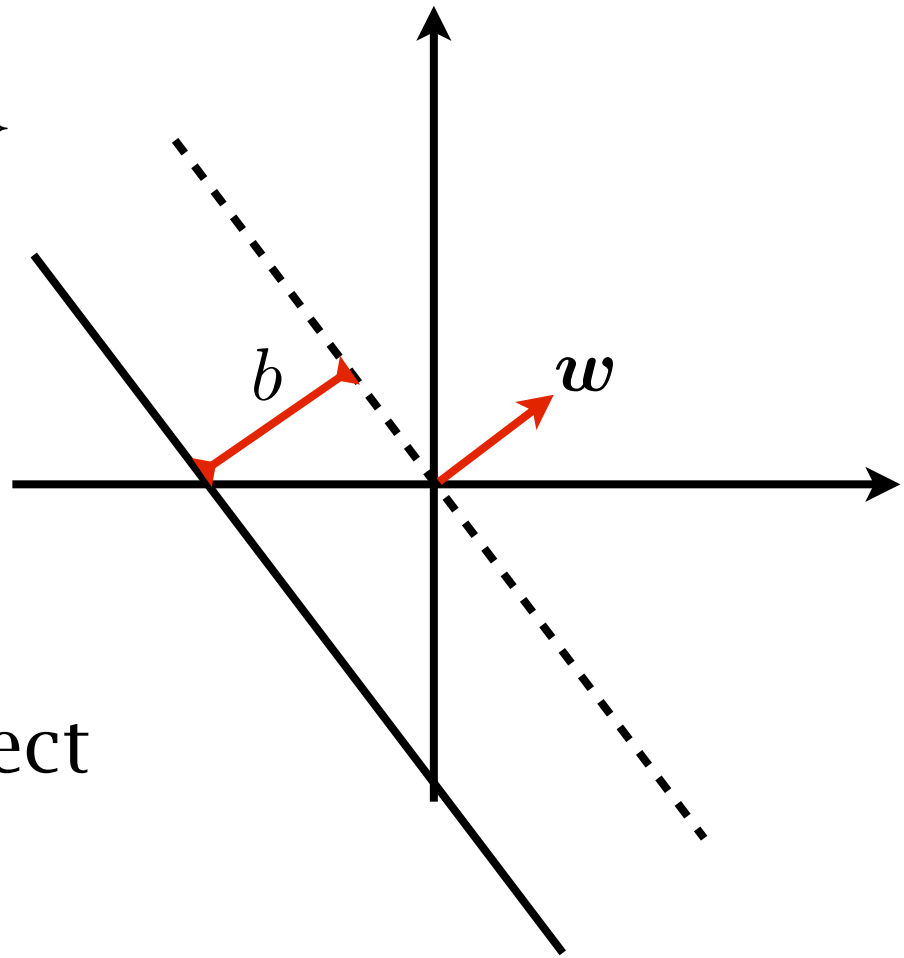
$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

for classification $y \in \{-1, +1\}$

we predict an instance by

$$\text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

$$= \begin{cases} +1, & \boldsymbol{w}^\top \boldsymbol{x} + b > 0 \\ -1, & \boldsymbol{w}^\top \boldsymbol{x} + b < 0 \\ \text{random}, & otherwise \end{cases}$$



for an example $(\boldsymbol{x}, y)$, a correct prediction means

$$y(\boldsymbol{w}^\top \boldsymbol{x} + b) > 0$$

Prototype

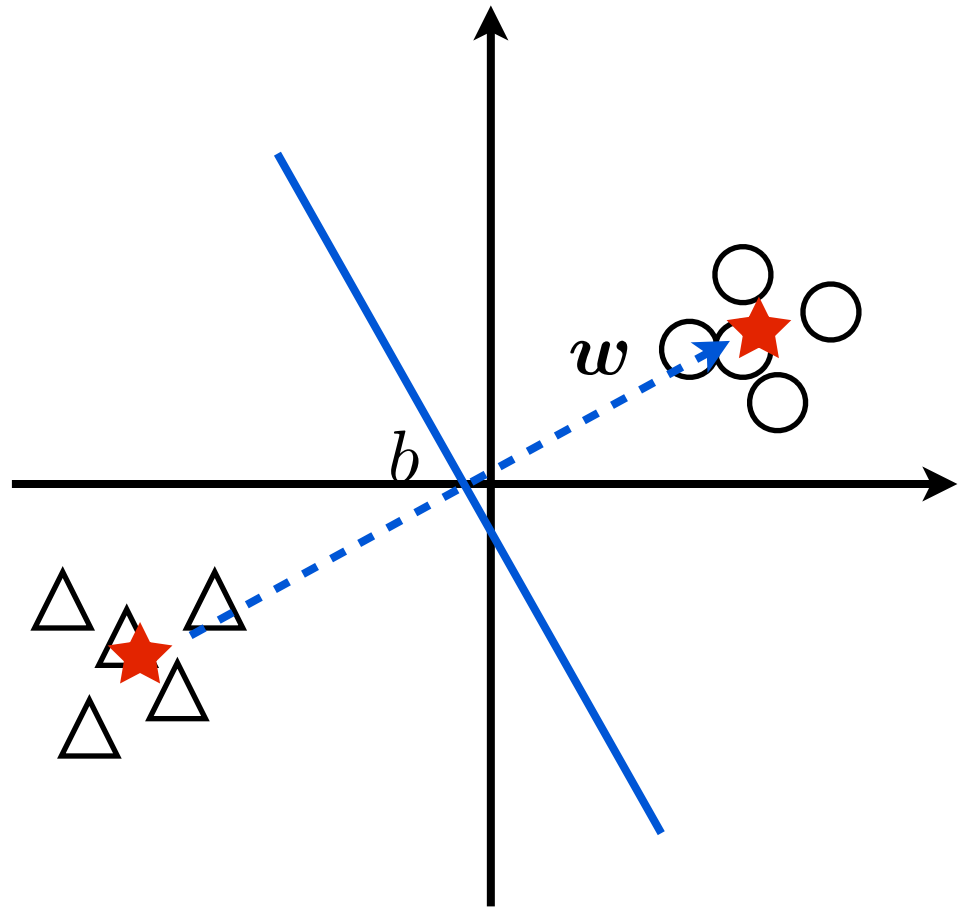
simple, but too restricted

$$\bar{\boldsymbol{x}}^{+} = \frac{1}{\sum_{i:y_i=+1} 1} \sum_{i:y_i=+1} \boldsymbol{x}_i$$

$$\bar{\boldsymbol{x}}^{-} = \frac{1}{\sum_{i:y_i=-1} 1} \sum_{i:y_i=-1} \boldsymbol{x}_i$$

$$\boldsymbol{w} = \bar{\boldsymbol{x}}^{+} - \bar{\boldsymbol{x}}^{-}$$

$$b = -\boldsymbol{w}^{\top} \cdot \frac{\bar{\boldsymbol{x}}^{+} + \bar{\boldsymbol{x}}^{-}}{2}$$

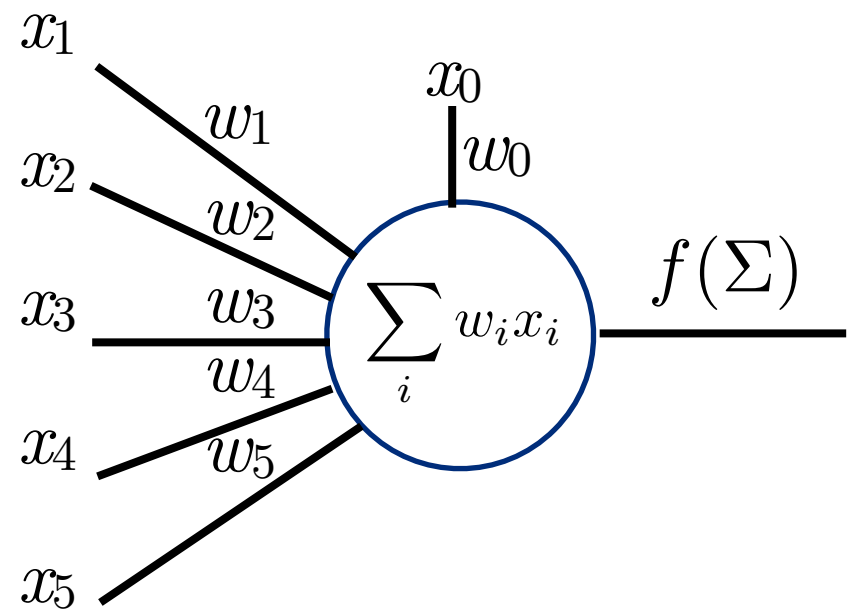


Perceptron

feed training examples one by one

1. $\boldsymbol{w} = 0$
2. for each example $(\boldsymbol{x}, y)$
   if $\text{sign}(y\boldsymbol{w}^\top \boldsymbol{x}) < 0$

$$\boldsymbol{w} = \boldsymbol{w} + y\boldsymbol{x}$$



$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

Perceptron

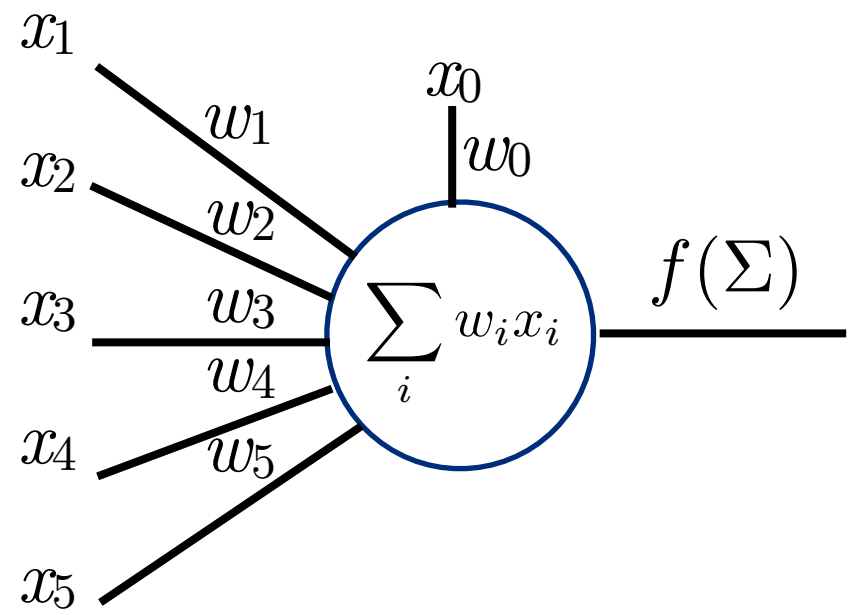
feed training examples one by one

1. $\boldsymbol{w} = 0$
2. for each example $(\boldsymbol{x}, y)$
   if $\text{sign}(y\boldsymbol{w}^\top \boldsymbol{x}) < 0$

$$\boldsymbol{w} = \boldsymbol{w} + y\boldsymbol{x}$$

gradient ascent

$$\frac{\partial y\boldsymbol{w}^\top \boldsymbol{x}}{\partial \boldsymbol{w}} = y\boldsymbol{x}$$



$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

Perceptron

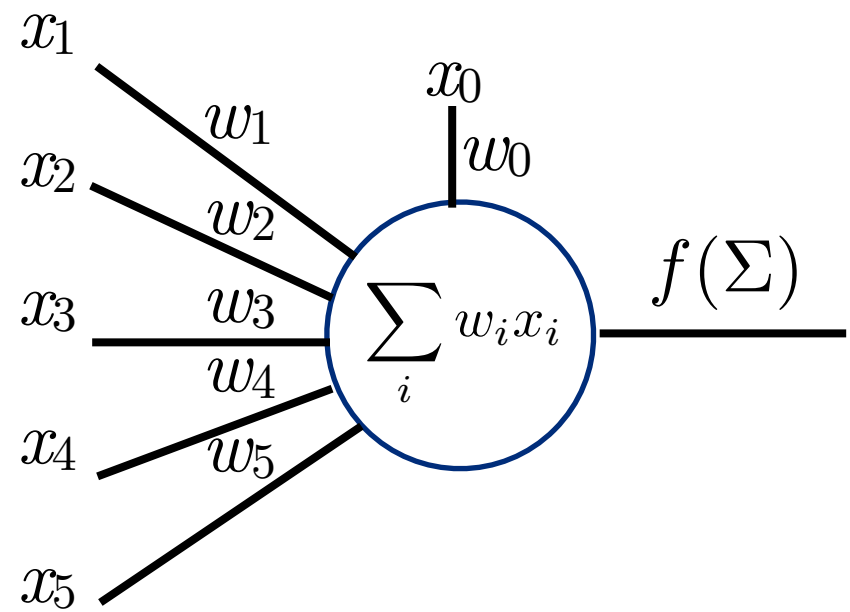
feed training examples one by one

1. $\boldsymbol{w} = 0$
2. for each example $(\boldsymbol{x}, y)$
   if $\text{sign}(y\boldsymbol{w}^\top \boldsymbol{x}) < 0$

$$\boldsymbol{w} = \boldsymbol{w} + y\boldsymbol{x}$$

gradient ascent

$$\frac{\partial y\boldsymbol{w}^\top \boldsymbol{x}}{\partial \boldsymbol{w}} = y\boldsymbol{x}$$



$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

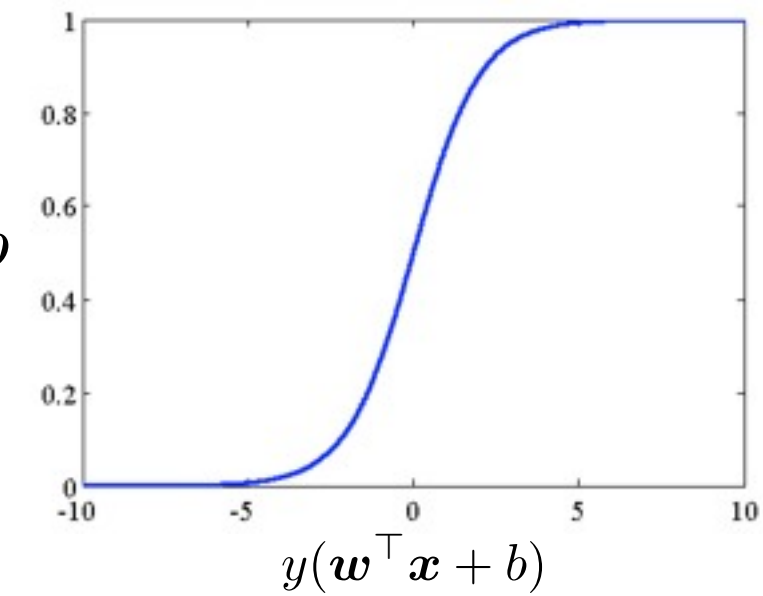
when all examples are with length 1 and are linearly separable by w^*, perceptron algorithm makes at most $\left(1/\min_{\boldsymbol{x}} \frac{|\boldsymbol{w}^{*\top}\boldsymbol{x}|}{\|\boldsymbol{x}\|_2}\right)^2$ mistakes

Logistic regression

assume logit model: for a positive example

$$\boldsymbol{w}^\top \boldsymbol{x} = \log \frac{p(+1 \mid \boldsymbol{x})}{1 - p(+1 \mid \boldsymbol{x})}$$

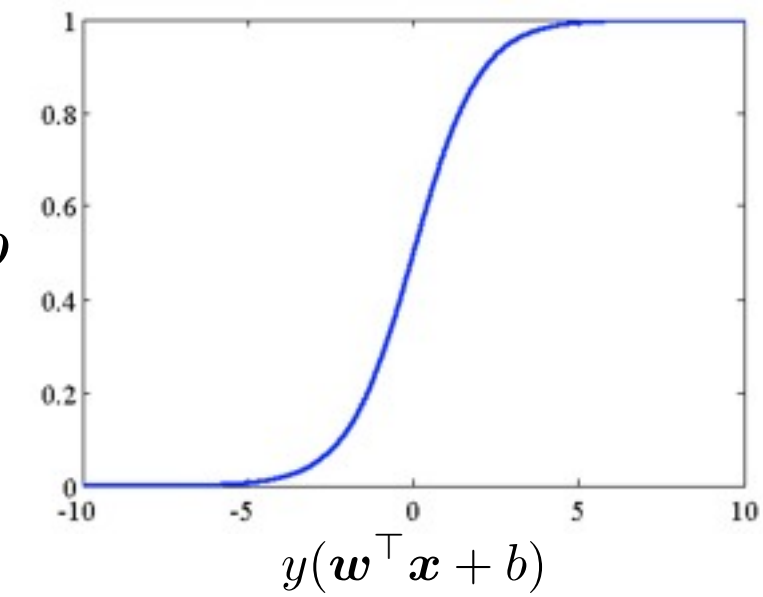
so that $p(y \mid \boldsymbol{x}, \boldsymbol{w}) = \dfrac{1}{1 + e^{-y(\boldsymbol{w}^\top \boldsymbol{x})}}$



Logistic regression

assume logit model: for a positive example

$$\boldsymbol{w}^\top \boldsymbol{x} = \log \frac{p(+1 \mid \boldsymbol{x})}{1 - p(+1 \mid \boldsymbol{x})}$$

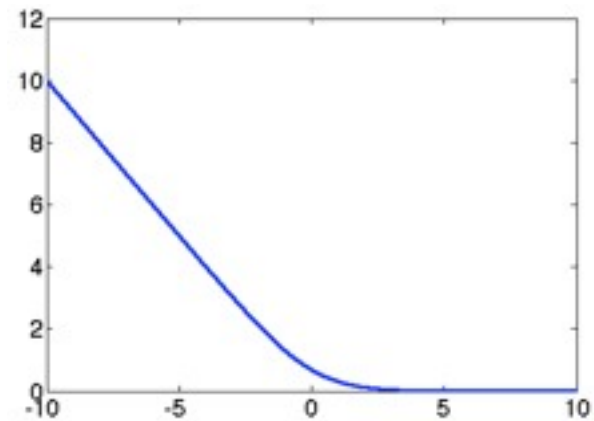


so that $p(y \mid \boldsymbol{x}, \boldsymbol{w}) = \dfrac{1}{1 + e^{-y(\boldsymbol{w}^\top \boldsymbol{x})}}$

minimize negative log-likelihood:

$$\arg\min_{\boldsymbol{w},b} -\log \prod_{i=1}^{m} p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}) = -\sum_i \log p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})$$

$$= \sum_i \log \left(1 + e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i)}\right)$$



convex

Logistic regression

Maximize a posterior (minimize negative a posterior)

$$\arg\min_{\boldsymbol{w},b} -\log\left(\prod_{i=1}^{m} p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})\right) p(\boldsymbol{w})$$

a prior: $\boldsymbol{w} \sim \mathcal{N}(0, \delta \boldsymbol{I})$

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; 0, \delta \boldsymbol{I}) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{\|\boldsymbol{w}-0\|_2^2}{2\delta^2}}$$

$$= -\sum_i \log p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}) - \log p(\boldsymbol{w})$$

$$= \sum_i \log\left(1 + e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i)}\right) + \frac{1}{2\delta^2}\|\boldsymbol{w}\|_2^2 + \text{const}$$

Logistic regression

Maximize a posterior (minimize negative a posterior)

$$\arg\min_{\boldsymbol{w},b} -\log\left(\prod_{i=1}^{m} p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})\right) p(\boldsymbol{w})$$

a prior: $\boldsymbol{w} \sim \mathcal{N}(0, \delta \boldsymbol{I})$

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; 0, \delta \boldsymbol{I}) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{\|\boldsymbol{w}-0\|_2^2}{2\delta^2}}$$

$$= -\sum_i \log p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}) - \log p(\boldsymbol{w})$$

$$= \sum_i \log\left(1 + e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i)}\right) + \frac{1}{2\delta^2}\|\boldsymbol{w}\|_2^2 + \text{const}$$

convex

regularized logistic regression

Linear classifier revisit

model space: $\mathbb{R}^{n+1}$

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

for classification $y \in \{-1, +1\}$

Original objective:

$$\arg\min_{\boldsymbol{w},b} \sum_i I(y(\boldsymbol{w}^\top \boldsymbol{x} + b) \leq 0)$$

0-1 loss
hard to optimize

Surrogate objective:

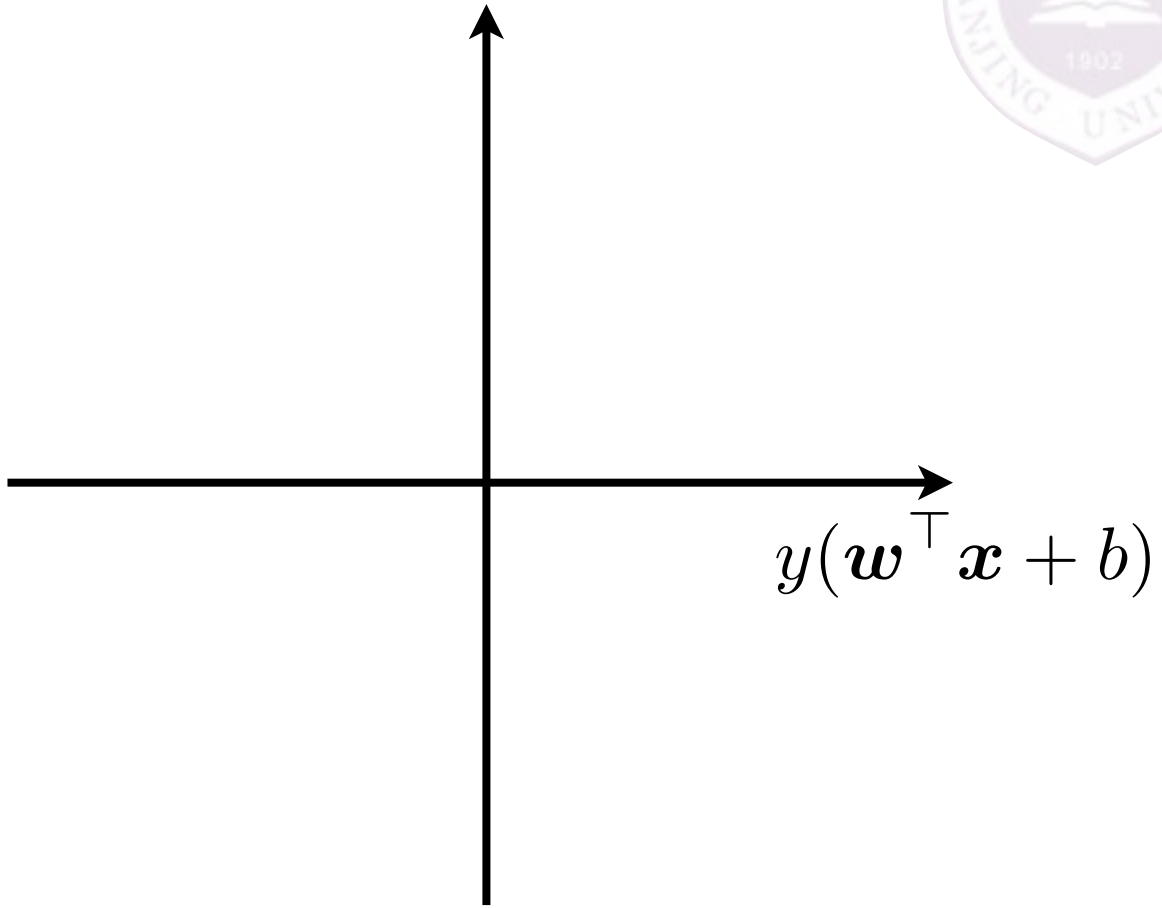
$$\arg\min_{\boldsymbol{w},b} \sum_i \log\left(1 + e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)}\right)$$

logistic regression

$$\arg\min_{\boldsymbol{w},b} \sum_i \max\{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b), 0\}$$

perceptron

Linear classifier revisit



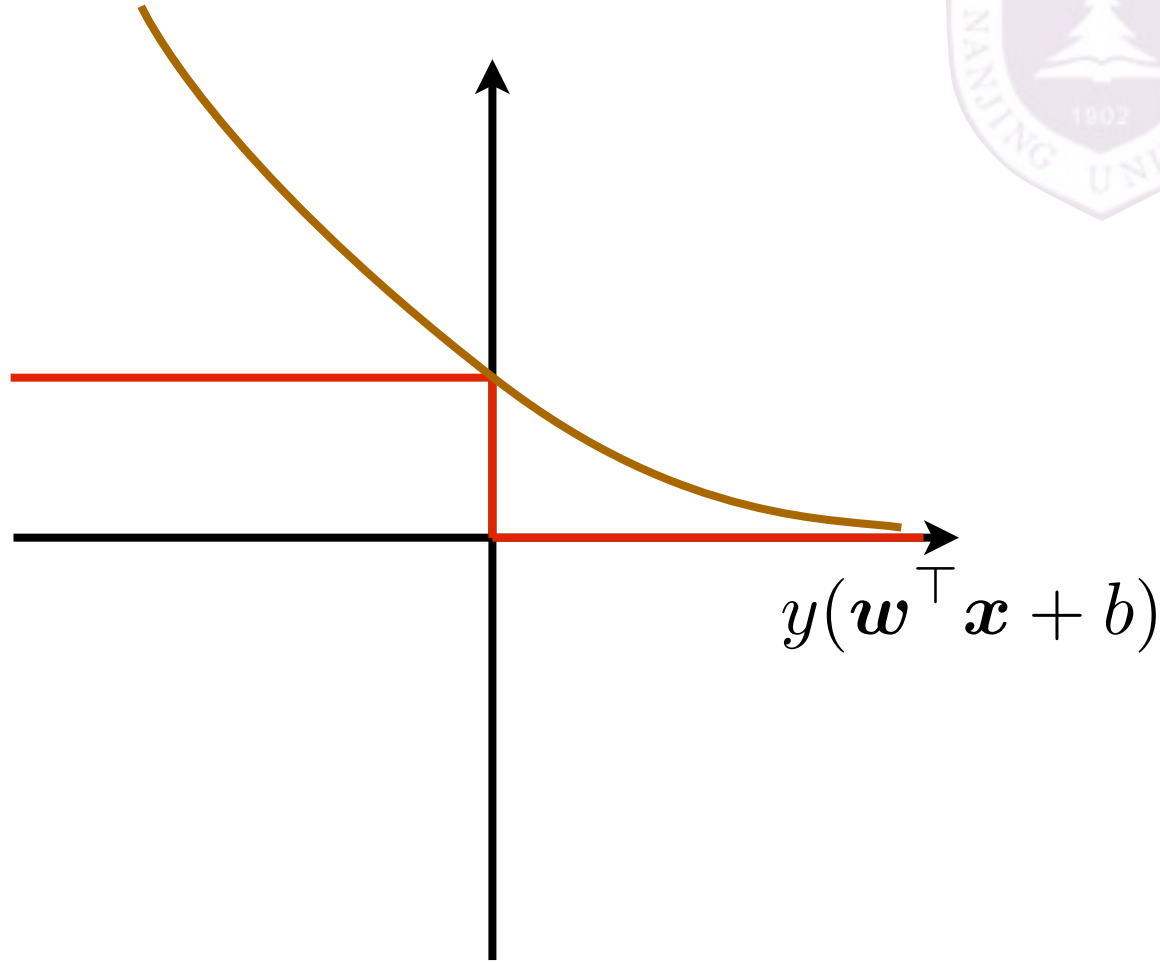
Linear classifier revisit

0-1 loss

$I(y(\boldsymbol{w}^\top \boldsymbol{x} + b) \leq 0)$

logistic regression

$\log_2(1 + e^{-y(\boldsymbol{w}^\top \boldsymbol{x} + b)})$



Linear classifier revisit

0-1 loss

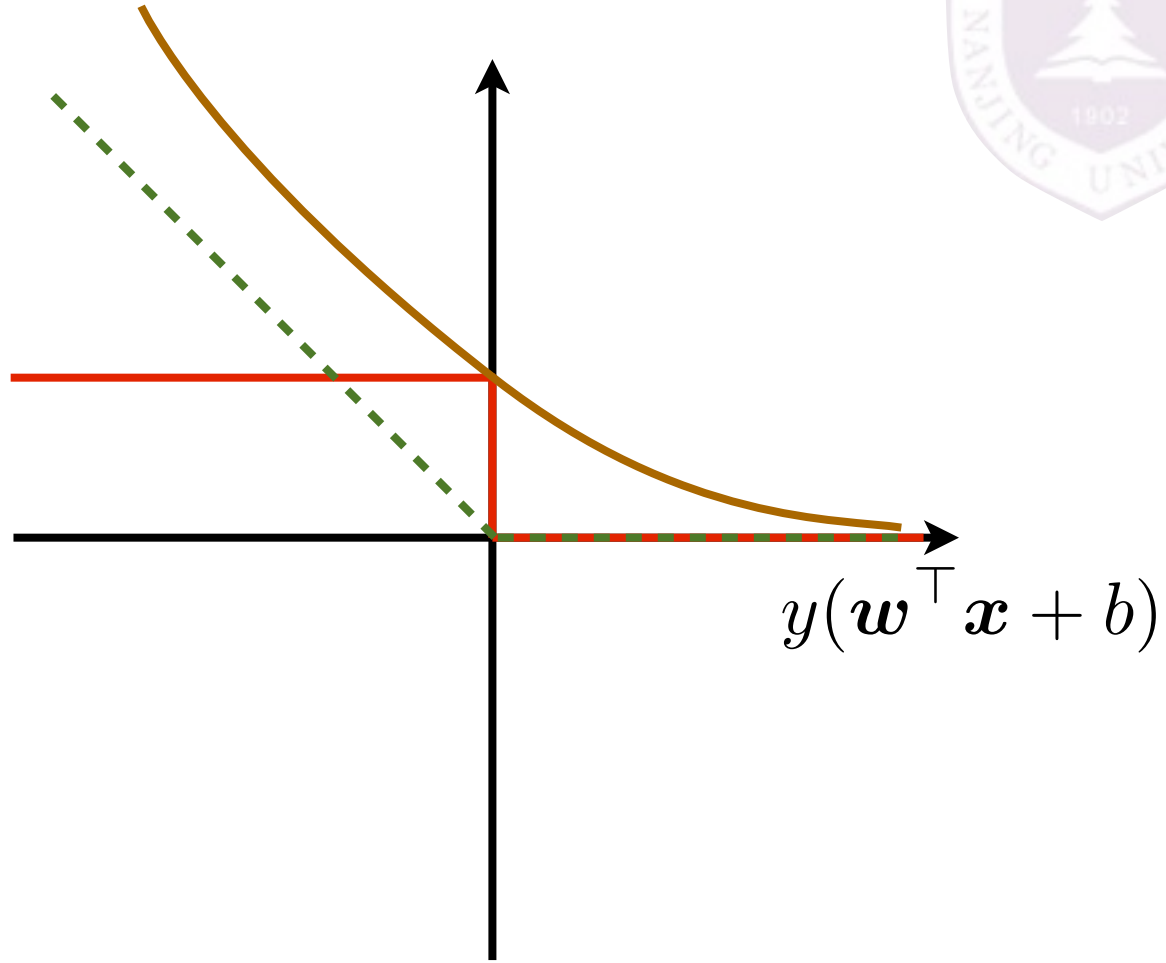
$I(y(\boldsymbol{w}^\top \boldsymbol{x} + b) \leq 0)$

logistic regression

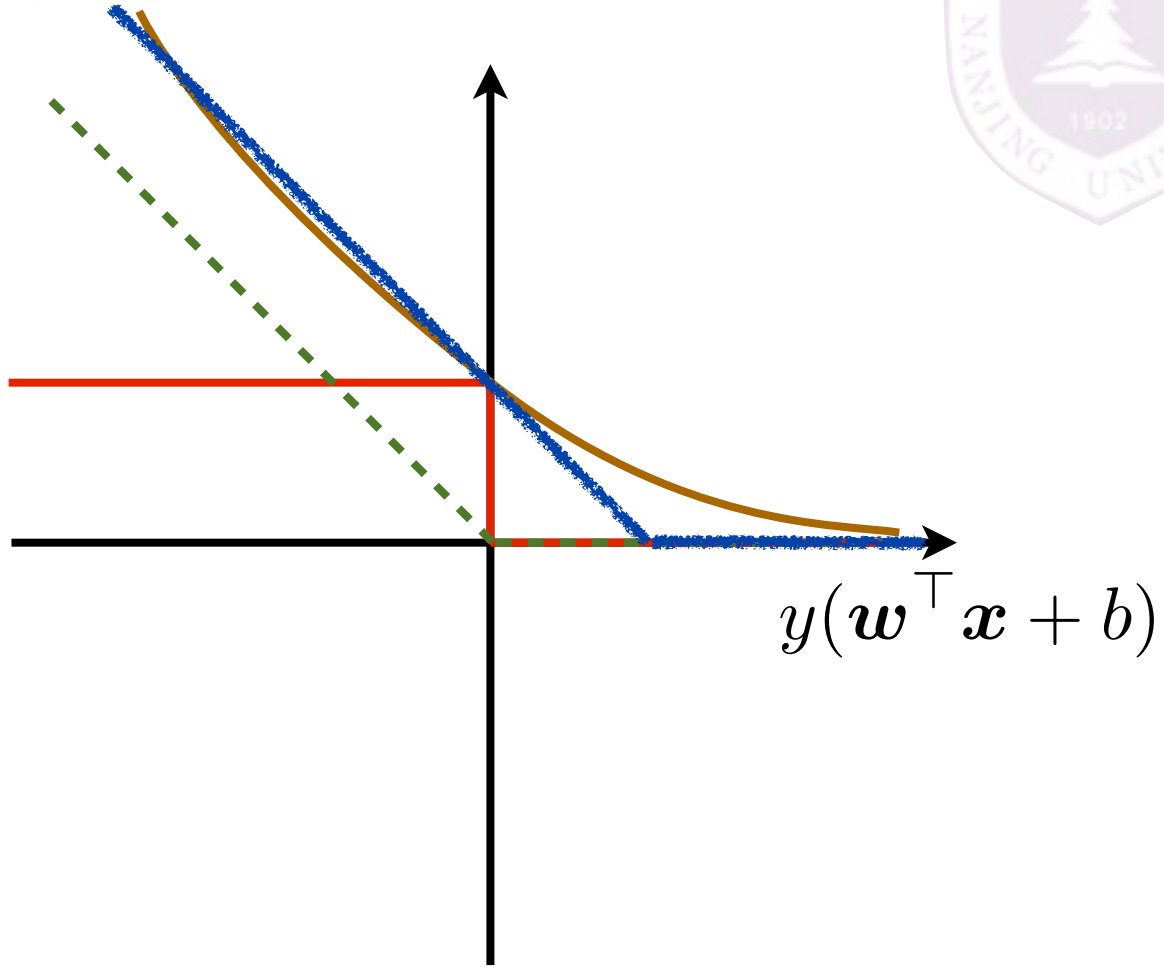
$\log_2(1 + e^{-y(\boldsymbol{w}^\top \boldsymbol{x} + b)})$

perceptron

$\max\{-y(\boldsymbol{w}^\top \boldsymbol{x} + b), 0\}$



Linear classifier revisit



0-1 loss

$I(y(\boldsymbol{w}^\top \boldsymbol{x} + b) \leq 0)$

logistic regression

$\log_2(1 + e^{-y(\boldsymbol{w}^\top \boldsymbol{x} + b)})$

perceptron

$\max\{-y(\boldsymbol{w}^\top \boldsymbol{x} + b), 0\}$

hinge loss

$\max\{1 - y(\boldsymbol{w}^\top \boldsymbol{x} + b), 0\}$

Support vector machines (SVM)

hinge loss + L2-norm

$$\arg\min_{\boldsymbol{w},b} \sum_i \max(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b), 0) + \lambda \|\boldsymbol{w}\|_2$$

Support vector machines (SVM)

hinge loss + L2-norm

$$\arg\min_{\boldsymbol{w},b} \sum_i \max(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b), 0) + \lambda \|\boldsymbol{w}\|_2$$

$$\max(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b), 0) = \xi_i$$
$$\xi_i \geq 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)$$
$$\xi_i \geq 0$$

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i$$

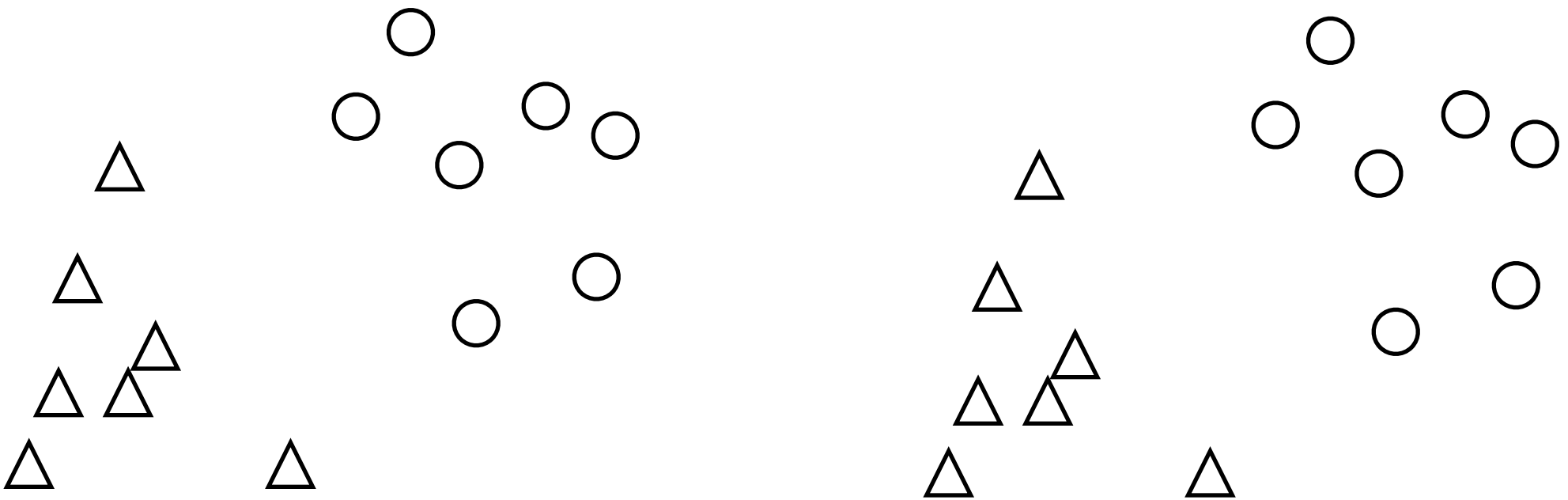
$$s.t. \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

quadratic

Support vector machines (SVM)

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

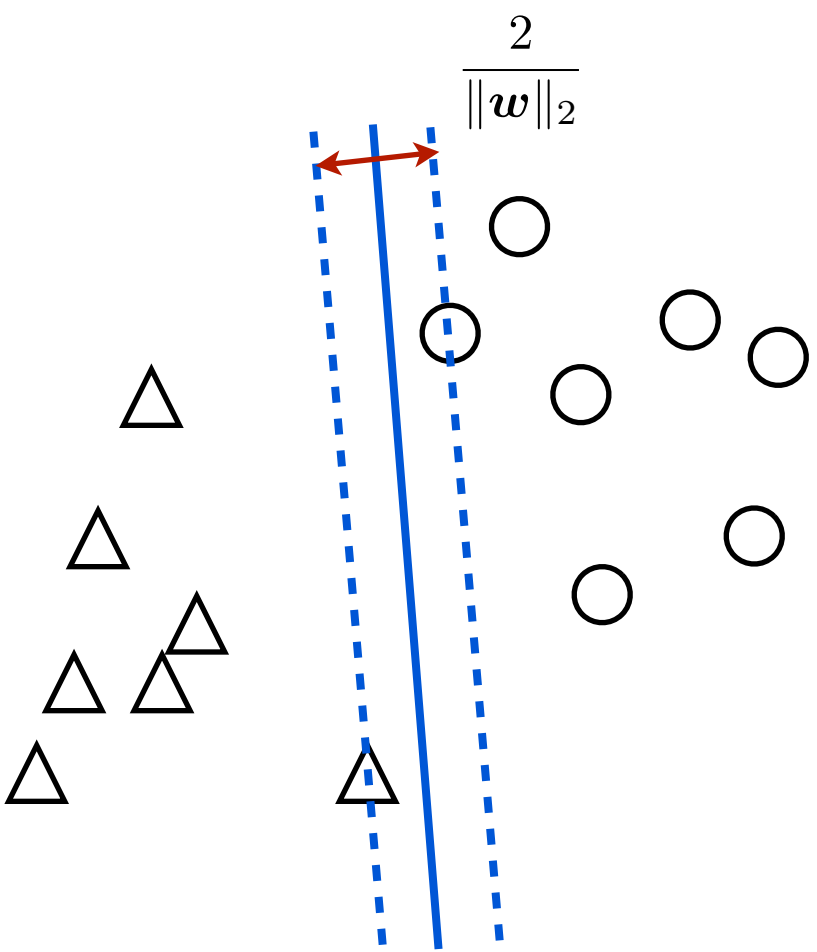
$$s.t. \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$$

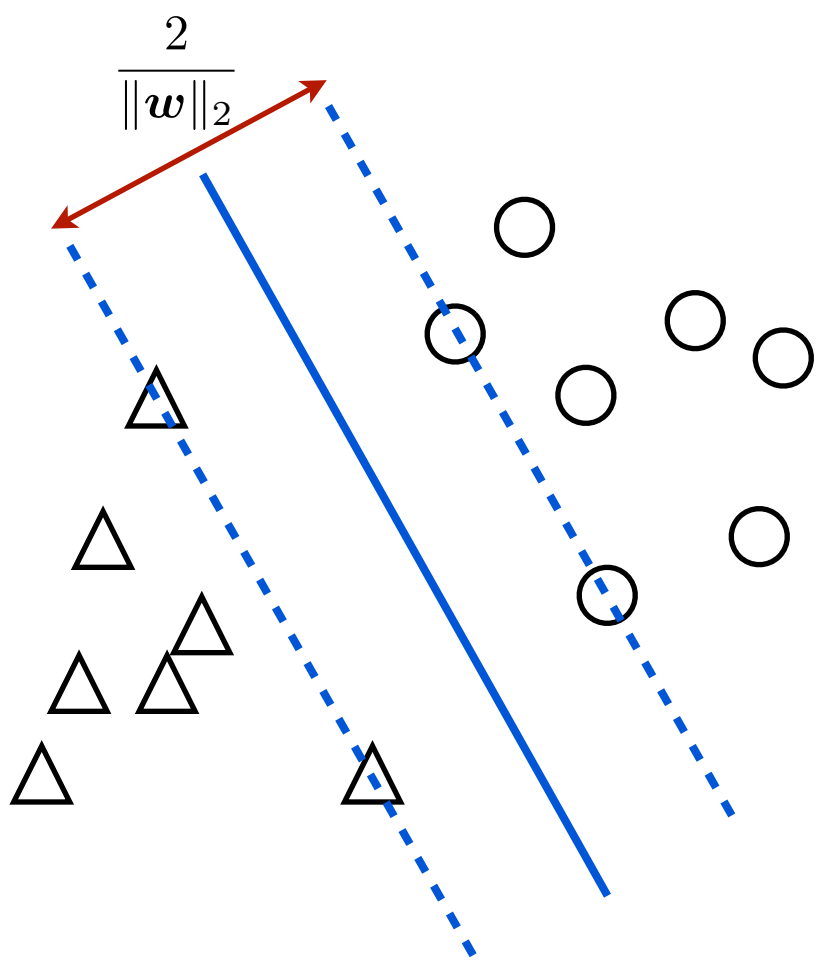


Support vector machines (SVM)

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$s.t. \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$$

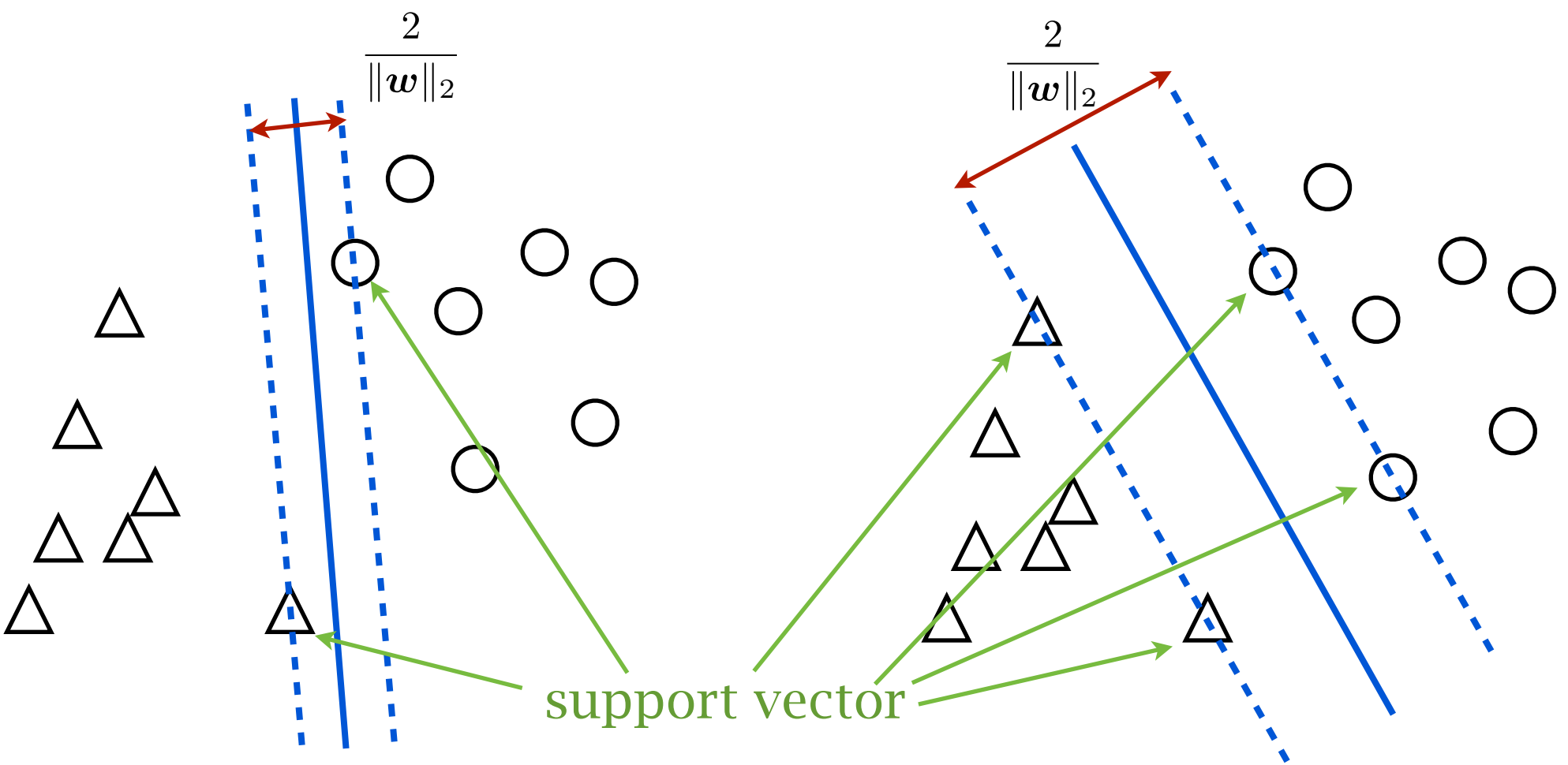




Support vector machines (SVM)

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$s.t. \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$$



Support vector machines (SVM)

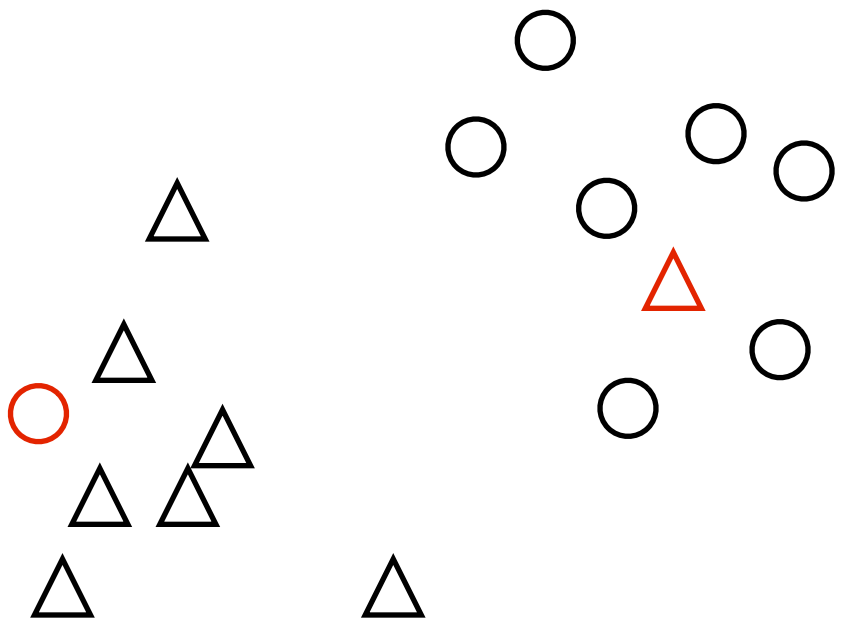
$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$s.t. \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$$

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i$$

$$s.t. \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$



Support vector machines (SVM)

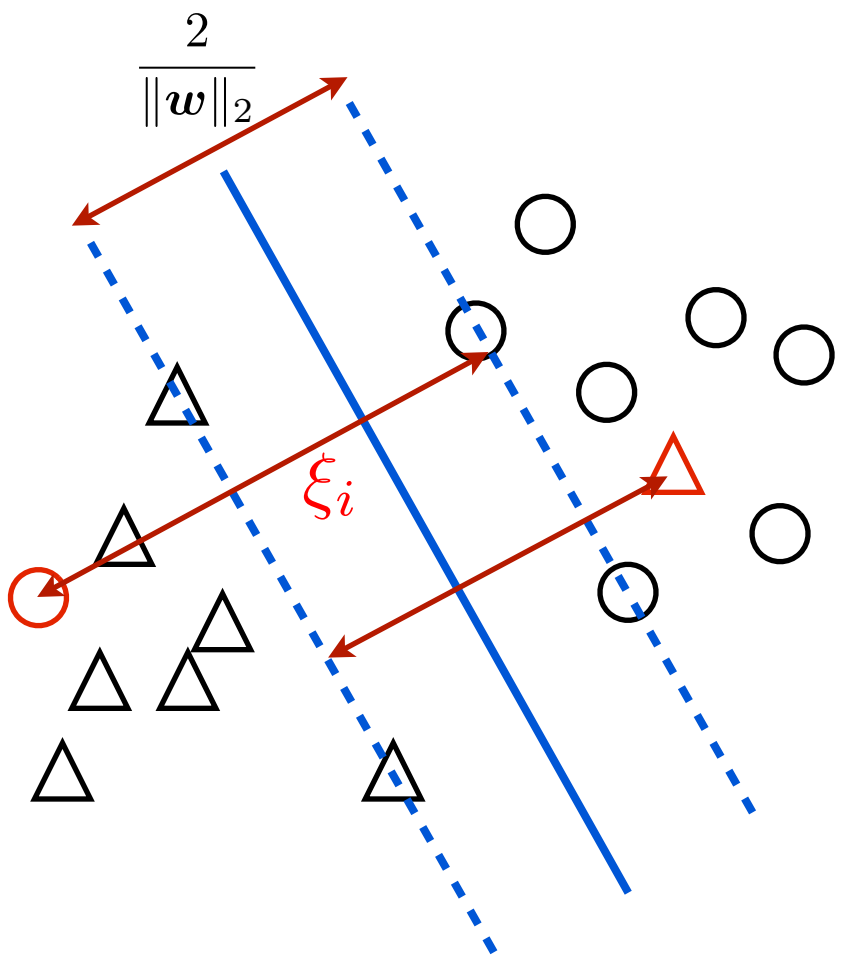
$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$s.t. \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$$

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i$$

$$s.t. \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$



slack variables

Scoring functions

$$\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 \quad \text{least square regression}$$

$$\frac{1}{m}\sum_{i=1}^{m}|\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i| \quad \text{LAD regression}$$

$$\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 + \lambda\|\boldsymbol{w}\|_2 \quad \text{ridge regression}$$

$$\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 + \lambda\|\boldsymbol{w}\|_1 \quad \text{LASSO}$$

Scoring functions

$$\sum_i I(y(\boldsymbol{w}^\top \boldsymbol{x} + b) > 0)$$ 0-1 loss

$$\sum_i \max\{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b), 0\}$$ perceptron

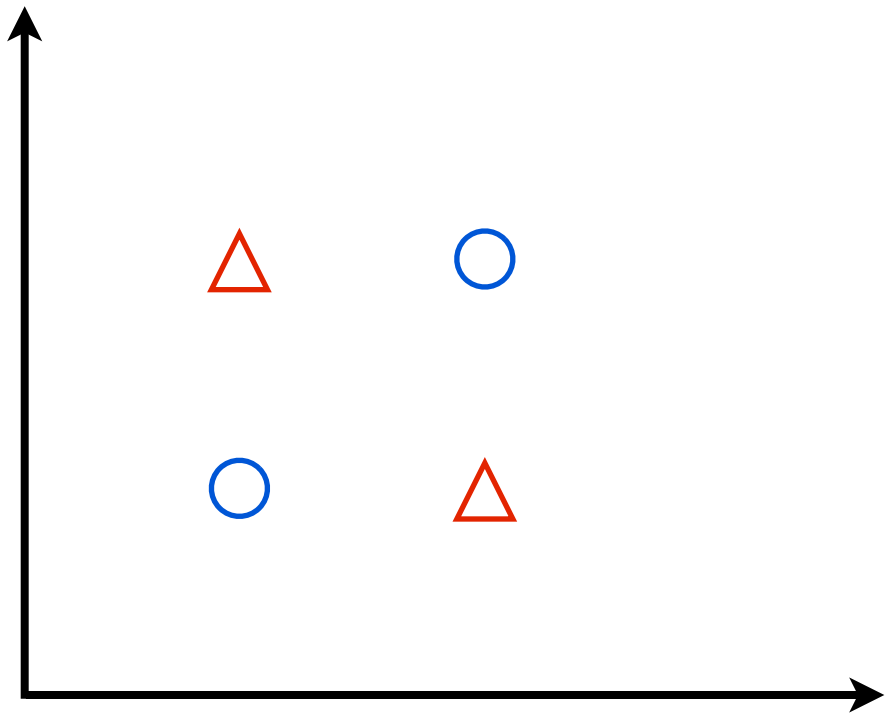
$$\sum_i \log\left(1 + e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)}\right)$$ logistic regression

$$\sum_i \log\left(1 + e^{-y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)}\right) + \lambda\|\boldsymbol{w}\|_2$$ regularized LR

$$\sum_i \max(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b), 0) + \lambda\|\boldsymbol{w}\|_2$$ SVM

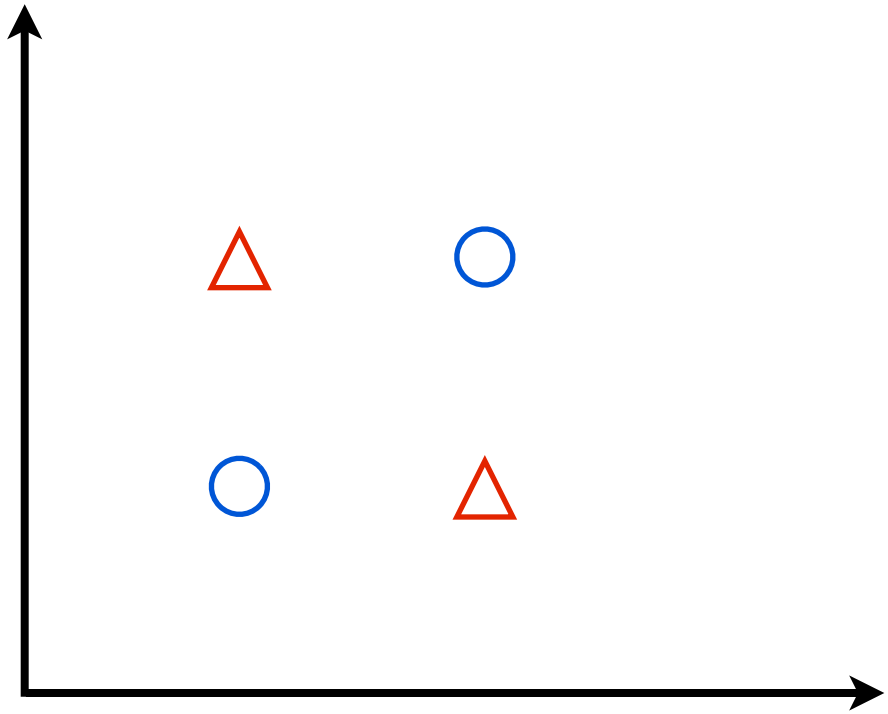
minimize loss + regularization

Linearity v.s. dimensionality

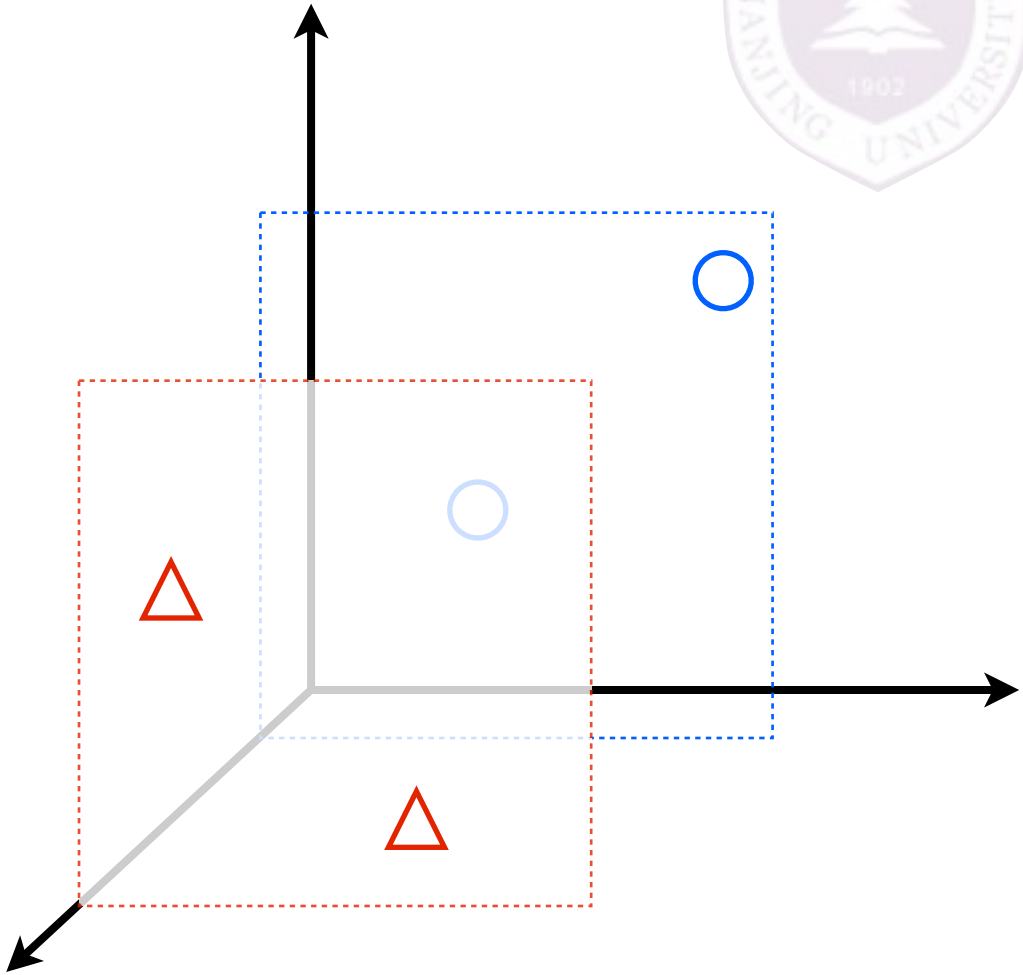


XOR in 2D

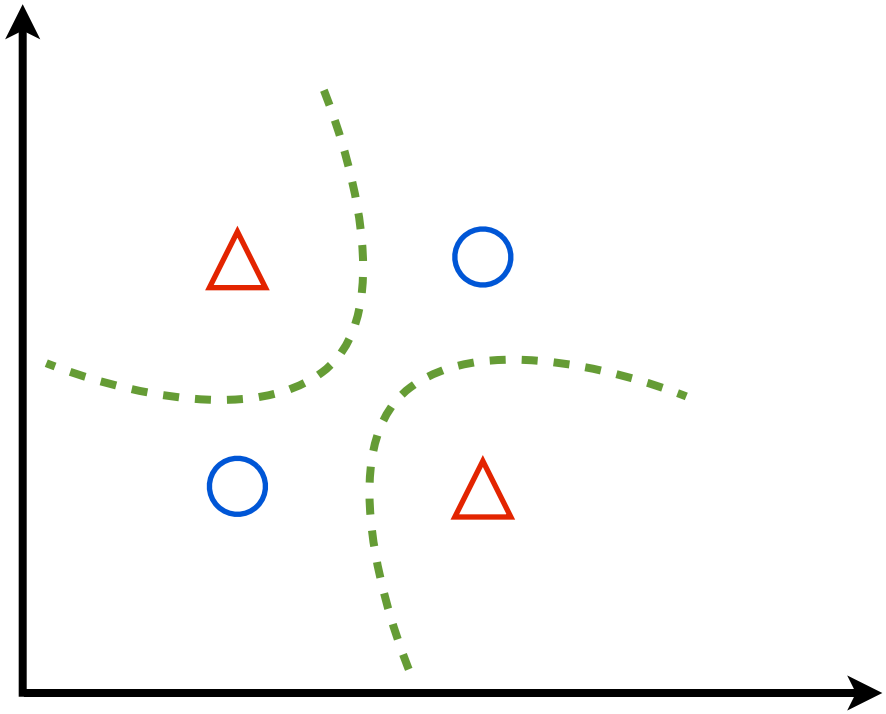
Linearity v.s. dimensionality



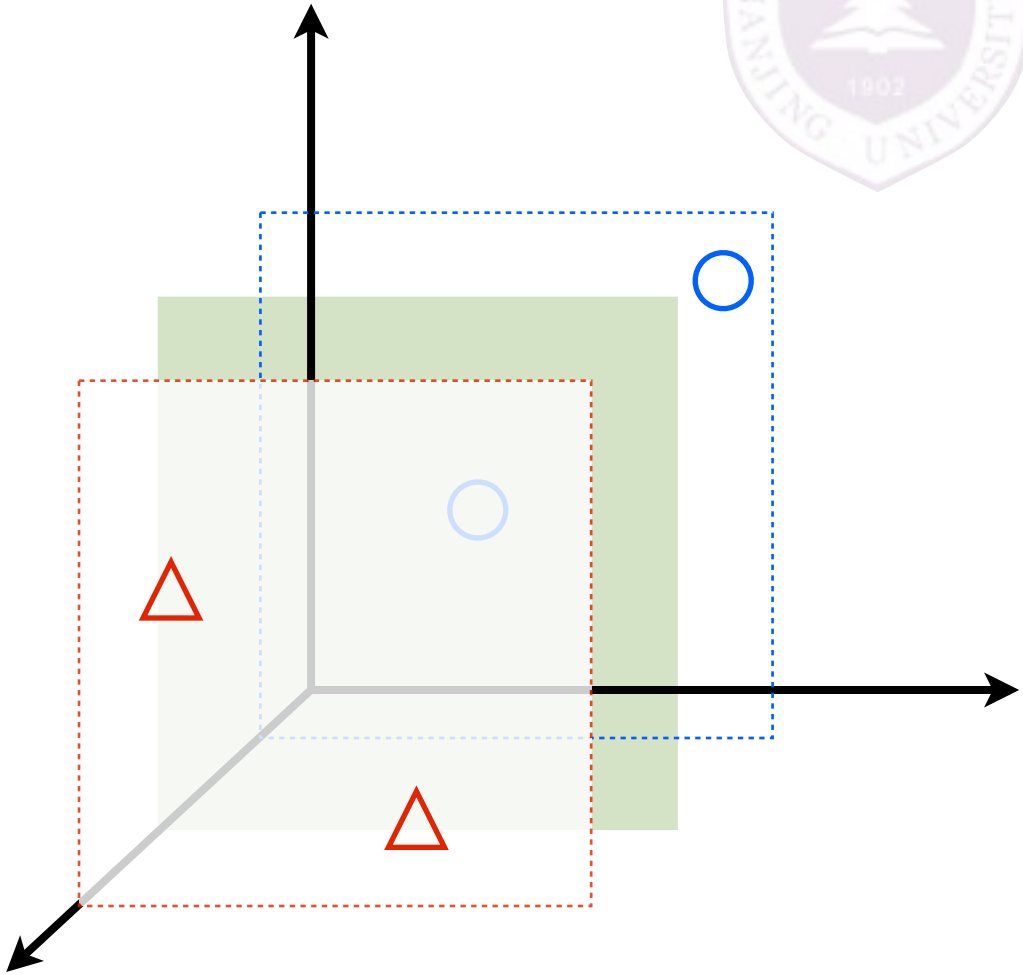
XOR in 2D



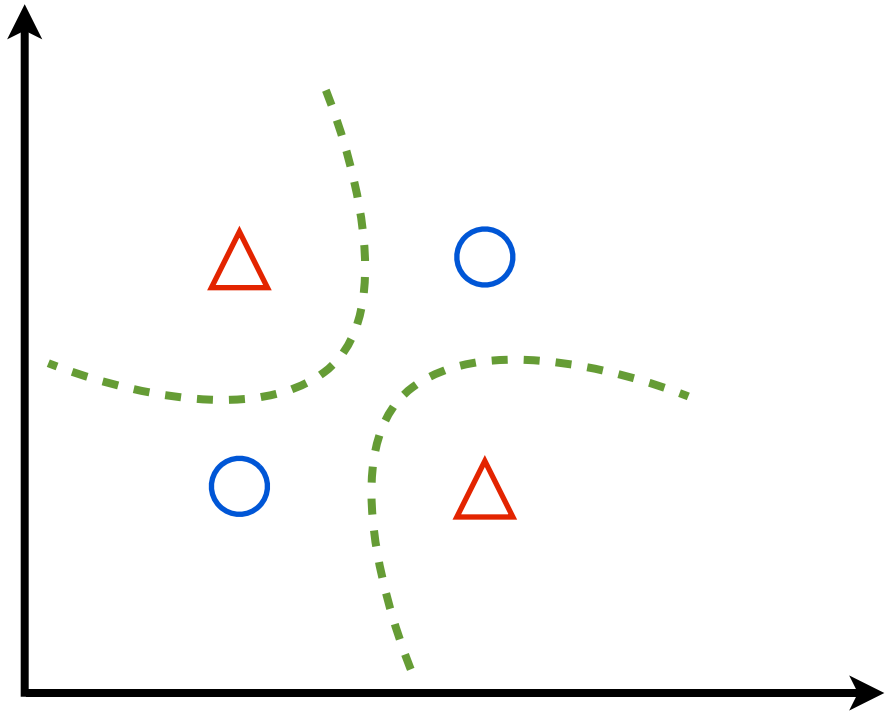
Linearity v.s. dimensionality



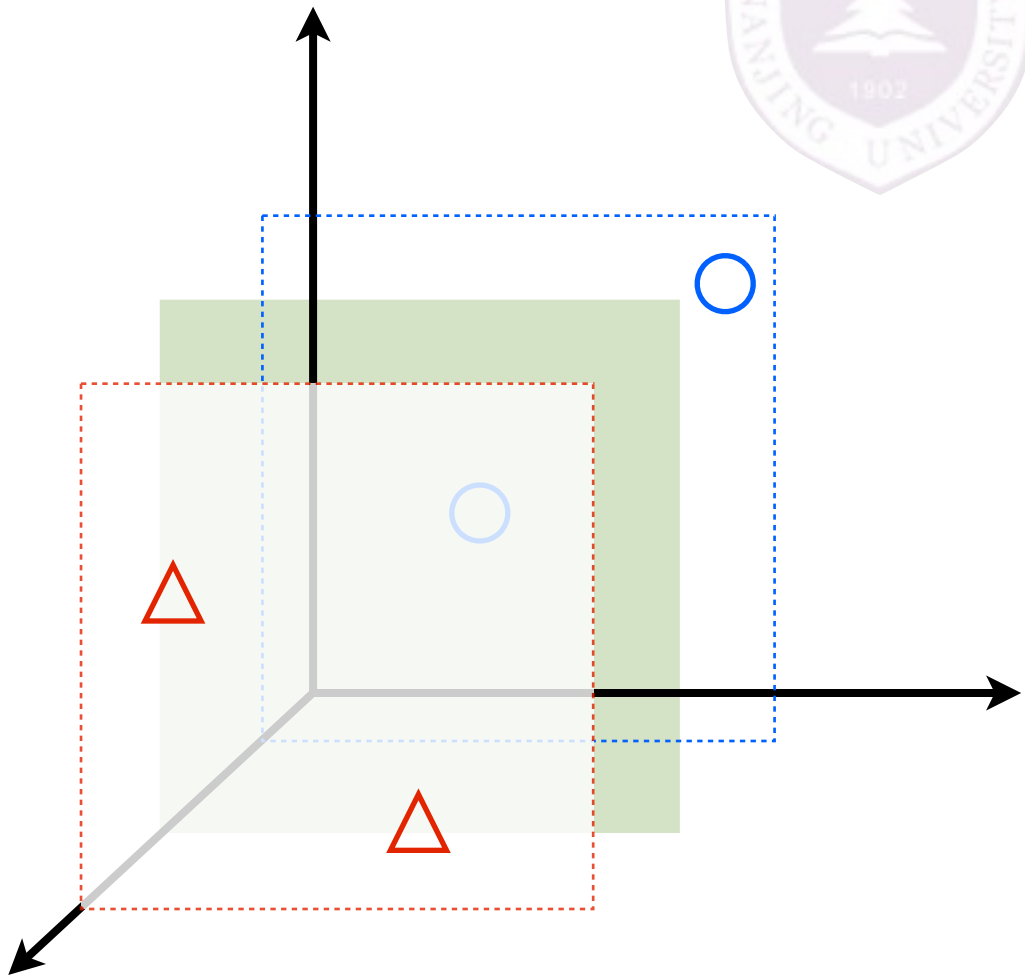
XOR in 2D



Linearity v.s. dimensionality



XOR in 2D



x_1	x_2	y
0	0	+1
0	1	-1
1	0	-1
1	1	+1

x_1	x_2	x_1x_2	y
0	0	0	+1
0	1	0	-1
1	0	0	-1
1	1	1	+1

$$\boldsymbol{w} = \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix}, b = -0.5$$

Kernelization

inner product by kernel distance

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = < \phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2) >$$

polynomial $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^\top \boldsymbol{x}_2)^n$

Gaussian radial basis $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}{\delta^2}}$

e.g. $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\boldsymbol{x}' = \begin{pmatrix} x_1' \\ x_2' \end{pmatrix}$ $\phi(\boldsymbol{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 x_2 \end{pmatrix}$

explicit inner product in higher dimension space:

$$< \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') >= x_1^2 x'^2_1 + x_2^2 x'^2_2 + 2x_1 x_2 x_1' x_2'$$

kernel function of the inner product in original space:

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^\top \boldsymbol{x}')^2 = (x_1 x_1' + x_2 x_2')^2$$

$$= x_1^2 x'^2_1 + x_2^2 x'^2_2 + 2x_1 x_2 x_1' x_2'$$

equal

this is easier to calculate

Kernelization

inner product by kernel distance

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = < \phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2) >$$

polynomial $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^\top \boldsymbol{x}_2)^n$

Gaussian radial basis $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}{\delta^2}}$

Kernelization

inner product by kernel distance

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = < \phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2) >$$

polynomial $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^\top \boldsymbol{x}_2)^n$

Gaussian radial basis $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}{\delta^2}}$

linear model in mapped feature space

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) = \sum_i \alpha_i < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) >$$

$$= \sum_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$$

Kernelization

inner product by kernel distance

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = < \phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2) >$$

polynomial $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^\top \boldsymbol{x}_2)^n$

Gaussian radial basis $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}{\delta^2}}$

linear model in mapped feature space

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) = \sum_i \alpha_i < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) >$$

$$= \sum_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$$

kernel ridge regression:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) = Y(K + \lambda \boldsymbol{I})^{-1} \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}) \\ \cdots \\ K(\boldsymbol{x}_m, \boldsymbol{x}) \end{pmatrix}$$

Change of basis

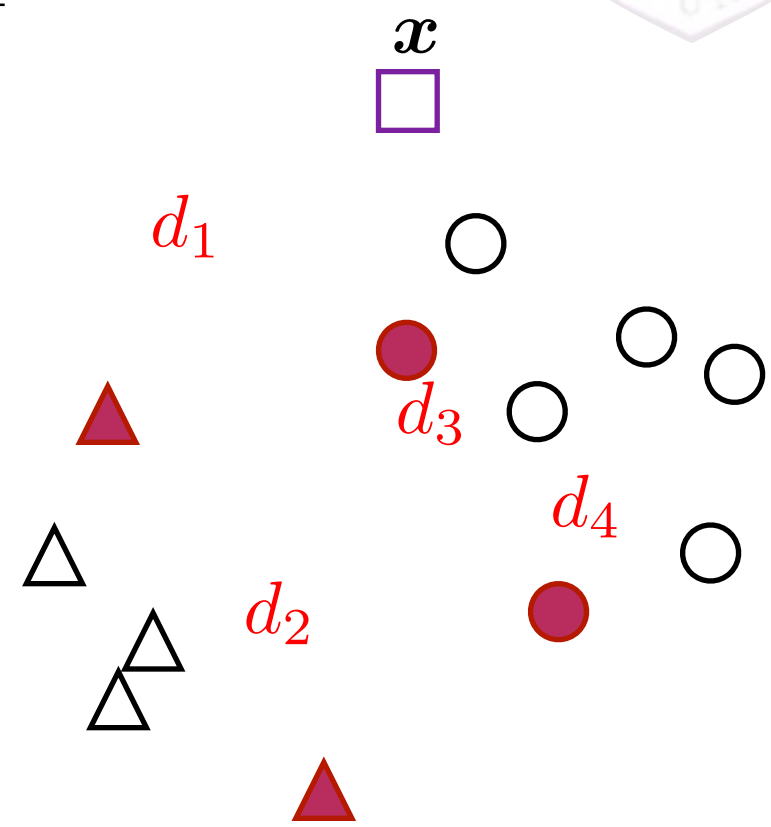
$$d = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_2\|_2^2}{\delta^2}}$$

new features:

$$\boldsymbol{x} \to \boldsymbol{z} = (d_1, d_2, d_3, d_4)$$

new decision function:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} \to f(\boldsymbol{z}) = \boldsymbol{\alpha}^\top \boldsymbol{d}$$



Change of basis

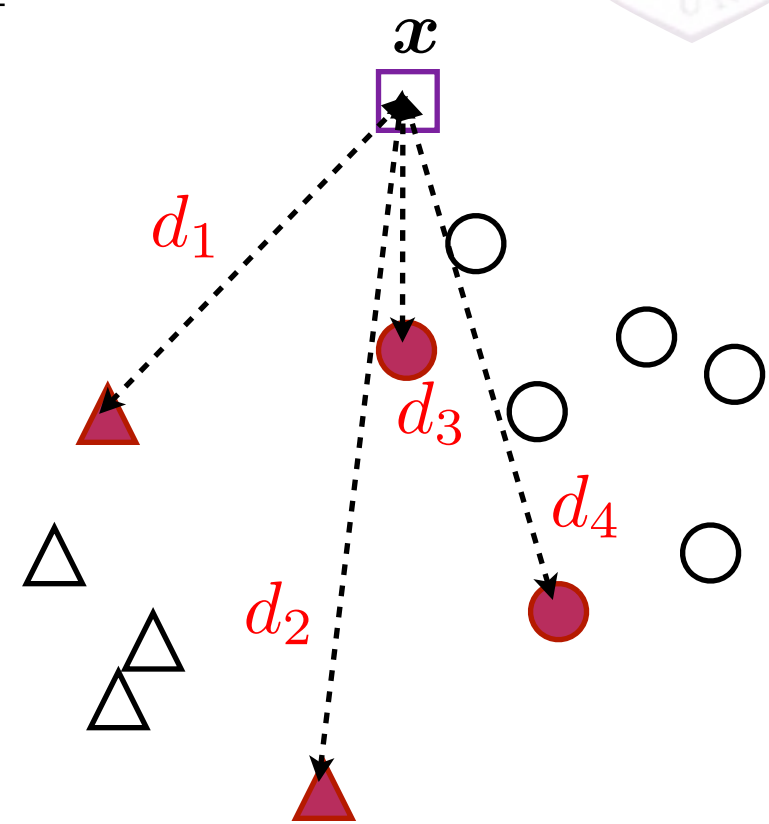
$$d = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_2\|_2^2}{\delta^2}}$$

new features:

$$\boldsymbol{x} \to \boldsymbol{z} = (d_1, d_2, d_3, d_4)$$

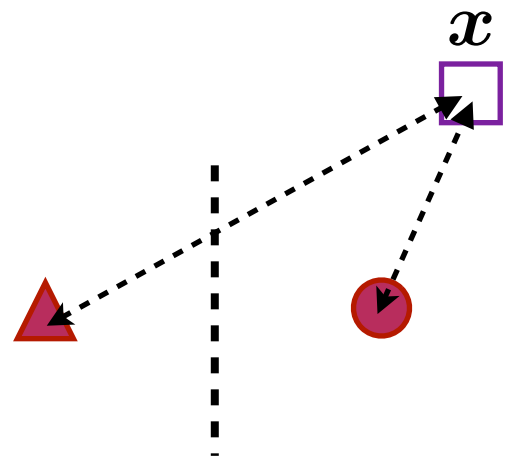
new decision function:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} \to f(\boldsymbol{z}) = \boldsymbol{\alpha}^\top \boldsymbol{d}$$

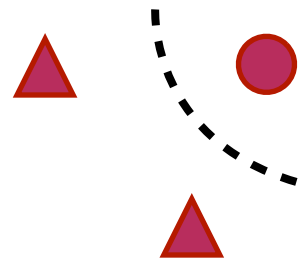


Change of basis

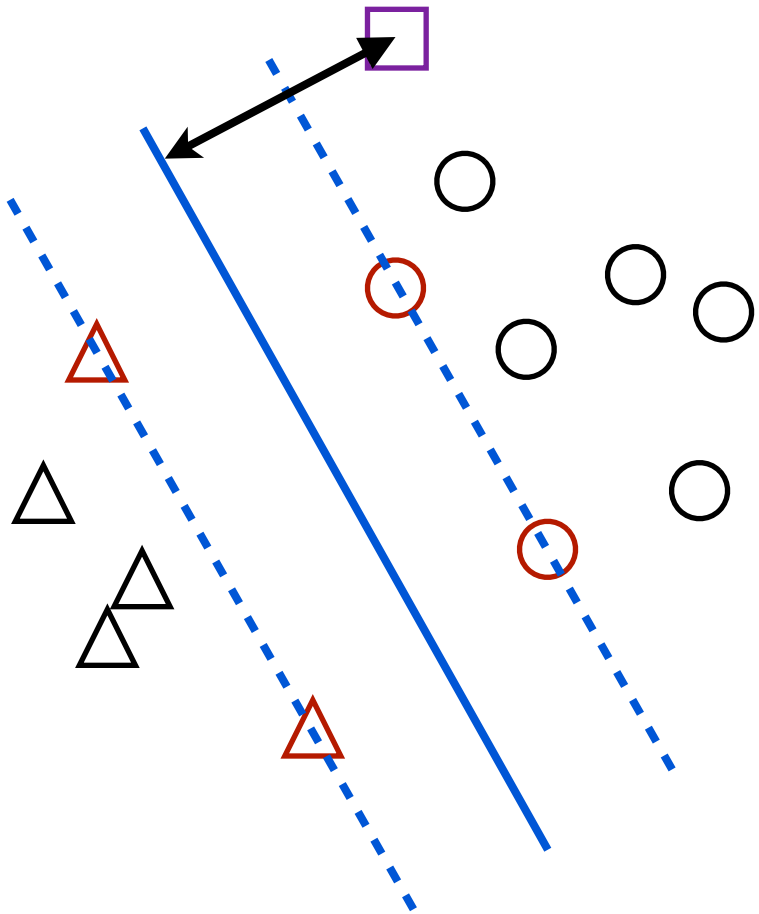
$$f(\boldsymbol{z}) = \alpha_1 \cdot d_1 + \alpha_2 \cdot d_2 \qquad (\alpha \in \mathbb{R})$$



$$f(\boldsymbol{z}) = \alpha_1 \cdot d_1 + \alpha_2 \cdot d_2 + \alpha_3 \cdot d_3$$



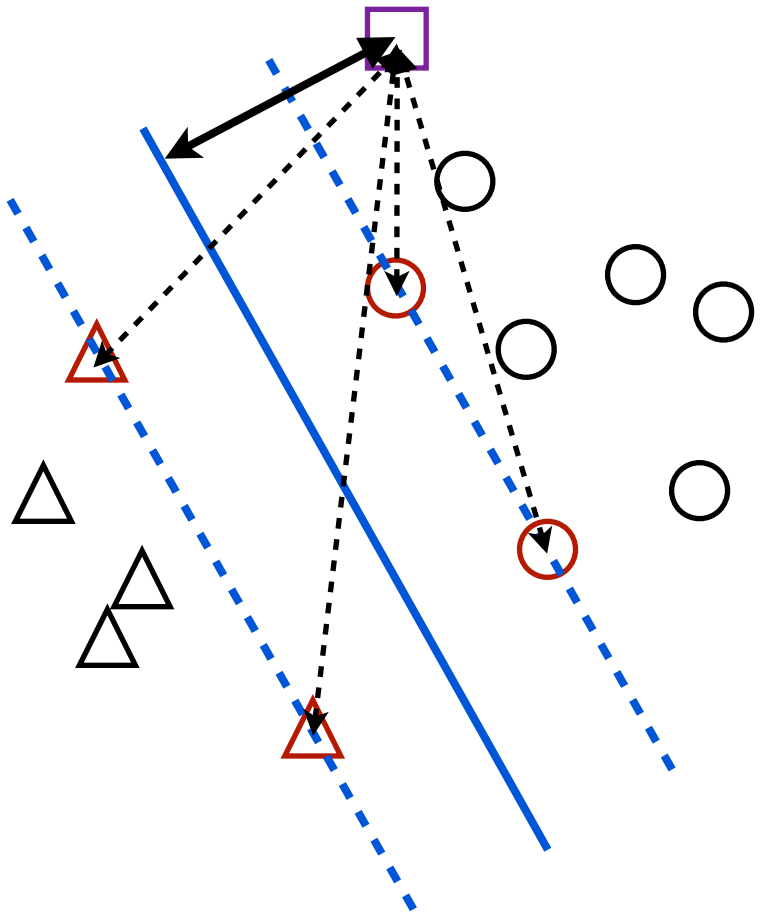
Representer theorem



the optimal function is in the form of

$$f^*(\cdot) = \sum_i \alpha_i K(\boldsymbol{x}_i, \cdot)$$

Representer theorem



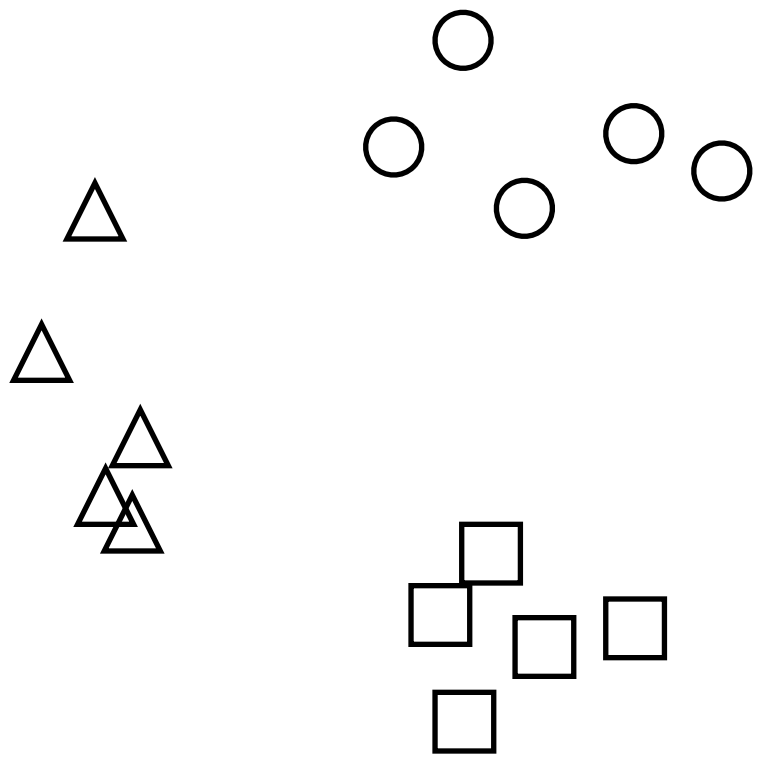
support vectors

the optimal function is in the form of

$$f^*(\cdot) = \sum_i \alpha_i K(\boldsymbol{x}_i, \cdot)$$

Multi-class classification

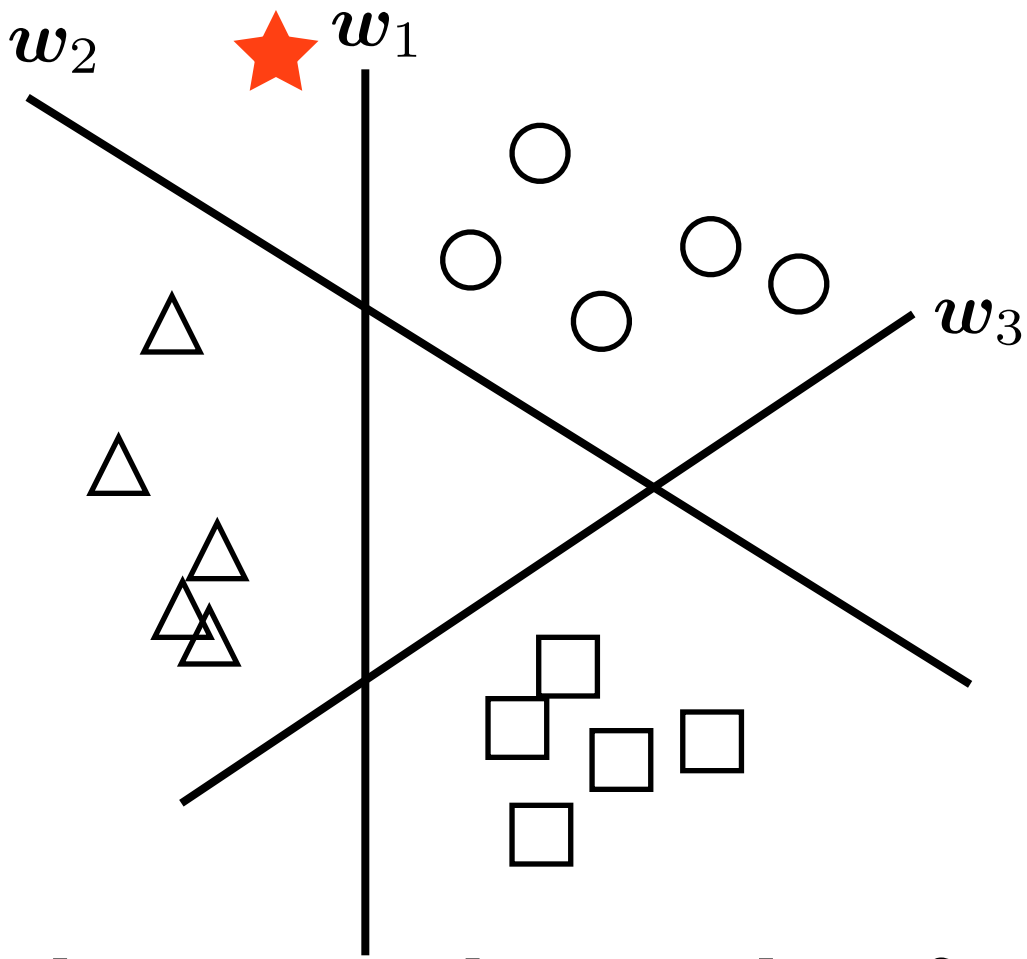
one-vs-rest



for C classes, need to train C binary classifiers

Multi-class classification

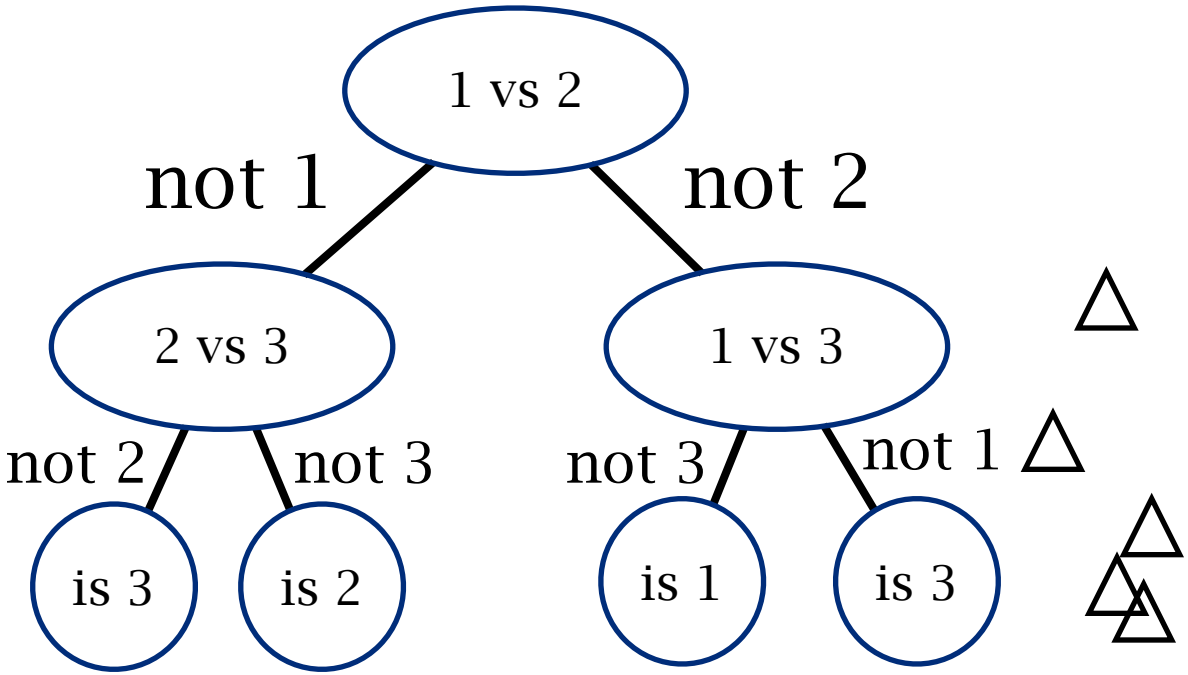
one-vs-rest

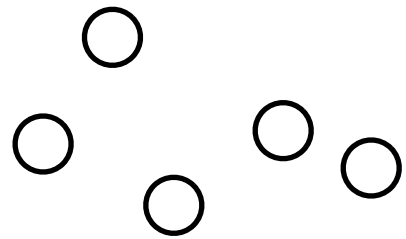


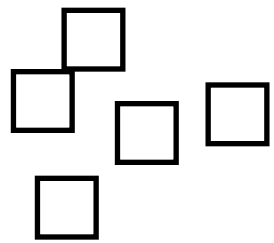
for C classes, need to train C binary classifiers

Multi-class classification

one-vs-one



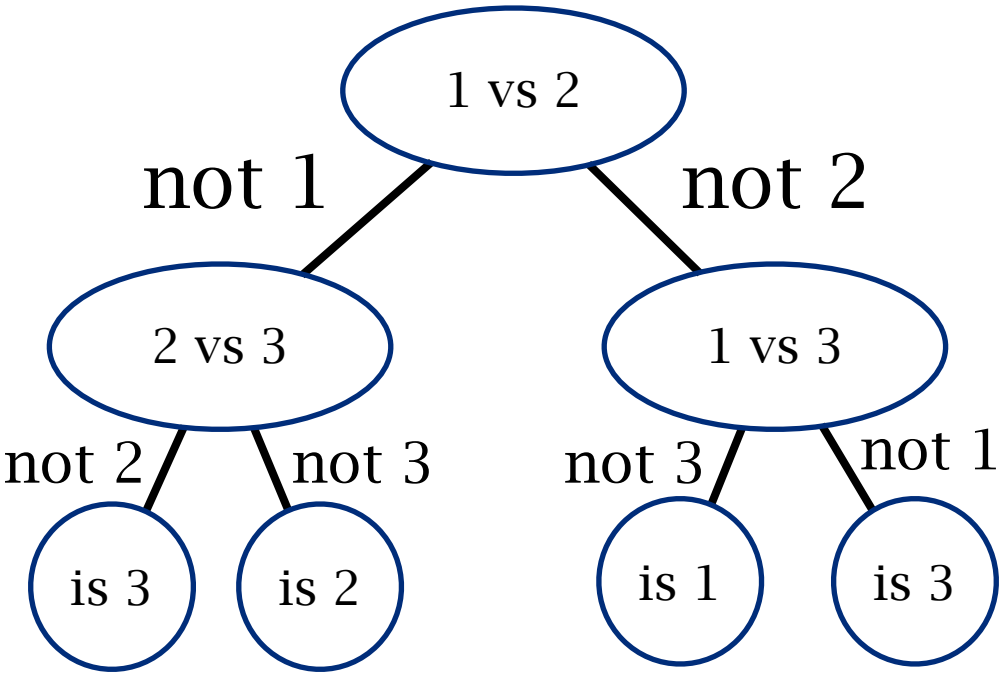


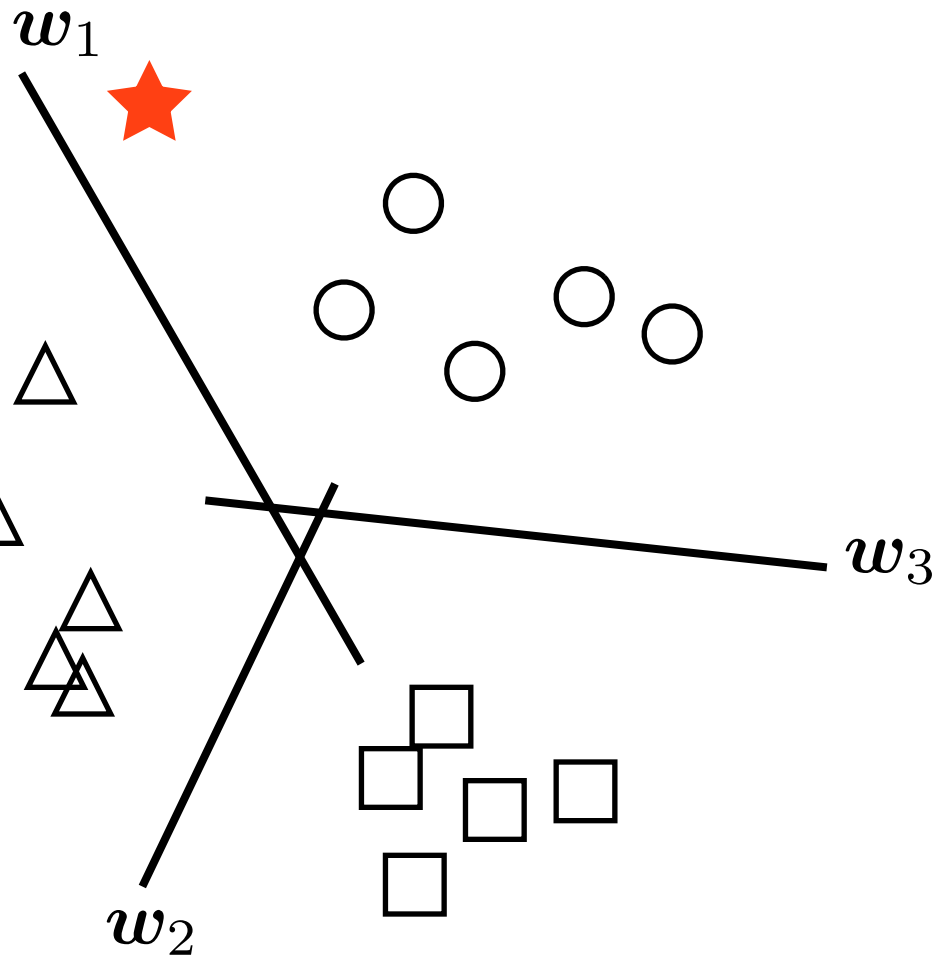


for C classes, need to train $C(C\text{-}1)$ binary classifiers

Multi-class classification

one-vs-one





for C classes, need to train $C(C\text{-}1)$ binary classifiers

习题

L1-norm作为正则化项(regularization)时为何会获得更稀疏(sparse)的解?

Logistic regression是用于回归还是分类?

在低维空间线性不可分的样本是否可以在高维空间线性可分?