

周志华 著

MACHINE  
LEARNING

# 机器学习

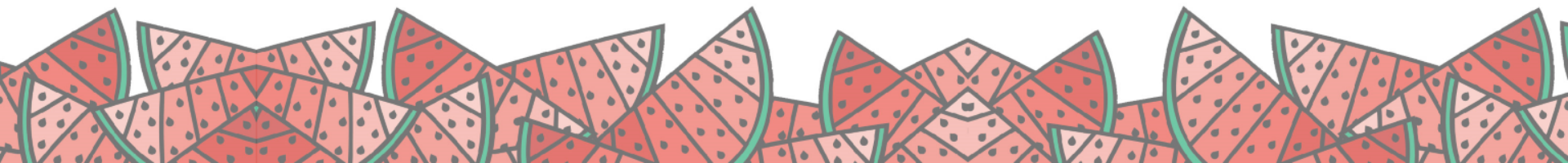
清华大学出版社

本章课件致谢..

张腾

本课件版权所有©LAMD A, 其他目的需征得本书作者同意

为本书教学目的可免费使用,



---

# 第六章：支持向量机

---

---

# 大纲

---

□ 间隔与支持向量

□ 对偶问题

□ 核函数

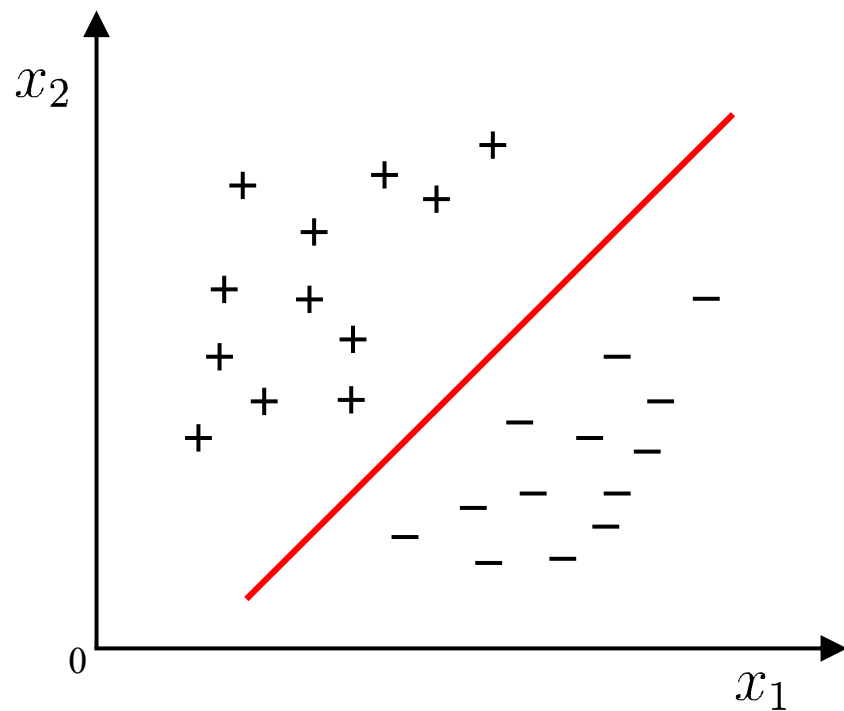
□ 软间隔与正则化

□ 支持向量回归

□ 核方法

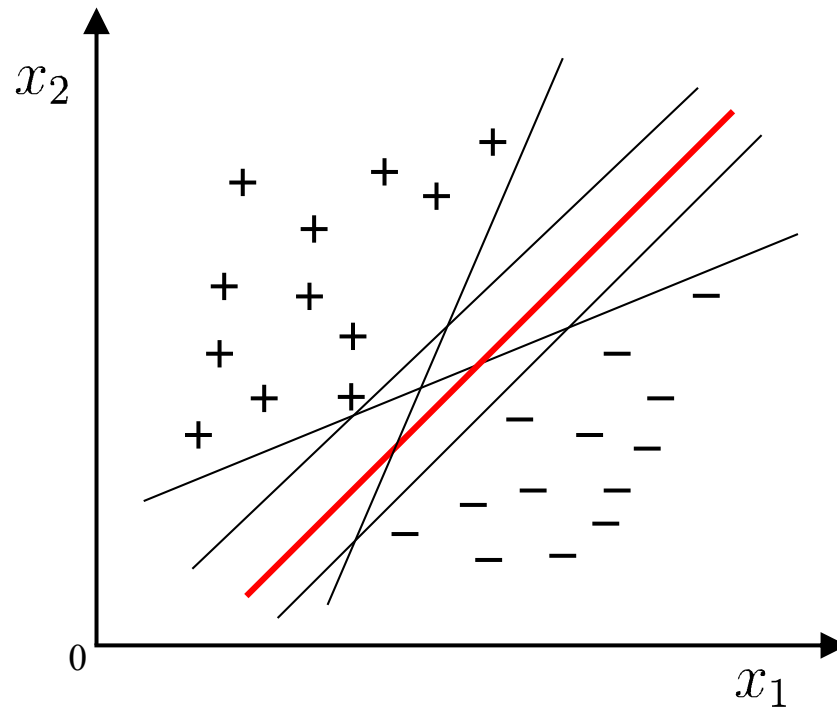
# 引子

线性模型：在样本空间中寻找一个超平面，将不同类别的样本分开。



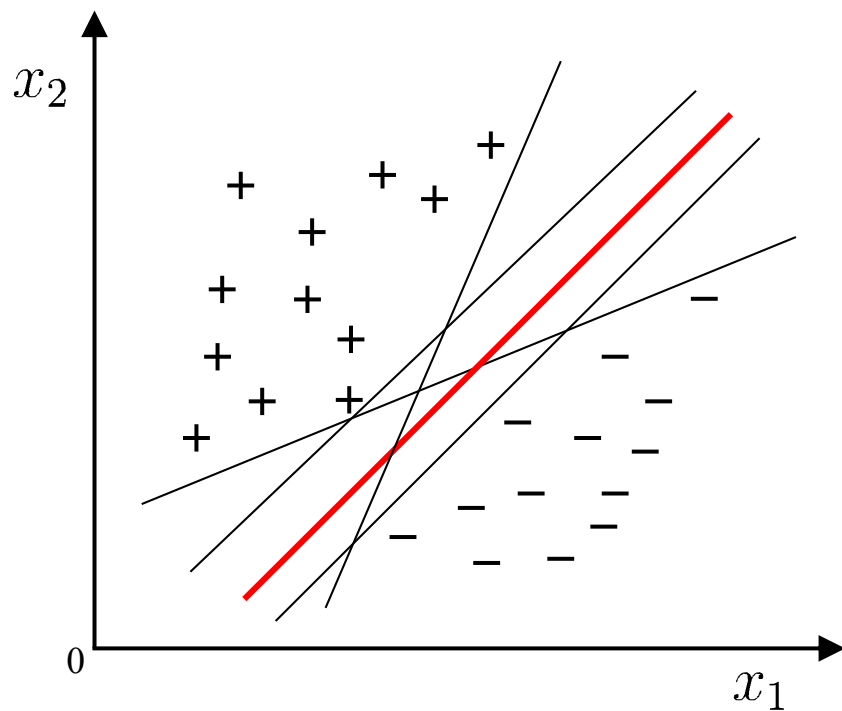
# 引子

-Q: 将训练样本分开的超平面可能有很多, 哪一个好呢?



# 引子

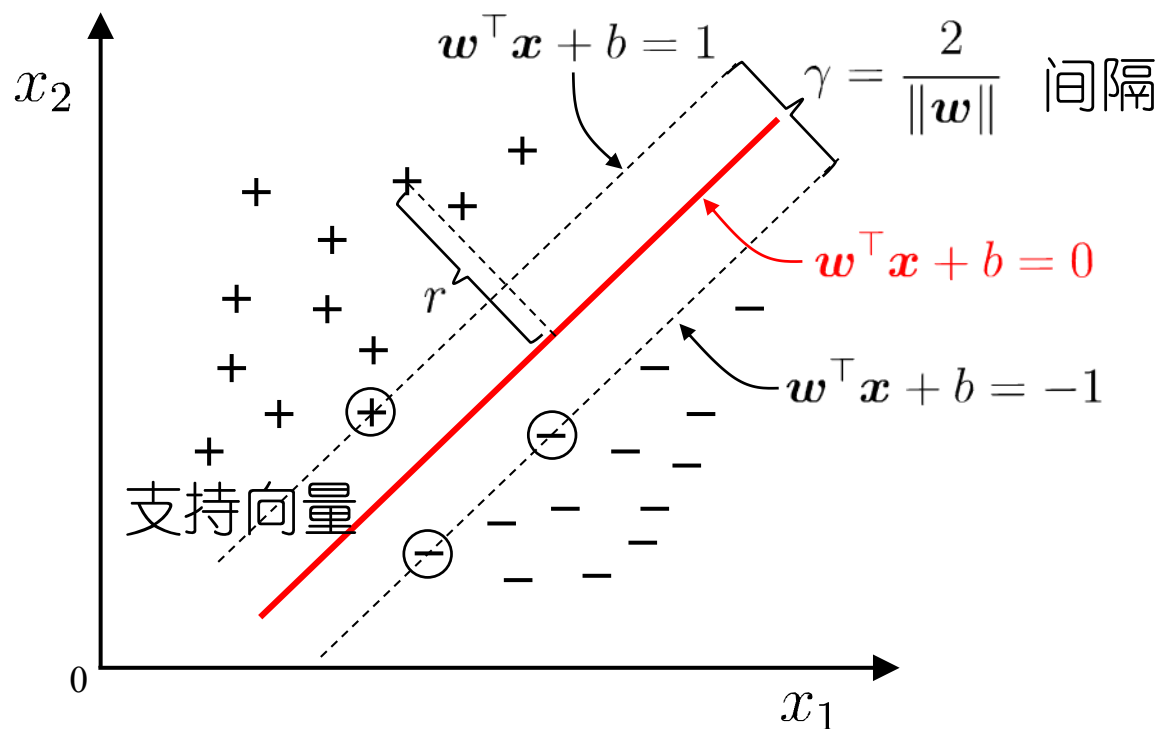
-Q: 将训练样本分开的超平面可能有很多, 哪一个好呢?



-A: 应选择“**正中间**”, 容忍性好, 鲁棒性高, 泛化能力最强.

# 间隔与支持向量

超平面方程:  $w^\top x + b = 0$



# 支持向量机基本型

□ 最大间隔：寻找参数  $\mathbf{w}$  和  $b$ , 使得  $\gamma$  最大.

$$\begin{aligned} \arg \max_{\mathbf{w}, b} \quad & \frac{2}{\|\mathbf{w}\|} \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$



$$\begin{aligned} \arg \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$



# 大纲

---

□ 间隔与支持向量

□ 对偶问题

□ 核函数

□ 软间隔与正则化

□ 支持向量回归

□ 核方法

# 对偶问题

## □ 拉格朗日乘子法

- 第一步：引入拉格朗日乘子  $\alpha_i \geq 0$  得到拉格朗日函数

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

- 第二步：令  $L(\mathbf{w}, b, \boldsymbol{\alpha})$  对  $\mathbf{w}$  和  $b$  的偏导为零可得

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^m \alpha_i y_i = 0.$$

- 第三步：回代

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - \sum_{i=1}^m \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, m. \end{aligned}$$

# 解的稀疏性

□ 最终模型:  $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = \boxed{\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}} + b$

□ KKT条件:

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\mathbf{x}_i) \geq 1, \\ \alpha_i (y_i f(\mathbf{x}_i) - 1) = 0. \end{cases}$$

$$y_i f(\mathbf{x}_i) > 1 \quad \Rightarrow \quad \alpha_i = 0$$

支持向量机解的稀疏性: 训练完成后, 大部分的训练样本都不需保留, 最终模型仅与支持向量有关.

# 求解方法 - SMO

□ 基本思路：不断执行如下两个步骤直至收敛。

- 第一步：选取一对需更新的变量  $\alpha_i$  和  $\alpha_j$ 。
- 第二步：固定  $\alpha_i$  和  $\alpha_j$  以外的参数，求解对偶问题更新  $\alpha_i$  和  $\alpha_j$ 。

□ 仅考虑  $\alpha_i$  和  $\alpha_j$  时，对偶问题的约束变为

$$\alpha_i y_i + \alpha_j y_j = - \sum_{k \neq i, j} \alpha_k y_k, \quad \alpha_i \geq 0, \quad \alpha_j \geq 0.$$

用一个变量表示另一个变量，回代入对偶问题可得一个单变量的二次规划，该问题具有闭式解。

□ 偏移项  $b$ ：通过支持向量来确定。

# 大纲

---

□ 间隔与支持向量

□ 对偶问题

□ 核函数

□ 软间隔与正则化

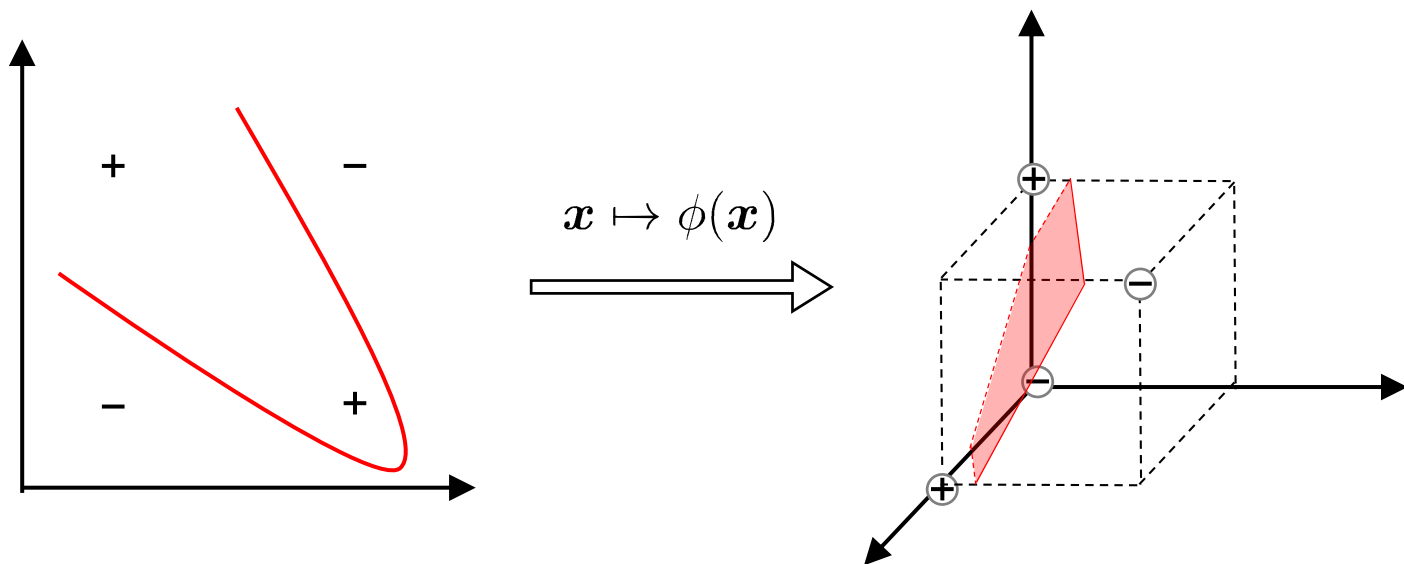
□ 支持向量回归

□ 核方法

# 线性不可分

-Q: 若不存在一个能正确划分两类样本的超平面, 怎么办?

-A: 将样本从原始空间映射到一个更高维的特征空间, 使得样本在这个特征空间内线性可分.



# 核支持向量机

□ 设样本  $\mathbf{x}$  映射后的向量为  $\phi(\mathbf{x})$ , 划分超平面为  $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$ .

原始问题

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i (\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$

对偶问题

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \boxed{\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)} - \sum_{i=1}^m \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad \text{只以内积的形式出现} \end{aligned}$$

预测

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{i=1}^m \alpha_i y_i \boxed{\phi(\mathbf{x}_i)^\top \phi(\mathbf{x})} + b$$

# 核函数

- 基本想法：不显式地设计核映射，而是设计核函数。

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

- Mercer定理**(充分非必要)：只要一个对称函数所对应的核矩阵半正定，则它就能作为核函数来使用。

- 常用核函数：

名称	表达式	参数
线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$	
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^d$	$d \geq 1$ 为多项式的次数
高斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\delta^2}\right)$	$\delta > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^\top \mathbf{x}_j + \theta)$	$\tanh$ 为双曲正切函数, $\beta > 0, \theta < 0$



# 大纲

---

□ 间隔与支持向量

□ 对偶问题

□ 核函数

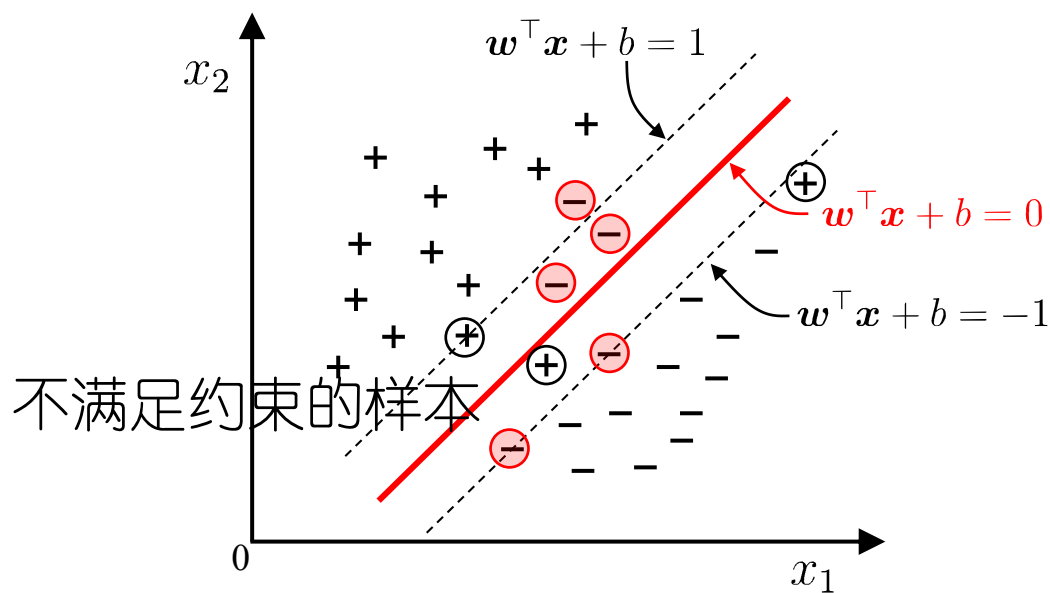
□ 软间隔与正则化

□ 支持向量回归

□ 核方法

# 软间隔

- Q: 现实中, 很难确定合适的核函数使得训练样本在特征空间中线性可分; 同时一个线性可分的结果也很难断定是否是有过拟合造成的.
- A: 引入“软间隔”的概念, 允许支持向量机在一些样本上不满足约束.



# 0/1损失函数

- 基本想法：最大化间隔的同时，让不满足约束的样本应尽可能少。

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m l_{0/1} (y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) - 1)$$

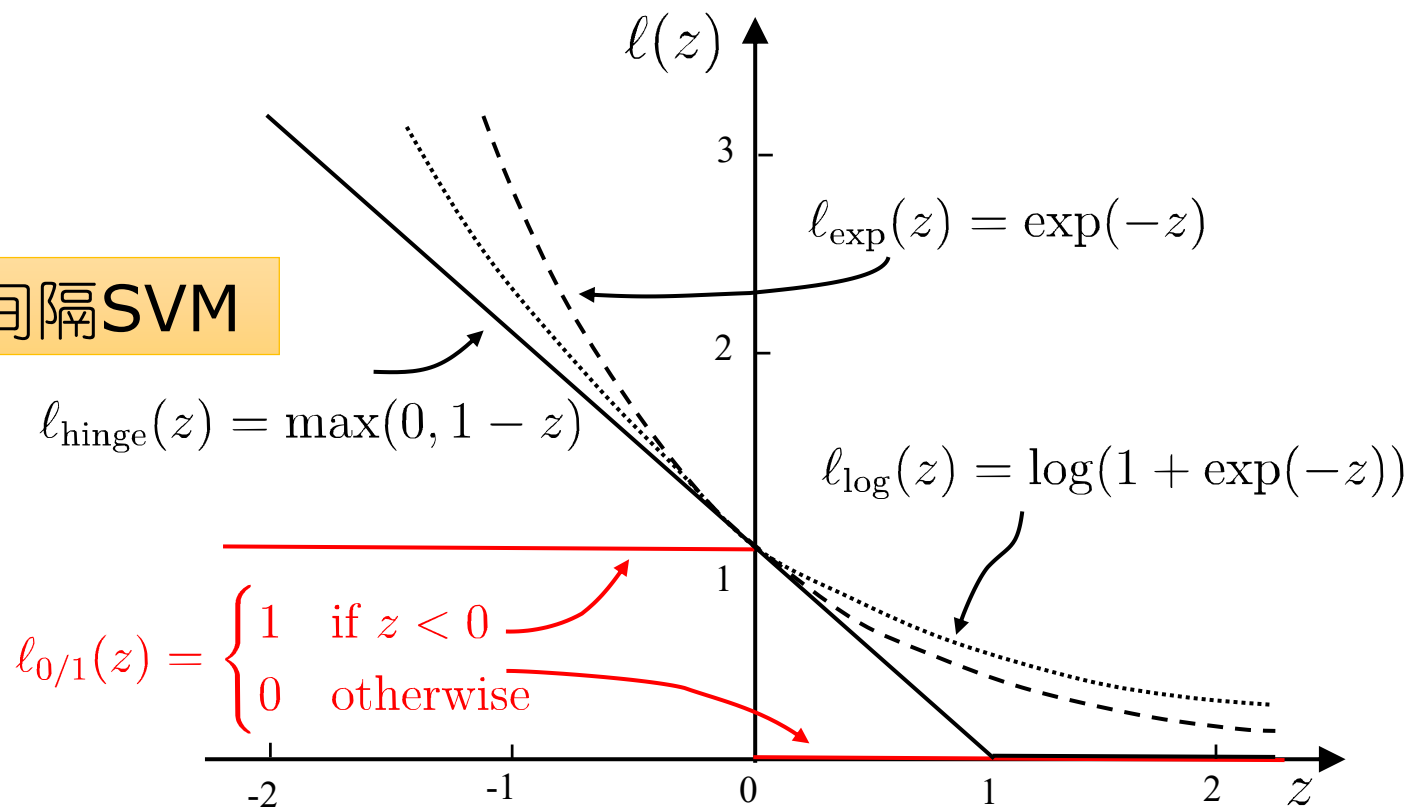
其中  $l_{0/1}$  是“0/1损失函数”

$$l_{0/1} = \begin{cases} 1 & z < 0 \\ 0 & otherwise \end{cases}$$

- 存在的问题：0/1损失函数非凸、非连续，不易优化！

# 替代损失

## 软间隔SVM



替代损失函数数学性质较好, 一般是**0/1**损失函数的上界

# 软间隔支持向量机

原始问题

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \max(0, 1 - y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b))$$

对偶问题

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) - \sum_{i=1}^m \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, m. \end{aligned}$$

根据KKT条件可推得最终模型仅与支持向量有关，也即hinge损失函数依然保持了支持向量机解的稀疏性。

# 正则化

- 支持向量机学习模型的更一般形式

$$\min_f \Omega(f) + C \sum_{i=1}^m l(f(\mathbf{x}_i), y_i)$$



结构风险, 描述模型的某些性质



经验风险, 描述模型与训练数据的契合程度

- 通过替换上面两个部分, 可以得到许多其他学习模型
  - 对数几率回归(Logistic Regression)
  - 最小绝对收缩选择算子(LASSO)
  - .....

# 大纲

---

□ 间隔与支持向量

□ 对偶问题

□ 核函数

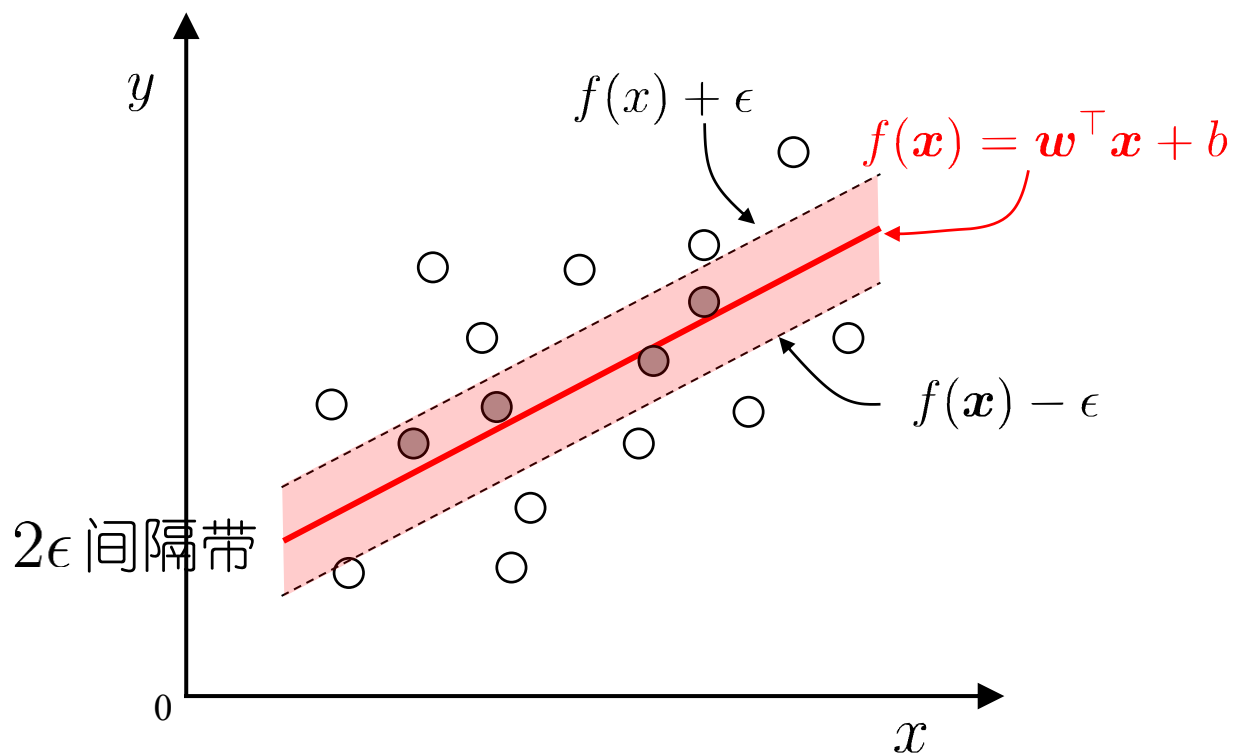
□ 软间隔与正则化

□ 支持向量回归

□ 核方法

# 支持向量回归

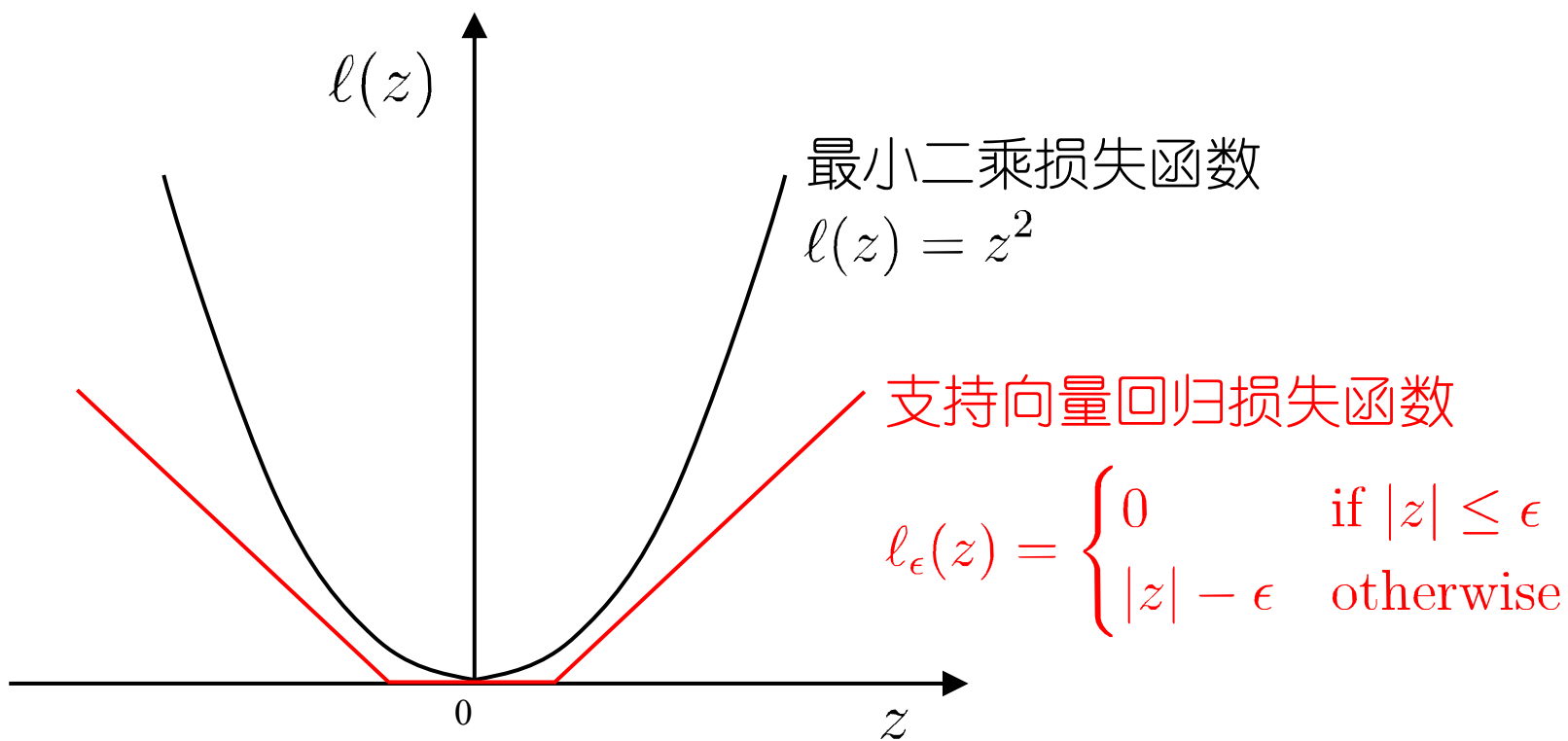
特点：允许模型输出和实际输出间存在  $2\epsilon$  的偏差。





# 损失函数

落入中间  $2\epsilon$  间隔带的样本不计算损失, 从而使得模型获得稀疏性.



# 形式化

原始问题

$$\begin{aligned} \min_{\mathbf{w}, b, \xi_i, \hat{\xi}_i} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m (\xi_i + \hat{\xi}_i) \\ \text{s.t.} \quad & y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) - b \leq \epsilon + \xi_i, \\ & y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) - b \geq -\epsilon - \hat{\xi}_i, \\ & \xi_i \geq 0, \hat{\xi}_i \geq 0, \quad i = 1, 2, \dots, m. \end{aligned}$$

对偶问题

$$\begin{aligned} \min_{\alpha, \hat{\alpha}} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j) \kappa(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^m (\alpha_i(\epsilon - y_i) + \hat{\alpha}_i(\epsilon + y_i)) \\ \text{s.t.} \quad & \sum_{i=1}^m (\alpha_i - \hat{\alpha}_i) = 0, \\ & 0 \leq \alpha_i \leq C, \quad 0 \leq \hat{\alpha}_i \leq C. \end{aligned}$$

预测

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$$

# 大纲

---

□ 间隔与支持向量

□ 对偶问题

□ 核函数

□ 软间隔与正则化

□ 支持向量回归

□ 核方法

# 表示定理

支持向量机  $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{i=1}^m \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$

支持向量回归  $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$

结论：无论是支持向量机还是支持向量回归，学得模型总可以表示成核函数的线性组合。

更一般的结论(表示定理)：对于任意单调增函数  $\Omega$  和任意非负损失函数  $l$ ，优化问题

$$\min_{h \in \mathbb{H}} F(h) = \Omega(\|h\|_{\mathbb{H}}) + l(h(\mathbf{x}_1), \dots, h(\mathbf{x}_m))$$

的解总可以写为  $h^* = \sum_{i=1}^m \alpha_i \kappa(\cdot, \mathbf{x}_i)$ 。

# 核线性判别分析

□ 通过表示定理可以得到很多线性模型的“核化”版本

- 核SVM
- 核LDA
- 核PCA
- .....

□ 核LDA: 先将样本映射到高维特征空间, 然后在此特征空间中做线性判别分析

$$\begin{aligned} \max_{\mathbf{w}} J(\mathbf{w}) &= \frac{\mathbf{w}^\top \mathbf{S}_b^\phi \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_w^\phi \mathbf{w}} \\ &\quad \downarrow \\ h(\mathbf{x}) &= \mathbf{w}^\top \phi(\mathbf{x}) = \sum_{i=1}^m \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) \\ \max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) &= \frac{\boldsymbol{\alpha}^\top \mathbf{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top \mathbf{N} \boldsymbol{\alpha}} \end{aligned}$$

# Take Home Message

---

- 支持向量机的“最大间隔”思想
- 对偶问题及其解的稀疏性
- 通过向高维空间映射解决线性不可分的问题
- 引入“软间隔”缓解特征空间中线性不可分的问题
- 将支持向量的思想应用到回归问题上得到支持向量回归
- 将核方法推广到其他学习模型

# 成熟的SVM软件包

---

- LIBSVM

<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

- LIBLINEAR

<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

- SVM<sup>light</sup>、SVM<sup>perf</sup>、SVM<sup>struct</sup>

[http://svmlight.joachims.org/svm\\_struct.html](http://svmlight.joachims.org/svm_struct.html)

- Pegasos

<http://www.cs.huji.ac.il/~shais/code/index.html>