

六、支持向量机

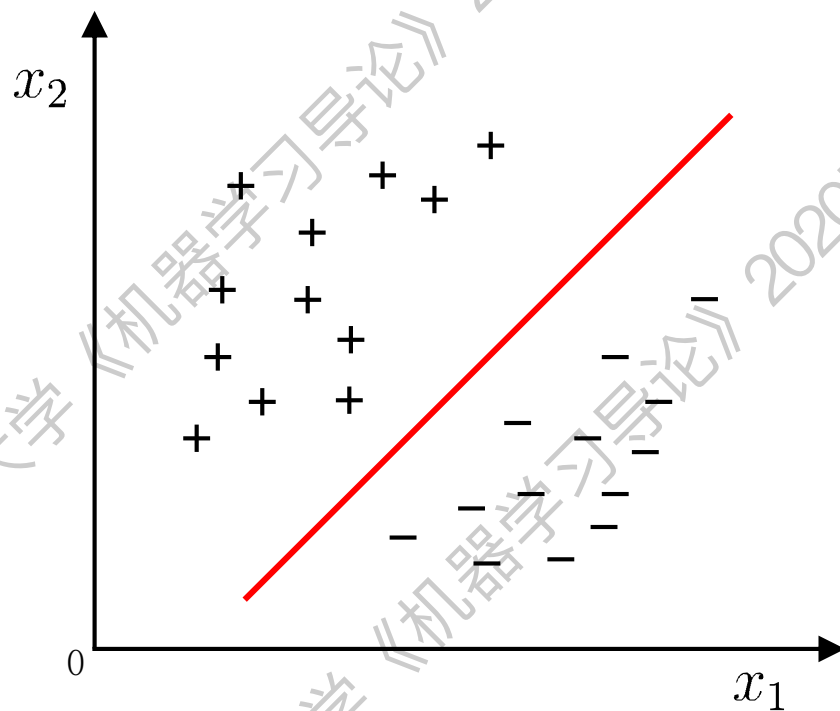
这个视频讲的比垃圾PPT好多了

<https://www.bilibili.com/video/BV1oJ411U7Y8?p=4>

主讲教师：周志华

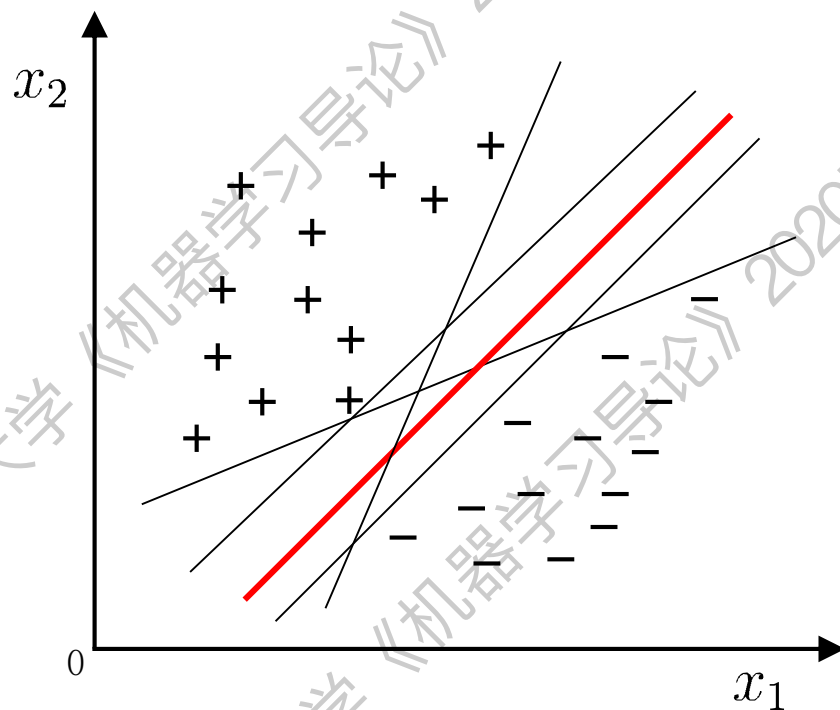
线性分类器回顾

在样本空间中寻找一个超平面，将不同类别的样本分开



线性分类器回顾

将训练样本分开的超平面可能有很多，哪一个更好呢？

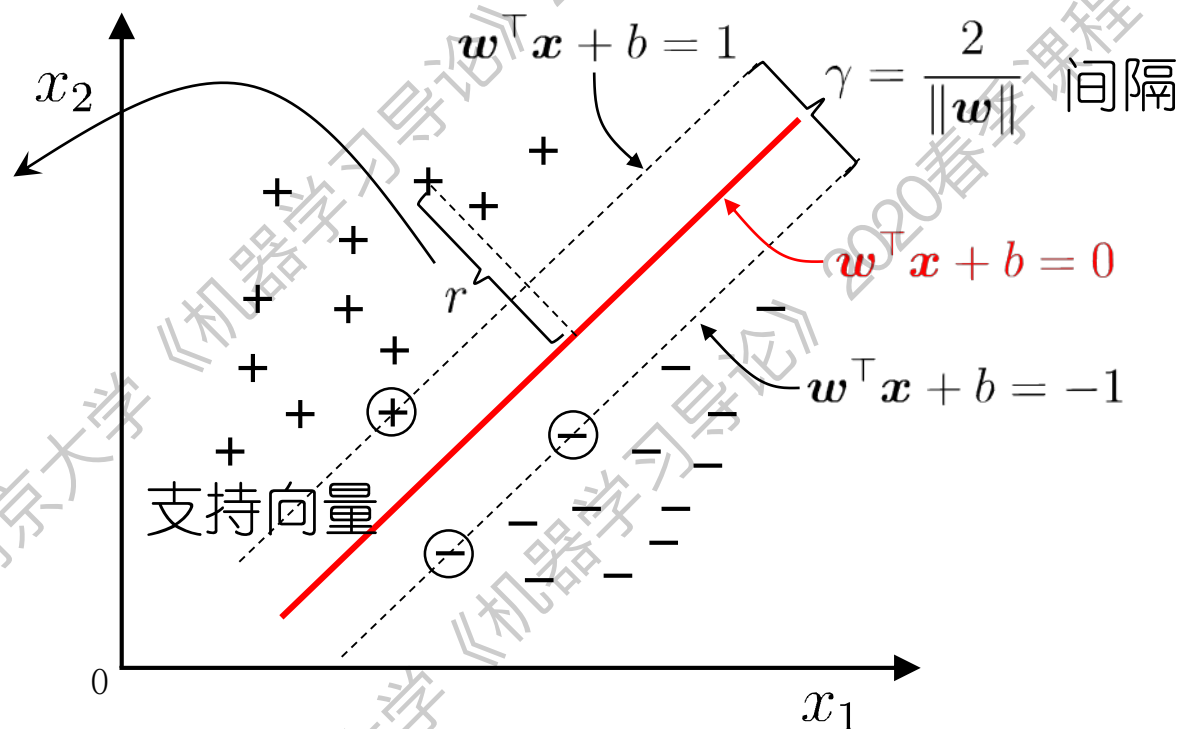


“正中间”的：鲁棒性最好，泛化能力最强

间隔(margin)与支持向量(support vector)

超平面方程: $w^T x + b = 0$

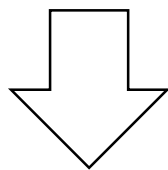
$$r = \frac{|w^T x + b|}{\|w\|}$$



支持向量机基本型

最大间隔：寻找参数 \mathbf{w} 和 b ，使得 γ 最大

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



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

凸二次规划问题，能用优化计算包求解，但可以有更高效的办法

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

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

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

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

□ 第三步：回代可得

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

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 α_i 和 α_j
- 第二步：固定 α_i 和 α_j 以外的参数，求解对偶问题更新 α_i 和 α_j

仅考虑 α_i 和 α_j 时，对偶问题的约束变为

$$\alpha_i y_i + \alpha_j y_j = c, \quad \alpha_i \geq 0, \quad \alpha_j \geq 0$$

用 α_i 表示 α_j ，代入对偶问题

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad \text{有闭式解!}$$

对任意支持向量 (\mathbf{x}_s, y_s) 有 $y_s f(\mathbf{x}_s) = 1$ ，由此可解出 b

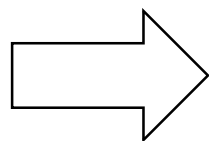
为提高鲁棒性，通常使用所有支持向量求解的平均值

解的稀疏性

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

KKT条件：

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



必有 $\alpha_i = 0$ 或
 $y_i f(\mathbf{x}_i) = 1$

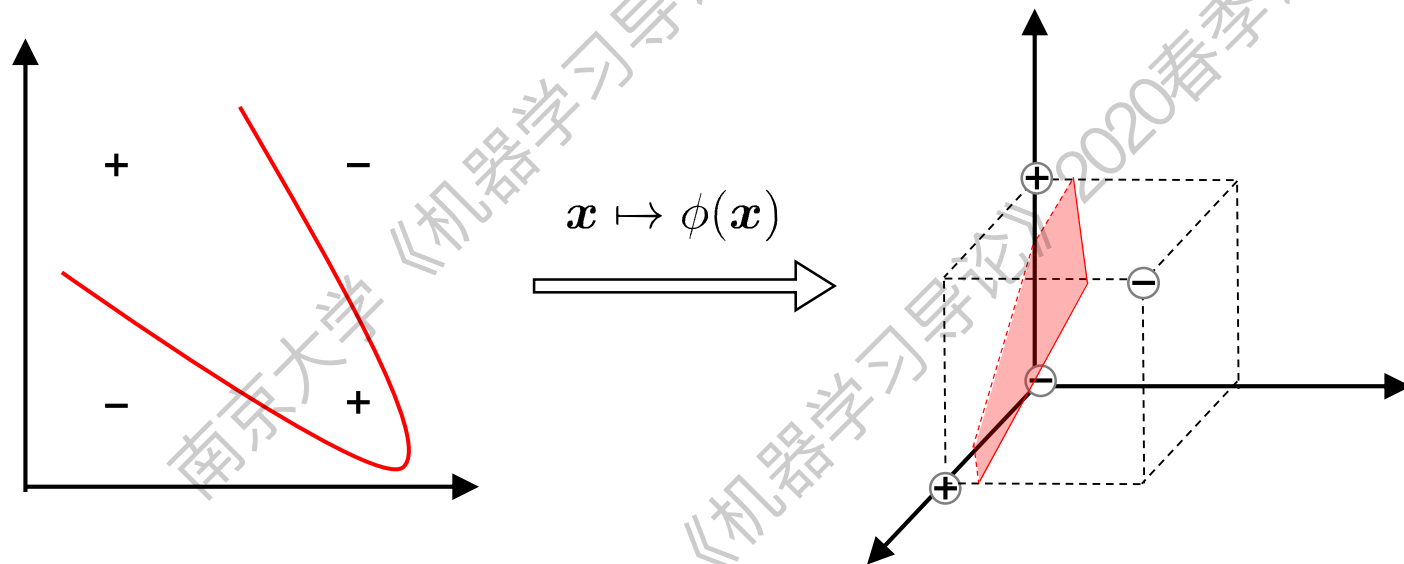
解的稀疏性：训练完成后，最终模型仅与支持向量有关

支持向量机(Support Vector Machine, SVM) 因此而得名

特征空间映射

若不存在一个能正确划分两类样本的超平面, 怎么办?

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



如果原始空间是有限维(属性数有限), 那么一定存在一个高维特征空间使样本可分

在特征空间中

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

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

预测

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

只以内积
形式出现

核函数 (kernel function)

基本思路：设计核函数

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

绕过显式考虑特征映射、以及计算高维内积的困难

Mercer 定理：若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

任何一个核函数，都隐式地定义了一个RKHS (Reproducing Kernel Hilbert Space, 再生核希尔伯特空间)

“核函数选择”成为决定支持向量机性能的关键！

核函数

常用核函数

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

基本经验：文本数据常用线性核，
情况不明时可先尝试高斯核

可通过函数组合得到：

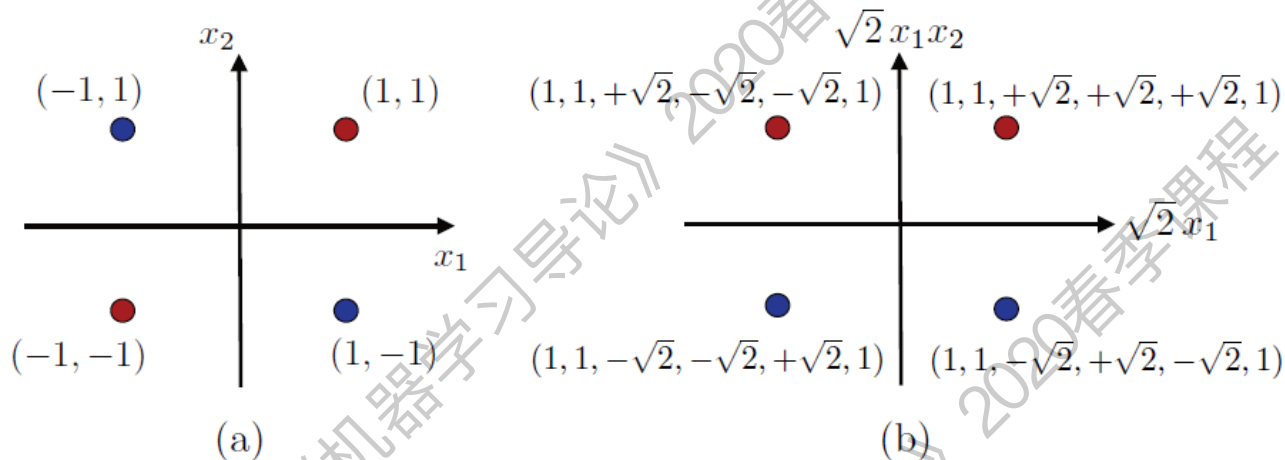
若 κ_1 和 κ_2 是核函数，则对任意正数 γ_1 、 γ_2 和任意函数 $g(\mathbf{x})$ ，

均为核函数

$$\left\{ \begin{array}{l} \gamma_1 \kappa_1 + \gamma_2 \kappa_2 \\ \kappa_1 \otimes \kappa_2(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z}) \kappa_2(\mathbf{x}, \mathbf{z}) \\ \kappa(\mathbf{x}, \mathbf{z}) = g(\mathbf{x}) \kappa_1(\mathbf{x}, \mathbf{z}) g(\mathbf{z}) \end{array} \right.$$

核函数-举例（多项式）

核函数相当于特征的高维映射



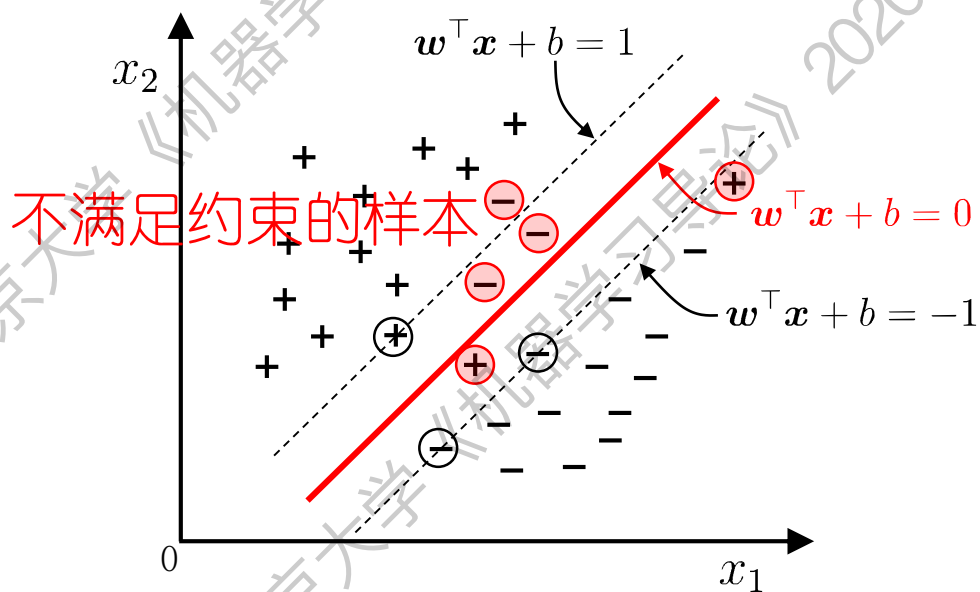
多项式核的应用

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^2, \quad K(\mathbf{x}, \mathbf{x}') = (x_1x'_1 + x_2x'_2 + c)^2 =$$
$$\begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} x_1'^2 \\ x_2'^2 \\ \sqrt{2}x_1'x_2' \\ \sqrt{2c}x_1' \\ \sqrt{2c}x_2' \\ c \end{bmatrix}.$$

软间隔

现实中很难确定合适的核函数，使训练样本在特征空间中线性可分
即便貌似线性可分，也很难断定是否是因过拟合造成的

引入**软间隔** (soft margin), 允许在一些样本上不满足约束



软间隔支持向量机

引入“松弛变量” (slack variables) ξ_i

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

对偶问题

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

与“硬间隔SVM”
的区别

根据 KKT 条件可知，最终模型仅与支持向量有关，也即采用 hinge 损失函数后仍保持了 SVM 解的稀疏性

从优化目标着手

基本思路：最大化间隔的同时，

让不满足约束 $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ 的样本尽可能少

原始问题

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

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

其中 $\ell_{0/1}$ 是 0/1 损失函数 (0/1 loss function):

$$\ell_{0/1}(z) = \begin{cases} 1, & \text{if } z < 0; \\ 0, & \text{otherwise.} \end{cases}$$

障碍：0/1 损失函数非凸、非连续，不易优化！

替代损失 (surrogate loss)

软间隔SVM

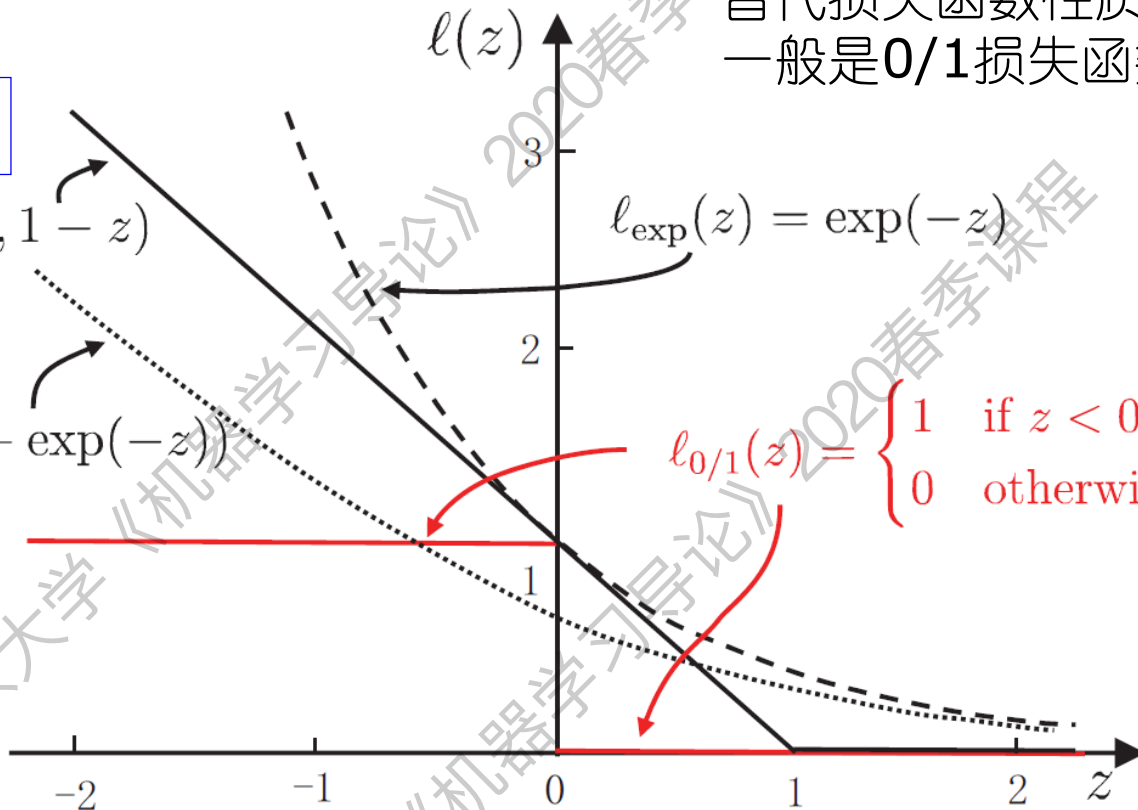
$$\ell_{\text{hinge}}(z) = \max(0, 1 - z)$$

$$\ell_{\log}(z) = \log(1 + \exp(-z))$$

$\ell(z)$

$$\ell_{\exp}(z) = \exp(-z)$$

$$\ell_{0/1}(z) = \begin{cases} 1 & \text{if } z < 0 \\ 0 & \text{otherwise} \end{cases}$$

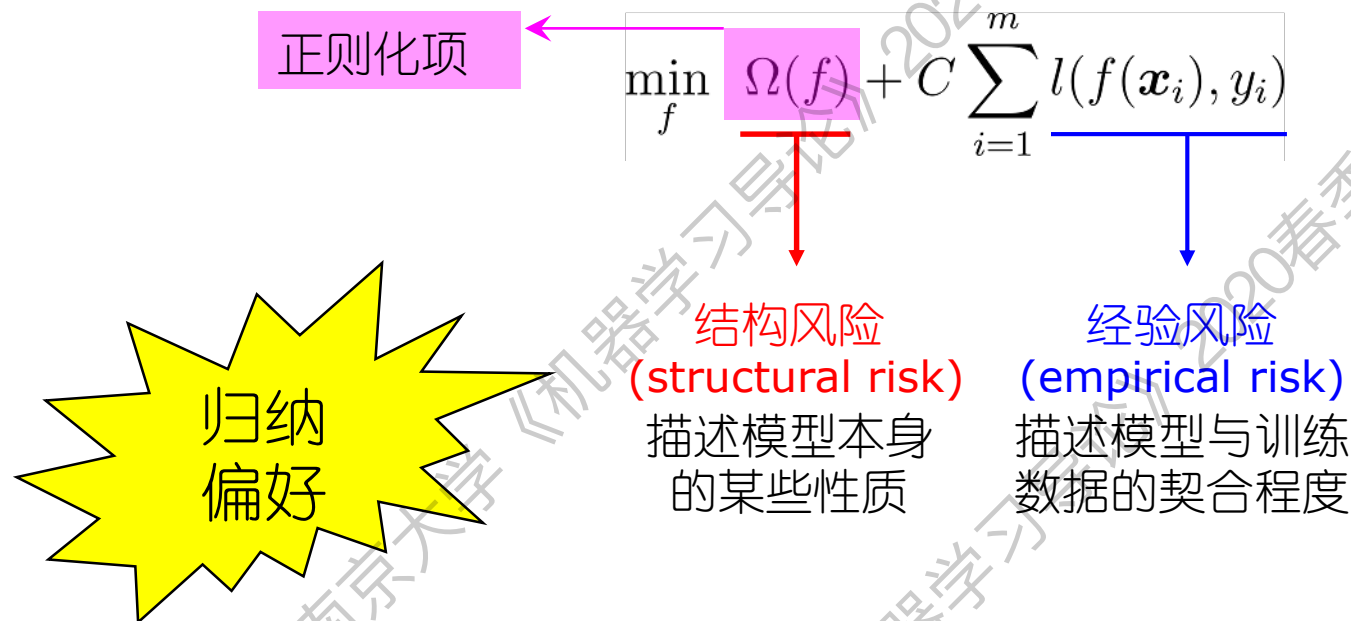


替代损失函数性质较好，
一般是0/1损失函数的上界

- 采用替代损失函数，是在解决困难问题时的常见技巧
- 求解替代函数得到的解是否仍是原问题的解？理论上称为替代损失的“一致性” (consistency) 问题

正则化 (regularization)

统计学习模型（例如 SVM）的更一般形式



□ 正则化可理解为“罚函数法”

通过对不希望的结果施以惩罚，使得优化过程趋向于希望目标

□ 从贝叶斯估计的角度，则可认为是提供了模型的先验概率

SVM简史

线性判别分析 **LDA** (R. Fisher, 1936)



感知机 **Perceptron** (F. Rosenblatt, 1958)

Generalized Portrait Methods (1962, V. Vapnik & A. Chervonenkis)



SVM简史(续)

VC Dimension (1968, V. Vapnik & A. Chervonenkis)

Structural risk minimization (1974, V. Vapnik & A. Chervonenkis)

Support Vector Networks (C. CORTES & V. Vapnik, 1995)



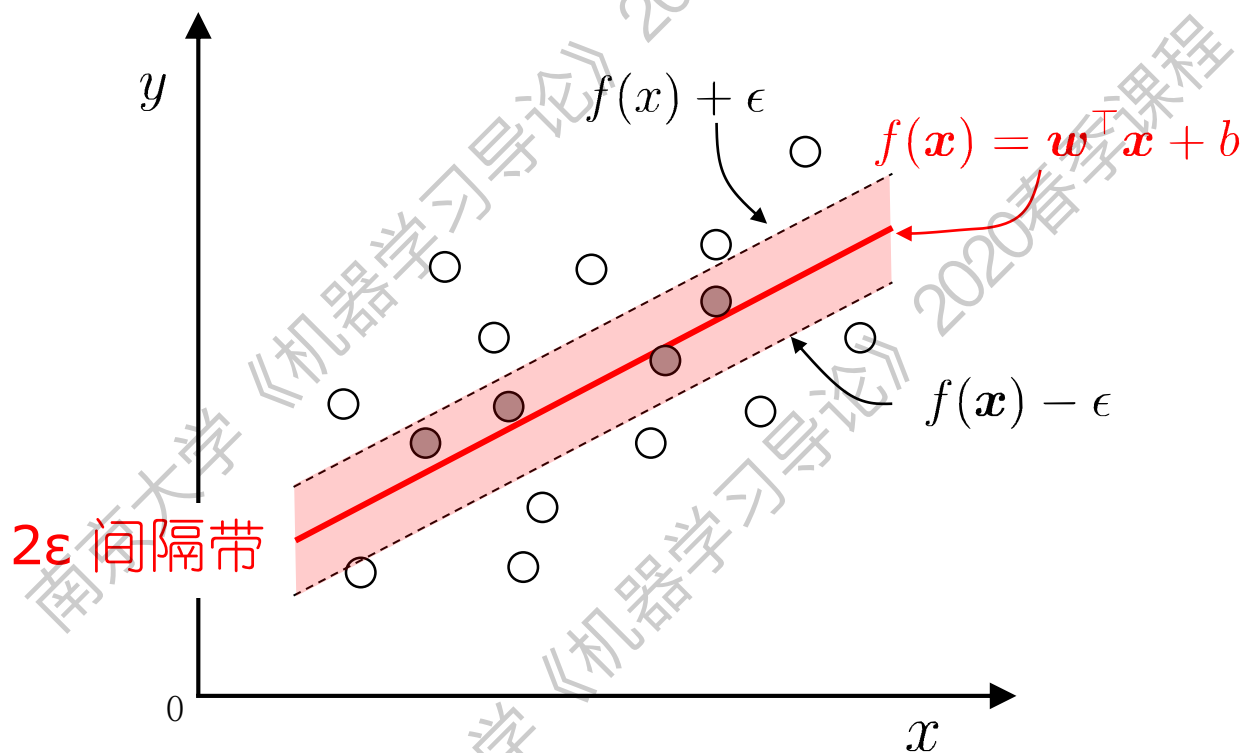
优化：如何快速求解？

应用：Large Margin概念的泛化？

如何使用SVM 解决自己特定的任务？

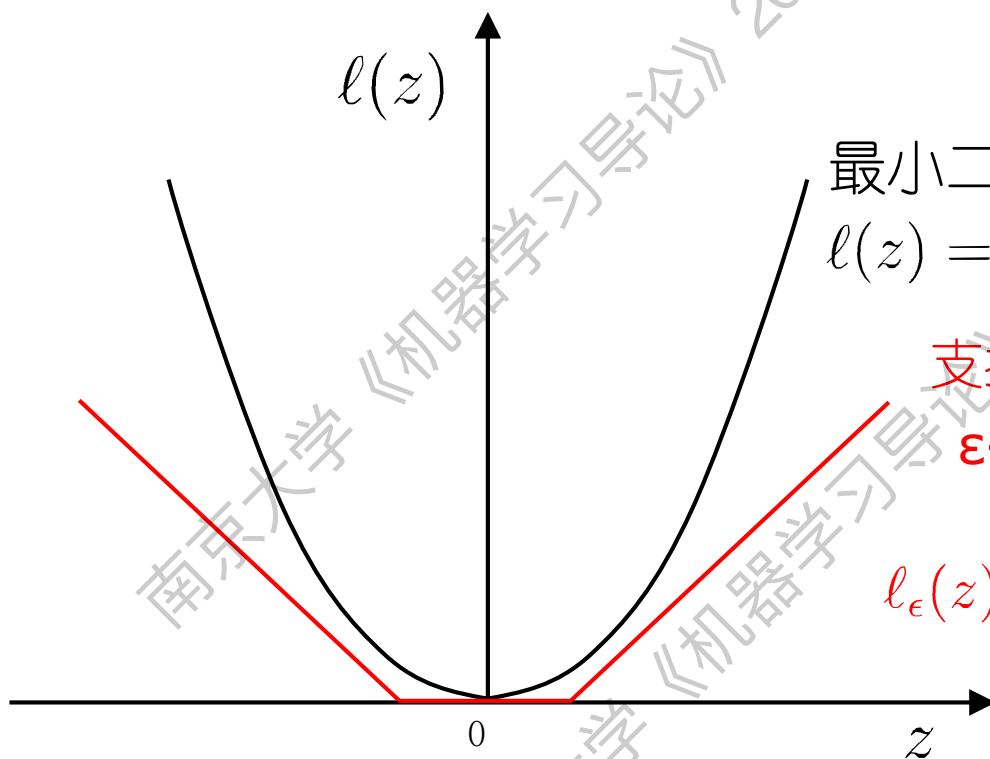
以回归学习为例

基本思路：允许模型输出与实际输出间存在 2ε 的差别



ϵ -不敏感(insensitive)损失函数

落入 2ϵ 间隔带的样本不计算损失



最小二乘损失函数

$$l(z) = z^2$$

支持向量回归使用的

ϵ -不敏感损失函数

$$l_\epsilon(z) = \begin{cases} 0 & \text{if } |z| \leq \epsilon \\ |z| - \epsilon & \text{otherwise} \end{cases}$$

支持向量回归 (SVR)

原始问题

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

对偶问题

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

预测

$$f(\mathbf{x}) = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) \mathbf{x}_i^T \mathbf{x} + b$$

如何使用SVM ?

- 入门级—— 实现并使用各种版本**SVM**
- 专业级—— 尝试、组合核函数
- 专家级—— 根据问题而设计目标函数、替代损失、进而.....

根据当前任务“度身定制”是关键

表示定理 (Representer Theorem)

观察 {

$$\begin{aligned} \text{核 SVM: } f(\mathbf{x}) &= \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{i=1}^m \alpha_i y_i \kappa(\mathbf{x}, \mathbf{x}_i) + b \\ \text{核 SVR: } f(\mathbf{x}) &= \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) \kappa(\mathbf{x}, \mathbf{x}_i) + b \end{aligned}$$

无论SVM还是SVR, 学得模型总能表示成核函数的线性组合

更一般的结论(表示定理): 对于任意单调递增函数 $\Omega : [0, \infty] \mapsto \mathbb{R}$ 和任意非负损失函数 $\ell : \mathbb{R}^m \mapsto [0, \infty]$, 优化问题

$$\min_{h \in \mathbb{H}} F(h) = \Omega(\|h\|_{\mathbb{H}}) + \ell(h(\mathbf{x}_1), h(\mathbf{x}_2), \dots, h(\mathbf{x}_m))$$

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

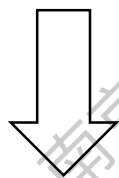
核方法 (Kernel methods)

基于表示定理能得到很多线性模型的“核化” (kernelized)版本

例如 KLDA (Kernelized LDA):

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

$$\max_w J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\phi \mathbf{w}}$$



$$h(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \sum_{i=1}^m \alpha_i \kappa(\mathbf{x}, \mathbf{x}_i)$$

$$\max_{\alpha} J(\alpha) = \frac{\alpha^T \mathbf{M} \alpha}{\alpha^T \mathbf{N} \alpha}$$

“核技巧” (kernel trick)
是机器学习中处理非线性问题的基本技术之一

支持向量机常用软件/工具包

□ LIBSVM

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

□ LIBLINEAR

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

□ SVM^{light}、SVM^{perf}、SVM^{struct}

http://svmlight.joachims.org/svm_struct.html

□ Pegasos

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

□

前往第七站.....

