

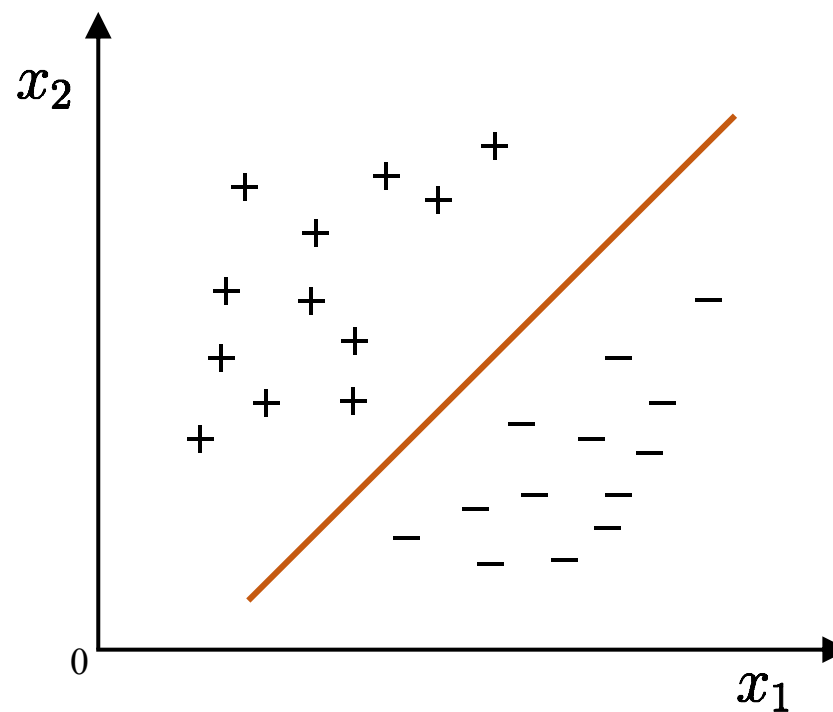
六、支持向量机

主讲教师：周志华

机器学习导论

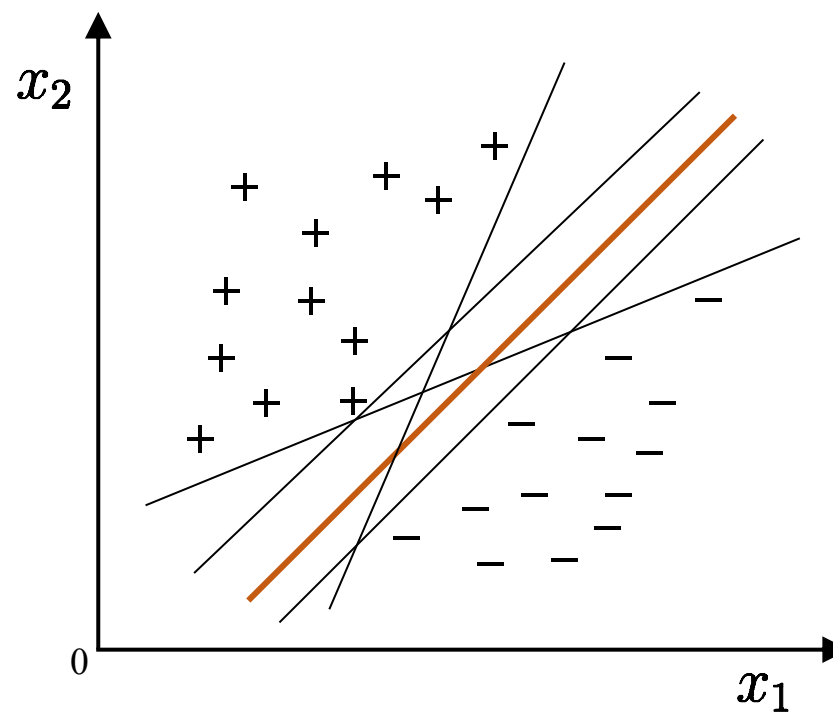
线性分类器回顾

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?

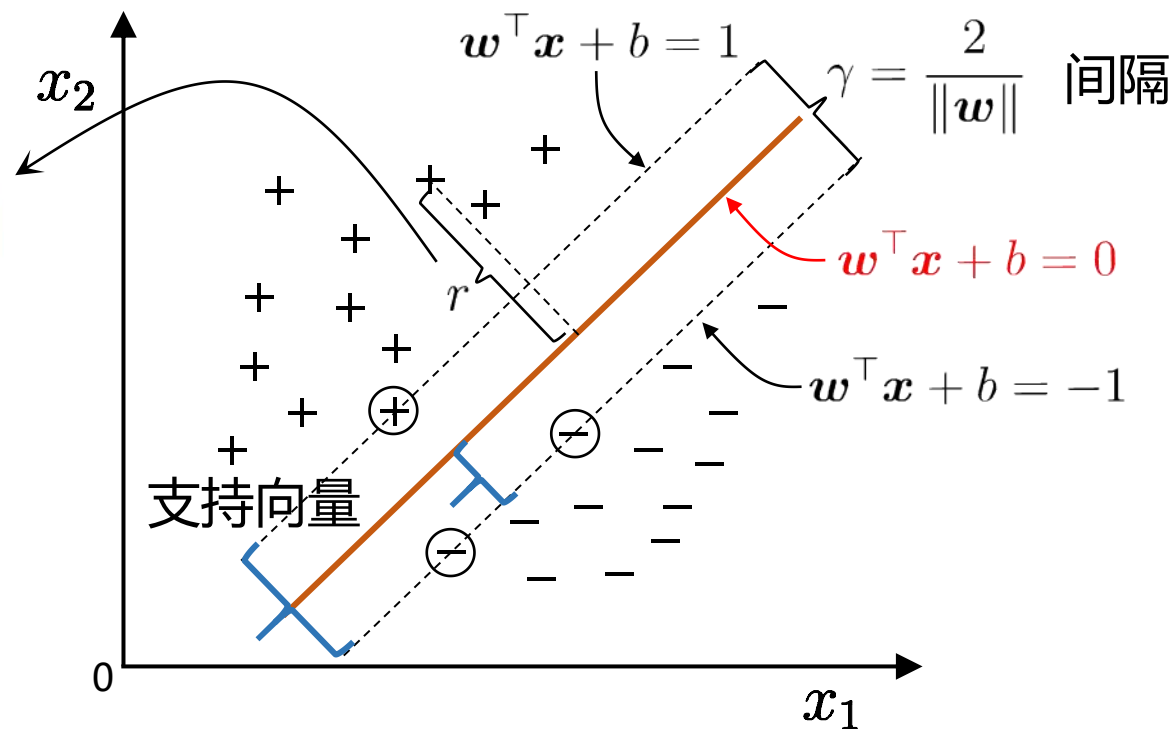


“**正中间**” 的: 鲁棒性最好, 泛化能力最强

间隔(Margin) 与 支持向量(Support Vector)

超平面方程: $w^T x + b = 0$

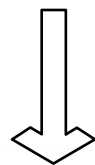
$$r = \frac{|w^T x + b|}{\|w\|}$$



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

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

解的特性

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:

$$\begin{cases} \alpha_i \geq 0; \\ 1 - y_i f(\mathbf{x}_i) \leq 0; \\ \alpha_i (1 - y_i f(\mathbf{x}_i)) = 0. \end{cases} \quad \longrightarrow \quad \begin{array}{l} \text{必有 } \alpha_i = 0 \text{ 或} \\ y_i f(\mathbf{x}_i) = 1 \end{array}$$

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

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 $0 = \sum_{i=1}^m \alpha_i y_i$ 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

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

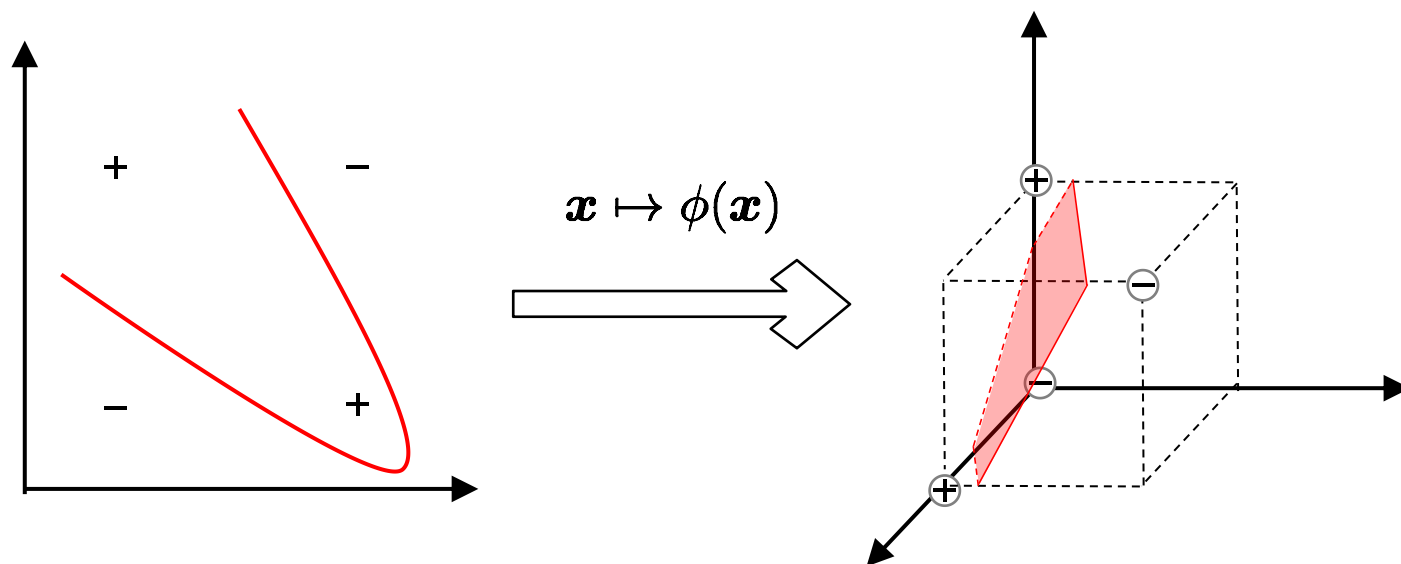
对任意支持向量 (\mathbf{x}_s, y_s) 有 $y_s f(\mathbf{x}_s) = 1$, 由此可解出 b

为提高鲁棒性, 通常使用所有支持向量求解的平均值

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线性可分

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

在特征空间中

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

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

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

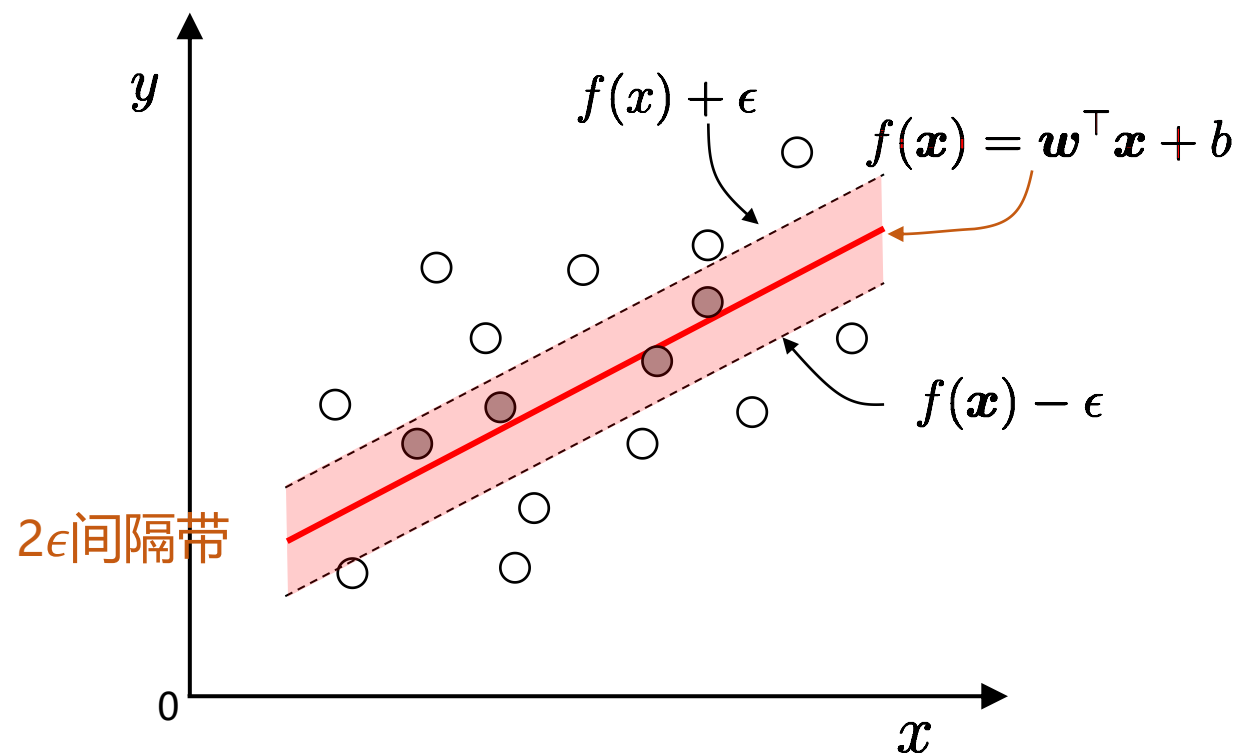
任何一个核函数，都隐式地定义了一个RKHS (Reproducing Kernel Hilbert Space, 再生核希尔伯特空间)

“核函数选择” 成为决定支持向量机性能的关键！

如何使用SVM
解决自己特定的任务？

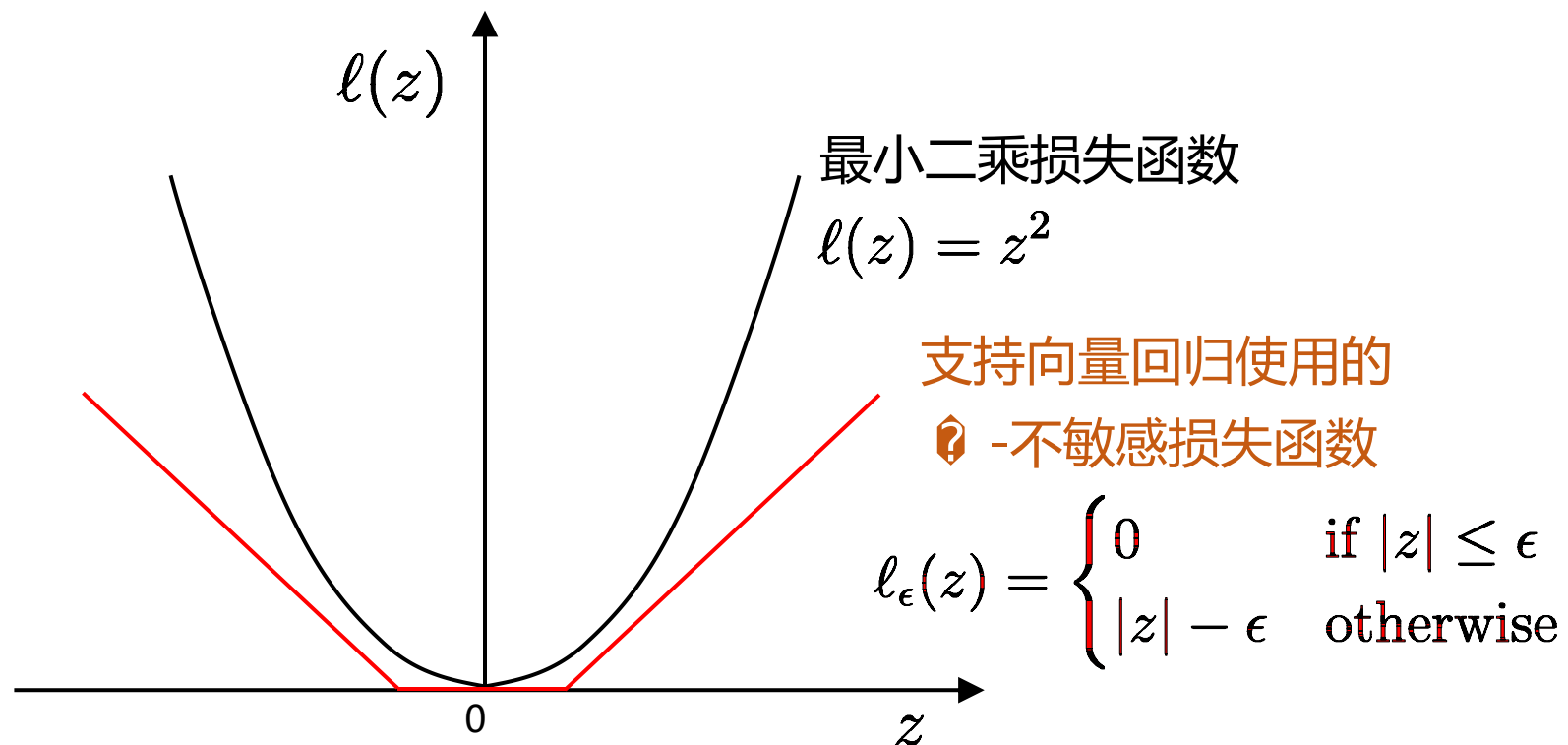
以回归学习为例

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



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

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

现实应用中

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