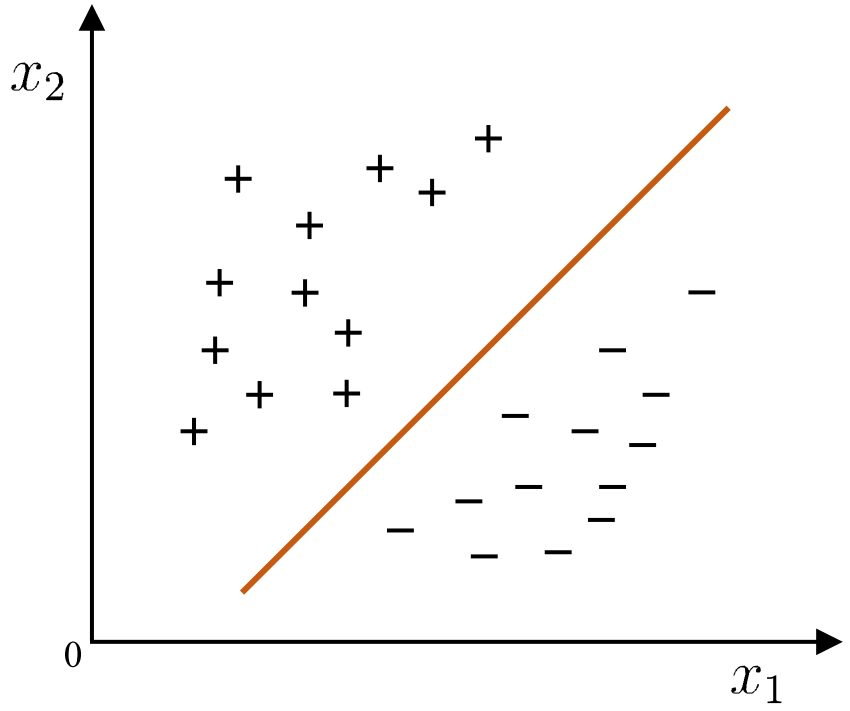
六、支持向量机

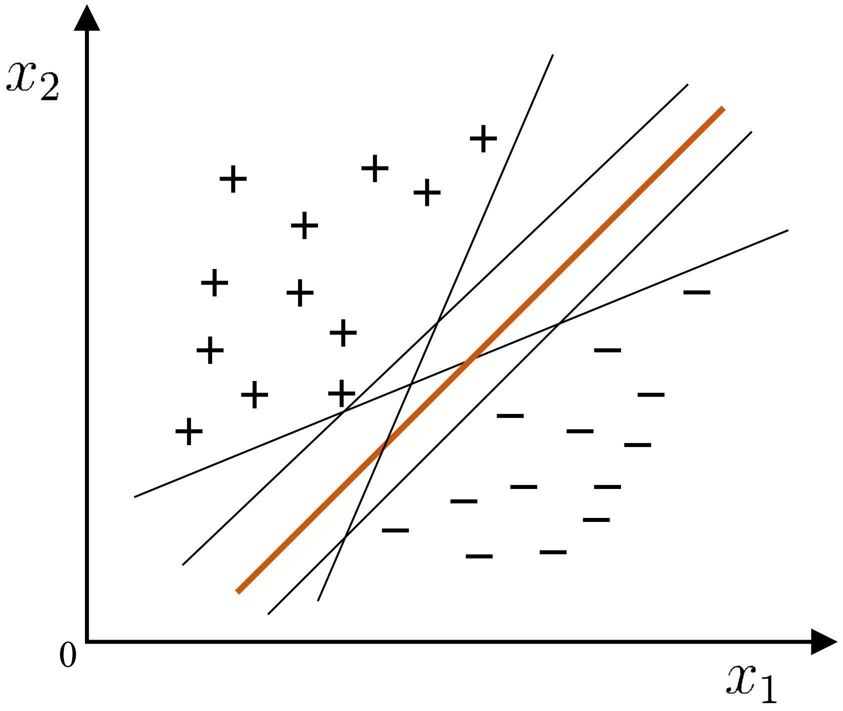
线性分类器回顾

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

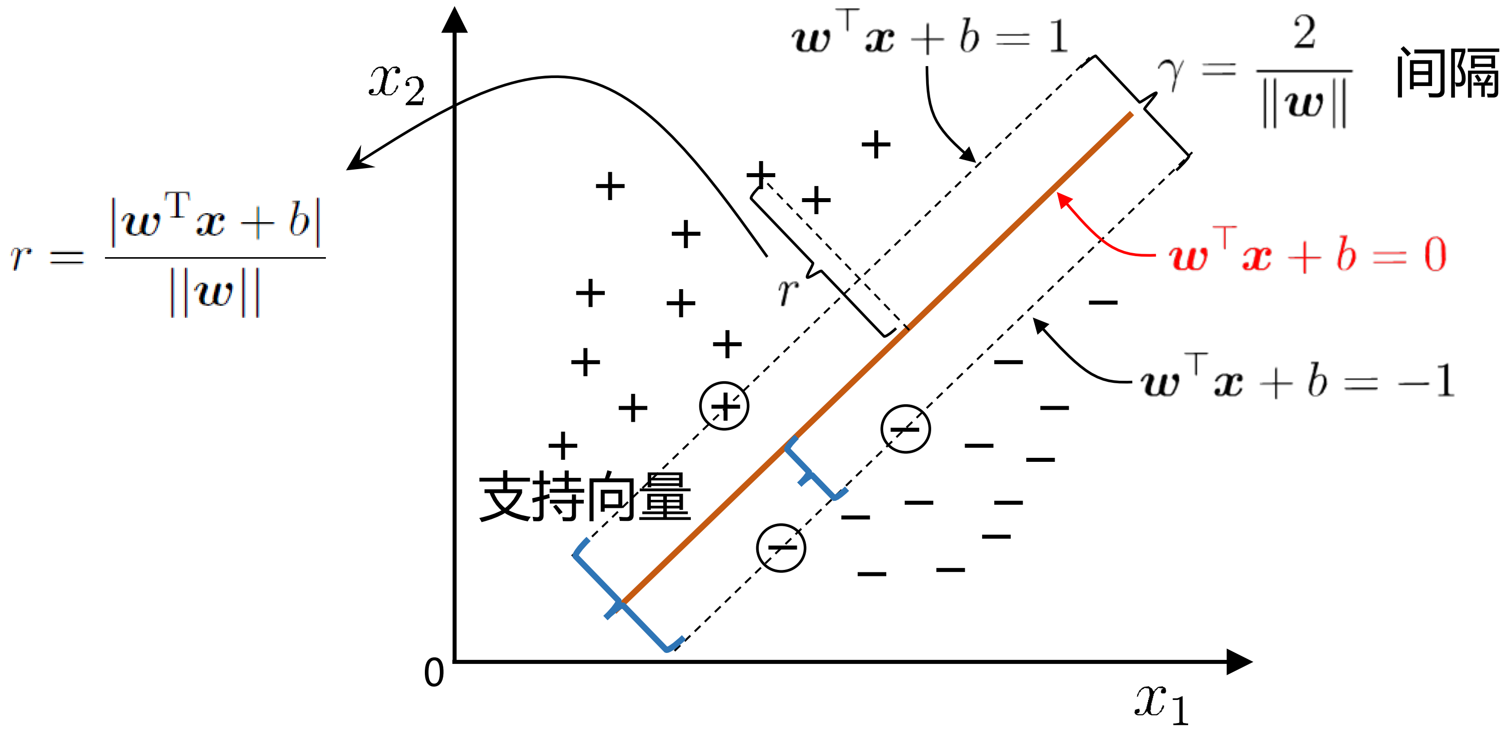
将训练样本分开的超平面可能有很多, 哪一个更好呢?



“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: $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$



支持向量机基本型

最大间隔: 寻找参数 $\boldsymbol{w}$ 和 b , 使得 γ 最大

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

$$\Downarrow$$

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

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

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

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

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

- 第三步：回代可得

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \ , \quad \alpha_i \geqslant 0 \ , \quad i = 1, 2, \ldots, m$$

解的特性

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

KKT条件：

$$\begin{cases} \alpha_i \geq 0\,; \\ 1 - y_i f(\boldsymbol{x}_i) \leq 0\,; \\ \alpha_i \left(1 - y_i f(\boldsymbol{x}_i)\right) = 0\,. \end{cases} \Longrightarrow \text{必有 } \alpha_i = 0 \text{ 或 } y_i f(\boldsymbol{x}_i) = 1$$

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

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

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

用 α_i 表示α_j , 代入对偶问题

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

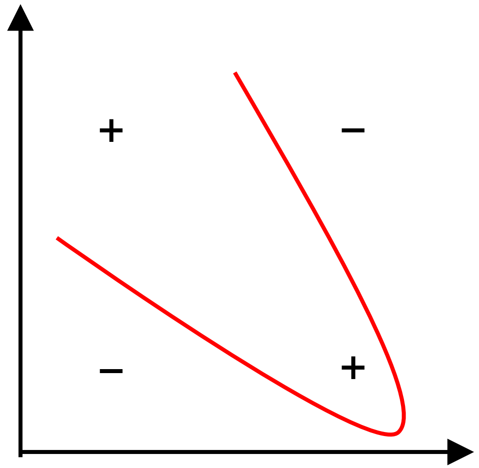
对任意支持向量$(\boldsymbol{x}_s, y_s)$ 有 $y_s f(\boldsymbol{x}_s) = 1$ ，由此可解出 b

为提高鲁棒性，通常使用所有支持向量求解的平均值

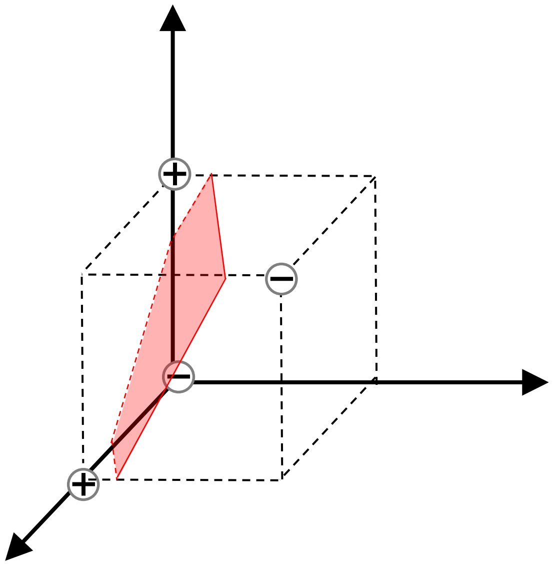
特征空间映射

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



$\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$



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

在特征空间中

原始问题

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

对偶问题

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boxed{\phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j)} \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0 \ , \quad \alpha_i \geqslant 0 \ , \quad i = 1, 2, \ldots, m \end{aligned}$$

预测

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

只以内积形式出现

核函数
(Kernel Function)

基本思路：设计核函数

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

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!

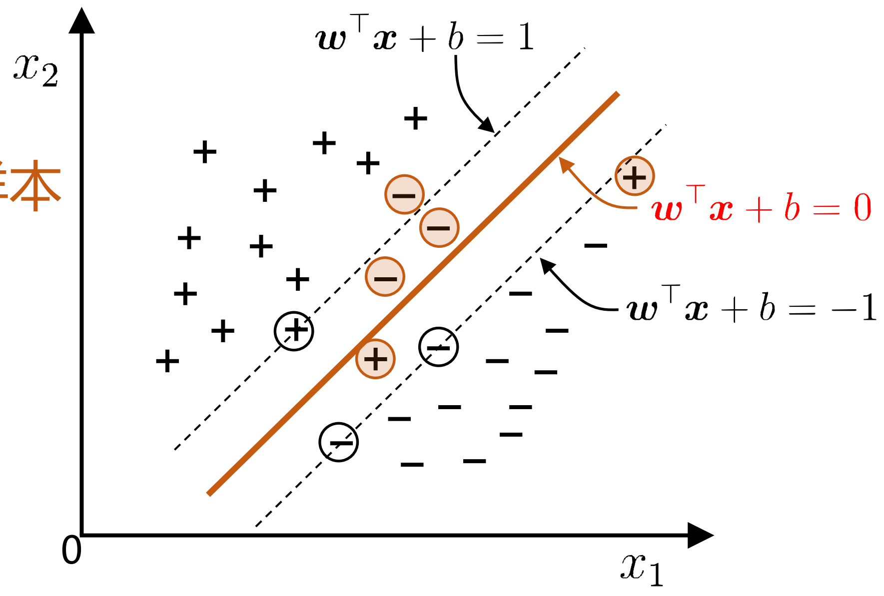
软间隔

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

不满足约束的样本



优化目标

基本思路：最大化间隔的同时，让不满足约束 $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b) \geqslant 1$ 的样本尽可能少

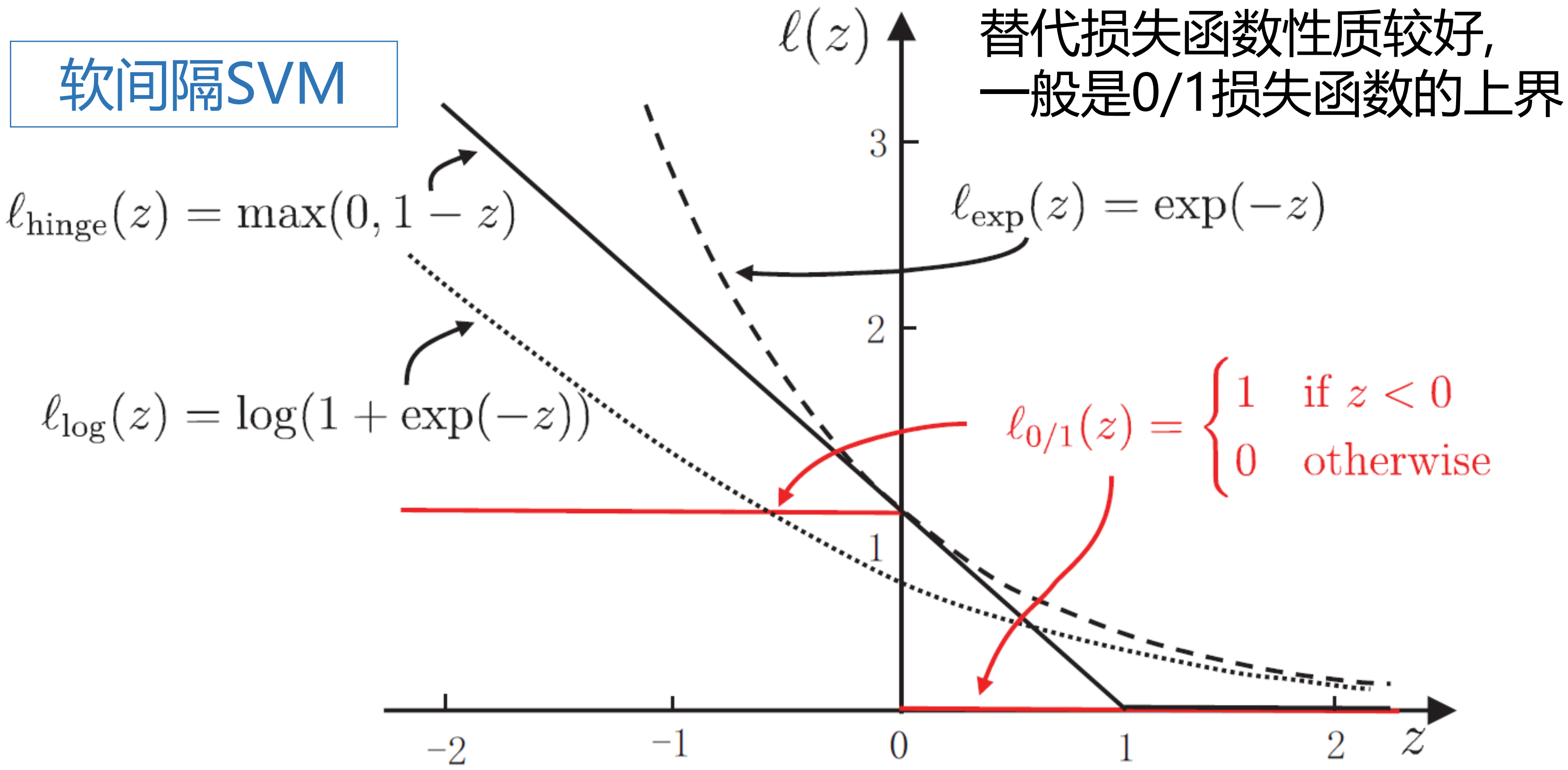
$$\boxed{\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_{0/1}\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)-1\right)}$$

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)



- 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

软间隔支持向量机

原始问题

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

引入“松弛量”

(Slack Variables)

ξ_i

$$\min_{\boldsymbol{w},b,\xi_i} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i$$

$$\text{s.t.}\quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1 - \xi_i \quad \xi_i \geqslant 0\ ,\ i = 1, 2, \ldots, m.$$

对偶问题

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$$

$$\text{s.t.} \sum_{i=1}^{m} \alpha_i y_i = 0\ ,\ 0 \leqslant \alpha_i \leqslant C\ ,\quad i = 1, 2, \ldots, m\ .$$

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

正则化
(Regularization)

统计学习模型（例如 SVM）的更一般形式

正则化项

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

归纳偏好

结构风险
(Structural Risk)
描述模型本身的某些性质

经验风险
(Empirical Risk)
描述模型与训练数据的契合程度

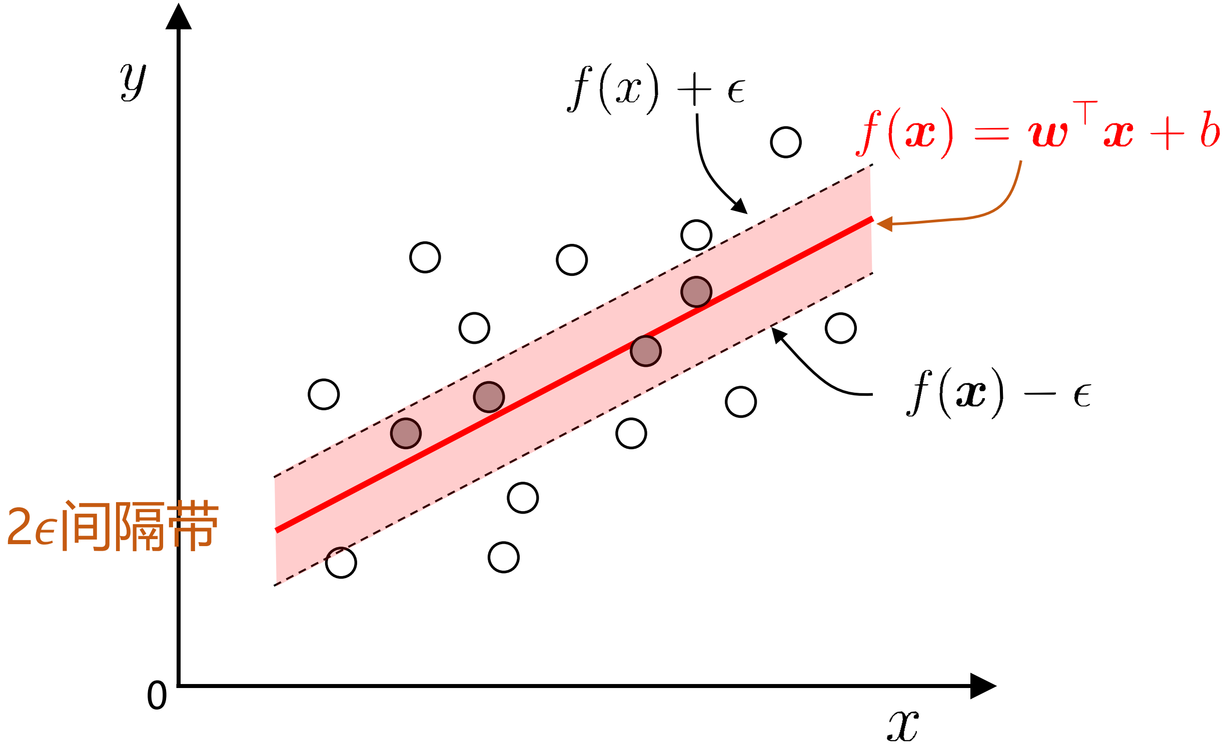
- 正则化可理解为“罚函数法”
  通过对不希望的结果施以惩罚，使得优化过程趋向于希望目标
- 从贝叶斯估计的角度，则可认为是提供了模型的先验概率

如何使用SVM

解决自己特定的任务？

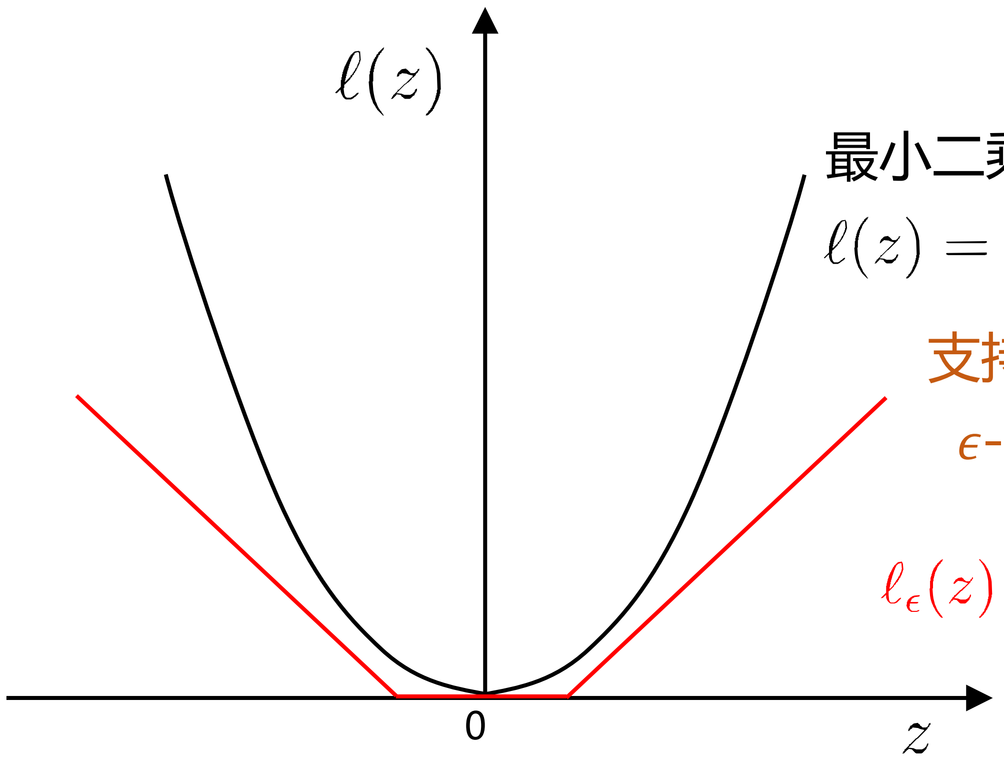
以回归学习为例

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

$$\ell(z) = z^2$$

支持向量回归使用的

ϵ-不敏感损失函数

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

支持向量回归 (SVR)

原始问题

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

对偶问题

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

预测

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

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