六、支持向量机

主讲教师：赵鹏

大纲

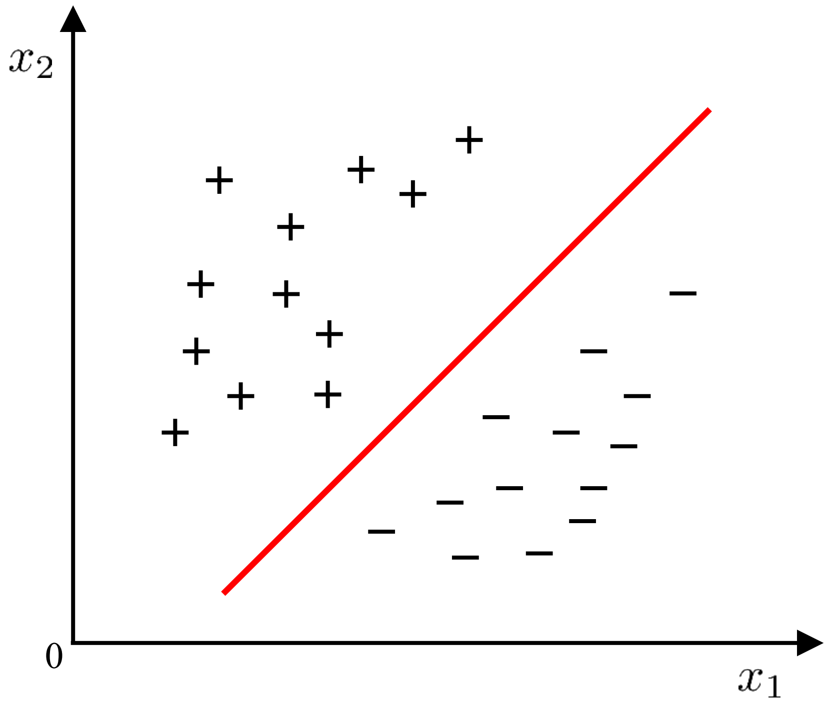
- 硬间隔支持向量机
- 核方法
- 软间隔支持向量机

大纲

- 硬间隔支持向量机
- 核方法
- 软间隔支持向量机

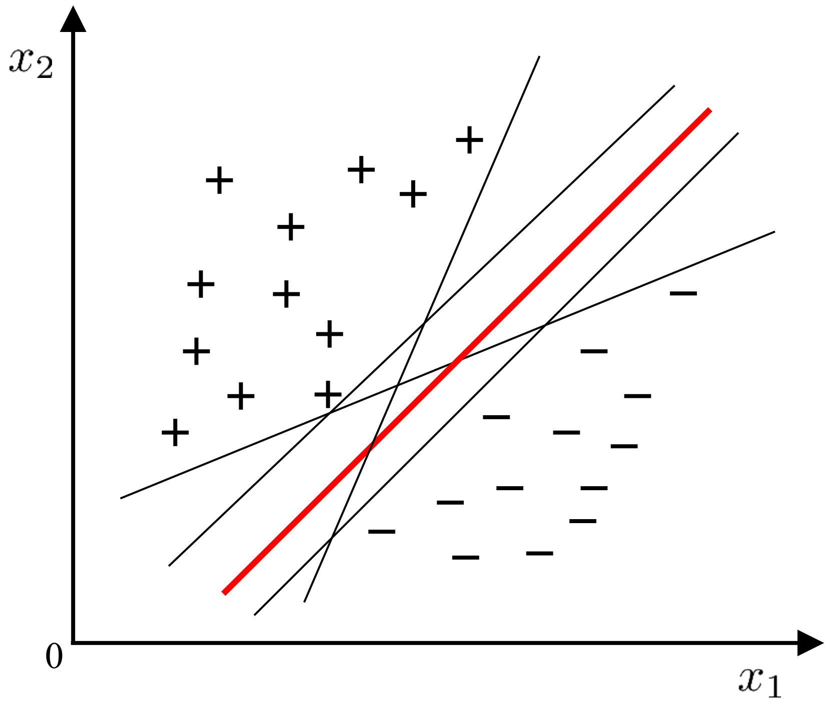
线性分类器回顾

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

形式化

线性可分样本集: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$

线性超平面: $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$

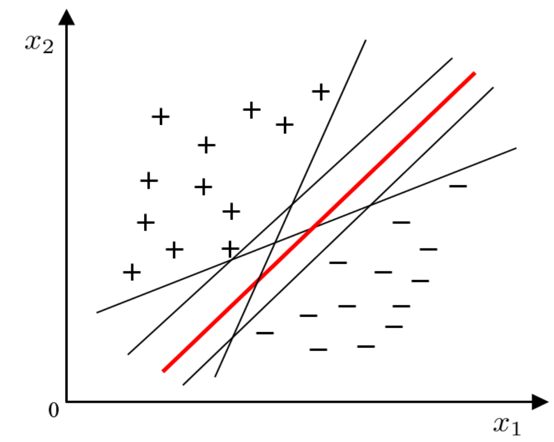
线性分类器: $\boldsymbol{x} \mapsto \mathbf{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$

- 任意点 $\boldsymbol{x}_0$ 到超平面距离为 $r(\boldsymbol{x}_0) = \dfrac{|\boldsymbol{w}^\top \boldsymbol{x_0} + b|}{\|\boldsymbol{w}\|}$
- 最大化样本中相较超平面的最小距离

$$\max_{\boldsymbol{w},b} \min_{i\in[m]} \frac{|\boldsymbol{w}^\top \boldsymbol{x_i} + b|}{\|\boldsymbol{w}\|}$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 0, \ i \in [m].$$

$$\Rightarrow \max_{\boldsymbol{w},b} \min_{i\in[m]} \frac{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)}{\|\boldsymbol{w}\|}$$



"正中间"的: 鲁棒性最好, 泛化能力最强

注意到分子分母对$(\boldsymbol{w}, b)$缩放同倍数不影响

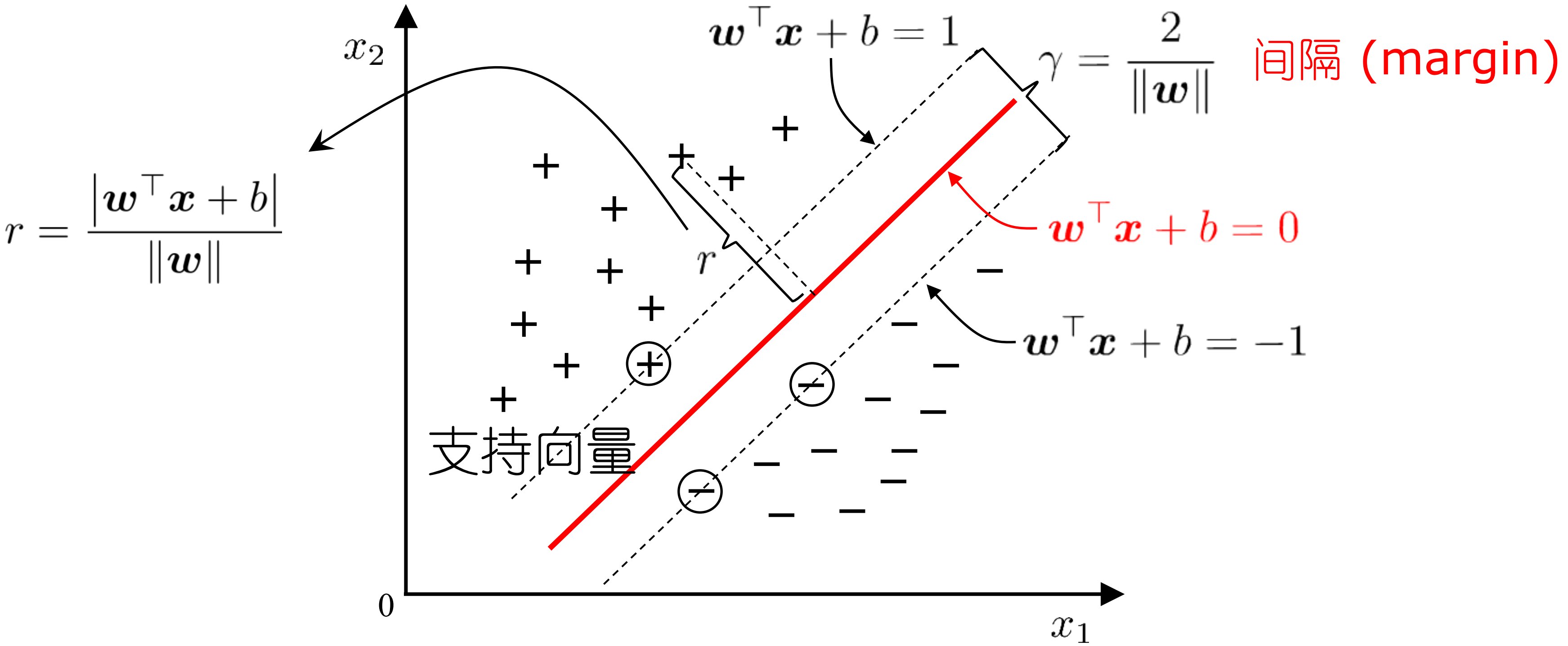
不妨缩放之使得 $\min_{i\in[m]} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1$

$$\max_{\boldsymbol{w},b} \quad \frac{1}{\|\boldsymbol{w}\|}$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i \in [m].$$

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 ，使得间隔最大

$$\underset{\boldsymbol{w},b}{\arg\max} \quad \frac{1}{\|\boldsymbol{w}\|}$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i \in [m].$$

$$\Downarrow$$

$$\underset{\boldsymbol{w},b}{\arg\min} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i \in [m].$$

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

支持向量机分析：对偶问题

$$\arg\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t.} \quad y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1, \ \ i \in [m].$$

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

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

- 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

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j$$
$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \quad \alpha_i \geqslant 0, \quad i \in [m].$$

支持向量机分析：对偶问题

$$\begin{aligned}\arg\min_{\boldsymbol{w},b}\ & \frac{1}{2}\|\boldsymbol{w}\|^2\\ \text{s.t.}\ & y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i+b\right)\geq 1,\ i\in[m].\end{aligned}$$

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

回顾KKT条件性质

KKT Conditions

- If f_i are convex, h_i are affine, $\tilde{x},\tilde{\lambda},\tilde{\nu}$ satisfy

$$\begin{aligned} f_i(\tilde{x}) &\leq 0, & i\in[m]\\ h_j(\tilde{x}) &= 0, & j\in[n]\\ \tilde{\lambda}_i^\star &\geq 0, & i\in[m]\\ \tilde{\lambda}_i^\star f_i(\tilde{x}) &= 0, & i\in[m]\end{aligned}$$

互补松弛

$$\nabla f_0(\tilde{x})+\sum_{i=1}^m\tilde{\lambda}_i^\star\nabla f_i(\tilde{x})+\sum_{j=1}^n\tilde{\nu}_j^\star\nabla h_j(\tilde{x})=0$$

$\Rightarrow$ Then, $\tilde{x}$ and $\tilde{\lambda},\tilde{\nu}$ are primal and dual optimal, with strong du

Sufficient Condition

For any *convex* optimization problem wi
differentiable objective and constraint functions, a
points that satisfy the KKT conditions are prim
and dual optimal, and they enjoy strong duality.

KKT Conditions

- Suppose strongly duality holds

$$\begin{aligned} f_0(x^\star) &= g(\lambda^\star,\nu^\star)\\ &= \inf_x\left(f_0(x)+\sum_{i=1}^m\lambda_i^\star f_i(x)+\sum_{j=1}^n\nu_j^\star h_j(x)\right)\\ &= f_0(x^\star)+\sum_{i=1}^m\lambda_i^\star f_i(x^\star)+\sum_{j=1}^n\nu_j^\star h_j(x^\star)\\ &= f_0(x^\star)\end{aligned}$$

强对偶满足，
则互补松弛

– Equality in the third line implies $x^\star$ minimizes $L(x,\lambda^\star,\nu^\star)$

– Equality in the last line implies $\sum_{i=1}^m\lambda_i^\star f_i(x^\star)=0$

$\Rightarrow$ Other expressions $\lambda_i^\star>0\Rightarrow f_i(x^\star)=0$
$f_i(x^\star)<0\Rightarrow\lambda_i^\star=0$

This is also called “Complementary Slackness”

支持向量机分析：对偶问题

$$\arg\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t.} \quad y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1, \; i \in [m].$$

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

解的特性：

KKT条件：

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

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”

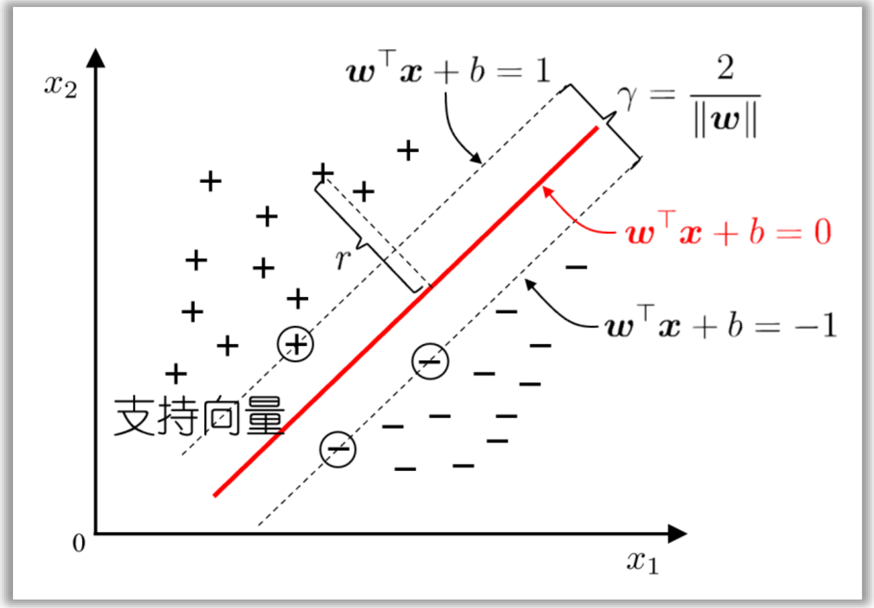
由对偶问题及KKT条件分析可知：

对于任意 $i \in [m]$ 必有 $\alpha_i = 0$ 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

算法核心复杂度主体与支持向量个数有关



对偶问题的求解

对偶问题

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

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

仍然是一个二次规划（QP）问题，可以用一般的QP solver进行求解

计算/存储复杂度均与样本总数 m 有关

利用结构特性可以做得更快

SMO (Sequential Minimal Optimization) [Platt, 1998]

对偶问题的求解 – SMO方法

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

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

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

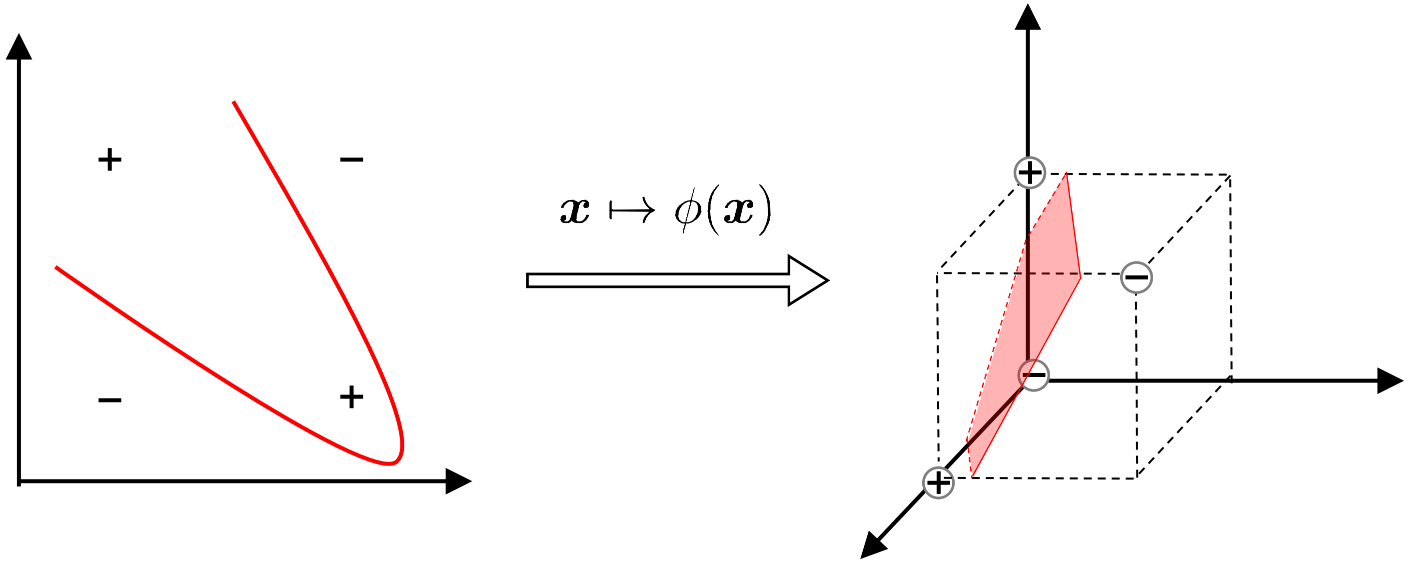
大纲

- 硬间隔支持向量机
- 核方法
- 软间隔支持向量机

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

记样本 $\boldsymbol{x}$ 映射后向量为 $\phi(\boldsymbol{x})$，划分超平面对应模型为 $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b$

原始问题

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

对偶问题

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

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 $\kappa\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \phi\left(\boldsymbol{x}_i\right)^{\top} \phi\left(\boldsymbol{x}_j\right)$

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

定理 6.1 (核函数) 令 $\mathcal{X}$ 为输入空间，$\kappa(\cdot,\cdot)$ 是定义在 $\mathcal{X} \times \mathcal{X}$ 上的对称函数，则 κ 是核函数当且仅当对于任意数据 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$，“核矩阵” (kernel matrix) $\mathbf{K}$ 总是半正定的:

$$\mathbf{K} = \begin{bmatrix} \kappa(\boldsymbol{x}_1,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_i,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_m,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_m) \end{bmatrix}.$$

核函数 (kernel function)

基本思路：设计核函数 $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$

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！

核函数

常用核函数

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)^d$	$d\geqslant 1$ 为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\lVert\boldsymbol{x}_i-\boldsymbol{x}_j\rVert^2}{2\sigma^2}\right)$	$\sigma>0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\lVert\boldsymbol{x}_i-\boldsymbol{x}_j\rVert}{\sigma}\right)$	$\sigma>0$
Sigmoid 核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\tanh(\beta\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j+\theta)$	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 $g(\boldsymbol{x})$，

均为核函数
$$\begin{cases}\gamma_1\kappa_1+\gamma_2\kappa_2\\ \kappa_1\otimes\kappa_2(\boldsymbol{x},\boldsymbol{z})=\kappa_1(\boldsymbol{x},\boldsymbol{z})\kappa_2(\boldsymbol{x},\boldsymbol{z})\\ \kappa(\boldsymbol{x},\boldsymbol{z})=g(\boldsymbol{x})\kappa_1(\boldsymbol{x},\boldsymbol{z})g(\boldsymbol{z})\end{cases}$$

大纲

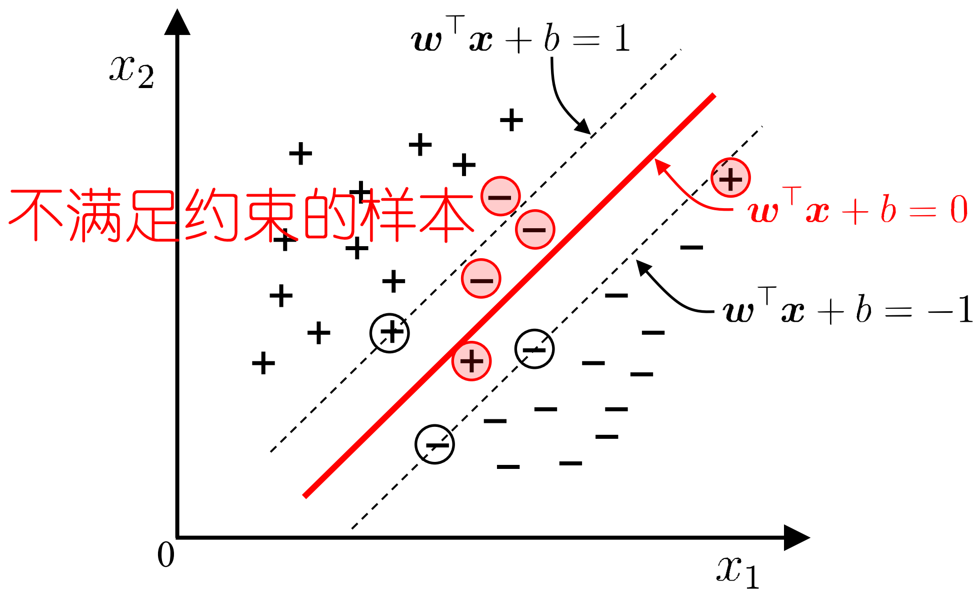
- 硬间隔支持向量机
- 核方法
- 软间隔支持向量机

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



优化目标

"硬间隔" 支持向量机基本型

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t.} \quad y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1, \; i \in [m].$$

"软间隔" 基本思路：

最大化间隔的同时，让 $y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1$约束违背程度尽可能少

⇨ 引入 "松弛变量" (slack variables)

$\xi_1, \xi_2, \ldots, \xi_m$，其中$\xi_i \geq 0$度量第 i个约束违背程度

硬约束 $y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1$ ⇨ 软约束 $y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1 - \xi_i$

优化目标

“硬间隔” 支持向量机基本型

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

“软间隔” 支持向量机优化式

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi_i} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0,\ i \in [m]. \end{aligned}$$

“软间隔”支持向量机

优化目标

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

$$\text{s.t.} \quad y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i \in [m].$$

可等价地写为：

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

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

软间隔支持向量机

原始问题

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

$$\text{s.t.} \quad y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1 - \xi_i,\ \ \xi_i \geq 0,\ \ i \in [m].$$

对偶问题

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

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

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

理解“软间隔”支持向量机

“软间隔”支持向量机优化目标：

$$\boxed{\min_{\boldsymbol{w},b} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_{\text{hinge}}\left(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\right)}$$

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

hinge损失是0/1损失的上界，且性质较好

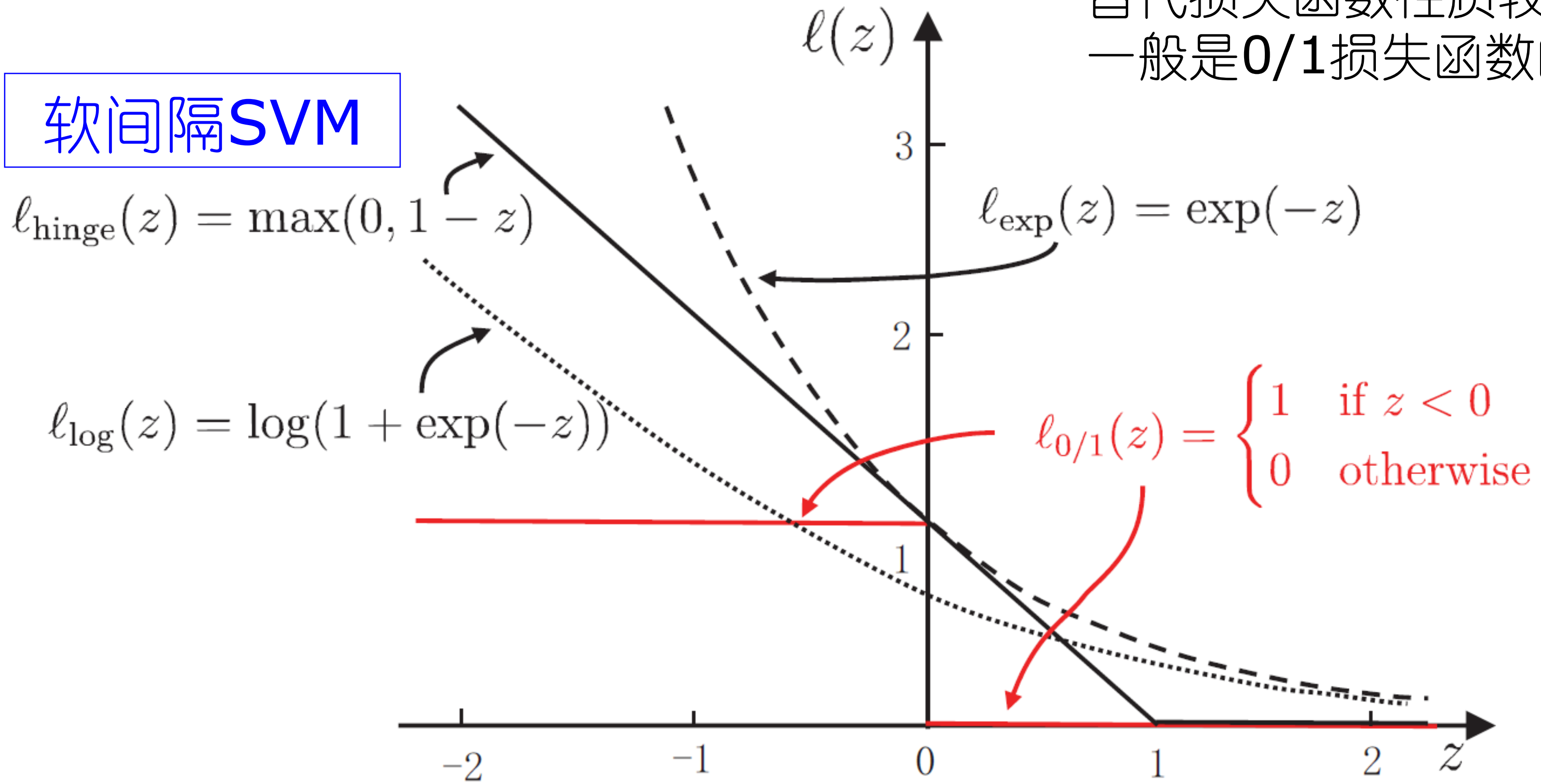
可以认为是0/1损失优化问题的“凸放松”

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

0/1损失 $\ell_{0/1}(z) = \mathbf{1}[z<0]$

替代损失 (surrogate loss)

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

正则化项

$$\min_{f} \underbrace{\Omega(f)}_{\text{结构风险}} + C \underbrace{\sum_{i=1}^{m} l(f(\boldsymbol{x}_i), y_i)}_{\text{经验风险}}$$

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

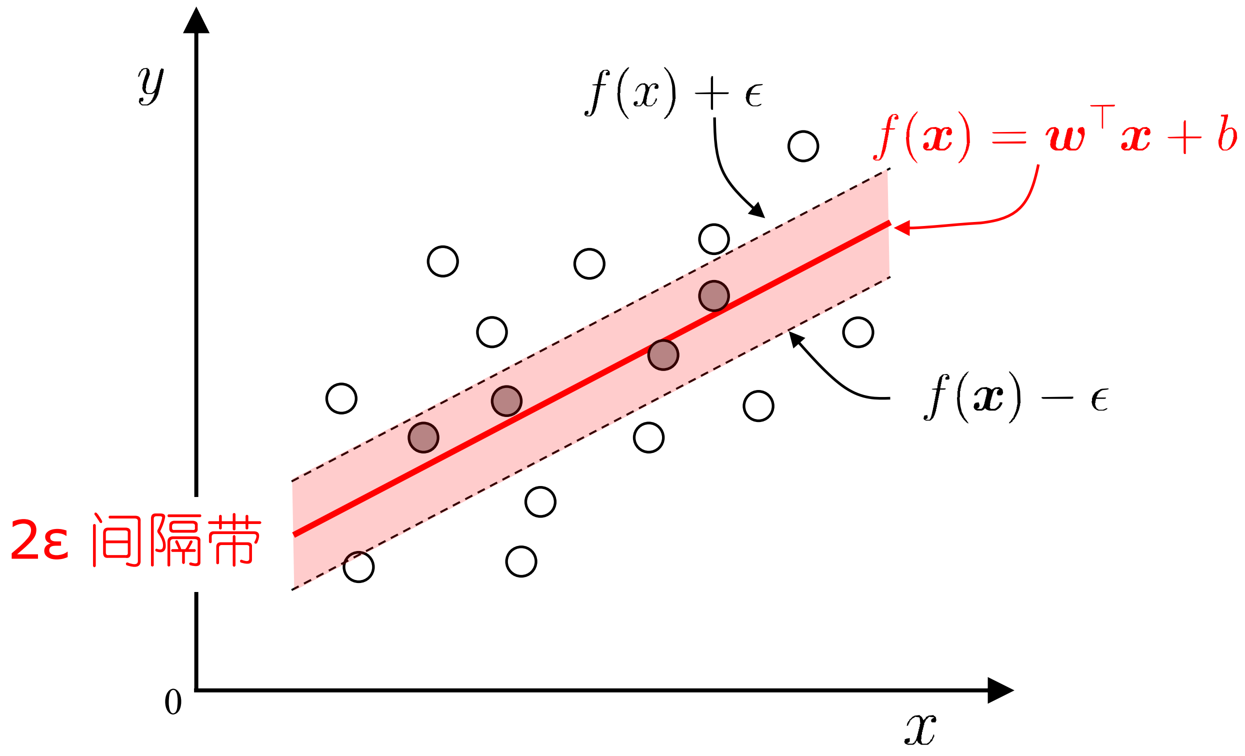
归纳
偏好

- 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？

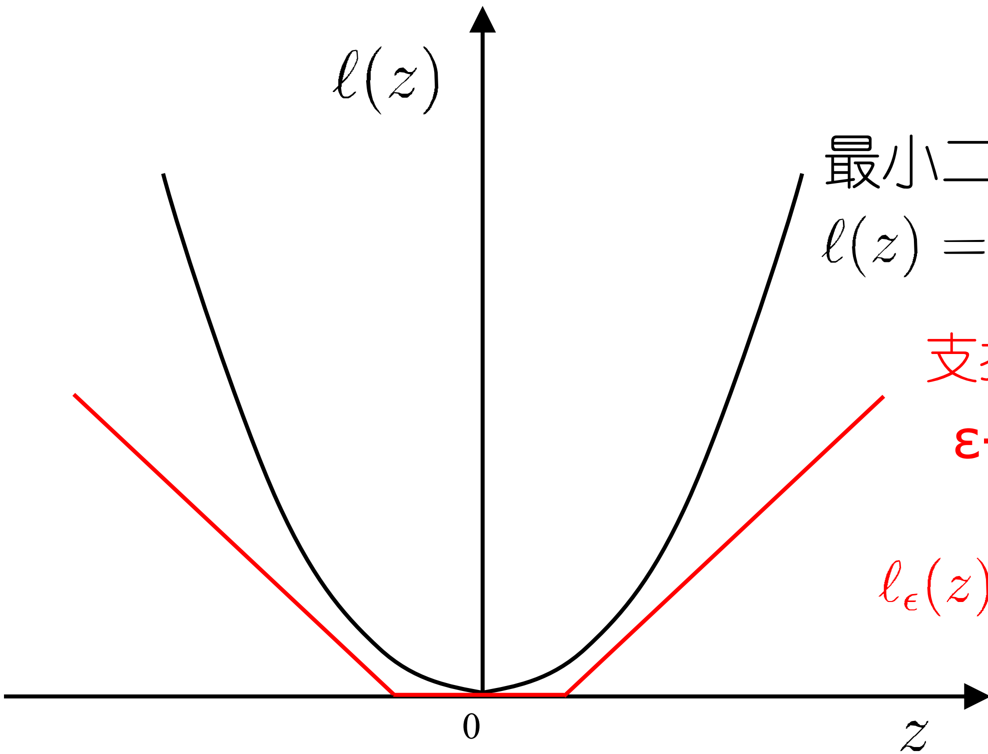
以回归学习为例

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

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

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

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

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

表示定理 (Representer Theorem)

观察
- 核 SVM: $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b \ = \sum_{i=1}^{m} \alpha_i y_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i) + b$
- 核 SVR: $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b \ = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i) \kappa(\boldsymbol{x}, \boldsymbol{x}_i) + b$

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

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

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

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

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

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

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

“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

SVM 与统计学习简史

V. Vapnik (1936-)

1963：Vapnik 提出支持向量的概念

1968：Vapnik 和 Chervonenkis 提出 VC 维

1974：提出结构风险最小化原则

… …　苏联解体前一年(1990), Vapnik 来到美国

1995：Support Vector Network 文章发表

1995: 《The Nature of Statistical Learning》出版

1998: SVM 在文本分类上取得巨大成功

1998: 《Statistical Learning Theory》出版

… …

“ Nothing is more practical than a good theory ”

-- V. Vapnik

前往第七站……