SVM与LR（逻辑回归）本质区别是有损失函数

- 对率回归优势：输出有概率意义，不止是给出预测标记，而且能直接用于多分类任务
- SVM无概率意义
- 对率回归劣势：光滑单减函数不能导出类似支持向量的概念，依赖于更多的训练样本，训练开销更大

一、基础模型

任意点x到超平面(w,b)距离

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

间隔（margin）：两异类支持向量到超平面距离

$$\gamma = \frac{2}{||\boldsymbol{w}||} ,$$

基本模型：最大间隔

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

二、对偶问题

> 每个SVM变种都要对偶

- 原问题m个一次不等约束，目标二次，有d+1个优化变量
- 对偶问题m个优化变量，目标二次，m个线性等和不等约束

> m>d:大数据问题，用原问题
>
> d>m:高纬度问题，用对偶

1、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

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

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

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

KKT条件：

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

2、对偶的稀疏性质

> 训练完成后，大部分样本无关，只剩支持向量

- αi=0即w不受x影响
- yi*f（xi）=1即xi是支持向量

3、解对偶--SMO算法

> 高效解决二次规划，因为固定了其他之后好算

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

解b

用所有支持向量平均最好

$$b=\frac{1}{|S|}\sum_{s\in S}\left(y_s-\sum_{i\in S}\alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_s\right)$$

三、核函数

> 有限维问题一定存在更改特征方案，高维特征空间使样本可分
>
> 基于对偶问题，针对原问题意义不大

- 映射函数不好算，用核函数换元
- 设计特征难（模型改良关键），先表示出来

> 但其实不需要知道怎么设计的，知道怎么算就可以了

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

核函数定义了一种相似性，类似欧氏距离的刻画尺度，因此隐式定义了再生核希尔伯特空间RKHS

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:

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

Mercer 定理:

> 把对函数的判定转化成对矩阵的判定

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

对于任一半正定核矩阵、总能找到一个与之对应的映射φ

- a*b就是对称函数，因为有交换性
- 核矩阵：矩阵第ij项=k（xi，xj）

常用核函数

> 模型参数选择空间很大

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	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\Vert\boldsymbol{x}_i-\boldsymbol{x}_j\Vert^2}{2\sigma^2}\right)$	$\sigma>0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\Vert\boldsymbol{x}_i-\boldsymbol{x}_j\Vert}{\sigma}\right)$	$\sigma>0$
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, $\beta>0$, $\theta<0$

经验：文本用线性核，不明情况先试高斯核

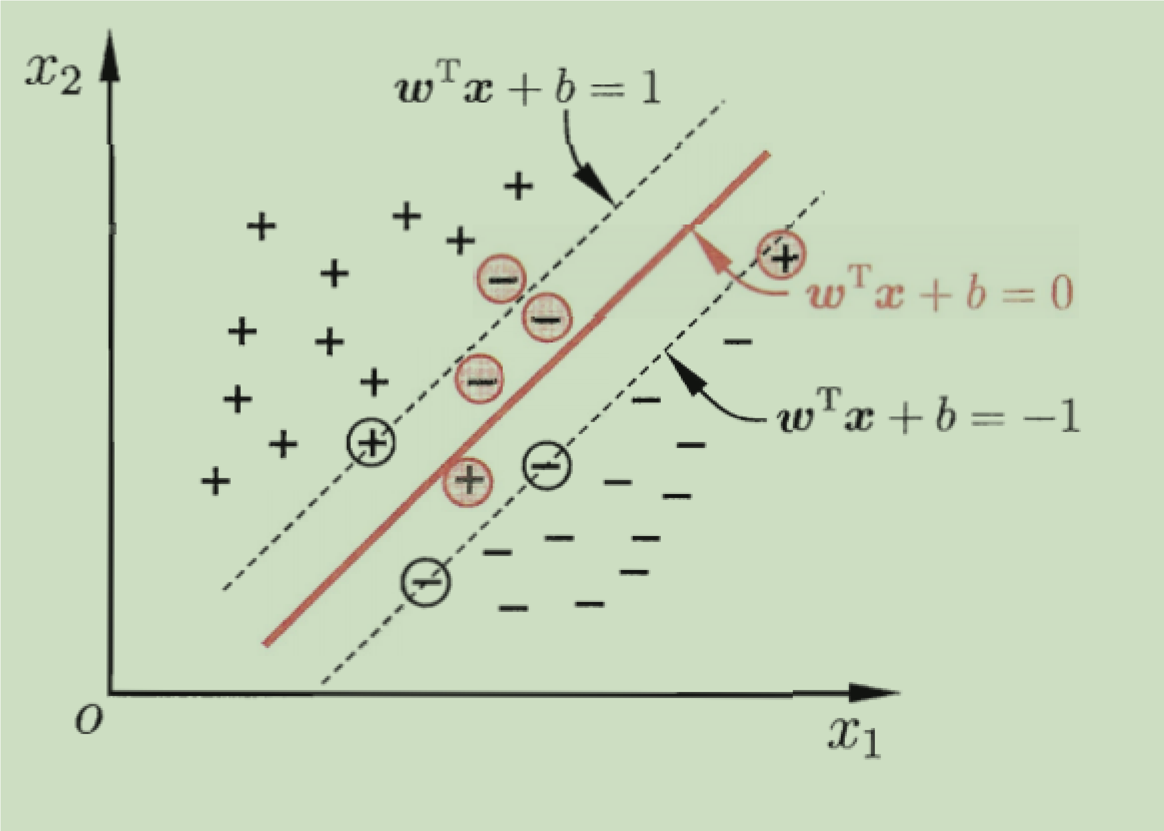
> 多项式d= 1 时退化为线性核.
>
> 高斯核亦称RBF核.

自由组合生成

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 $g(\boldsymbol{x})$，

$$\text{均为核函数}\ \left\{\begin{array}{l}\gamma_1\kappa_1+\gamma_2\kappa_2\\ \kappa_1\otimes\kappa_2(\boldsymbol{x},\boldsymbol{z})=\kappa_1(\boldsymbol{x},\boldsymbol{z})\kappa_2(\boldsymbol{x},\boldsymbol{z})\\ \kappa(\boldsymbol{x},\boldsymbol{z})=g(\boldsymbol{x})\kappa_1(\boldsymbol{x},\boldsymbol{z})g(\boldsymbol{z})\end{array}\right.$$

四、软间隔



原因：

- 难找核函数使得训练样本在特征空间中线性可分
- 找到了也不知道是不是过拟合

因此允许一部分样本不满足约束（分类错误&在硬间隔内部）

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

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

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

- C有限时允许出错：C的大小代表了对出错的容忍程度

替代损失函数surrogate loss

> 01损失函数非凸非连续，数学性质不好

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

指数损失(exponential loss): $\ell_{exp}(z) = \exp(-z)$;

对率损失(logistic loss): $\ell_{log}(z) = \log(1 + \exp(-z))$.

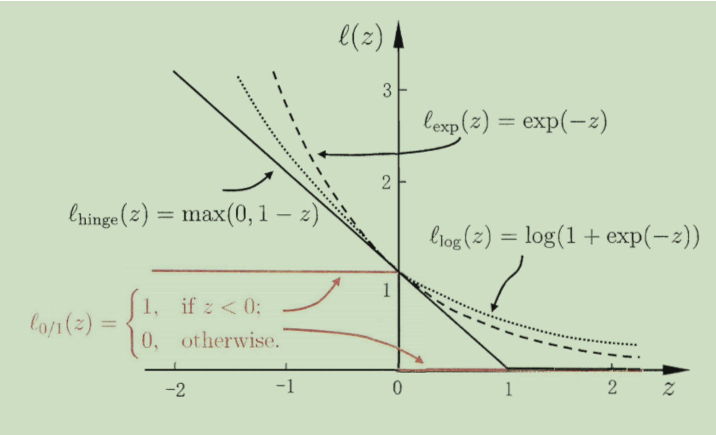


图 6.5 三种常见的替代损失函数: hinge损失、指数损失、对率损失

引入松弛变量（slack）

- 换元，刻画不满足约束程度
- 松弛变量都大于等于0

原始问题

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

引入“松弛变量”(slack variables) ξ_i

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

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

结论

对偶问题

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

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

与“硬间隔SVM”的区别

仍然只与支持向量有关（αi>0）

类似式(6.13)，对软间隔支持向量机，KKT 条件要求

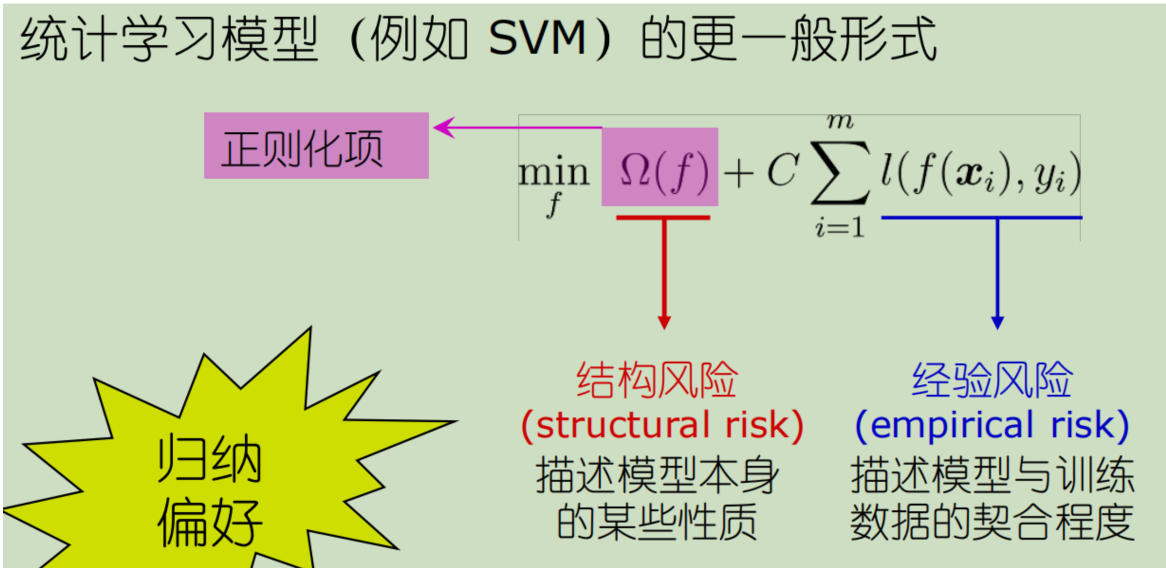
$$
\begin{cases}
\alpha_i \geqslant 0\,, \quad \mu_i \geqslant 0\,, \\
y_i f(\boldsymbol{x}_i) - 1 + \xi_i \geqslant 0\,, \\
\alpha_i \left(y_i f\left(\boldsymbol{x}_i\right) - 1 + \xi_i\right) = 0\,, \\
\xi_i \geqslant 0\,, \quad \mu_i \xi_i = 0\,.
\end{cases}
\tag{6.41}
$$

于是，对任意训练样本 $(\boldsymbol{x}_i, y_i)$，总有 $\alpha_i = 0$ 或 $y_i f(\boldsymbol{x}_i) = 1 - \xi_i$. 若 $\alpha_i = 0$，则该样本不会对 $f(\boldsymbol{x})$ 有任何影响；若 $\alpha_i > 0$，则必有 $y_i f(\boldsymbol{x}_i) = 1 - \xi_i$，即该样本是支持向量：由式(6.39) 可知，若 $\alpha_i < C$，则 $\mu_i > 0$，进而有 $\xi_i = 0$，即该样本恰在最大间隔边界上；若 $\alpha_i = C$，则有 $\mu_i = 0$，此时若 $\xi_i \leqslant 1$ 则该样本落在最大间隔内部，若 $\xi_i > 1$ 则该样本被错误分类. 由此可看出，软间隔支持向量机的最终模型仅与支持向量有关，即通过采用 hinge 损失函数仍保持了稀疏性.

正则化

> 一种替代损失函数
>
> 罚函数思想

- 经验风险小需要模型拟合好数据：比如准确率比较高，或对数几率优化的比较好，回归问题中平方比较小
- 结构风险和模型本身有关：比如w复杂度，w小模型简单，用w范数表示，比如决策树深度



> C：正则化常数
>
> Lp范数：常用正则化项

- L2使w分量取值均衡，即非零分量个数尽量稠密
- L0和L1使w分量尽量系数，即非零分量个数尽量少

五、支持向量回归SVR

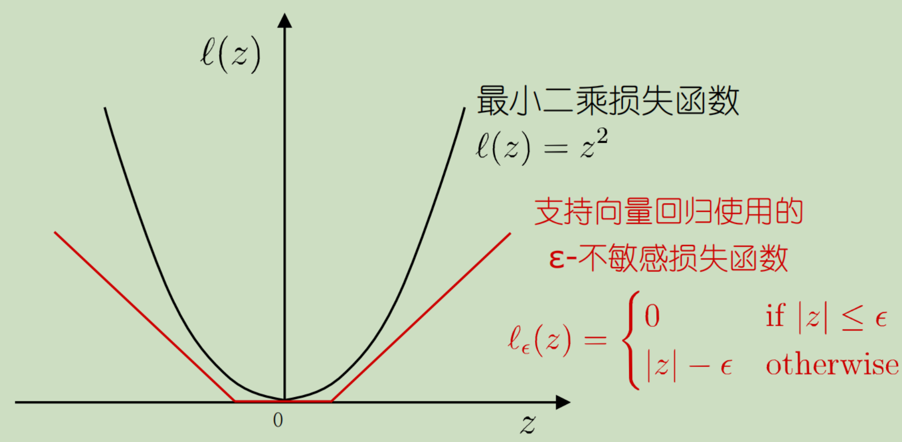
> 当f（x）与y差别绝对值大于ε时才计算损失

1、SVR原问题

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\ell_\epsilon(f(\boldsymbol{x}_i) - y_i)$$

ε-不敏感损失函数

落入 2ε 间隔带的样本不计算损失



引入松弛变量

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

$$\text{s.t. } f(\boldsymbol{x}_i) - y_i \leqslant \epsilon + \xi_i\ ,$$

$$y_i - f(\boldsymbol{x}_i) \leqslant \epsilon + \hat{\xi}_i\ ,$$

$$\xi_i \geqslant 0,\ \hat{\xi}_i \geqslant 0\ ,\ i = 1, 2, \ldots, m$$

2、SVR对偶问题

$$\max_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \quad \sum_{i=1}^{m} y_i(\hat{\alpha}_i - \alpha_i) - \epsilon(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$$

$$\text{s.t.} \quad \sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i) = 0\ , \quad 0 \leqslant \alpha_i, \hat{\alpha}_i \leqslant C$$

KKT性质

$$\begin{cases} \alpha_i(f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i) = 0\ , \\ \hat{\alpha}_i(y_i - f(\boldsymbol{x}_i) - \epsilon - \hat{\xi}_i) = 0\ , \\ \alpha_i\hat{\alpha}_i = 0\ ,\ \xi_i\hat{\xi}_i = 0\ , \\ (C - \alpha_i)\xi_i = 0\ ,\ (C - \hat{\alpha}_i)\hat{\xi}_i = 0\ . \end{cases} \tag{6.52}$$

可以看出，当且仅当 $f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i = 0$ 时 α_i 能取非零值，当且仅当 $y_i - f(\boldsymbol{x}_i) - \epsilon - \hat{\xi}_i = 0$ 时 $\hat{\alpha}_i$ 能取非零值. 换言之，仅当样本 $(\boldsymbol{x}_i, y_i)$ 不落入 ϵ-间隔带中，相应的 α_i 和 $\hat{\alpha}_i$ 才能取非零值. 此外，约束 $f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i = 0$ 和 $y_i - f(\boldsymbol{x}_i) - \epsilon - \hat{\xi}_i = 0$ 不能同时成立，因此 α_i 和 $\hat{\alpha}_i$ 中至少有一个为零.

解

> 带入wx+b

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

- 使系数α`i-αi不为0的样本是SVR的支持向量，落在ε-间隔带之外

由 KKT 条件(6.52)可看出，对每个样本 $(\boldsymbol{x}_i, y_i)$ 都有 $(C - \alpha_i)\xi_i = 0$ 且 $\alpha_i(f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i) = 0$. 于是，在得到 α_i 后，若 $0 < \alpha_i < C$，则必有 $\xi_i = 0$，进而有

$$b = y_i + \epsilon - \sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i)\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}\ . \tag{6.54}$$

- 考虑映射

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

其中 $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$ 为核函数.

六、核方法

> 基于核函数的学习方法

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

无论SVM还是SVR，学得模型总能表示成核函数的线性组合

> 算新样本和训练样本的核函数，然后线性组合，实现预测

1、表示定理

> 统一以上，解总是可以写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 $h^*(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i)$

- 只需Ω单增，不必凸
- 说明了核函数的强大

只要是优化经验风险和结构风险之和的问题，解（预测结果）都可以写成核函数形式

2、核线性判别分析KLDA

- 将样本映射到高维特征空间, 在此特征空间做线性判别分析
- “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

可以将Sb和Sw改写

工具包：

libSVM：不够快和高效

liblinear：无函数，解线性SVM，随机优化，快速