第七章：支持向量机

Support Vector Machine, SVM

- 线性支持向量机
- 核支持向量机
- 序列最小优化算法

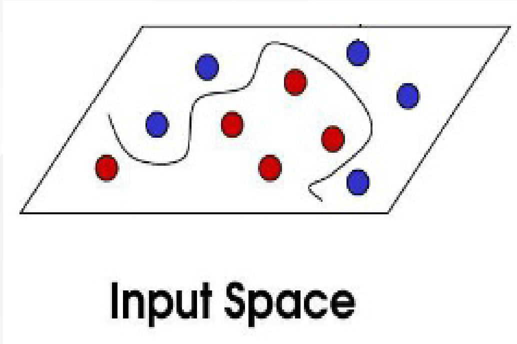
非线性SVM

■ 软间隔SVM原问题：

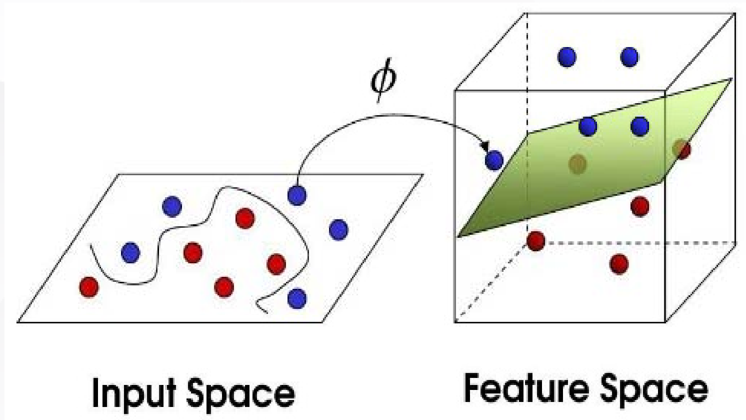
$$\min_{w,b}\frac{1}{2}\|\boldsymbol{w}\|_2^2+C\sum_{i=1}^{N}\xi_i$$

$$s.t.\ \ y^i\left(\boldsymbol{w}^T\boldsymbol{x}^i+b\right)\geq 1-\xi_i,\ \ i=1,\ldots,N,$$

$$\xi_i\geq 0,\ \ \ i=1,\ldots,N.$$



■ 非线性函数复杂！



■ 低维 ⇨ 高维。

非线性 ⇨ 线性

非线性SVM-核方法

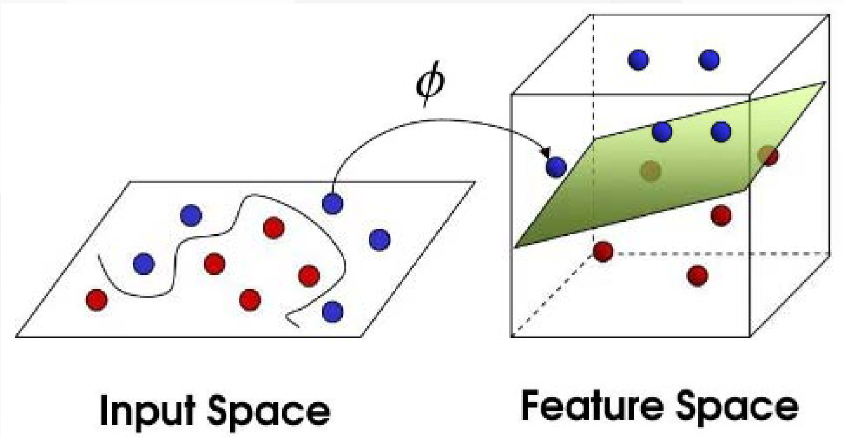
■非线性变换: $\boldsymbol{z} = \phi(\boldsymbol{x})$

■SVM最优化问题:

$$\min_{w,b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i$$

$$s.t.\ \ y^i\left(\boldsymbol{w}^T\boldsymbol{z}^i + b\right) \geq 1-\xi_i,\ \ i=1,\dots,N,$$

$$\xi_i \geq 0,\quad i=1,\dots,N.$$



SVM对偶问题

■对偶问题:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^i y^j \alpha_i \alpha_j \left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j$$

$$s.t.\ 0 \leq \alpha_i \leq C, i = 1,.., N,$$

$$\sum_{i=1}^{N} \alpha_i y^i = 0.$$

■最优解

$$\boldsymbol{w}^* = \sum_{i=1}^{N} \alpha_i^* y^i \boldsymbol{x}^i,\ b^* = y^j - \sum_{i=1}^{N} \alpha_i^* y^i \boxed{\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j}.$$

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \left(\boldsymbol{w}^*\right)^T \boldsymbol{x} + b^* = \sum_{i=1}^{N} \alpha_i^* y^i \boxed{\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}} + b^*$$

SVM对偶问题

■对偶问题:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^i y^j \alpha_i \alpha_j \left(\phi(\boldsymbol{x}^i)\right)^T \phi(\boldsymbol{x}^j)$$

$$s.t.\ 0 \le \alpha_i \le C, i = 1,..,N,$$

$$\sum_{i=1}^{N} \alpha_i y^i = 0.$$

■最优解

$$\boldsymbol{w}^* = \sum_{i=1}^{N} \alpha_i^* y^i \phi(\boldsymbol{x}^i),\ b^* = y^j - \sum_{i=1}^{N} \alpha_i^* y^i \boxed{\left(\phi(\boldsymbol{x}^i)\right)^T \phi(\boldsymbol{x}^j)}.$$

有什么问题?

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \left(\boldsymbol{w}^*\right)^T \phi(\boldsymbol{x}) + b^* = \sum_{i=1}^{N} \alpha_i^* y^i \boxed{\left(\phi(\boldsymbol{x}^i)\right)^T \phi(\boldsymbol{x})} + b^*$$

SVM对偶问题

- 如果映射函数选取得当，使得存在一个函数 $K()$，对于任意的x^i, x^j,都有 $K(\boldsymbol{x}^i,\boldsymbol{x}^j)=\left(\phi(\boldsymbol{x}^i)\right)^T\phi(\boldsymbol{x}^j)$ ，则函数$K()$称为**核函数**。
- 核函数$K()$与映射函数$\phi()$ 的对应关系?
- 例子：$\boldsymbol{X}=R^2,\phi:R^2\to H,K(\boldsymbol{x},\boldsymbol{z})=<\boldsymbol{x},\boldsymbol{z}>^2$ $\boldsymbol{x}=(x_1,x_2)$ $\boldsymbol{z}=(z_1,z_2)$

$$<\boldsymbol{x},\boldsymbol{z}>^2=(x_1z_1+x_2z_2)^2=(x_1z_1)^2+2x_1z_1x_2z_2+(x_2z_2)^2$$

① $\boldsymbol{H}=R^3,\phi(\boldsymbol{x})=(x_1^2,\sqrt{2}x_1x_2,x_2^2)^T$

② $\boldsymbol{H}=R^3,\phi(\boldsymbol{x})=\frac{1}{\sqrt{2}}((x_1-x_2)^2,2x_1x_2,(x_1+x_2)^2)^T$

③ $\boldsymbol{H}=R^4,\phi(\boldsymbol{x})=(x_1^2,x_1x_2,x_1x_2,x_2^2)^T$

核技巧（Kernel Trick）与支持向量机

- 核技巧: 学习与预测时只需使用 $K(\boldsymbol{x},\boldsymbol{z})$ ϕ

- SVM对偶问题:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} y^i y^j \alpha_i \alpha_j K\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right)$$

$$s.t. \sum_{i=1}^{N} \alpha_i y^i = 0,$$

$$0 \le \alpha_i \le C, i = 1,..,N.$$

- 分离超平面及最大间隔分类器

$$\boldsymbol{w}^* = \sum_{i=1}^{N}\alpha_i^* y^i \phi(\boldsymbol{x}^i),\ b^* = y^j - \sum_{i=1}^{N}\alpha_i^* y^i K\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right).$$

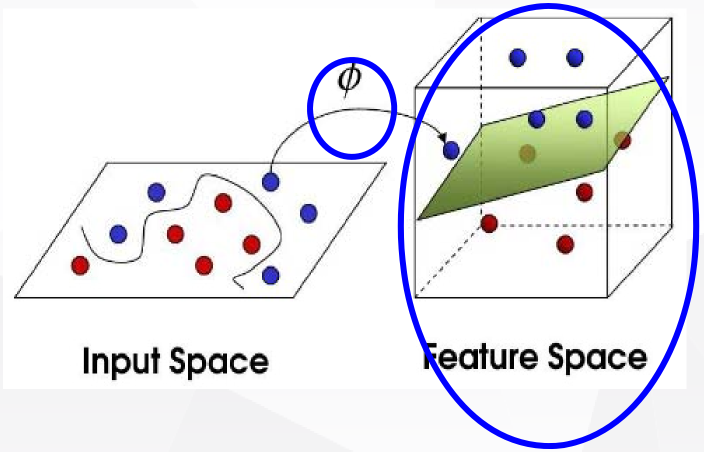
$$f_{\boldsymbol{w}^*,b^*} = (\boldsymbol{w}^*)^T \phi(\boldsymbol{x}) + b^* = \sum_{i=1}^{N}\alpha_i^* y^i K(\boldsymbol{x}^i, \boldsymbol{x}) + b^*.$$

核 SVM

- 相当于在映射后的特征空间学习线性SVM。

- 给定核函数 $K(x, z)$, 可以使用线性SVM方法找到非线性可分数据集的判别函数。

- 学习过程是在映射后的特征空间进行的。



- 核函数优势：

  线性方法 解决 **非线性问题**

核函数

定理（核函数）

令 χ 为输入空间，$K(\cdot,\cdot)$ 是定义在 $\chi\times\chi$ 上的对称的函数，则 $K(\cdot,\cdot)$ 是核函数**当且仅当**对于任意数据 $D=\{\boldsymbol{x}^1,\boldsymbol{x}^2,\ldots,\boldsymbol{x}^N\}$，核矩阵 $\mathbb{K}$ 总是半正定的。

$$\mathbb{K}=\begin{bmatrix} K(\boldsymbol{x}^1,\boldsymbol{x}^1) & \cdots & K(\boldsymbol{x}^1,\boldsymbol{x}^j) & \cdots & K(\boldsymbol{x}^1,\boldsymbol{x}^N) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\boldsymbol{x}^i,\boldsymbol{x}^1) & \cdots & K(\boldsymbol{x}^i,\boldsymbol{x}^j) & \cdots & K(\boldsymbol{x}^i,\boldsymbol{x}^N) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\boldsymbol{x}^N,\boldsymbol{x}^1) & \cdots & K(\boldsymbol{x}^N,\boldsymbol{x}^j) & \cdots & K(\boldsymbol{x}^N,\boldsymbol{x}^N) \end{bmatrix}.$$

格拉姆（Gram）矩阵

常用核函数

多项式核:

$$K(\boldsymbol{x},\boldsymbol{z})=(\boldsymbol{x}^T\boldsymbol{z}+1)^p$$

- $p=2$， $\boldsymbol{x}=(x_1,x_2)$

$$K(\boldsymbol{x},\boldsymbol{z})=(x_1z_1+x_2z_2+1)^2=1+2x_1z_1+2x_2z_2+2x_1z_1x_2z_2+x_1^2z_1^2+x_2^2z_2^2$$

- 映射函数: $\phi(\boldsymbol{x})=(1,\sqrt{2}x_1,\sqrt{2}x_2,\sqrt{2}x_1x_2,x_1^2,x_2^2)^T$
- 多项式分类器 $f(\boldsymbol{x})=\mathrm{sgn}(\sum_{i=1}^{n}\alpha_i^*y^i((\boldsymbol{x}^i)^T\boldsymbol{x}+1)^p+b^*).$

常用核函数

■高斯核

$$K(\boldsymbol{x},\boldsymbol{z})=\exp\{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}\}.$$

■径向基函数 (Radial Basis Function,RBF)

$$K(\boldsymbol{x},\boldsymbol{z})=\exp\{-\frac{Dist(\boldsymbol{x},\boldsymbol{z})}{2\sigma^2}\}.\qquad K(\boldsymbol{x},\boldsymbol{z})=\exp\{-\gamma\times Dist(\boldsymbol{x},\boldsymbol{z})\}.$$

■分类器是高斯径向基函数

$$f(\boldsymbol{x})=\sum_{i=1}^{n}\alpha_i^* y^i(\exp\{-\frac{\|\boldsymbol{x}^i-\boldsymbol{x}\|^2}{2\sigma^2}\}+b^*$$

势函数 $K_{k+1}(x)=\sum_{\hat{x}_j} a_j K(x,\hat{x}^j)$

其中

$$a_j=\begin{cases}+1 & for\ \hat{x}^j\in\omega_1\\ -1 & for\ \hat{x}^j\in\omega_2\end{cases}$$

常用核函数

- 拉普拉斯核

$$K(\boldsymbol{x},\boldsymbol{z})=\exp\{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma}\}.$$

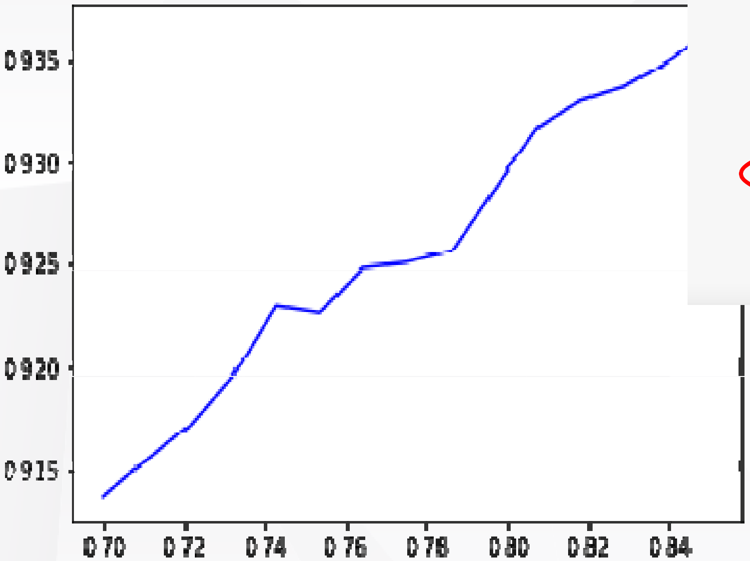
- Sigmoid核

$$K(\boldsymbol{x},\boldsymbol{z})=\tanh(\beta\boldsymbol{x}^T\boldsymbol{z}+\theta)$$

- 核函数的线性组合仍是核函数。
- $g(\boldsymbol{x})K(\boldsymbol{x},\boldsymbol{z})\ g(\boldsymbol{z})$仍是核函数，$g(.)$是任意函数。
- **高斯核在实际运用中特别多。**
  超参少，**有限维→无限维**

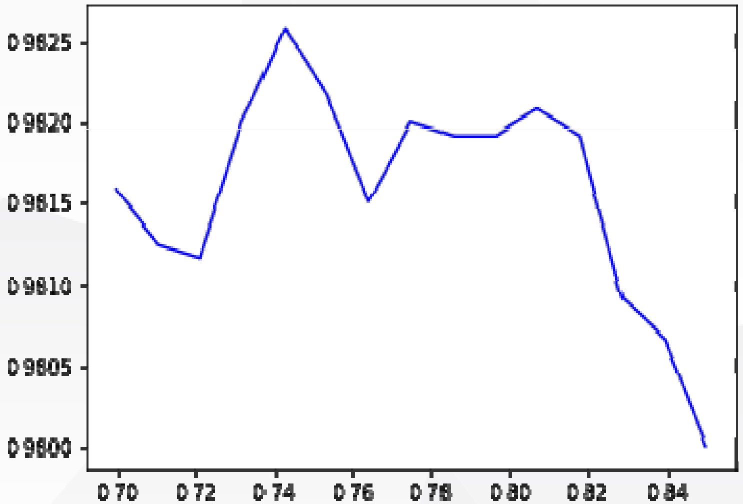
Kernel SVM+KL

```python
X_val_pca = pca.transform(X_val)

# 利用svc训练
print('SVC begin')
#kernel : 核函数，默认是rbf，可以是'linear', 'poly', 'rbf', 'sigmoi
clf1 = svm.SVC(kernel='sigmoid')
clf1.fit(X_train_pca, y_train)

# 返回accuracy
```



保留的信息量
降维后，特征维度越高，
保留的信息量越多

- 10类 准确率：0.98209820



- 10类 准确率：(RBF)0.98209820
98

序列最小优化算法 Sequential Minimal Optimization, SMO

- SVM的原问题可能通过传统的凸二次规划方法来获得**全局最优解。**
- 算法慢，尤其是训练数据集很大时。
- SMO(John Platt ,1998) ：高效求解SVM对偶问题。
- 对偶问题：

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^i y^j \alpha_i \alpha_j \left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j$$

$$s.t. \sum_{i=1}^{N} \alpha_i y^i = 0,$$

$$0 \le \alpha_i \le C, i = 1, .., N.$$

SMO动机:坐标梯度上升

■无约束最优化问题:

$$\max_{\boldsymbol{\alpha}} W(\alpha_1, \ldots, \alpha_N)$$

■坐标上升优化算法（Coordinate Ascent Optimization Algorithm）：

$$
\begin{aligned}
&\text{Loop until convergence : } \{ \\
&\quad \text{For } i = 1, \cdots, N \ : \{ \\
&\quad\quad \boxed{\alpha_i = \max_{\hat{\alpha}_i} W(\alpha_1, \cdots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \cdots, \ \alpha_N).} \\
&\quad \} \\
&\}
\end{aligned}
$$

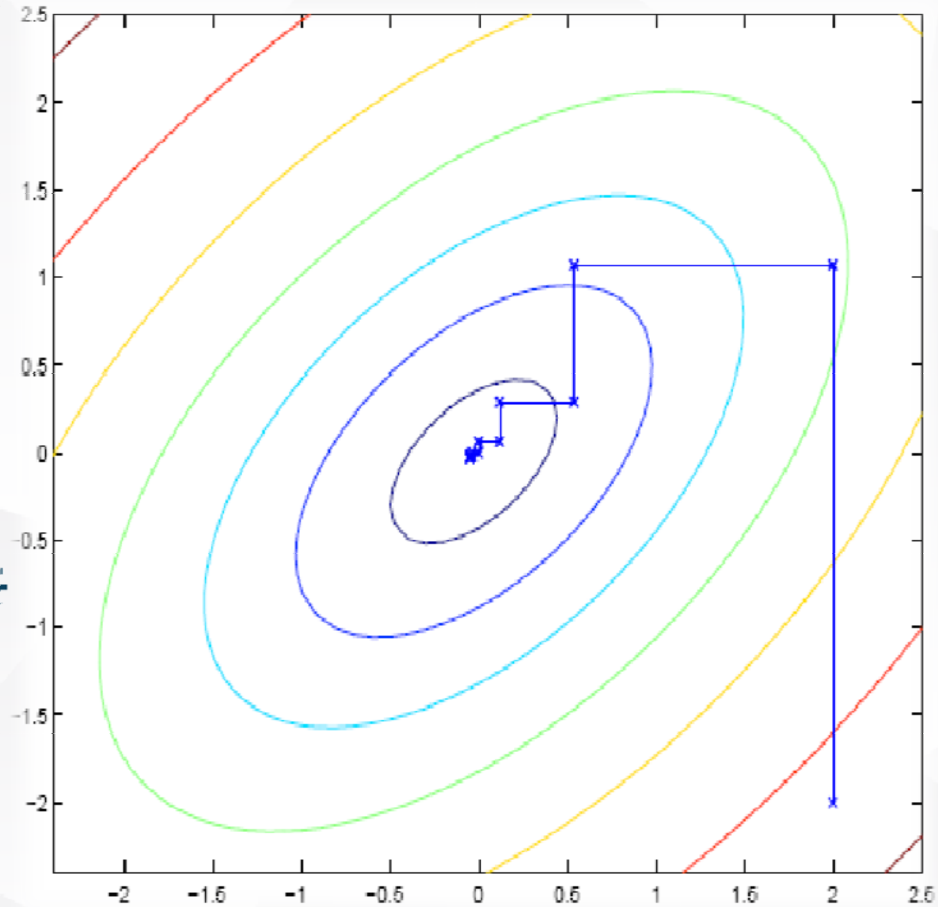
■每次只关于一个参数 α_i 优化目标函数。

坐标梯度上升/下降

■椭圆：目标函数的等高线

■初始点：(2,-2)

■坐标梯度上升法：每一步沿着坐标轴方向移动。



SVM坐标梯度下降求解

■对偶问题:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^i y^j \alpha_i \alpha_j \left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j$$

$$\boxed{s.t. \sum_{i=1}^{N} \alpha_i y^i = 0,}$$

$$0 \le \alpha_i \le C, i = 1, .., N.$$

■每次固定N-1个变量 $\alpha_2, \ldots, \alpha_N$ ，关于 变量 α_1 优化目标函数?

SMO: 更新一对变量

- 为了保证满足约束条件，每次更新**两个**变量。 ⟹ SMO
- Repeat until convergence : {

(1) 选择要更新的一对变量 α_i 和 α_j

（启发式选择：选择使目标函数值 改变最大的变量）

(2) 关于变量 α_i 和 α_j 优化目标函数 $W(\alpha)$

- SMO高效：更新 α_i, α_j 的方法计算高效!

SMO: 关于两个变量的优化方法

- 假定 $\alpha_i s$ 满足约束条件，固定 $\alpha_3,\ldots,\alpha_N$ ，关于变量 α_1 和 α_2 优化目标函数：

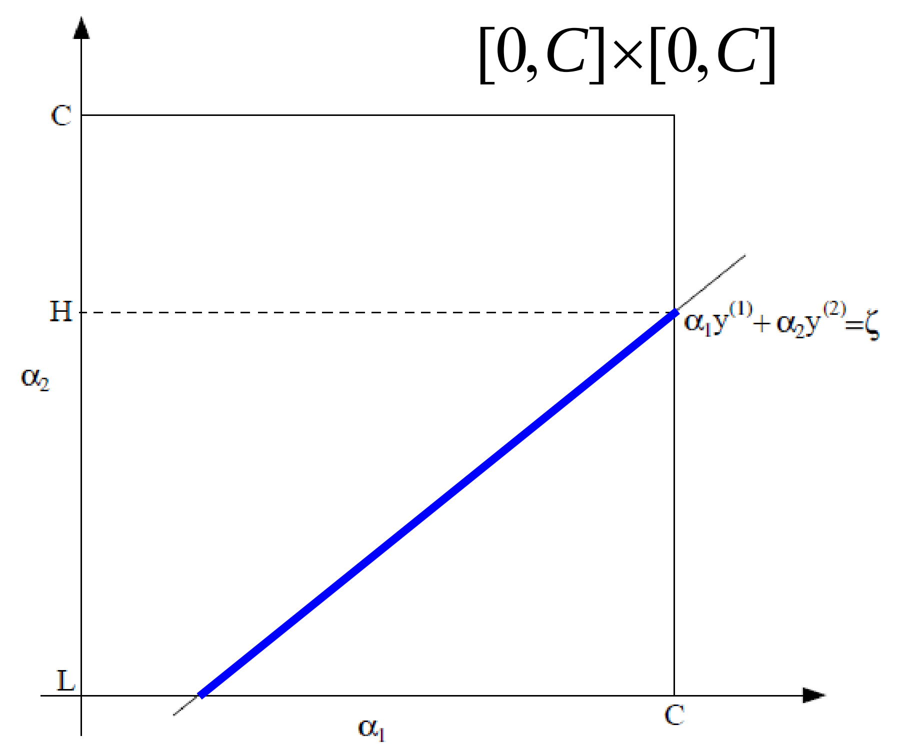
$$W(\alpha_1,\ldots,\alpha_N)$$

- 约束条件：$\alpha_1 y^1 + \alpha_2 y^2 = \boxed{-\sum_{i=3}^{N} \alpha_i y^i}$

常数 ζ

有界的

$L \le \alpha_2 \le H$.



SMO: 关于一个变量优化

■将 α_1 写成关于 α_2 的等式: $\alpha_1=(\zeta-\alpha_2 y^2)y^1$

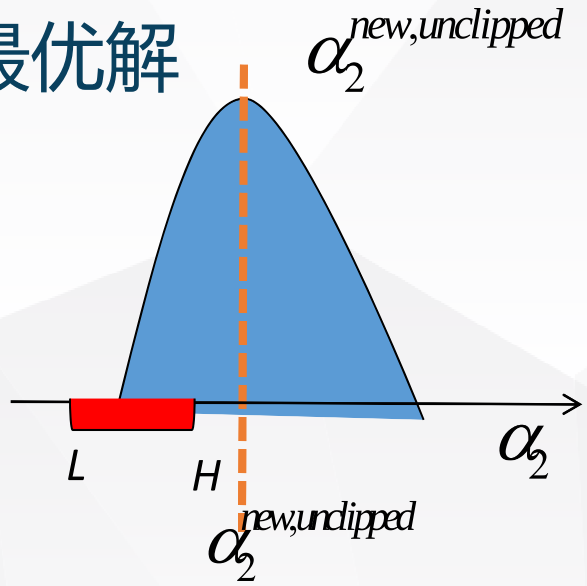
■目标函数 $W(\alpha)$ ： $W(\alpha_1,\ldots,\alpha_N)=W((\zeta-\alpha_2 y^2)y^1,\alpha_2,\ldots,\alpha_N)$

■关于变量 α_2 的二次函数。

■不考虑 α_2 的取值范围，可求全局最优解

■考虑约束条件： $L\le\alpha_2\le H$

$$\alpha_2^{new}=\begin{cases} H & \text{if } \alpha_2^{new,unclipped}>H \\ \alpha_2^{new,unclipped} & \text{if } L\le\alpha_2^{new,unclipped}\le H \\ L & \text{if } \alpha_2^{new,unclipped}<L \end{cases}$$



SMO: 两个变量的最优解

- 得到 α_2^{new}, 根据 $\alpha_1=(\zeta-\alpha_2 y^2)y^1$ 可得:

$$\Downarrow$$

$$\zeta=\alpha_1^{old}y^1+\alpha_2^{old}y^2=\alpha_1^{new}y^1+\alpha_2^{new}y^2$$

$$\Downarrow$$

$$\alpha_1^{new}=\alpha_1^{old}+y^1y^2(\alpha_2^{old}-\alpha_2^{new})$$

$$\alpha_2^{new}=\begin{cases} H & \text{if } \alpha_2^{new,unclipped}>H \\ \alpha_2^{new,unclipped} & \text{if } \mathrm{L}\le\alpha_2^{new,unclipped}\le H \\ L & \text{if } \alpha_2^{new,unclipped}<L \end{cases}$$

SMO: 第一个变量的选择

- 主要思想: 从训练样本中选择**违背KKT 条件**的样本，进而确定 α_i

- KKT 条件: $\boxed{\begin{aligned} &\alpha_i = 0 \Leftrightarrow y^i g(\boldsymbol{x}^i) \geq 1 \\ &0 < \alpha_i < C \Leftrightarrow y^i g(\boldsymbol{x}^i) = 1 \\ &\alpha_i = C \Leftrightarrow y^i g(\boldsymbol{x}^i) \leq 1 \end{aligned}}$

$$g(\boldsymbol{x}^i) = \sum_{j=1}^{N} \alpha_j y^j K(\boldsymbol{x}^i, \boldsymbol{x}^j) + b$$

- 检查在决策边界上的支持向量： $0 < \alpha_i < C$

- 检查所有训练样本。

SMO: 第二个变量的选择

- 选 α_2 的标准?

定理

$$\alpha_2^{new,unclipped} = \alpha_2^{old} + \frac{y^2(E_1 - E_2)}{\eta}$$

where $\eta = K_{11} + K_{22} - 2K_{12} = \left\| \phi(\boldsymbol{x}^1) - \phi(\boldsymbol{x}^2) \right\|^2, E_i = g(\boldsymbol{x}^i) - y^i.$

- 选择使 $|E_1 - E_2|$ 最大的 α_2
- α_1 固定，E_1 固定。

$$E_1 > 0 \Rightarrow 选最小的E_i作为E_2$$

$$E_1 < 0 \Rightarrow 选最大的E_i作为E_2$$

- E_i很关键，怎么样使得算法高效?

SMO：*b* 及对偶值 E_i

- 更新完变量后，更新 *b*.
- $0<\alpha_1^{new}<C$ ，KKT条件：$\sum_{i=1}^{N}\alpha_i y^i K_{i1}+b=y^1$

$$\Downarrow$$

$$b_1^{new}=y^1-\sum_{i=3}^{N}\alpha_i y^i K_{i1}-\alpha_1^{new}y^1K_{11}-\alpha_2^{new}y^2K_{21}$$

- 引入：$E_1=\sum_{i=3}^{N}\alpha_i y^i K_{i1}+\alpha_1^{old}y^1K_{11}+\alpha_2^{old}y^2K_{21}+b^{old}-y^1.$

$$\left\{\begin{aligned}b_1^{new}&=-E_1-y^1K_{11}(\alpha_1^{new}-\alpha_1^{old})-y^2K_{21}(\alpha_2^{new}-\alpha_2^{old})+b^{old}.\\ b_2^{new}&=-E_2-y^1K_{12}(\alpha_1^{new}-\alpha_1^{old})-y^2K_{22}(\alpha_2^{new}-\alpha_2^{old})+b^{old}.\end{aligned}\right.$$

- 如果 $0<\alpha_i^{new}<C,\ i=1,2\quad b_1^{new}$? b_2^{new}.
- 如果 $\alpha_1^{new},\alpha_2^{new}$ 是0或*C*，$\forall b\in[\min\{b_1^{new},b_2^{new}\},\max\{b_1^{new},b_2^{new}\}]$ 都满足KKT条件，取 $b^{new}=\dfrac{b_1^{new}+b_2^{new}}{2}$
- 更新 E_i $\quad E_i^{new}=\sum_{S}y^j\alpha_jK(\boldsymbol{x}^i,\boldsymbol{x}^j)+b^{new}-y^i$

支持向量集合

SMO 算法

■输入: 训练数据集 $S=\{(x^i, y^i), i=1,\dots,N\}$ ，误差 ε;

■输出: $\hat{\alpha}=(\hat{\alpha}_1,\dots,\hat{\alpha}_N)$

① 初始化: $\alpha^{(0)}=0, k=0,$ 并计算偏移量$b^{(0)}$

② 初始化误差项 $E_i=g(\boldsymbol{x}^i)-y^i$.

③ 选择待优化的变量: $\alpha_1^{(k)}$、$\alpha_2^{(k)}$，求解优化问题的解 $\alpha_1^{(k+1)}$、$\alpha_2^{(k+1)}$，

$$\alpha_2^{new,unclipped}=\alpha_2^{(k)}+\frac{y^2(E_1-E_2)}{\eta},$$

$$\alpha_2^{(k+1)}=\begin{cases} H & \text{if } \alpha_2^{new,unclipped}>H \\ \alpha_2^{new,unclipped} & \text{if } L\le\alpha_2^{new,unclipped}\le H \\ L & \text{if } \alpha_2^{new,unclipped}<L \end{cases}$$

$$\eta=K_{11}+K_{22}-2K_{12}$$

$$\alpha_1^{(k+1)}=\alpha_1^{(k)}+y^1y^2(\alpha_2^{(k)}-\alpha_2^{(k+1)})$$

④ 更新α 为 $\alpha^{(k+1)}$，更新 $E_i=g(\boldsymbol{x}^i)-y^i$ ，计算$b^{(k+1)}$

⑤ 如果达到**终止条件**，则停止算法；否则k=k+1,转到第③步

#满足KKT条件或误差项均小于 ε

SMO小结

- **启发式、迭代式**算法。
- 解满足KKT条件时，得到最优解；否则选择两个变量来优化。
- 最优化问题转换为关于两个选定变量的QP问题，该问题通常有闭式解，**计算高效，收敛快**。
- 变量的选择方法:
  - 第一个变量：违背KKT条件程度最大的变量
  - 第二个变量：使目标函数值减小最快的变量
- 把一个 最优化问题不换转化为若干个子问题，通过对子问题的求解得到原问题的解。

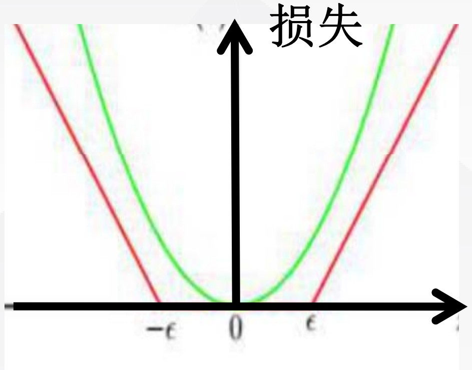
支持向量回归(Support Vector Regression, SVR)

■ SVM：

$$\min_{w,b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i$$

$$s.t.\ \ y^i\left(\boldsymbol{w}^T\boldsymbol{x}^i + b\right) \geq 1-\xi_i,\ i=1,\dots,N,$$

$$\xi_i \geq 0,\quad i=1,\dots,N.$$

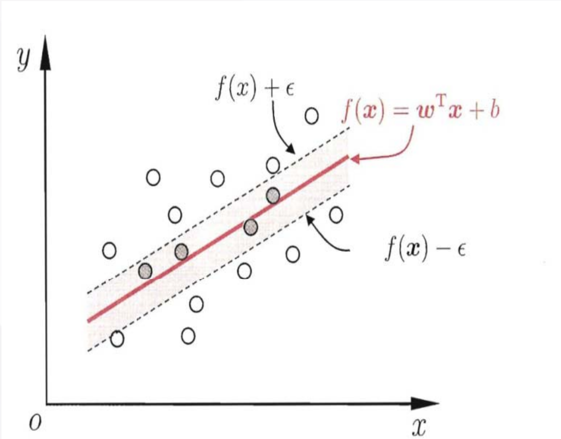


y-$f(\boldsymbol{x})$

■ 回归问题：均方误差损失

■ SVC：

$$\varepsilon_i(y^i, f(\boldsymbol{x}^i)) = \begin{cases} 0 & \text{if } |y^i - f(\boldsymbol{x}^i)| \leq \varepsilon \\ |y^i - f(\boldsymbol{x}^i)| - \varepsilon & \text{otherwise} \end{cases}$$



ε 不敏感损失

■ 性质

•误差在 ε 内，可以接受。

•误差大于ε 时，对于损失的影响是线性的（不是二次的），对噪声更鲁棒。

支持向量回归: 原问题

■SVR原问题:

$$\min_{w,b} \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$s.t.\ \ \left|y^i-(\boldsymbol{w}^T\boldsymbol{x}^i+b)\right|\le\varepsilon, i=1,\dots,N,$$

■松弛变量:

$$-\varepsilon-\xi_i^-\le y^i-(\boldsymbol{w}^T\boldsymbol{x}^i+b)\le\varepsilon+\xi_i^+, i=1,\dots,N,$$

正偏移 $\xi_i^+\ge0$ $\xi_i^-\ge0$ 负偏移

■带松弛变量的SVR原问题:

$$\min_{w,b} \frac{1}{2}\|\boldsymbol{w}\|_2^2+C\sum_{i=1}^{N}(\xi_i^+ +\xi_i^-)$$

$$s.t.\ \ y^i-(\boldsymbol{w}^T\boldsymbol{x}^i+b)\le\varepsilon+\xi_i^+,\ i=1,\dots,N,$$

$$\boldsymbol{w}^T\boldsymbol{x}^i+b-y^i\le\varepsilon+\xi_i^-,\ i=1,\dots,N,$$

$$\xi_i^+,\xi_i^-\ge0,\ \ i=1,\dots,N.$$

SVR: 对偶及核

■拉格朗日函数：

$$L(\boldsymbol{w},b,\boldsymbol{\alpha}^{+},\boldsymbol{\alpha}^{-},\boldsymbol{\mu}^{+},\boldsymbol{\mu}^{-})=\frac{1}{2}\|\boldsymbol{w}\|_2^2+C\sum_{i=1}^{N}\left(\xi_i^{+}+\xi_i^{-}\right)$$
$$+\sum_{i=1}^{N}\alpha_i^{+}\left(y^i-\boldsymbol{w}^T\boldsymbol{x}^i-b-\varepsilon-\xi_i^{+}\right)+\sum_{i=1}^{N}\alpha_i^{-}\left(\boldsymbol{w}^T\boldsymbol{x}^i+b-y^i-\varepsilon-\xi_i^{-}\right)$$
$$-\sum_{i=1}^{N}\mu_i^{+}\xi_i^{+}-\sum_{i=1}^{N}\mu_i^{-}\xi_i^{-}$$

■求偏导数:

$$\frac{\partial L}{\partial \boldsymbol{w}}=0\Rightarrow \boldsymbol{w}=\sum_{i=1}^{N}(\alpha_i^{+}-\alpha_i^{-})\boldsymbol{x}^i=0$$

$$\frac{\partial L}{\partial b}=0\Rightarrow \sum_{i=1}^{N}(\alpha_i^{+}-\alpha_i^{-})=0$$

$$\frac{\partial L}{\partial \xi_i^{-}}=0\Rightarrow C-\alpha_i^{-}-\mu_i^{-}=0$$

$$\frac{\partial L}{\partial \xi_i^{+}}=0\Rightarrow C-\alpha_i^{+}-\mu_i^{+}=0$$

SVR: 对偶及核

■SVR对偶问题：$\max\limits_{\alpha^+,\alpha^-}\sum_{i=1}^{N}\left(y^i(\alpha_i^+-\alpha_i^-)-\varepsilon(\alpha_i^++\alpha_i^-)\right)-\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^+-\alpha_i^-)(\alpha_j^+-\alpha_j^-)\left(\boldsymbol{x}^i\right)^T\boldsymbol{x}^j$

$$\boxed{\begin{aligned}&\min_{\alpha^+,\alpha^-}\sum_{i=1}^{N}\left(y^i(\alpha_i^--\alpha_i^+)+\varepsilon(\alpha_i^++\alpha_i^-)\right)+\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^+-\alpha_i^-)(\alpha_j^+-\alpha_j^-)\left(\boldsymbol{x}^i\right)^T\boldsymbol{x}^j\\&\sum_{i=1}^{N}(\alpha_i^+-\alpha_i^-)=0\\&0\le\alpha_i^+\le C, i=1,2,\ldots,N\\&0\le\alpha_i^-\le C, i=1,2,\ldots,N\end{aligned}}$$

■KKT条件：

$$\alpha_i^+\alpha_i^-=0,\ \ \xi_i^+\xi_i^-=0$$

$$C-\alpha_i^+-\mu_i^+=0, C-\alpha_i^--\mu_i^-=0$$

$$\mu_i^+\xi_i^+=0, \mu_i^-\xi_i^-=0$$

$$-\varepsilon-\xi_i^-\le y^i-(\boldsymbol{w}^T\boldsymbol{x}^i+b)\le\varepsilon+\xi_i^+, i=1,\ldots,N,$$

$$\alpha_i^+\ge0,\alpha_i^-\ge0,\xi_i^+\ge0,\xi_i^-\ge0$$

$$\boxed{\begin{aligned}&\alpha_i^+(\varepsilon+\xi_i^+-y^i+\boldsymbol{w}^T\boldsymbol{x}^i+b)=0\\&\alpha_i^-(\varepsilon+\xi_i^++y^i-\boldsymbol{w}^T\boldsymbol{x}^i-b)=0\end{aligned}}$$

$\alpha_i^+>0$或$\alpha_i^+>0$ 时， 样本 $(\boldsymbol{x}^i,y^i)$ 为支持向量

SVR: 对偶及核

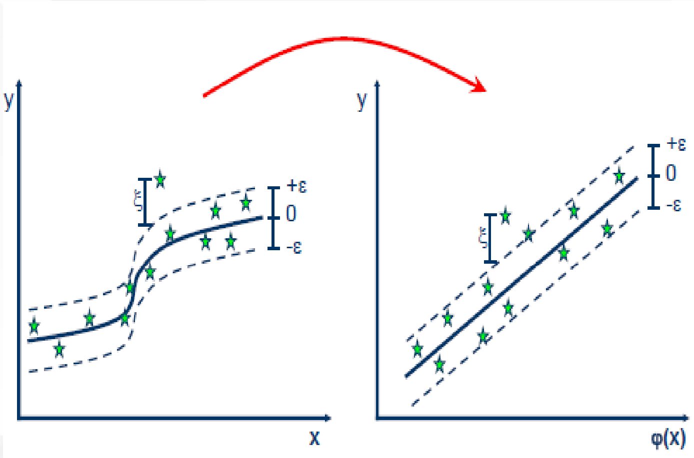
- SVR对偶问题:

$$\min_{\alpha^+,\alpha^-}\sum_{i=1}^{N}\left(y^i(\alpha_i^- - \alpha_i^+) + \varepsilon(\alpha_i^+ + \alpha_i^-)\right) + \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j$$

$$\sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0$$

$$0 < \alpha_i^+ < C, i = 1, 2, \ldots, N$$

$$0 < \alpha_i^- < C, i = 1, 2, \ldots, N$$



- 非线性回归—核技巧。
- 落在ε 隔离带边界及之外的样本，才是SVR的支持向量。
- SVR的支持向量仅是训练样本的一部分,即其解仍具有稀疏性.

多类 SVM:

- 多类问题转换：一对多，多类情况三.

- 训练数据： $S=\{(\boldsymbol{x}^i, y^i), i=1,\ldots,N\},\ \boldsymbol{x}^i\in R^D,\ y^i\in\{1,\ldots,m\}$

- 最优化问题:

$$
\begin{aligned}
&\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|_2^2+C\sum_{i=1}^{N}\xi_i\\
&s.t.\quad \forall j\neq y^1:\quad \boldsymbol{w}_{y^1}^T\boldsymbol{x}^1\geq\boldsymbol{w}_j^T\boldsymbol{x}^1+1-\xi_1,\\
&\qquad\ldots\\
&\qquad\forall j\neq y^N:\quad \boldsymbol{w}_{y^N}^T\boldsymbol{x}^N\geq\boldsymbol{w}_j^T\boldsymbol{x}^N+1-\xi_N,,\\
&\qquad\xi_i\geq 0,\quad i=1,\ldots,N.
\end{aligned}
$$

学习理论：泛化误差上界

■定理

假设空间为k个函数的集合，固定 δ 和N，在概率1−δ下，对于任假设空间的任意一个函数 $\hat{f}$, 都有

$$E(\hat{f}) \le \min_{f \in F} \hat{E}(f) + 2\sqrt{\frac{\log 2k/\delta}{2N}}$$

- 假定我们把假设空间 $\mathcal{F}$ 换为更大的假设空间，即 $\mathcal{F}' \supseteq \mathcal{F}$ ，那么第一项减小，偏差降低。
- 当k增加时，第二项增大。方差增加。
- **假设空间为无限情形**：**VC维**，增长函数，打散函数。

假设空间无限

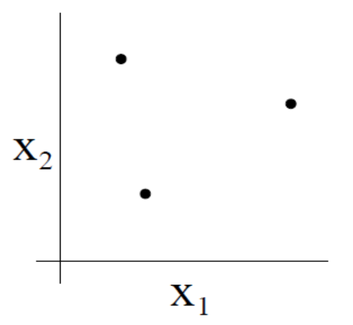
- 给定训练数据集 $S=\{\boldsymbol{x}^1,\ldots,\boldsymbol{x}^d\}$ ，如果对于**任何一个可能的label 集合**，都能够从假设空间*H*中找到**找到一个假设***h*，将训练数据正确地分开，那么就称***H*打散(shatter) *S*。**

- 对于给定的假设空间*H*，*H* ***能打散***的**最大的训练数据集中的样本**的数目称为*H*的VC维，记为VC(*H*)。它度量假设类*H*的学习能力。

- 如果*H*能打散包含无穷多样本的数据集，则VC(*H*)=∞

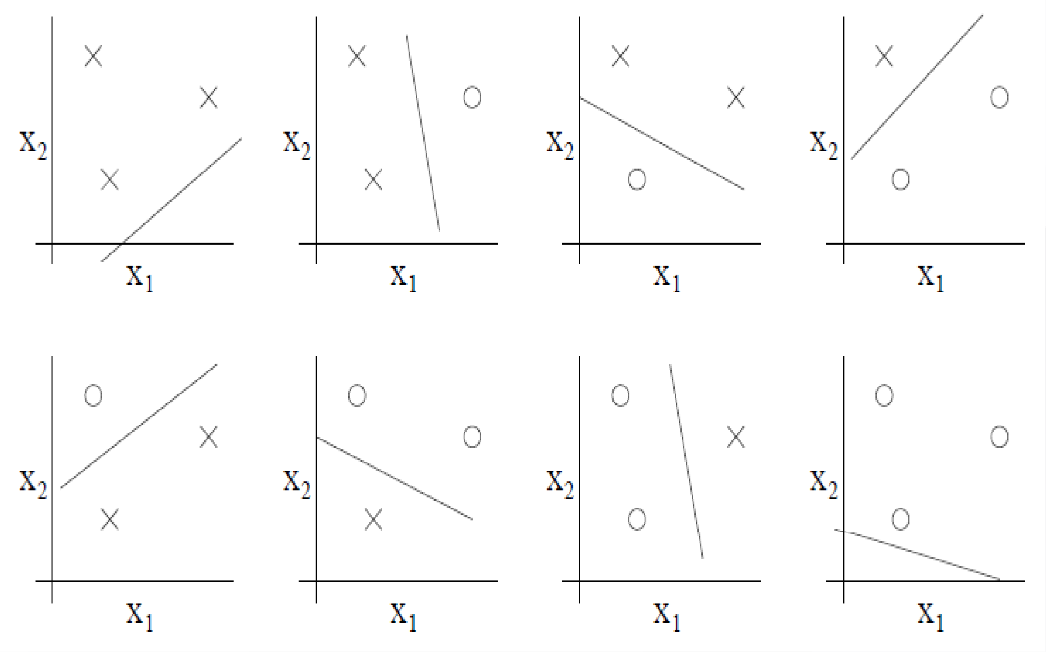
VC 维

- 例 子:

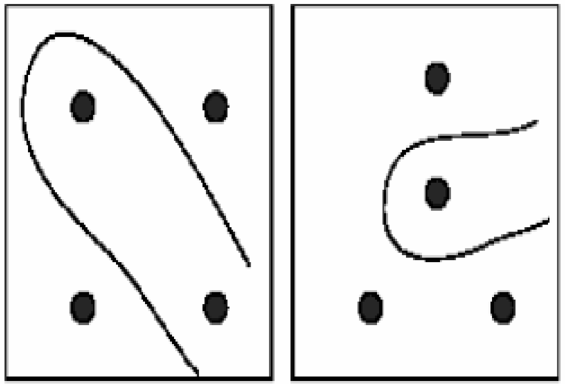


线性分类器$\boldsymbol{H}$ $h(x)=I_{\{\theta_0+\theta_1 x_1+\theta_2 x_2\geq 0\}}$
能否打散该数据 集？

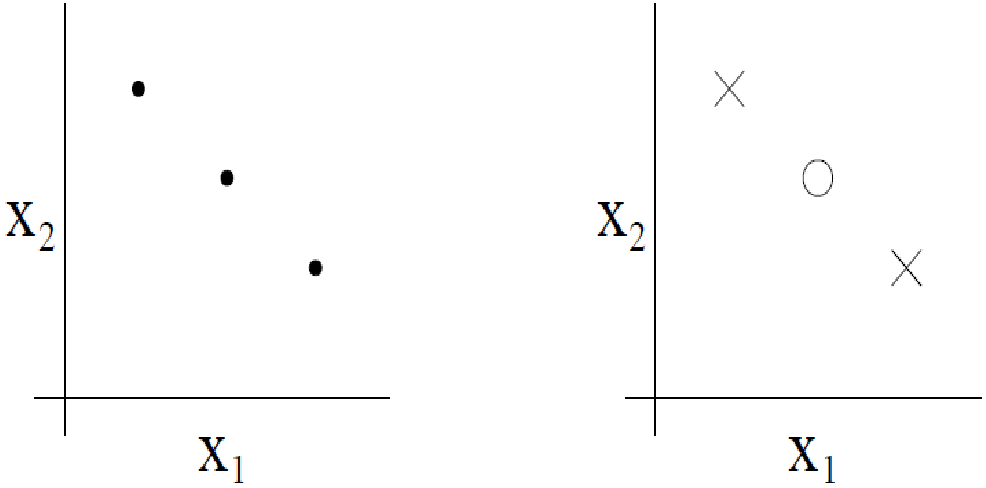
- 对于任意可能的label，都可以找到能将其正确分类的线性分类器。



VC 维



*H*能打散吗？



*H*能打散吗？

- VC(***H***) = 3。
- 若VC(***H***)=*d*,则至少存在一组可以被打散的*d*,个样本点，但是通常并不是每组*d*个样本点都可以被打散。
- 直观上，参数越多，VC维越大，反之亦然。

VC 维

定理

给定$\boldsymbol{H}$和 δ ，令 $d=VC(\boldsymbol{H})$ ，对于任意 $h\in\boldsymbol{H}$, 下式以至少 $1-\delta$ 的概率成立：

$$|\varepsilon(h)-\hat{\varepsilon}(h)|\leq O\left(\sqrt{\frac{d}{N}\log\frac{N}{d}+\frac{1}{N}\log\frac{1}{\delta}}\right)$$

记 $h^*=\arg\min_{h\in\boldsymbol{H}}(\varepsilon(h))$ $\hat{h}=\arg\min_{h\in\boldsymbol{H}}(\hat{\varepsilon}(h))$.

$$\varepsilon(\hat{h})\leq\varepsilon(h^*)+O\left(\sqrt{\frac{d}{N}\log\frac{N}{d}+\frac{1}{N}\log\frac{1}{\delta}}\right)$$

- 如果假设空间有有限的VC维，则当N变大时，一致性收敛的可能性更高。
- 可以利用$\hat{\varepsilon}(h)$ 给 $\varepsilon(h)$ 估计一个上界。

VC 维

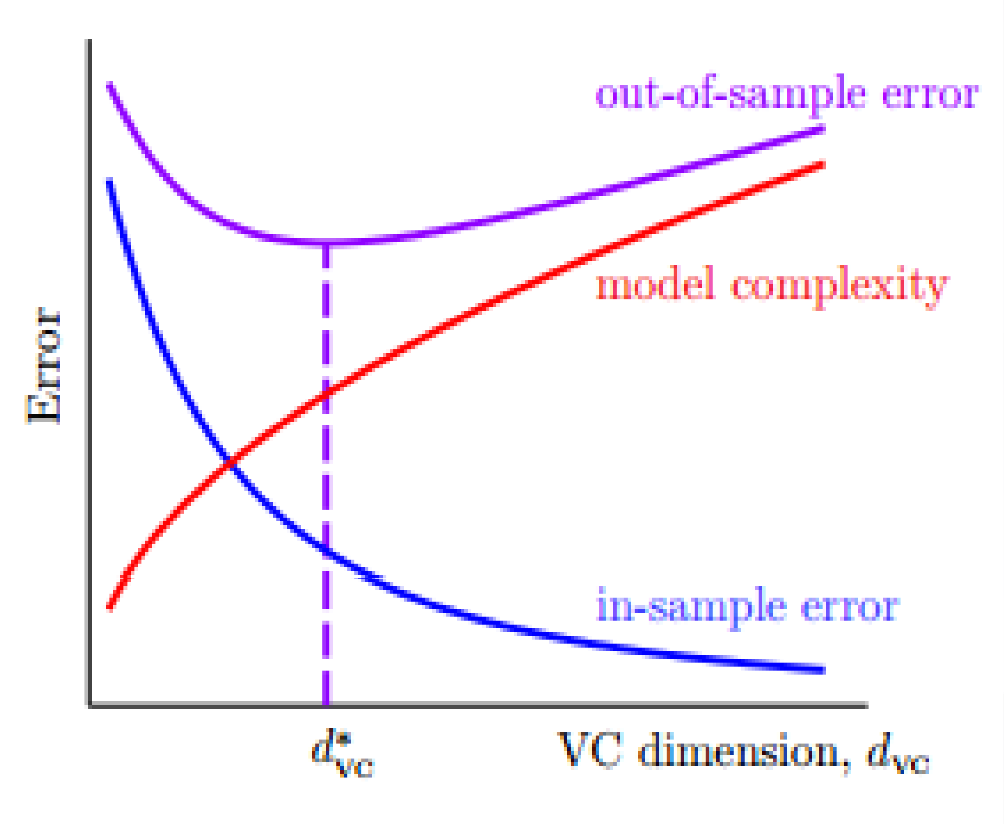
推论

如果使 $|\varepsilon(h)-\hat{\varepsilon}(h)|\leq\gamma$ 对于任意的 $h\in\boldsymbol{H}$ 都以 $1-\delta$ 的概率成立，这样 $\varepsilon(\hat{h})\leq\varepsilon(h^*)+2\gamma$ ，这就需要 $N=O_{\gamma,\delta}(d)$ 。

- 需要的训练 样本的数量约是 $O(VC(H))$。
- 在多数假设空间中，$VC(H)\sim O$ (参数的数目）。
- 需要的训练 样本的数量 $N\sim O$ (参数的数目）。

VC维：小结

- VC：度量模型复杂度。



- N大时, 能够支持复杂的模型(VC 维大).

SVM泛化分析

- 间隔与SVM分类器复杂 度的联系（参考Vapnik (1982)）

Theorem: Vapnik 1982

The class of optimal linear separators has VC dimension h bounded from above as:

$$h \le \min\{\left\lceil \frac{4r^2}{\rho^2} \right\rceil, m\} + 1,$$

where ρ is the margin, r is the radius of the smallest sphere that can enclose all of the training examples, and m is the dimensionality of $\boldsymbol{X}$.

- 不考虑 m ， 最小化VC维 h 等价于最大化间隔 。

- 使分类器的复杂度小！

SVR应用-共享单车骑行量预测

■ Scikit-learn：SVR函数

■ 731天共享单车骑行量，以及每天的天气特征。共22维

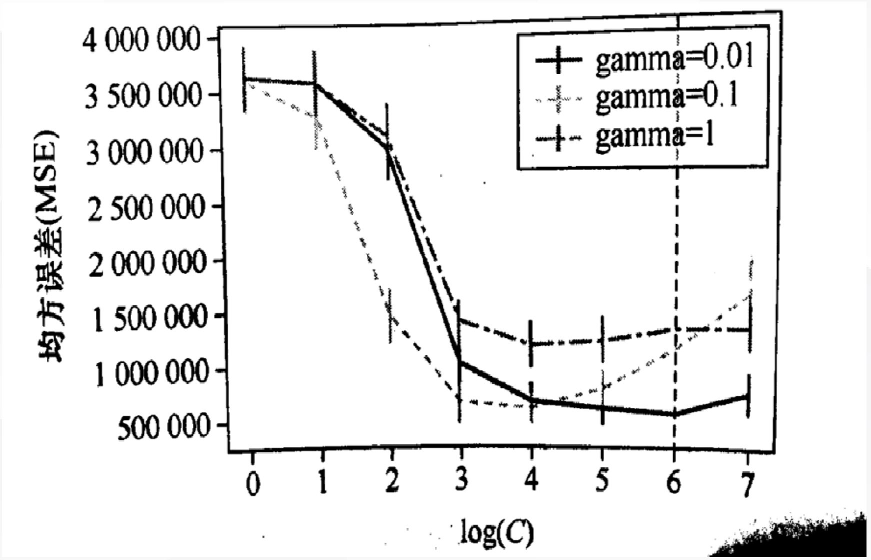
- 离散特征：（日期，年，季节，月份，星期，是否节假日）和天气（晴、阴、雨、雪）
- 连续特征温度、体感温度、湿度、风速

■ 数据预处理：标准化、变量编码等

■ 80%训练，20%测试

■ 选用径向基函数

超参C，gamma



作业

- Show that, irrespective of the dimensionality of the data space, a data set consisting of just two data points (call them $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, one from each class) is sufficient to determine the maximum-margin hyperplane. Fully explain your answer, including giving an explicit formula for the solution to the hard margin SVM (i.e., $\mathbf{w}$) as a function of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$.

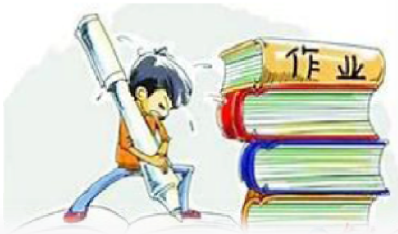
- Gaussian kernel takes the form:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

  Try to show that the Gaussian kernel can be expressed as the inner product of an infinite-dimensional feature vector.

- **Hint:** Making use of the following expansion, and then expanding the middle factor as a power series.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{x^T x}{2\sigma^2}\right)\exp\left(\frac{x^T x'}{\sigma^2}\right)\exp\left(-\frac{(x')^T x'}{2\sigma^2}\right)$$



SVM小结

- SVM(支持向量机)
  - 名词解释
    - SVM — 是一种二类分类模型。基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机。（感知机追求最大程度正确划分，最小化错误，很容易造成过拟合。支持向量机追求大致正确分类的同时，间隔最大也一定程度上避免过拟合）
    - 支持向量 — 线性可分时位于间隔边界的样本，模型的学习只也与支持向量有关
    - 间隔 — 超平面离支持向量的距离。距离越大，表示分类预测的正确性和确信度越大。
  - 模型
    - 线性可分支持向量机 — 硬间隔最大化
    - 线性支持向量机 — 软间隔最大化
    - 非线性支持向量机
      - 软间隔最大化
      - 核技巧
  - 训练过程
    - 将原问题转化为对偶问题（已推出）
    - 用SMO算法求解对偶问题（凸二次规划问题）
    - 确定分离超平面和决策函数
  - 测试过程
    - 决策函数判断wx+b，大于0，记为正例，反之为负例。