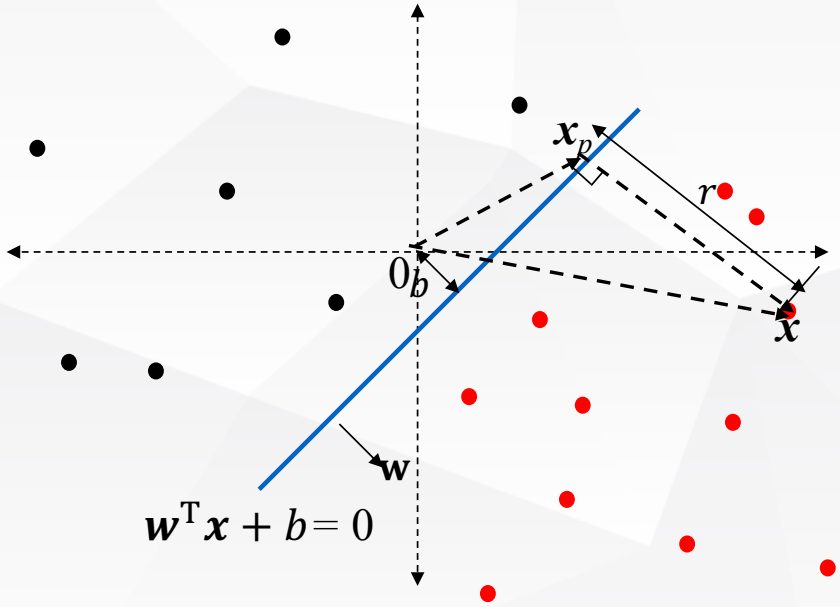
第7章 支持向量机
（Support Vector Machine, SVM）

Outline

- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- 核化SVM模型
- SVM回归（SVR）
- Sklearn中的SVM的API
- SVM应用案例

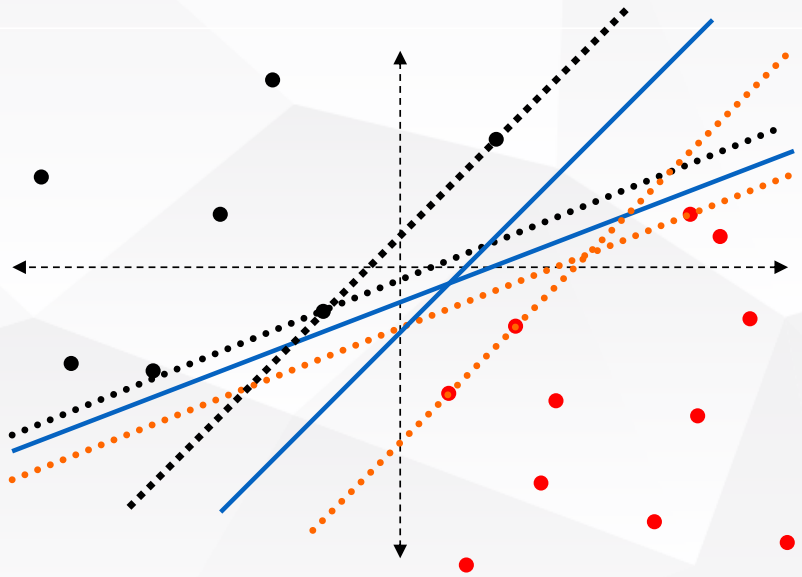
SVM：从几何出发的分类模型

- 假定线性判别函数为：$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$
- 如果$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0$ ，那么$\boldsymbol{x}$是位于超平面上的点
- 不妨假设：所有满足$f(\boldsymbol{x}) < 0$ 的点，其对应的y等于-1
- 所有满足$f(\boldsymbol{x}) > 0$ 的点，其对应的y等于1



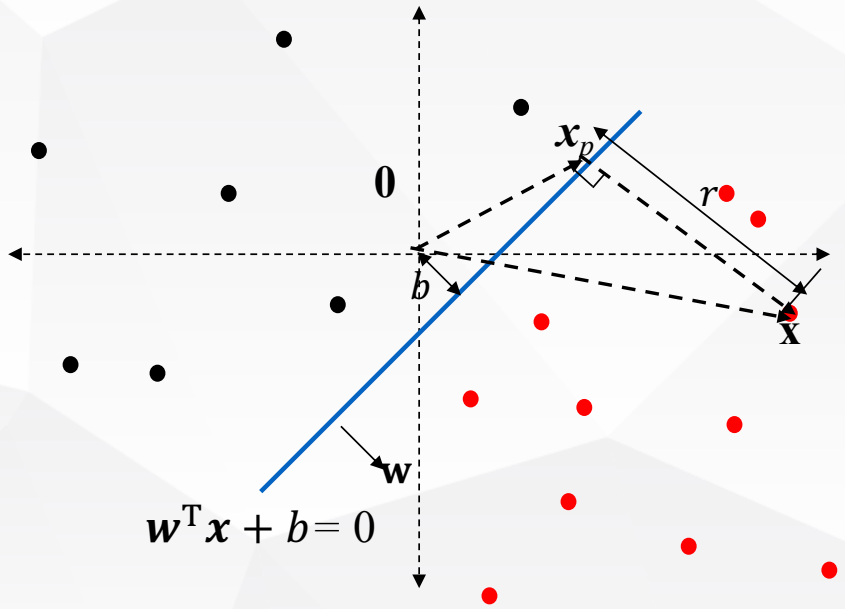
间隔

- 最大间隔原则：最大化两个类最近点之间的距离
  - 这个距离被称为间隔(*margin*)
- 我们先假设分类器是线性可分的



间隔

- 令线性分类面为：$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$
- 则有 $\boldsymbol{x} = \boldsymbol{x}_p + \gamma \dfrac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$
- 其中x到分类面的距离γ。



间隔

■代入得到： $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$

$$= \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{x}_p + \gamma\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}\right) + b$$

$f(\boldsymbol{x}_p) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_p + b = 0$

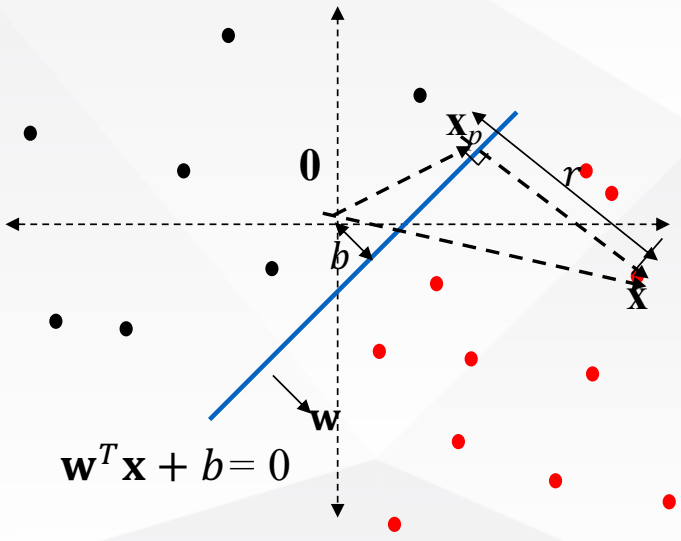
$$= \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_p + \gamma\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{\|\boldsymbol{w}\|_2} + b$$

$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} = \|\boldsymbol{w}\|_2^2$

$$= f(\boldsymbol{x}_p) + \gamma\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$$

$$= \gamma\|\boldsymbol{w}\|_2$$

$$\boldsymbol{x} = \boldsymbol{x}_p + \gamma\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$$

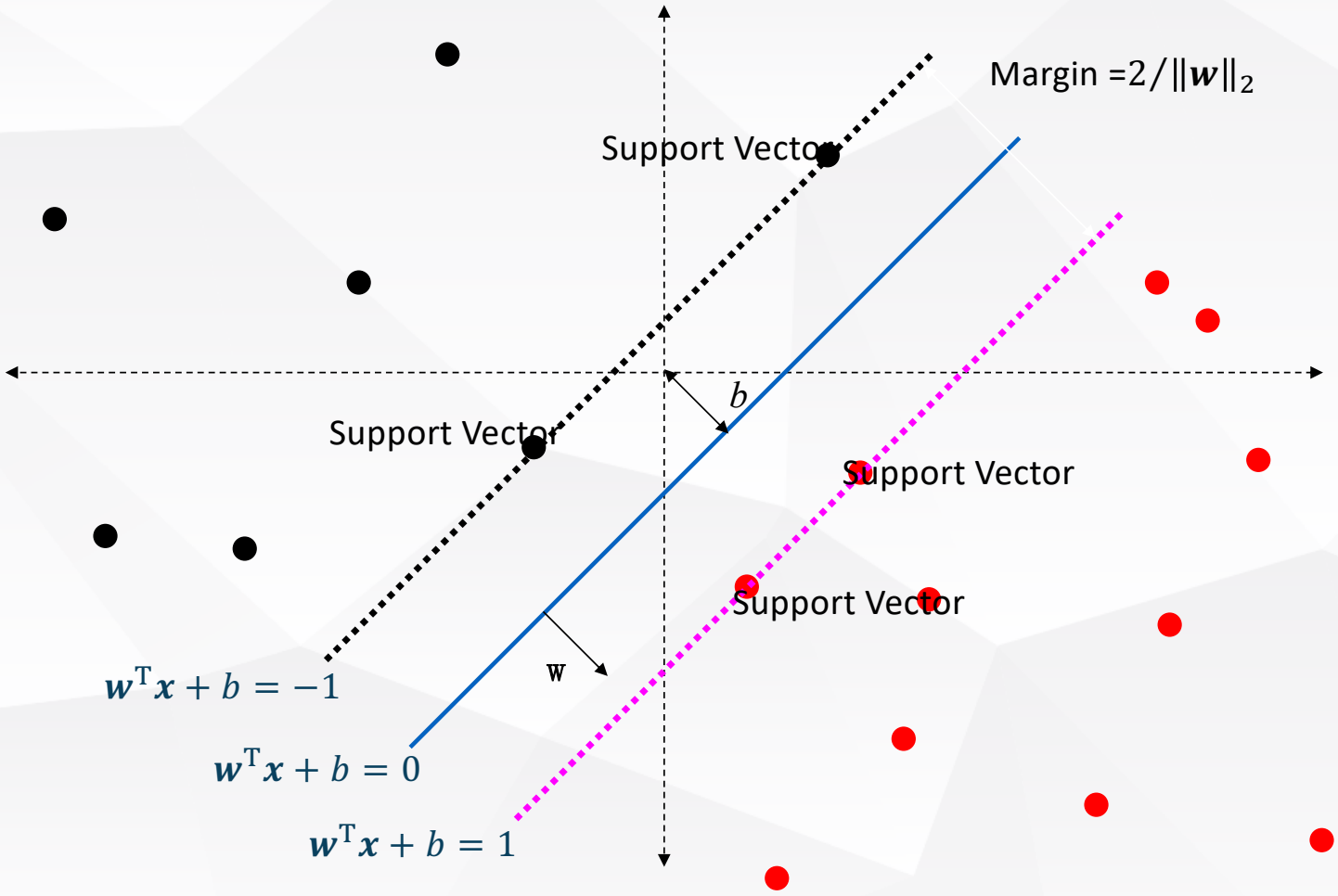


■当x=0时，原点到分类面的距离:$\gamma_0 = \frac{f(\mathbf{0})}{\|\boldsymbol{w}\|_2} = \frac{b}{\|\boldsymbol{w}\|_2}$

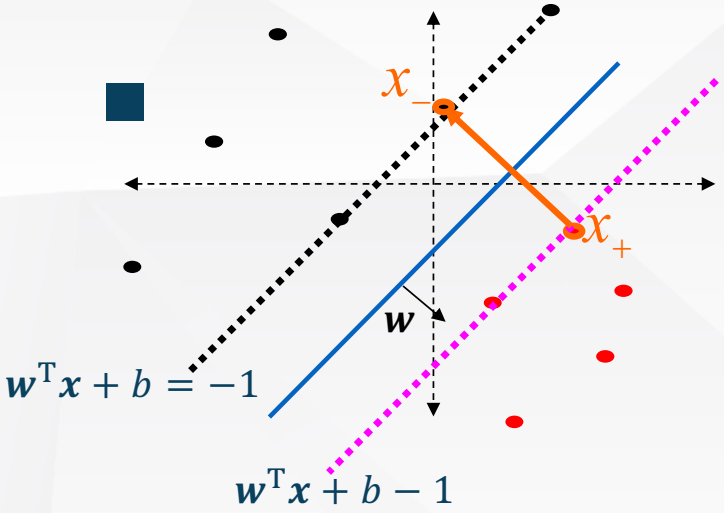
线性判别函数

- 线性判别函数利用一个超平面把特征空间分隔成两个区域.
- 超平面的方向由法向量$\boldsymbol{w}$确定，它的位置由阈值b确定。
- 判别函数$f(\boldsymbol{x})$正比于$\boldsymbol{x}$点到超平面的代数距离（带正负号）
  - 当$\boldsymbol{x}$点在超平面的正侧时，$f(\boldsymbol{x}) > 0$
  - 当$\boldsymbol{x}$点在超平面的负侧时，$f(\boldsymbol{x}) < 0$
  - $\boldsymbol{x}$点到超平面的距离$\frac{yf(\boldsymbol{x})}{\|\boldsymbol{w}\|_2}$可视为对$\boldsymbol{x}$判别的置信度，其中$y \in \{1, -1\}$

SVM 符号表示


Margin =2/‖w‖2
Support Vector
Support Vector
Support Vector
Support Vector
b
w
wᵀx + b = −1
wᵀx + b = 0
wᵀx + b = 1

间隔计算



$$margin = \|x_+ - x_-\|_2$$

$$= \|\lambda w\|_2 \qquad x_+ = x_- + \lambda w$$

$$= \frac{2}{\|w\|_2^2}\|w\|_2 \qquad \lambda = \frac{2}{\|w\|_2^2}$$

$$= \frac{2}{\|w\|_2}$$

$$\left.\begin{matrix} w^T x_+ + b = 1 \\ w^T x_- + b = -1 \end{matrix}\right\} \Rightarrow w^T(x_+ - x_-) = 2$$

$$\Rightarrow w^T \lambda w = 2 \Rightarrow \lambda = \frac{2}{\|w\|_2^2}$$

SVM：最大间隔

■最大化间隔的超平面为：

$$\max \frac{2}{\|\boldsymbol{w}\|_2}$$

$$\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1, \qquad i = 1,2,\ldots,N$$

■等价于

$$\min \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1, \qquad i = 1,2,\ldots,N$$

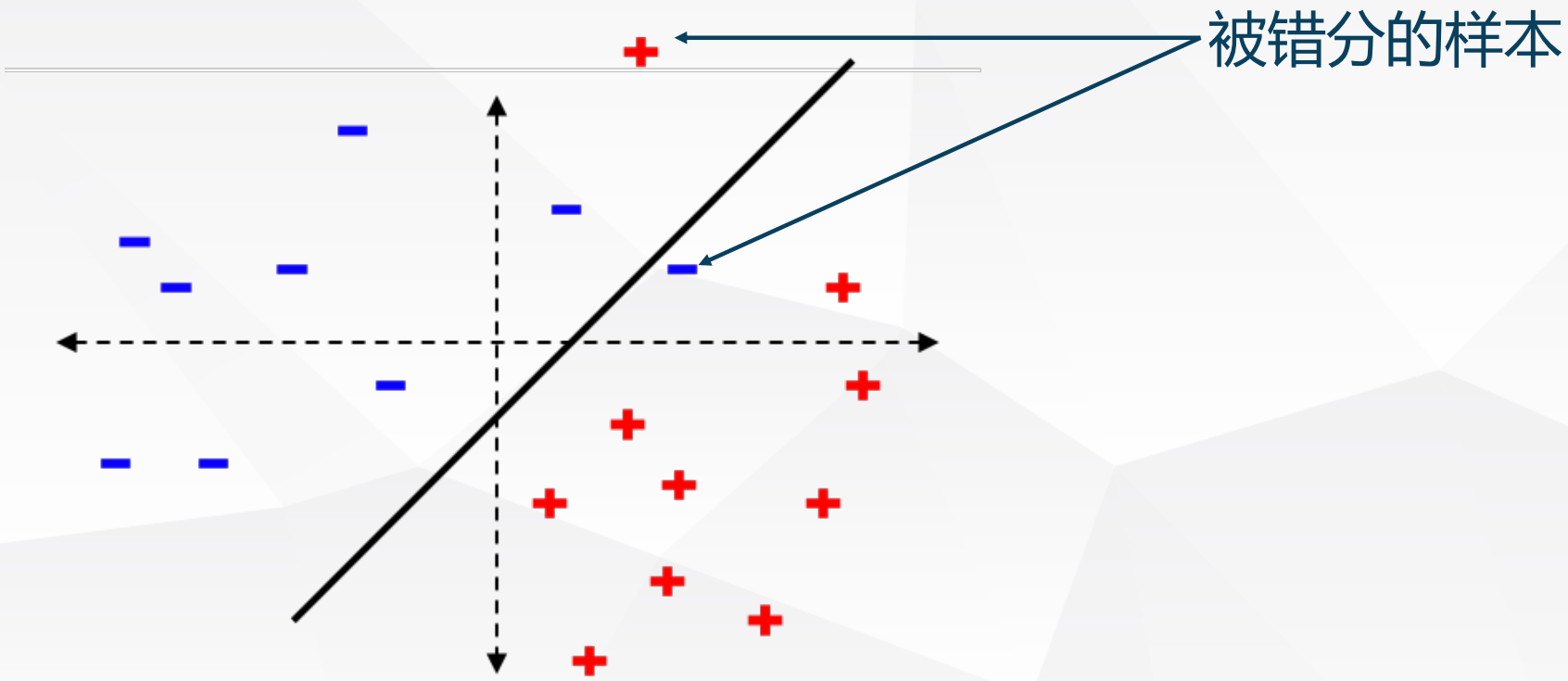
- 二次规划问题(目标函数为二次函数，约束为线性约束)
- 变量数为$D+1$，约束项的数目为N

Outline

- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- 核化SVM模型
- SVM回归（SVR）
- Sklearn中的SVM的API
- SVM应用案例

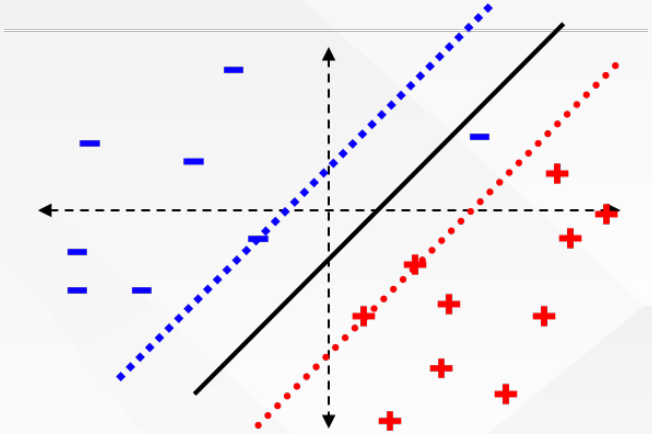
数据不完全线性可分

■在实际问题中，数据不一定完全线性可分。

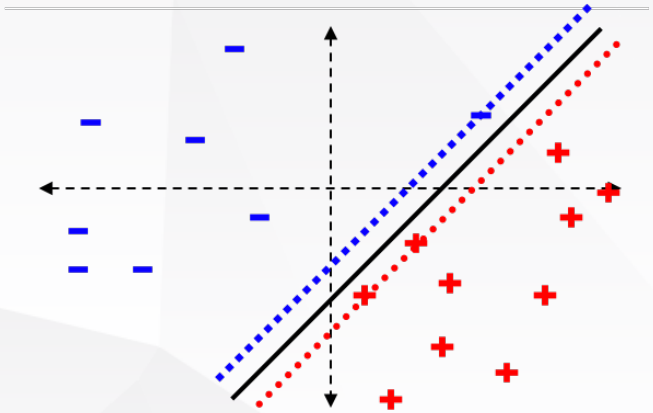


数据可完全线性可分，但间隔小

■或者数据完全线性可分，但完全分开训练样本的分类器间隔小。



少量样本被错分，但间隔大



样本被完全分对，但间隔小

C-SVM

- 当样本可以完全线性可分时：$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1$
- 在实际问题中，数据不一定完全线性可分。
- 因此解决方案引入软间隔（ soft margin ），允许一些样本出错，即允许某些样本不满足约束，将约束放松为

$$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1-\xi_i$$

- 其中ξ_i 称为松弛变量（ *slack variables* ）且$\xi_i\geq 0$。

C-SVM

- 当然松弛变量是有成本的， 样本不满足约束的程度越低越好。
- 每一个松弛变量对应一个代价 ，得到软间隔最大化的SVM（C-SVM）的目标函数：

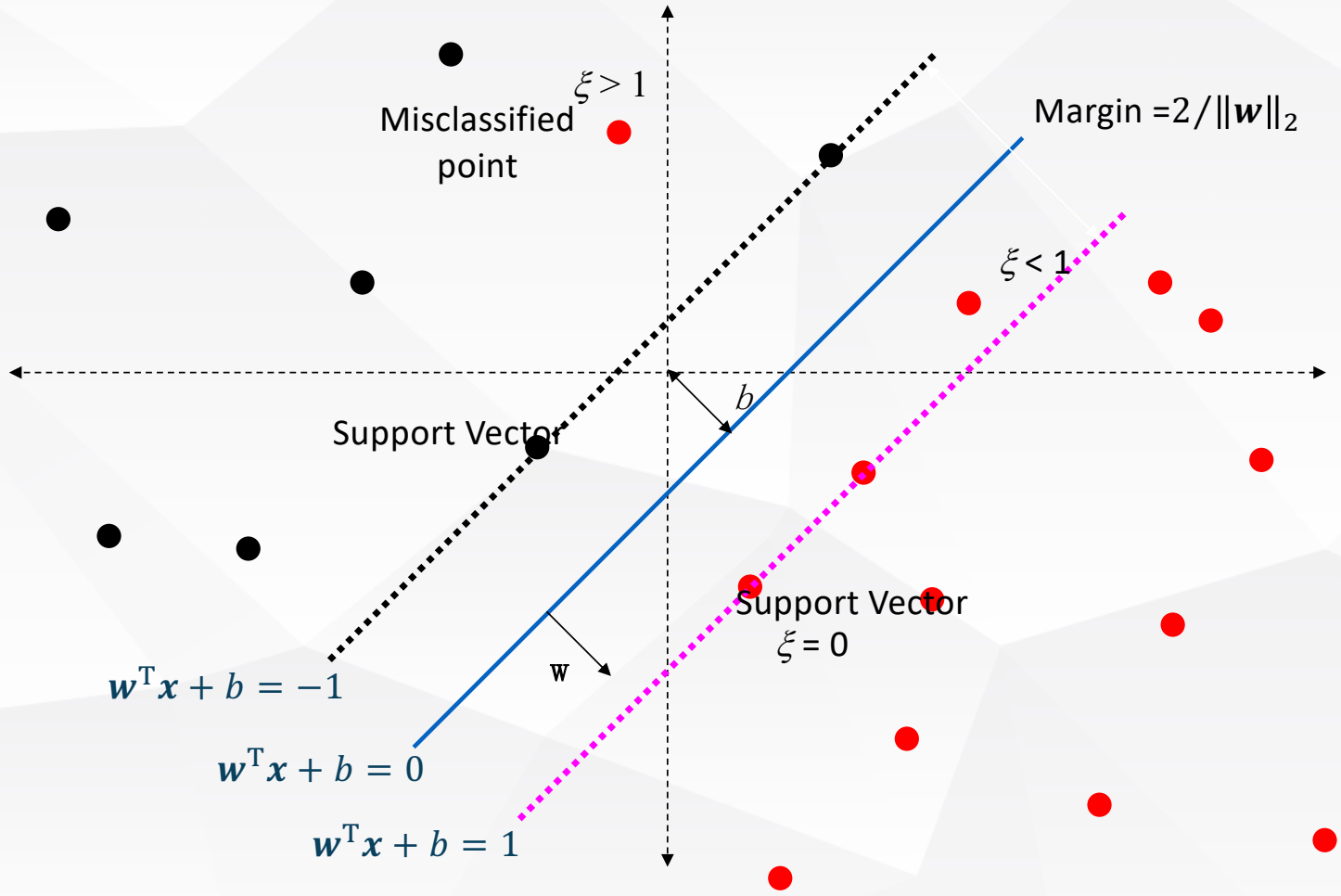
$$J(\boldsymbol{w}, b, C) = C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, \qquad i = 1,2,\ldots,N$$

$$\xi_i \geq 0, \quad i = 1,2,\ldots,N$$

- 即间隔尽可能大，同时样本被误分类的程度尽可能低.
- 其中参数C控制间隔和松弛变量惩罚项之间的平衡，C越大，对误分类的惩罚越大，$\|\boldsymbol{w}\|_2^2$越大，间隔越小。

数据不完全线性可分：松弛变量



C-SVM

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

$$\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, \qquad i = 1,2,\dots,N$$

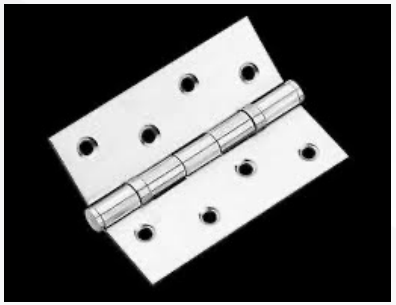
$$\xi_i \geq 0, \quad i = 1,2,\dots,N$$

- 形式与带正则的线性回归或Logistic回归的目标函数类似

$$J(\boldsymbol{w}, \lambda) = C\sum_{i=1}^{N} L(y_i, f(\boldsymbol{x}_i, \boldsymbol{w})) + R(\boldsymbol{w})$$

- 事实上，被误分的样本点的$\xi_i \geq 1$，因此$\sum_{i=1}^{N}\xi_i$为被误分样本数的上界，可视为训练误差。
- 因此参数C可视为控制最小训练误差和模型复杂度的参数。

合页损失（Hinge Loss）

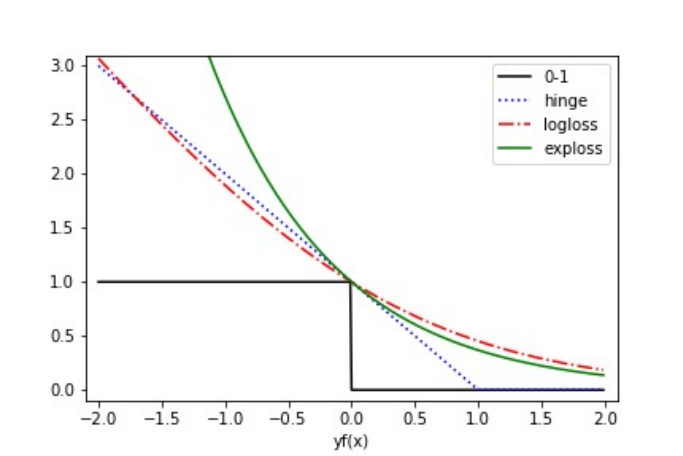


- 在C-SVM中
  - 当$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1,\ \ \xi_i=0$
  - 其他点：$\xi_i=1-y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)$

- 因此得到替代损失函数

$$\xi=L_{Hinge}(y,\hat{y})=\begin{cases}0 & y\hat{y}\geq 1\\ 1-y\hat{y} & otherwise\end{cases}$$

- 该损失函数称为合页损失(Hinge Loss)。



C-SVM：合页损失+L2正则

■将合页损失代入C-SVM的目标函数

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

$$= \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}L_{Hinge}(y_i, f(\boldsymbol{x}_i, \boldsymbol{w}, b))$$

■对比一般机器学习模型的目标函数

$$J(\boldsymbol{\theta}, \lambda) = C\sum_{i=1}^{N}L(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta})) + R(\boldsymbol{\theta})$$

■形式相同。

注意：SVM也可扩展到采用合页损失+L1正则。

Outline

- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- 核化SVM模型
- SVM回归（SVR）
- Sklearn中的SVM的API
- SVM应用案例

回忆：SVM分类模型

- C-SVM的目标函数为

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

$$\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, \qquad i = 1,2,\dots,N$$

$$\xi_i \geq 0, \quad i = 1,2,\dots,N$$

- 为带不等式约束的优化问题。
- 带不等式约束的优化问题可采用拉格朗日乘子变成非约束的优化问题，再进一步变成对偶问题（dual problem）。
- 通过求解与原问题等价的对偶问题(dual problem)得到原始问题的最优解。优点
  - 1. 对偶问题往往更容易求解
  - 2. 可以自然的引入核函数，进而推广到非线性模型

拉格朗日乘子法

■C-SVM原问题目标函数： $J(\boldsymbol{w}, b, C) = C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\|\boldsymbol{w}\|_2^2$

$$\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i, \qquad i = 1,2,\dots,N$$

$$\xi_i \geq 0, \quad i = 1,2,\dots,N$$

■写成标准的不等式约束问题： $\min C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\|\boldsymbol{w}\|_2^2$

$$\text{s.t. } 1 - \xi_i - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \leq 0, \qquad i = 1,2,\dots,N$$

$$-\xi_i \leq 0, \quad i = 1,2,\dots,N$$

■对应的广义拉格朗日函数：

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

$$\text{s.t. } \alpha_i \geq 0, \alpha_i \geq 0, \qquad \mu_i \geq 0, \ \xi_i \geq 0, \ i = 1,2,\dots,N$$

对偶性(Duality)

■一般地，原问题：

$$P = \min_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$\text{s.t.} \quad g_i(\boldsymbol{x}) \le 0, 1 \le i \le K$$
$$h_j(\boldsymbol{x}) = 0, 1 \le j \le L$$

■ 广义拉格朗日函数：

$$L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\boldsymbol{x}) + \sum_{i=1}^{K} \alpha_i \, g_i(\boldsymbol{x}) + \sum_{j=1}^{L} \beta_j \, h_j(\boldsymbol{x})$$

$$\text{s.t.} \quad \alpha_i > 0, 1 \le i \le K$$

对偶性

- 拉格朗日函数：$L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\boldsymbol{x}) + \sum_{i=1}^{K} \alpha_i \, g_i(\boldsymbol{x}) + \sum_{j=1}^{L} \beta_j \, h_j(\boldsymbol{x})$
- 令$\theta_P(\boldsymbol{x}) = \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}: \, \alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$，其中$P$表示原问题（Primal）。
- 如果某个约束条件不满足（$h_j(\boldsymbol{x}) \neq 0$或$g_i(\boldsymbol{x}) > 0$）时

$$\theta_P(\boldsymbol{x}) = \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}: \, \alpha_i \geq 0} \left[f(\boldsymbol{x}) + \sum_{i=1}^{K} \alpha_i \, g_i(\boldsymbol{x}) + \sum_{j=1}^{L} \beta_j \, h_j(\boldsymbol{x}) \right] = \infty$$

  - 只要令不满足约束条件的拉格朗日乘子$\beta_j = \infty$或$\alpha_i = \infty$即可。
- 当所有约束条件都满足时，则有：$\theta_P(\boldsymbol{x}) = f(\boldsymbol{x})$
- 亦即我们最初要最小化的函数。

对偶性

- 考虑最小化问题：$\min_{\boldsymbol{x}} \theta_P(\boldsymbol{x}) = \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ ，
- 这个问题的解和我们最初的原问题的解相同。
- 用$p^* = \min_{\boldsymbol{x}} \theta_P(\boldsymbol{x}) = \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$表示这是原问题的最优解。

- 接下来我们考虑另一个不同的问题：$\theta_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ ，其中D表示对偶问题（Dual）。

- 定义对偶问题的优化问题：$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} \theta_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} \min_{\mathbf{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$。
- 用$d^* = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} \theta_D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$表示对偶问题的最优解。

注意：和原问题相比，对偶问题交换了max和min的顺序。

对偶性

- 原问题和对偶问题直接有什么关系呢？
- 容易证明：

$$d^* = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta}:\, \alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = p^*$$

  - 直观上，最大值中最小的一个，总比最小值中最大的一个要大。
- 对偶问题的最优值d^*提供了一个原问题的最优值p^*的一个下界。
- 在满足一定条件时，$d^* = p^*$，可以通过求解对偶问题来间接地求解原问题。

对偶性

- 假设f, g_i是凸函数，h_j是仿射函数（形式为$h_j(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} + b$）
- 同时约束g_i是严格可行的（存在$\boldsymbol{x}$使得$g_i(\boldsymbol{x}) \leq 0$），
- 则必然存在$\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$，使得
  - $\boldsymbol{x}^*$是原问题的最优解
  - $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$是对偶问题的最优解
  - $d^* = p^* = L(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$

- 同时$\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$满足如下Karush-Kuhn-Tucker (KKT) 条件。

对偶性

■最优解$\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$满足如下Karush-Kuhn-Tucker (KKT) 条件

$$\frac{\partial L(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{x}} = 0$$

$$\frac{\partial L(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \beta_j} = 0 \text{ , } 1 \le j \le L$$

$$\alpha_i^* \ge 0, \qquad 1 \le i \le K$$

$$g_i(\boldsymbol{x}^*) \le 0, \qquad 1 \le i \le K$$

$$\alpha_i^* g_i(\boldsymbol{x}^*) = 0, \qquad 1 \le i \le K$$

■对偶互补条件是SVM的很多重要性质（如支持向量）的来源

- 如果要满足$\alpha_i^* g_i(\boldsymbol{x}^*) = 0$，必须$\alpha_i^* = 0$或者$g_i(\boldsymbol{x}^*) = 0$
- $\alpha_i^* = 0$ 时，约束$g_i^*(\boldsymbol{x}) < 0$不起作用；$g_i(\boldsymbol{x}^*) = 0$ 时，是等式约束问题。

SVM的对偶问题

■拉格朗日函数$L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu})$

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

■对$\boldsymbol{w}, b, \xi_i$最小化（一阶导数为0）：

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial \boldsymbol{w}} = 0 \Longrightarrow \boldsymbol{w} = \sum_{i=1}^{N}\alpha_i\, y_i \boldsymbol{x}_i$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{N}\alpha_i\, y_i = 0$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu})}{\partial \xi_i} = 0 \Longrightarrow C = \alpha_i + \mu_i$$

SVM的对偶问题

■将上述结论代入拉格朗日函数$L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu})$

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu} = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\,(y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{N}\mu_i\,\xi_i$$

$C = \alpha_i + \mu_i$

$$= \frac{1}{2}\|\boldsymbol{w}\|_2^2| - \sum_{i=1}^{N}\alpha_i\,(y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 + \xi_i) + \sum_{i=1}^{N}\alpha_i\,\xi_i$$

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

$$= \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} - \sum_{i=1}^{N}\alpha_i\,y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - \sum_{i=1}^{N}\alpha_i\,y_i b + \sum_{i=1}^{N}\alpha_i$$

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

$$= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j - b\sum_{i=1}^{N}\alpha_i\,y_i + \sum_{i=1}^{N}\alpha_i$$

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

SVM的对偶问题

- 从而得到对偶变量$\boldsymbol{\alpha}$的优化问题：

$$\max\left(\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j\right)$$

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

$$0\le\alpha_i\le C,\qquad i=1,2,\ldots,N$$

- 可以看出，SVM中满足： $d^*=p^*$ ，可以通过求解对偶问题来得到原问题的最优解。
- 对偶问题仍然是一个QP问题：变量数为N，约束项的数目为$(N+1)$
  - 当N较大时，对偶问题的复杂度可能比原问题更高
  - 但对偶问题可利用核技巧(kernel trick)与核方法结合
  - 对此问题有高效的求解算法：SMO（Sequential Minimal Optimization）

w, b的计算

- 得到最佳的$\boldsymbol{\alpha}$后，可计算$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \, y_i \boldsymbol{x}_i$
- 用任意一个支持向量即可求得b： $b = y_i - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i$
- 为了得到更稳定的解，对所有支持向量求平均

$$b = \frac{1}{N_S} \sum_{\boldsymbol{x}_{\mathrm{m}} \in S} \left(y_m - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_{\mathrm{m}} \right)$$

SVM的优化算法SMO迭代更新$\boldsymbol{\alpha}$，对$\boldsymbol{w}, b$也是迭代更新。

SVM的对偶问题

■得到$\boldsymbol{\alpha}, \boldsymbol{w}, b$ 后，可计算SVM模型：

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

$$= \left(\sum_{i=1}^{N} \alpha_i\, y_i \boldsymbol{x}_i\right)^{\mathrm{T}} \boldsymbol{x} + b$$

$$= \sum_{i=1}^{N} \alpha_i\, y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x} + b$$

$$= \sum_{i=1}^{N} \alpha_i\, y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b$$

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

对于新点$\boldsymbol{x}$的预测，只需要计算它与训练数据的内积。
这一点是我们使用核方法进行非线性推广的基本前提。

■点$\boldsymbol{x}$ 的标签可根据$f(\boldsymbol{x})$的符号得到：$\hat{y} = \mathrm{sign}(f(\boldsymbol{x}))$

α的稀疏性

- SVM的KKT条件中的对偶互补条件为：

$$\alpha_i \left(1 - \xi_i - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right) = 0$$

$$\mu_i \xi_i = 0$$

- 因此，对每个训练样本点：$\alpha_i \left(1 - \xi_i - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right) = 0$

$$\alpha_i = 0 \quad \text{或} \quad 1 - \xi_i - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 0$$

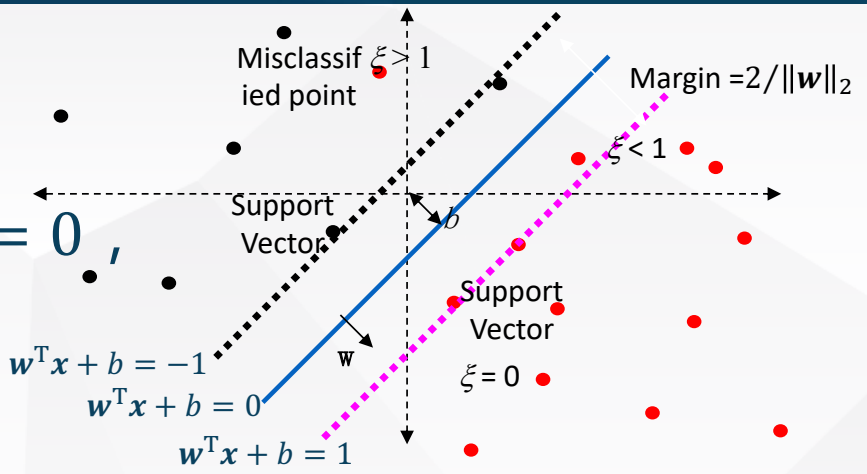
- 当$\alpha_i = 0$ 时，该点在决策函数

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = \sum_{i=1}^{N} \alpha_i\, y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b$$

- 中不起作用（大多数点可以抛掉， 不必保存）。

α的稀疏性

- $\alpha_i\left(1-\xi_i-y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\right)=0$
- 当$\alpha_i \neq 0$时，$1-\xi_i-y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)=0$，这些样本点被称为支持向量。



（1）$\alpha_i < C$：则$\mu_i > 0$，进而根据$\mu_i\xi_i = 0 \Rightarrow \xi_i = 0$，则$(\boldsymbol{x}_i, y_i)$恰好在最大间隔边界$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 1-\xi_i = 1$上；

（2）$\alpha_i = C$：则$\mu_i = 0$，此时$\xi_i \geq 0$：

①若$\xi_i \leq 1$，则$(\boldsymbol{x}_i, y_i)$落在在最大间隔边界内部，$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 1-\xi_i \geqslant 0$，仍能被正确分类；

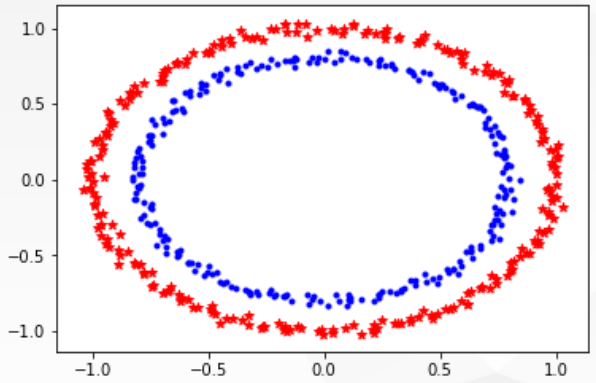
②若$\xi_i > 1$，则$(\boldsymbol{x}_i, y_i)$落在在最大间隔边界外部，$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 1-\xi_i \leqslant 0$，样本被错误分类。

Outline

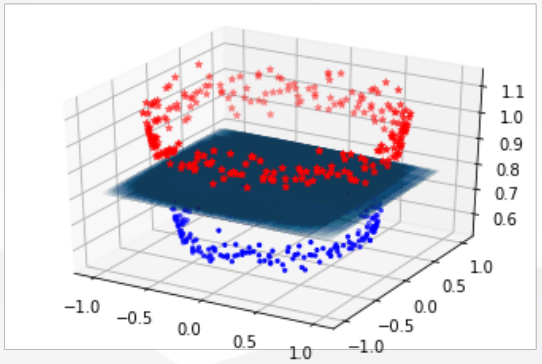
- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- **核化SVM模型**
- SVM回归（SVR）
- Sklearn中的SVM的API
- SVM应用案例

核方法

- 前面我们用超平面（线性模型）来分开不同类的训练样本。
- 但在实际任务中，原始样本空间也许并不存在一个超平面能将训练样本分开。
- 例如：



$$(z_1, z_2, z_3) = (x, y, x^2 + y^2)$$



- 对这类问题，我们可以将原始空间映射到一个更高维特征空间，使得在这个特征空间数据线性可分。

核方法

- 令$\phi(\boldsymbol{x})$ 表示将$\boldsymbol{x}$映射后的特征向量，则在特征空间中划分超平面对应的模型可表示为

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b$$

- 根据之前SVM的推导，得到特征映射后的SVM目标函数

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

$$\text{s.t. } 1 - \xi_i \le y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_i) + b\right), \qquad i = 1,2,\dots,N$$

$$\xi_i \ge 0, \quad i = 1,2,\dots,N$$

核方法——对偶

■相应的对偶问题为：

$$\max\left(\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j y_i y_j\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle\right)$$

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

$$0\le\alpha_i\le C,\qquad i=1,2,\ldots,N$$

■求得偶问题的解$\boldsymbol{\alpha}$后，可计算$\boldsymbol{w},b$，从而得到分类判别函数：

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b$$

$$=\sum_{i=1}^{N}\alpha_i\,y_i\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x})\rangle+b$$

核技巧（Kernel Trick）

- 将问题变为对偶问题：只需计算点积
- 在SVM中，最大化下列目标函数

$$W(\boldsymbol{\alpha}) = \sum_{i=0}^{N} \alpha_i - \frac{1}{2} \sum_{i=0}^{N} \sum_{j=0}^{N} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}_j) \rangle$$

- 判别函数为：$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}) + b = \sum_{i=1}^{N} \alpha_i \, y_i \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}) \rangle + b$
- 由于特征空间维数可能很高（甚至无穷维），直接计算特征空间的点积通常是困难的
- 核函数：高维空间中的点积可写成核(*kernel*)的形式

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}_j) \rangle$$

核技巧（Kernel Trick）

- 选定核函数，无需计算映射$\phi(\boldsymbol{x})$ 就可以计算点积
- SVM核化目标函数为

$$W(\boldsymbol{\alpha}) = \sum_{i=0}^{N} \alpha_i - \frac{1}{2} \sum_{i=0}^{N} \sum_{j=0}^{N} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- 预测模型为：$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x}) + b = \sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b$

构造核函数

- 一种核函数的构造方式是显式地定义一个特征映射$\phi(.)$，将每个输入$\boldsymbol{x}$映射到$\phi(\boldsymbol{x})$。从而得到核函数的间接定义

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$$

- 其中$\phi(.)$称为基函数。

构造核函数

- 另一种可选的方式是直接定义核函数。
- 此时需保证函数是有效核。

- 在显式定义特征映射的情况下，核函数为特征空间（可能为无限维）中的内积。

核函数

- 令$\mathcal{X}$为输入空间，$k(.,.)$是定义在$\mathcal{X} \times \mathcal{X}$上的对称函数，则$k$是核函数的充要条件是对任意数据$\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$，核矩阵$\boldsymbol{K}$总是半正定的：

$$\boldsymbol{K} = \begin{bmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \kappa(\boldsymbol{x}_2, \boldsymbol{x}_1) & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_N, \boldsymbol{x}_1) & \kappa(\boldsymbol{x}_N, \boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$

- 对于一个半正定核矩阵，总是能找到一个与之对应的映射ϕ
  - 任何一个核函数都隐式定义了一个再生Hilbert空间

多项式核

- 多项式核： $k(\boldsymbol{x}, \boldsymbol{x}') = \left(\gamma \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}' + r\right)^{M}$
- 当$M = 1$，$\gamma = 1$，$r = 0$时，为线性核，

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

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}'$$

RBF核

- 径向基函数 (RBF)：： $k(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\|\boldsymbol{x} - \boldsymbol{x}'\|)$
- 指数平方(Squared exponential, SE)/高斯核

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{x}')\right)$$

- 当$\boldsymbol{\Sigma}$为对角阵时

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\sum_{j=1}^{D}\frac{(x_j - x_j')^2}{2\sigma_j^2}\right)$$

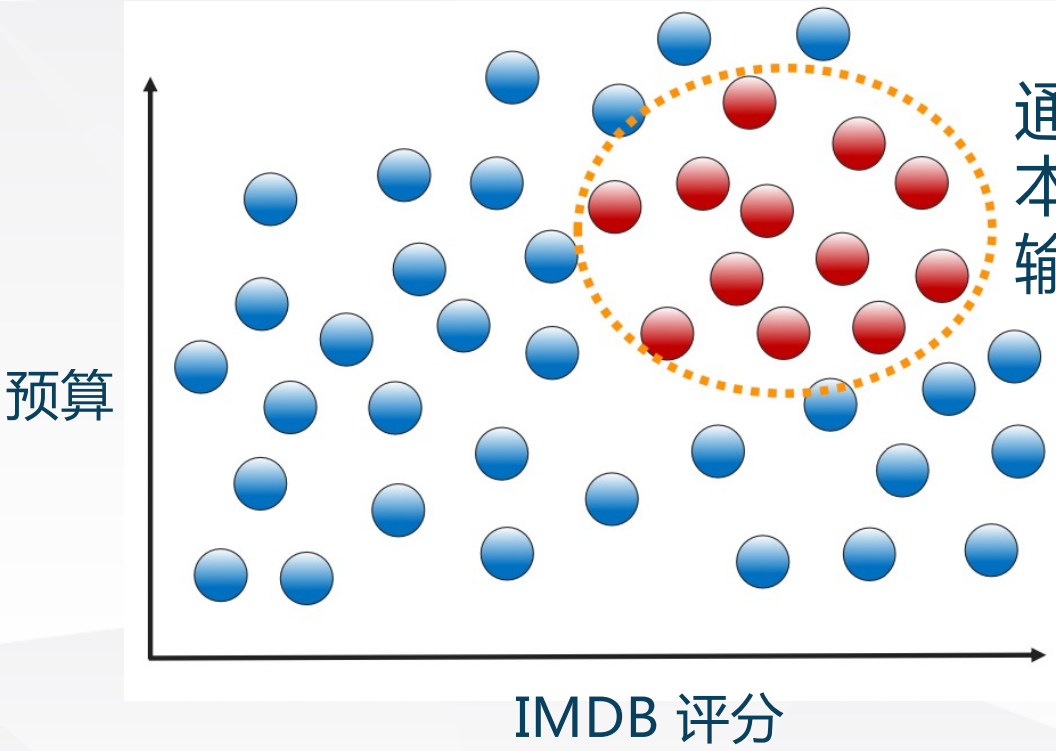
- 当$\boldsymbol{\Sigma}$为球形时，得到各向同性核

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\sigma^2}\right) = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2)$$

- σ称为核函数的带宽。 γ为核函数的带宽的倒数

RBF核

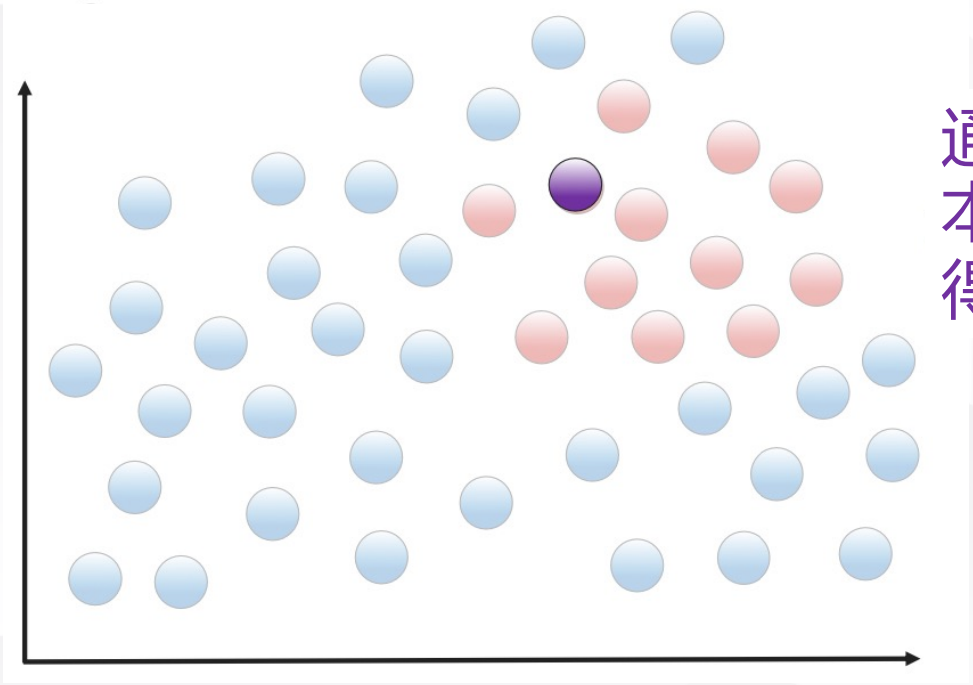
■例：金棕榈奖(Palme d'Or) 电影预测



通过计算观测样本与训练样本/支持向量的相似性，将输入变换到无穷维空间

RBF核

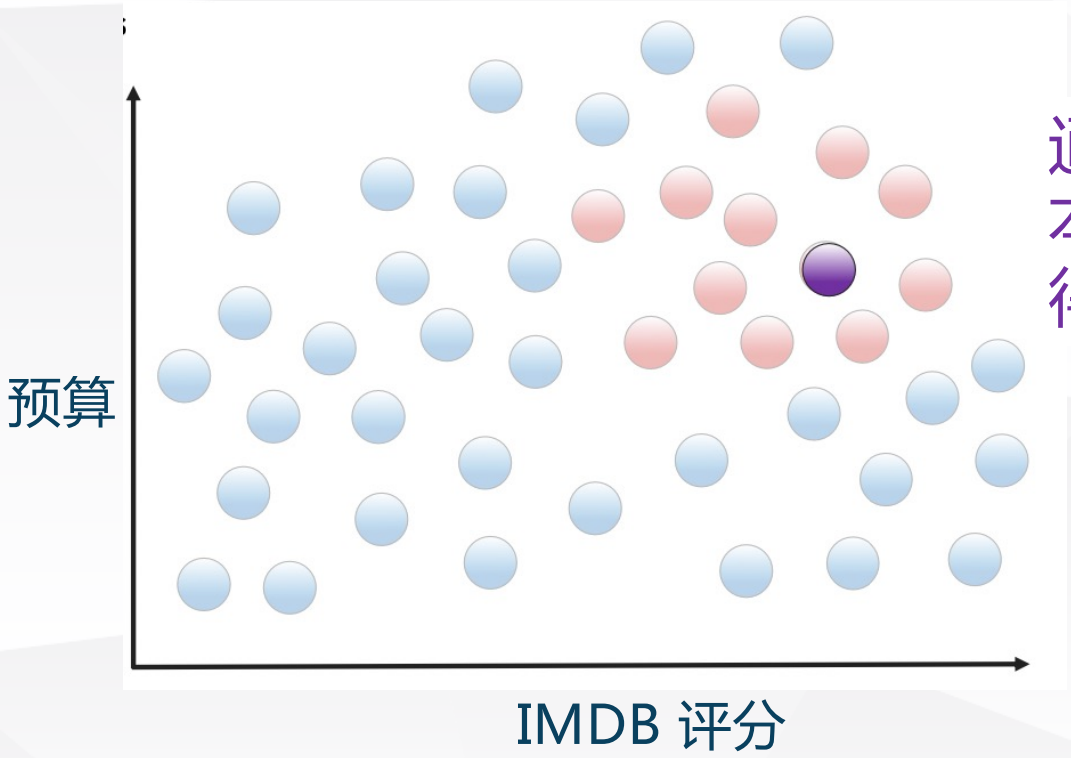
■例：金棕榈奖(Palme d'Or) 电影预测



通过计算观测样本与训练样本“Pulp Fiction” 的相似性，得到第1维特征

RBF核

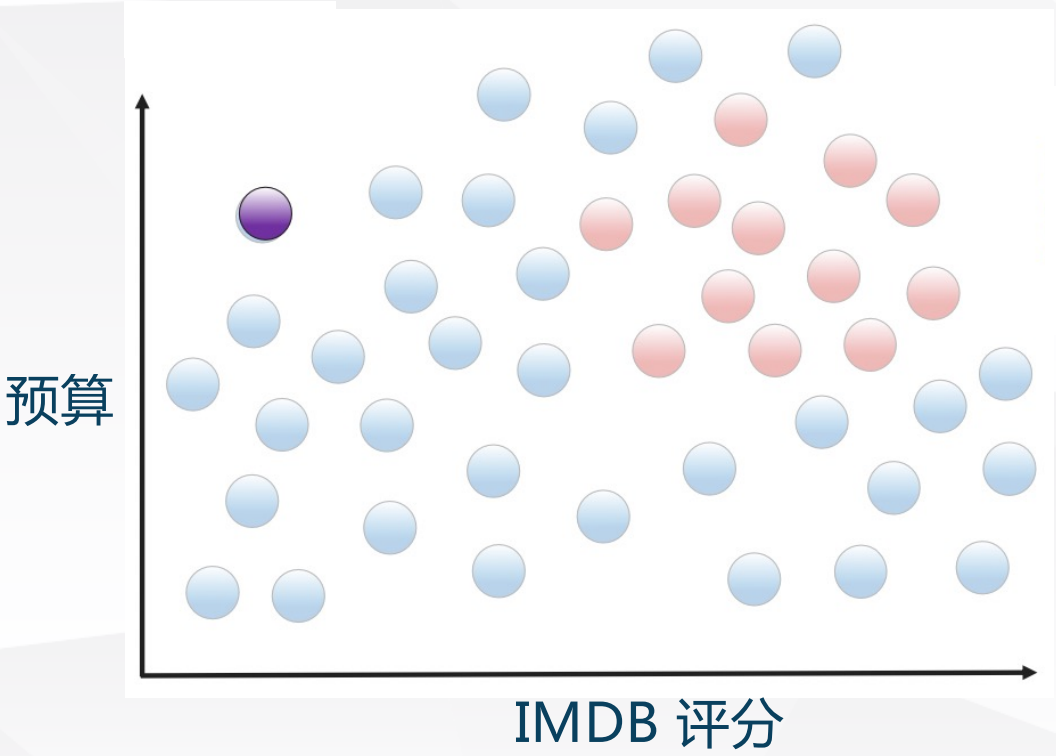
■例：金棕榈奖(Palme d'Or) 电影预测



通过计算观测样本与训练样本“Black Swan ”的相似性，得到第2维特征

RBF核

■例：金棕榈奖(Palme d'Or) 电影预测

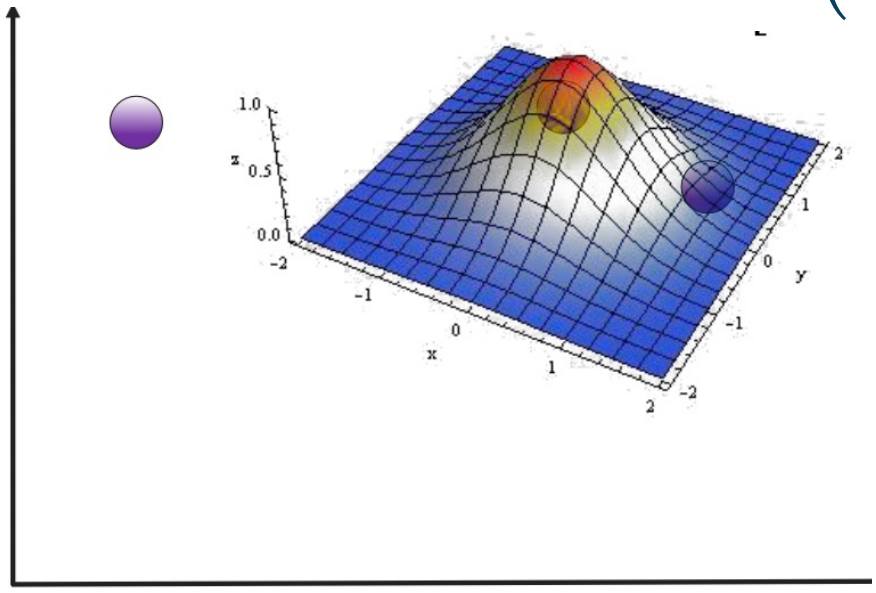


通过计算观测样本与训练样本“Transformer” 的相似性，得到第3维特征

RBF核

■例：金棕榈奖(Palme d'Or) 电影预测

$$a_1(\boldsymbol{x}_i) = \exp\left(-\gamma \left\| \boldsymbol{x}_i - \boldsymbol{x}_{\text{Pulp Fiction}} \right\|_2^2\right)$$



对第1维特征
创建一个高斯函数

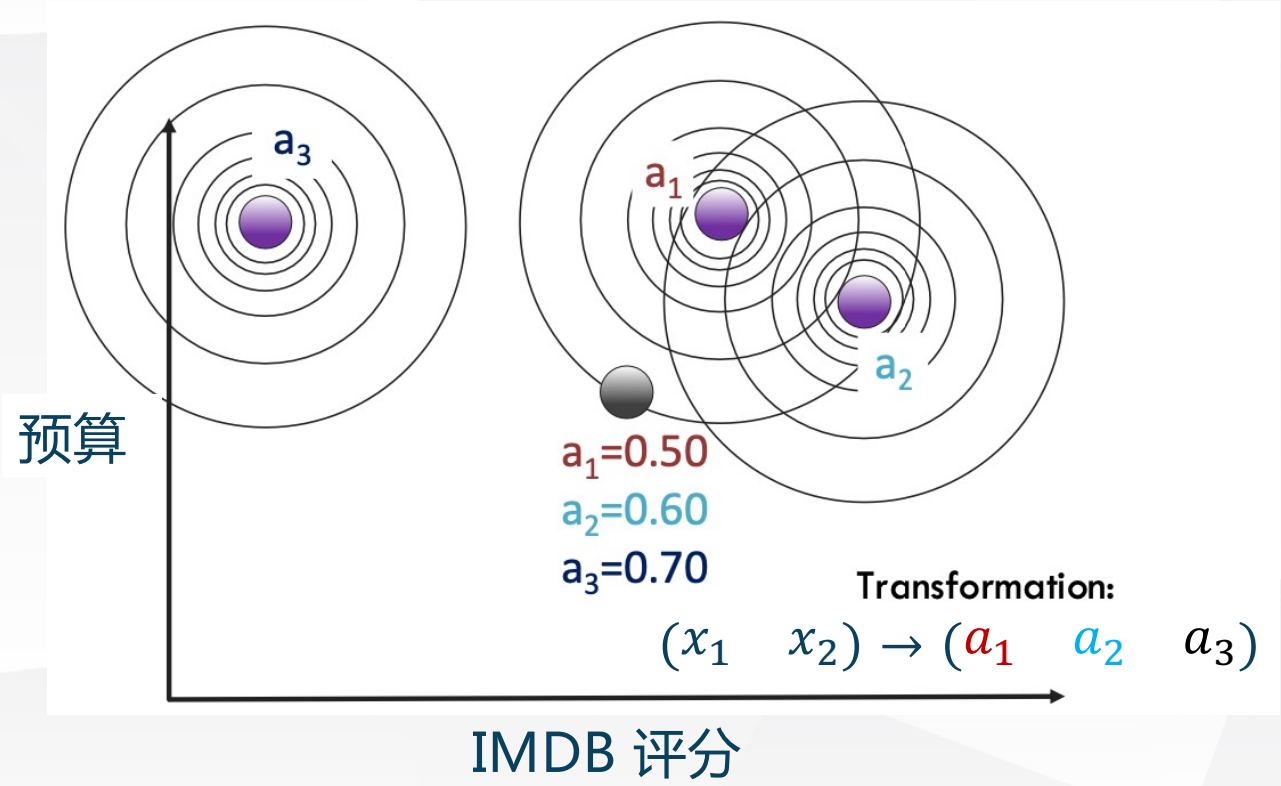
对第2维特征
创建一个高斯函数

对第3维特征
创建一个高斯函数

$$a_3(\boldsymbol{x}_i) = \exp\left(-\gamma \left\| \boldsymbol{x}_i - \boldsymbol{x}_{\text{Transformer}} \right\|_2^2\right)$$

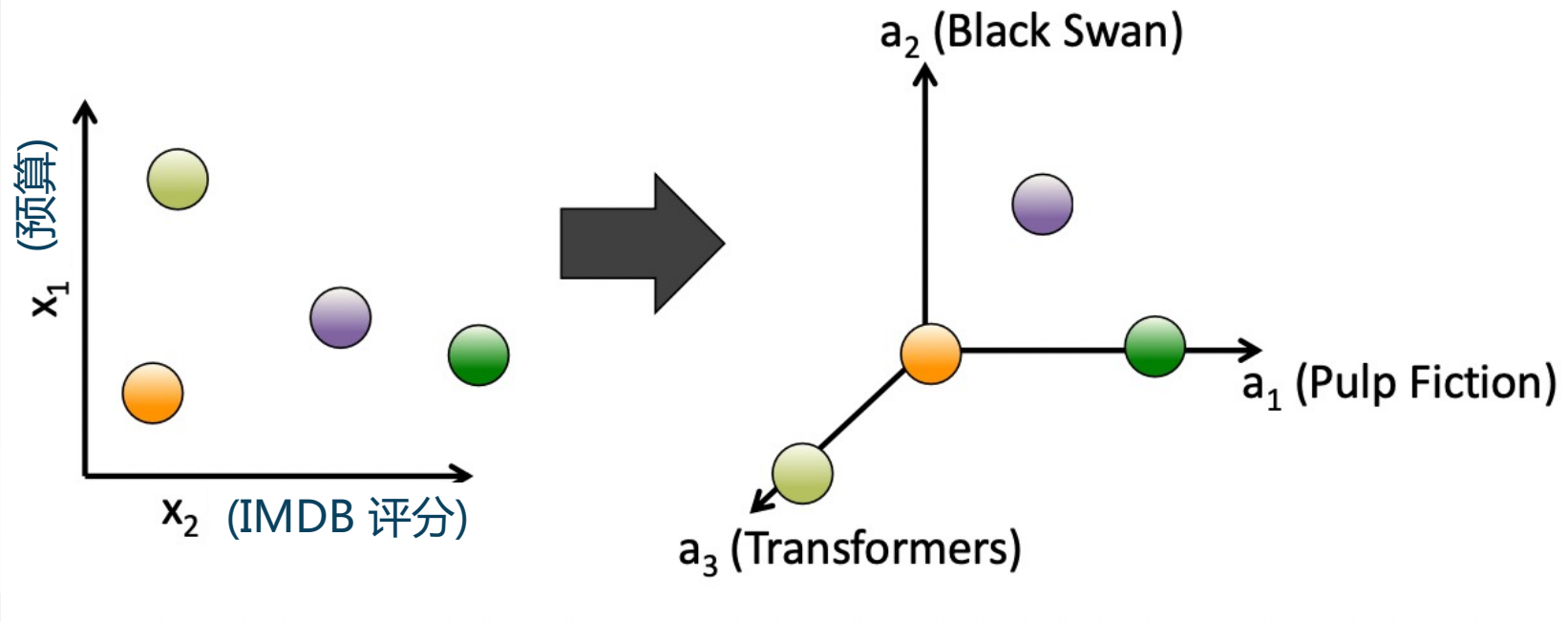
RBF核

■例：金棕榈奖(Palme d'Or) 电影预测



RBF核

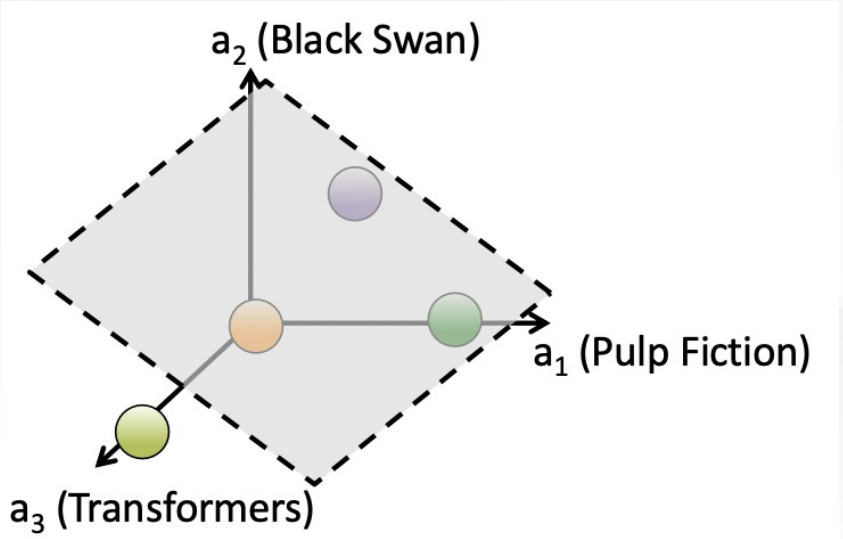
■例：金棕榈奖(Palme d'Or) 电影预测

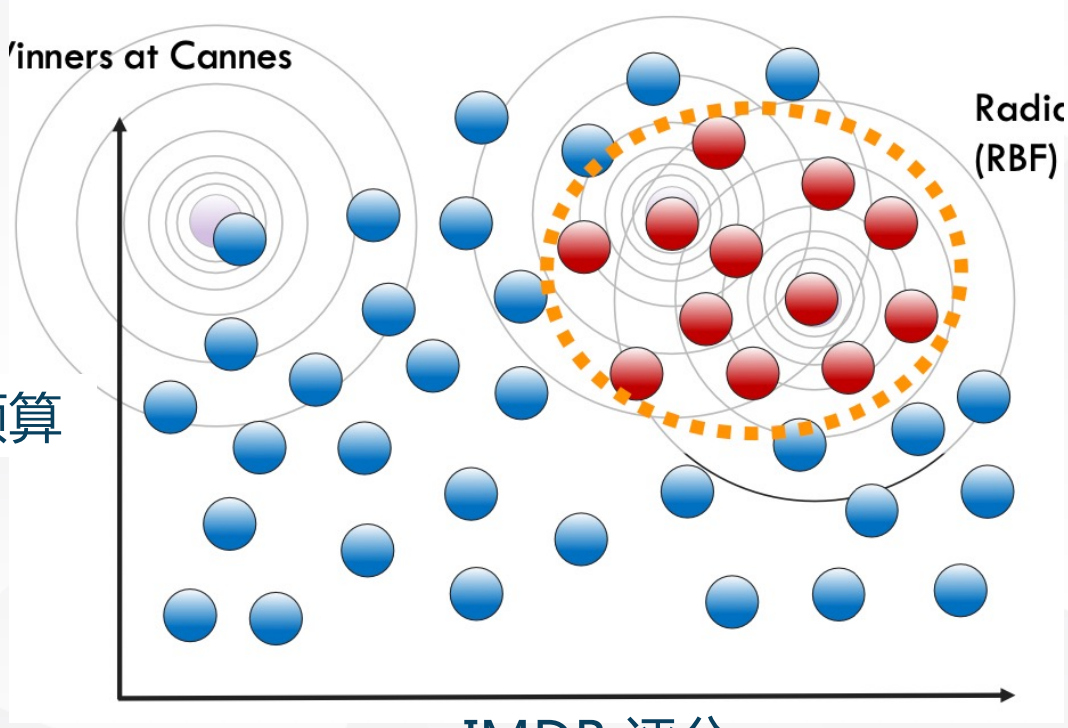


$$(x_1 \quad x_2) \rightarrow (a_1 \quad a_2 \quad a_3)$$

RBF核

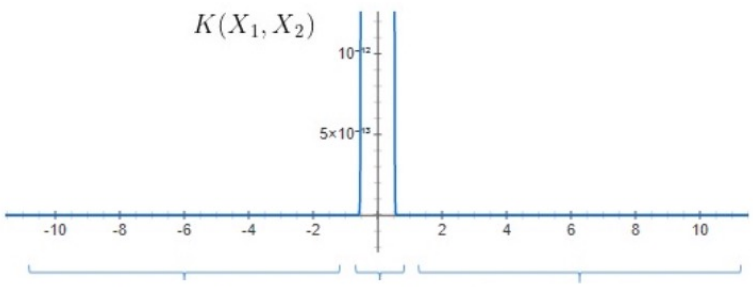
■例：金棕榈奖(Palme d'Or) 电影预测



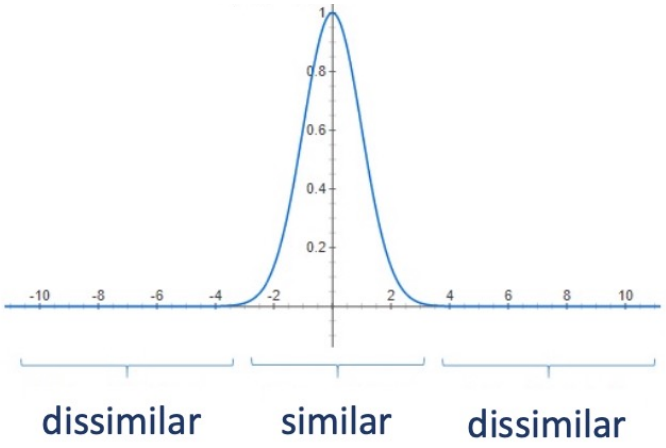


RBF：核函数的宽度

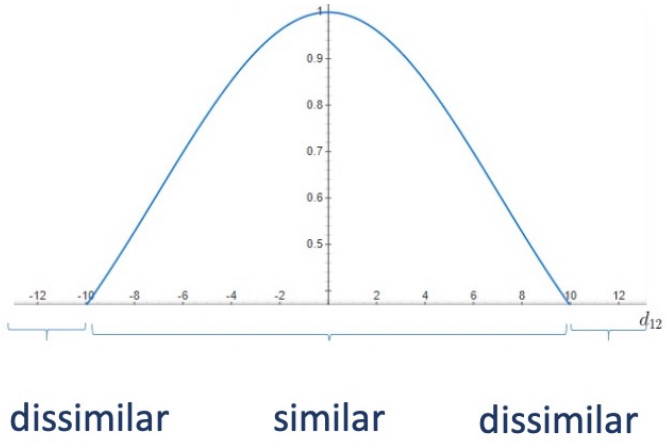
$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{0.01}\right)$$



$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2}\right)$$



$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{100}\right)$$



更多核函数

■核的闭包性质：令 k_1 和 k_2 为定义在X * X核，则下列表示也是核：

$$k(\mathbf{x},\mathbf{z}) = k_1(\mathbf{x},\mathbf{z}) + k_2(\mathbf{x},\mathbf{z})$$

$$k(\mathbf{x},\mathbf{z}) = ak_1(\mathbf{x},\mathbf{z})$$

$$k(\mathbf{x},\mathbf{z}) = k_1(\mathbf{x},\mathbf{z})k_2(\mathbf{x},\mathbf{z})$$

$$k(\mathbf{x},\mathbf{z}) = k_1(\phi(\mathbf{x}),\phi(\mathbf{z})), \quad \text{where } \phi: X \to \mathbf{R}^n$$

$$k(\mathbf{x},\mathbf{z}) = \mathbf{x}^T\mathbf{B}\mathbf{z}, \quad \text{where } \mathbf{B} \text{ is SPD}$$

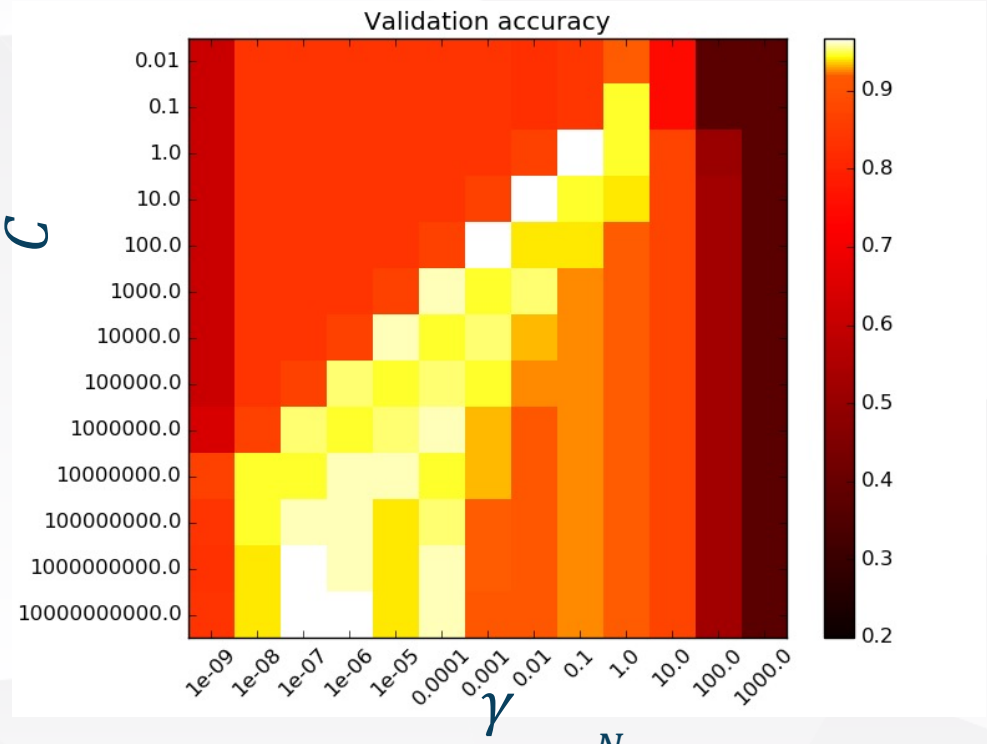
$$k(\mathbf{x},\mathbf{z}) = p(k_1(\mathbf{x},\mathbf{z})), \quad \text{where } p \text{ is a polynomial with positive coefficient}$$

$$k(\mathbf{x},z) = \exp(k_1(\mathbf{x},\mathbf{z}))$$

$$k(\mathbf{x},z) = \exp\left(\|\mathbf{x}-\mathbf{z}\|^2/2\right)$$

例：RBF核

在鸢尾花分类任务上（只取了前2维特征），
不同参数值的RBF核SVM分类器的交叉验证精度



C：训练样本有多重要
越大，越看重训练样本，模型越复杂

γ： RBF核宽度的倒数，
表示每个训练样本的影响范围

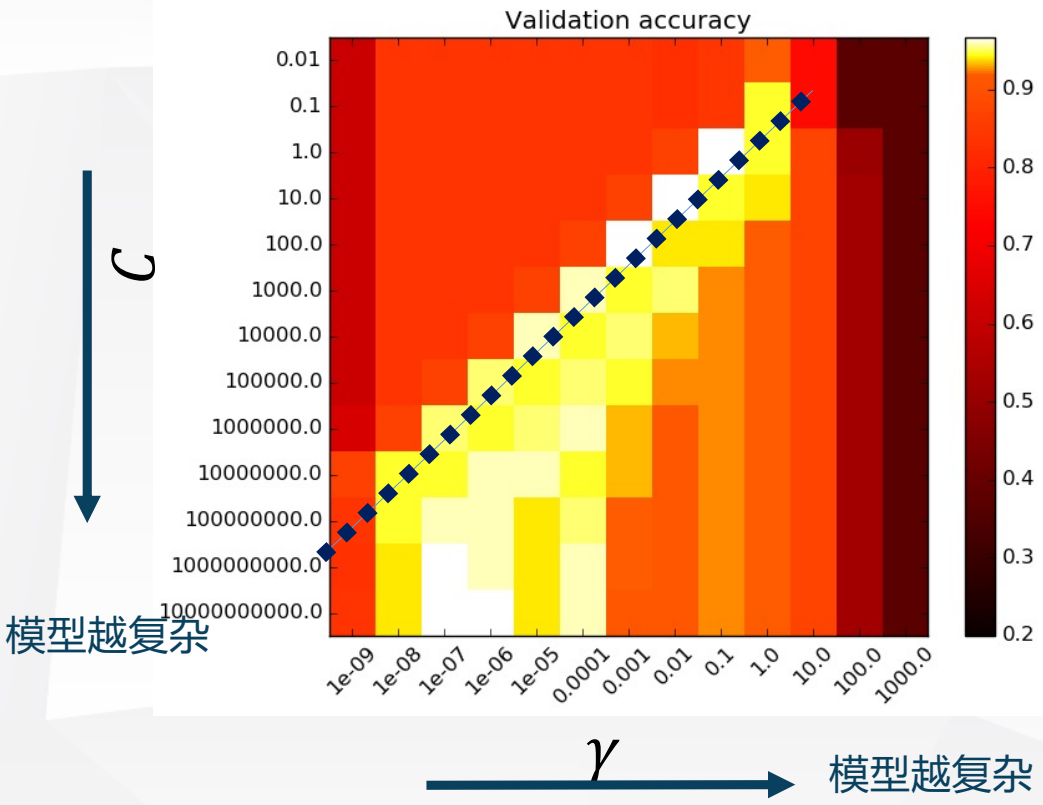
越大， RBF核宽度越小，
每个训练样本/支持向量的范围越小，
模型越复杂

Source: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

$$J(\boldsymbol{w}, b; C) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N} L_{Hinge}(y_i, f(\boldsymbol{x}_i; \boldsymbol{w}, b))$$

例：RBF核

在鸢尾花分类任务上（只取了前2维特征），
不同参数值的RBF核SVM分类器的交叉验证精度



对角线上可以找到好的模型：平滑模型（γ较小）可以通过增加正确分类每个点（较大的C）的重要性而变得更复杂，从而获得性能良好的模型。

当γ取一些中间值时，C非常大对应的模型性能几乎相同：不必通过强制更大的间隔来正则化，RBF核的半径本身就是一个很好的结构正则化器。

不过在实践中，较大的C意味着更少的支持向量，从而内存占用少、预测速度更快

Source: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

小结：核方法

- 核函数编码特征相似度的先验信息。
- 核函数的参数需要通过交叉验证选择或通过学习得到 。
- 非线性核有时能极大提高分类器的性能。
- 非线性核通常计算繁重（训练和预测）。

Outline

- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- 核化SVM模型
- **SVM优化求解：SMO**
- SVM回归（SVR）
- Sklearn中的SVM的API
- SVM应用案例

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：

$$\max\left(\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j\right)$$

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

$$0\le\alpha_i\le C,\qquad i=1,2,\dots,N$$

SMO动机: 坐标上升

■无约束最优化问题: $W(\alpha_1, \alpha_2, \dots, \alpha_N) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$

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

Loop until convergence

{

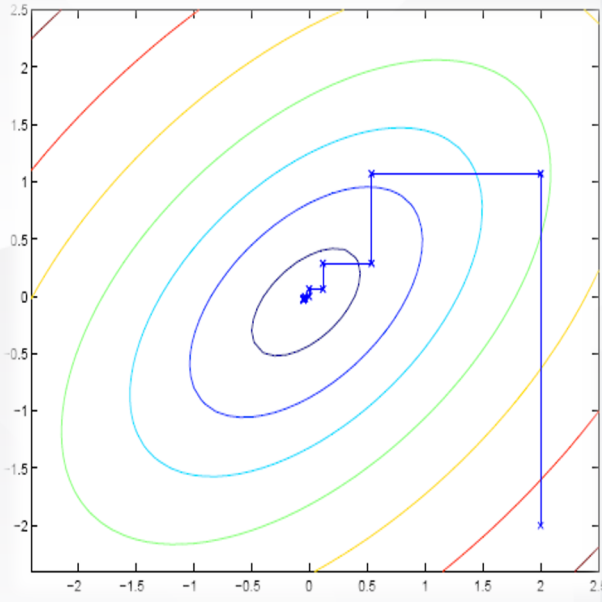
for $i = 1, 2, \dots, N$

{

$\alpha_i = \underset{\hat{\alpha}_i}{\operatorname{argmax}} W(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \dots, \alpha_N)$

}

}



■每次只关于一个参数α_i优化目标函数。

■坐标梯度上升法：每一步沿着坐标轴方向移动。

SVM坐标上升求解

■对偶问题:

$$\max\left(\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j\right)$$

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

$$0\le\alpha_i\le C,\qquad i=1,2,\dots,N$$

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

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

每次更新**两个**变量

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$W(\boldsymbol{\alpha})$
- SMO高效：更新α_i和α_j的方法计算高效！

SMO: 关于两个变量的优化方法

$$W(\alpha_1, \alpha_2, \dots, \alpha_N) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

- 假定α_1和α_2满足约束条件，固定$\alpha_3, \dots, \alpha_N$，关于变量$\alpha_1$和$\alpha_2$ 优化
  目标函数：$W(\alpha_1 , \alpha_2)$

$$W(\alpha_1, \alpha_2) = \frac{1}{2} \alpha_1^2 y_1^2 \boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_1 + \frac{1}{2} \alpha_2^2 y_2^2 \boldsymbol{x}_2^{\mathrm{T}} \boldsymbol{x}_2 + \alpha_1 \alpha_2 y_1 y_2 \boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_2 + \alpha_1 y_1 \underbrace{\sum_{i=3}^{N} \alpha_i y_i \boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_i}_{v_1} + \alpha_2 y_2 \underbrace{\sum_{i=3}^{N} \alpha_i y_i \boldsymbol{x}_2^{\mathrm{T}} \boldsymbol{x}_i}_{v_2} - \alpha_1 - \alpha_2$$

- 令$K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$

$$W(\alpha_1, \alpha_2) = \frac{1}{2} \alpha_1^2 y_1^2 K_{11} + \frac{1}{2} \alpha_2^2 y_2^2 K_{22} + \alpha_1 \alpha_2 y_1 y_2 K_{12} + \alpha_1 y_1 v_1 + \alpha_2 y_2 v_2 - \alpha_1 - \alpha_2$$

SMO: 关于两个变量的优化方法

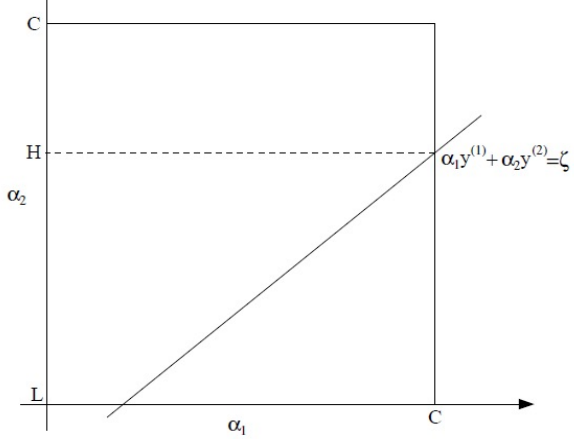
- 假定α_1和α_2满足约束条件，固定$\alpha_3, \ldots, \alpha_N$，关于变量$\alpha_1$和$\alpha_2$ 优化目标函数：$W(\alpha_1, \alpha_2)$
- 根据约束条件：$\sum_{i=1}^{N} \alpha_i y_i = 0$
- $\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i = \zeta$ 常数

$$\Rightarrow \alpha_1 = \zeta y_1 - \alpha_2 y_1 y_2$$

（两边同乘以y_1，且$y_1^2 = 1$）

有界

$$L \leq \alpha_2 \leq H$$



SMO: 关于一个变量优化

- 将 α_1写成关于α_2的等式：$\alpha_1 = \zeta y_1 - \alpha_2 y_1 y_2$
- 目标函数 $W(\boldsymbol{\alpha})$： $W(\alpha_1, \alpha_2, \ldots, \alpha_N) = W(\zeta y_1 - \alpha_2 y_1 y_2, \alpha_2)$
- 关于变量α_2的二次函数：

$$W(\alpha_2) = \frac{1}{2}(\zeta - \alpha_2 y_2)^2 K_{11} + \frac{1}{2}\alpha_2^2 K_{22} + y_2(\zeta - \alpha_2 y_2)\alpha_2 K_{12}$$
$$+(\zeta - \alpha_2 y_2)v_1 + \alpha_2 y_2 v_2 - \zeta y_1 + \alpha_2 y_1 y_2 - \alpha_2$$

- 不考虑α_2 的取值范围，可求全局最优解 $\alpha_2^{\text{new}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{K_{11} + K_{22} - 2K_{12}}$

求解α_2

$$W(\alpha_1,\alpha_2)=\frac{1}{2}K_{11}\alpha_1^2+\frac{1}{2}K_{22}\alpha_2^2+\alpha_1\alpha_2y_1y_2K_{12}+\alpha_1y_1v_1+\alpha_2y_2v_2-\alpha_1-\alpha_2+\text{常数}$$

s.t. $\alpha_1y_1+\alpha_2y_2=-\sum_{i=3}^{N}\alpha_i\,y_i=\zeta\,,\quad 0\le\alpha_i\le C,i=1,2$

其中 $v_1=\sum_{i=3}^{N}\alpha_i\,y_iK_{1i},\quad v_2=\sum_{i=3}^{N}\alpha_i\,y_iK_{2i},\quad K_{ij}=K(\boldsymbol{x}_i,\boldsymbol{x}_j)$

由$\alpha_1y_1+\alpha_2y_2=-\sum_{i=3}^{N}\alpha_i\,y_i=\zeta$得，$\alpha_1=\zeta y_1-\alpha_2y_1y_2$

$$W(\alpha_2)=\frac{1}{2}(\zeta-\alpha_2y_2)^2K_{11}+\frac{1}{2}\alpha_2^2K_{22}+y_2(\zeta-\alpha_2y_2)\alpha_2K_{12}+(\zeta-\alpha_2y_2)v_1+\alpha_2y_2v_2$$
$$-\zeta y_1+\alpha_2y_1y_2-\alpha_2+\text{常数}$$

$$\frac{\partial W(\alpha_2)}{\partial\alpha_2}=(K_{11}+K_{22}-2K_{12})\alpha_2-y_2(\zeta(K_{11}-K_{12})+v_1-v_2-y_1+y_2)=0$$

求解α_2

$$\frac{\partial W(\alpha_2)}{\partial \alpha_2} = (K_{11} + K_{22} - 2K_{12})\alpha_2 - y_2(\zeta(K_{11} - K_{12}) + v_1 - v_2 - y_1 + y_2) = 0$$

$$(K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{new}} = y_2(\zeta(K_{11} - K_{12}) + v_1 - v_2 - y_1 + y_2)$$

由于$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b$，　$y_1 \alpha_1^{old} + y_2 \alpha_2^{old} = \zeta$

定义$E_i = f(\boldsymbol{x}_i) - y_i$ 即 $E_1 = f(\boldsymbol{x}_1) - y_1$，$E_2 = f(\boldsymbol{x}_2) - y_2$

由于$v_1 = \sum_{i=3}^{N} \alpha_i y_i K_{1i}$，$v_2 = \sum_{i=3}^{N} \alpha_i y_i K_{2i}$

所以$v_1 = f(\boldsymbol{x}_1) - b - \sum_{i=1}^{2} \alpha_i^{old} y_i K_{1i}$，$v_2 = f(\boldsymbol{x}_2) - b - \sum_{i=1}^{2} \alpha_i^{old} y_i K_{2i}$

此外，$y_1 \alpha_1^{old} + y_2 \alpha_2^{old} = \zeta$，将$\zeta$，$v_1$，$v_2$代入$\frac{\partial W(\alpha_2)}{\partial \alpha_2} = 0$得

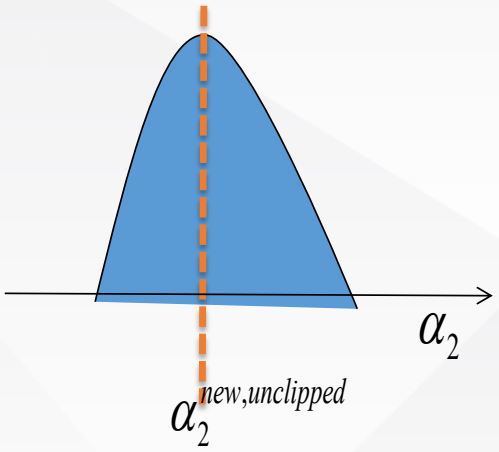
$$(K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{new}} = (K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{old}} + y_2(E_1 - E_2)$$

$$\alpha_2^{\text{new}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{K_{11} + K_{22} - 2K_{12}}$$

SMO: 关于一个变量优化

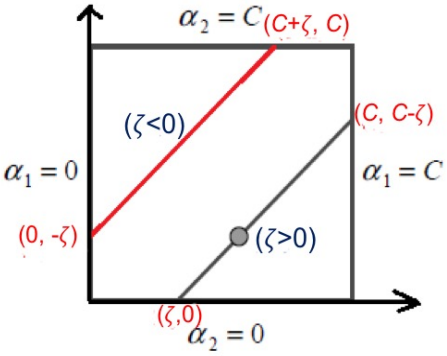
■考虑约束条件：

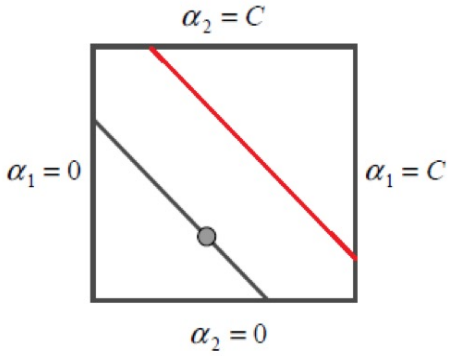
$$\alpha_2^{\text{new}} = \begin{cases} H & \alpha_2^{\text{new,unclipped}} > H \\ \alpha_2^{\text{new,unclipped}} & L \le \alpha_2^{\text{new,unclipped}} \le H \\ L & \alpha_2^{\text{new,unclipped}} < L \end{cases}$$



$$L = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$$

$$H = \min\left(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$$





$$L = \max(0, \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C)$$

$$H = \min(C, \alpha_2^{\text{old}} + \alpha_1^{\text{old}})$$

$$\alpha_1 y_1 + \alpha_2 y_2 = \zeta$$

SMO: 两个变量的最优解

■得到α_2^{new}，根据 $\alpha_1 = \zeta y_1 - \alpha_2 y_1 y_2$可得：

$$\alpha_1^{\text{old}} y_1 + \alpha_2^{\text{old}} y_2 = \alpha_1^{\text{new}} y_1 + \alpha_2^{\text{new}} y_2$$

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 \left(\alpha_2^{\text{old}} - \alpha_2^{\text{new}}\right)$$

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
  - 选择训练集中违反KKT条件最严重的样本点

- KKT 条件:

$$\begin{cases} \alpha_i = 0 & \Rightarrow y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1 \\ 0 < \alpha_i < C & \Rightarrow y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 1 \\ \alpha_i = C & \Rightarrow y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \leq 1 \end{cases}$$

- 首先选择违反$0 < \alpha_i < C \quad \Rightarrow y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 1$的点（决策边界上的支持向量）

- 如果上述支持向量都满足KKT条件，检查其他所有训练样本。

SMO: 第二个变量的选择

- 选 α_2的标准？

$$\alpha_2^{\text{new,unclipped}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta}$$

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

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

$$E_1 > 0 \Rightarrow \text{选最小的} E_i \text{作为} E_2$$

$$E_1 < 0 \Rightarrow \text{选最大的} E_i \text{作为} E_2$$

- E_i很关键，怎么样使得算法高效？

保存所有的E_i

SMO: b及对偶值 E_i

- 更新完变量后，更新b
- $\alpha_1^{\text{new}} > 0$，根据KKT条件：$y_1 f(\boldsymbol{x}_1) = y_1 \left(\sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_1) + b \right) = 1$

$$\Downarrow$$

$$b_1^{\text{new}} = y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i1} - \alpha_1^{\text{new}} y_1 K_{11} - \alpha_2^{\text{new}} y_2 K_{21}$$

- 引入：$E_1 = f(\boldsymbol{x}_1) - y_1 = \sum_{i=3}^{N} \alpha_i y_i K_{i1} + \alpha_1^{\text{old}} y_1 K_{11} + \alpha_2^{\text{old}} y_2 K_{21} + b^{\text{old}}$
- 得到 $\begin{cases} b_1^{\text{new}} = -E_1 - y_1 K_{11}\left(\alpha_1^{\text{new}} - \alpha_1^{\text{old}}\right) - y_2 K_{21}\left(\alpha_2^{\text{new}} - \alpha_2^{\text{old}}\right) + b^{\text{old}} \\ b_2^{\text{new}} = -E_2 - y_1 K_{12}(\alpha_1^{\text{new}} - \alpha_1^{\text{old}}) - y_2 K_{22}(\alpha_2^{\text{new}} - \alpha_2^{\text{old}}) + b^{\text{old}} \end{cases}$
- 如果$0 < \alpha_1^{\text{new}} < C$, $0 < \alpha_2^{\text{new}} < C$, $b^{\text{new}} = b_1^{\text{new}} = b_2^{\text{new}}$
- 如果α_1^{new}、 α_2^{new}同时是0或C ，$\forall b \in [\min\{b_1^{\text{new}}, b_2^{\text{new}}\}, \max\{b_1^{\text{new}}, b_2^{\text{new}}\}]$都满足KKT条件，取

$$b^{\text{new}} = \frac{b_1^{\text{new}} + b_2^{\text{new}}}{2}$$

- 更新 $E_i = \sum_{S} y_j \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b^{\text{new}} - y_i$

S：支持向量集合

SMO 算法

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

② 初始化误差项E_i

③ 选择待优化的变量：α_1和α_2求解优化问题的解

$$\alpha_2^{\text{new,unclipped}}=\alpha_2^{\text{old}}+\frac{y_2(E_1-E_2)}{\eta}$$

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

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

$$\alpha_1^{(t+1)}=\alpha_1^{(t)}+y_1y_2(\alpha_2^{(t)}-\alpha_2^{(t)}+1)$$

④ 更新$\boldsymbol{\alpha}$为 $\boldsymbol{\alpha}^{(\mathrm{t}+1)}$，更新 E_i，计算$b^{(t+1)}$

⑤ 如果达到**终止条件**，则停止算法；否则$t=t+1$,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。

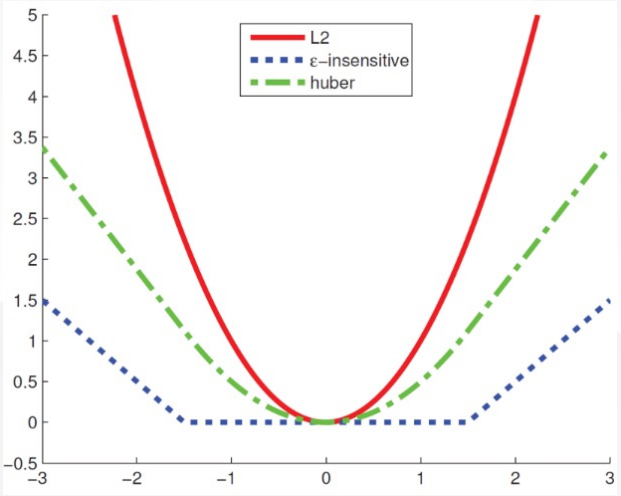
Outline

- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- 核化SVM模型
- SVM回归（SVR）
- Sklearn中的SVM的API
- SVM应用案例

ϵ不敏感损失函数

- 在之前的线性回归模型中，只有当真值y与预测$\hat{y}$完全相等时，我们才认为损失为0（L2损失、L1损失、Huber损失）。
- 在支持向量回归中，我们能容忍当真值y与预测$\hat{y}$有ϵ的偏差，即当y与$\hat{y}$与之间的差异大于ϵ时才计算损失，称为ϵ不敏感损失（ϵ insensitive loss ）。

$$L_\epsilon(y,\hat{y}) = \begin{cases} 0 & |y-\hat{y}| \le \epsilon \\ |y-\hat{y}| - \epsilon & otherwise \end{cases}$$



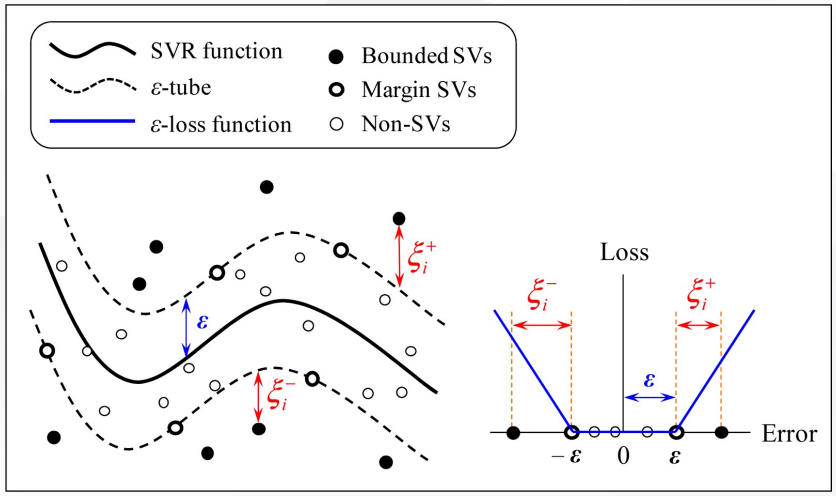
- 不惩罚小的损失
- 误差大于ϵ时，对损失的影响是线性的，对噪声鲁棒

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

- 假设回归函数为线性模型：$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$
- 同SVM分类类似， SVR的目标函数为

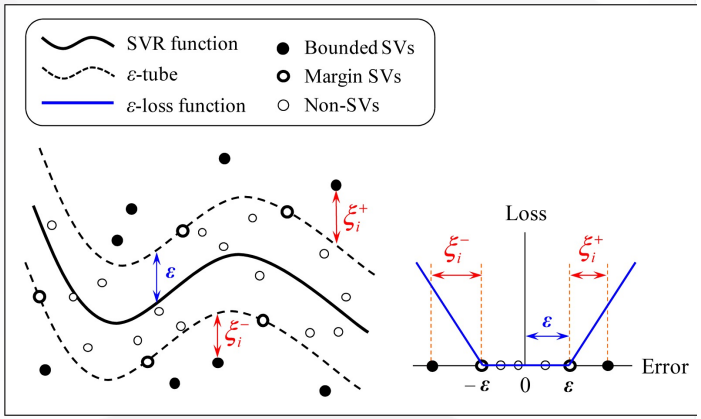
$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$\text{s.t.} \left| y_i - \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right| \le \epsilon, \qquad i = 1,2,\ldots,N$$



松弛变量

- 类似SVM分类模型，加入松弛变量$\xi_i \geq 0$，用于表示每个点在ϵ管道外的程度。
- 由于用的是绝对值，实际上是两个不等式，也就是说两边都需要松弛变量：$\xi_i^{\wedge}, \xi_i^{\vee}$。



- 则SVR模型的目标函数变为

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

$$\text{s.t.} -\epsilon - \xi_i^{\vee} \leq y_i - \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \leq \epsilon + \xi_i^{\wedge}, \qquad i = 1,2,\ldots,N$$

$$\xi_i^{\wedge} \geq 0, \qquad \xi_i^{\vee} \geq 0, \qquad i = 1,2,\ldots,N$$

拉格朗日函数

■与SVM类似，SVR的拉格朗日函数为：

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}, \boldsymbol{\xi}^{\vee}, \boldsymbol{\xi}^{\wedge}, \boldsymbol{\mu}^{\vee}, \boldsymbol{\mu}^{\wedge}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}(\xi_i^{\vee} + \xi_i^{\wedge})\Bigg)$$

$$+\sum_{i=1}^{N}\alpha_i^{\vee}\left(-\epsilon - \xi_i^{\vee} - y_i + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\Bigg)$$

$$+\sum_{i=1}^{N}\alpha_i^{\wedge}\left(y_i - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - b - \epsilon - \xi_i^{\wedge}\right)\Bigg)$$

$$-\sum_{i=1}^{N}\mu_i^{\vee}\,\xi_i^{\vee} - \sum_{i=1}^{m}\mu_i^{\wedge}\,\xi_i^{\wedge}$$

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

$$\text{s.t.}\ -\epsilon - \xi_i^{\vee} \le y_i - \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \le \epsilon + \xi_i^{\wedge}$$

$$\xi_i^{\wedge} \ge 0, \qquad \xi_i^{\vee} \ge 0$$

SVR的对偶问题

■拉格朗日函数$L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}, \boldsymbol{\xi}^{\vee}, \boldsymbol{\xi}^{\wedge}, \boldsymbol{\mu}^{\vee}, \boldsymbol{\mu}^{\wedge})$对$\mathbf{w}, b, \xi_i^{\wedge}, \xi_i^{\vee}$的一阶导数：

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}, \boldsymbol{\xi}^{\vee}, \boldsymbol{\xi}^{\wedge}, \boldsymbol{\mu}^{\vee}, \boldsymbol{\mu}^{\wedge})}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N} (\alpha_i^{\wedge} - \alpha_i^{\vee})\, \boldsymbol{x}_i$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}, \boldsymbol{\xi}^{\vee}, \boldsymbol{\xi}^{\wedge}, \boldsymbol{\mu}^{\vee}, \boldsymbol{\mu}^{\wedge})}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} (\alpha_i^{\wedge} - \alpha_i^{\vee}) = 0$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}, \boldsymbol{\xi}^{\vee}, \boldsymbol{\xi}^{\wedge}, \boldsymbol{\mu}^{\vee}, \boldsymbol{\mu}^{\wedge})}{\partial \xi_i^{\wedge}} = 0 \Rightarrow C - \alpha_i^{\wedge} + \mu_i^{\wedge}$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}, \boldsymbol{\xi}^{\vee}, \boldsymbol{\xi}^{\wedge}, \boldsymbol{\mu}^{\vee}, \boldsymbol{\mu}^{\wedge})}{\partial \xi_i^{\vee}} = 0 \Rightarrow C - \alpha_i^{\vee} + \mu_i^{\vee}$$

SVR的对偶问题

■将上述结论代入拉格朗日函数 $L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}, \boldsymbol{\xi}^{\vee}, \boldsymbol{\xi}^{\wedge}, \boldsymbol{\mu}^{\vee}, \boldsymbol{\mu}^{\wedge})$ ，得到对偶问题：

$$\max_{\boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}} \left(-\sum_{i=1}^{N} \left((\epsilon - y_i)\alpha_i^{\wedge} + (\epsilon + y_i)\alpha_i^{\vee} \right) - \sum_{i=1}^{N} \frac{1}{2} (\alpha_i^{\wedge} - \alpha_i^{\vee})(\alpha_j^{\wedge} - \alpha_j^{\vee}) \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \right)$$

$$\text{s.t.} \sum_{i=1}^{N} (\alpha_i^{\wedge} - \alpha_i^{\vee}) = 0$$

$$0 \le \alpha_i^{\wedge} \le C, \qquad i = 1,2, \dots, N$$

$$0 \le \alpha_i^{\vee} \le C, \qquad i = 1,2, \dots, N$$

■去掉负号，将上述目标函数中的max换成min，得到等价问题：

$$\min_{\boldsymbol{\alpha}^{\vee}, \boldsymbol{\alpha}^{\wedge}} \left(\sum_{i=1}^{N} \left((\epsilon - y_i)\alpha_i^{\wedge} + (\epsilon + y_i)\alpha_i^{\vee} \right) + \sum_{i=1}^{N} \frac{1}{2} (\alpha_i^{\wedge} - \alpha_i^{\vee})(\alpha_j^{\wedge} - \alpha_j^{\vee}) \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \right)$$

SVR模型

■求得α的最优值后，可计算$\boldsymbol{w}, b$ ，从而得到SVR模型

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = \sum_{i=1}^{N}(\alpha_i^{\wedge} - \alpha_i^{\vee})\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x} + b$$

$$= \sum_{i=1}^{N}(\alpha_i^{\wedge} - \alpha_i^{\vee})\langle \boldsymbol{x}_i, \boldsymbol{x}\rangle + b$$

■ $\boldsymbol{x}$ 与训练数据的点积$\langle \boldsymbol{x}_i, \boldsymbol{x}\rangle$换成核函数$k(\boldsymbol{x}_i, \boldsymbol{x})$，得到核化SVR模型

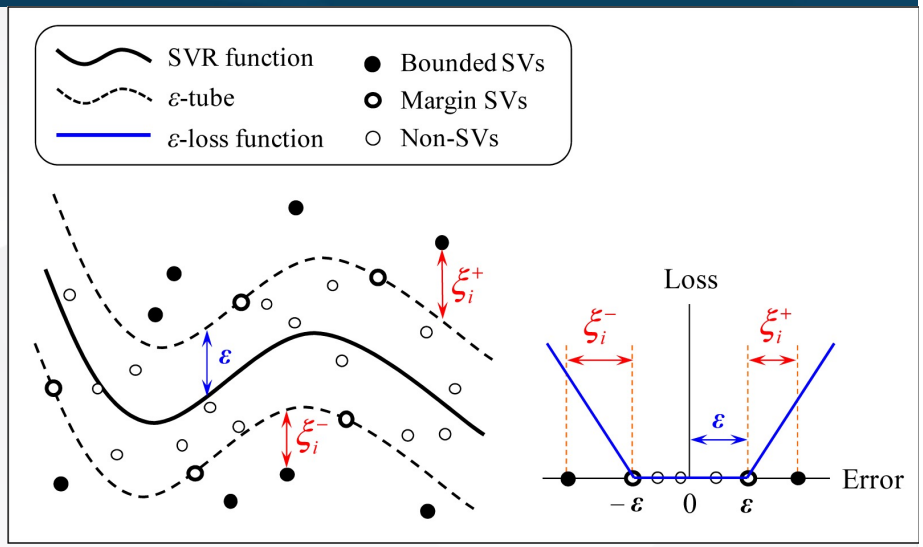
$$f(\boldsymbol{x}) = \sum_{i=1}^{N}(\alpha_i^{\wedge} - \alpha_i^{\vee})k(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

SVR的支持向量

■KKT条件之二为：

$$\alpha_i^{\vee}\left(\epsilon+\xi_i^{\vee}+y_i-\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\right)=0$$

$$\alpha_i^{\wedge}\left(\epsilon+\xi_i^{\wedge}-y_i+\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\right)=0$$



支持向量：

仅当样本$(\boldsymbol{x}_i, y_i)$落在ϵ间隔中，$\alpha_i^{\vee}$、$\alpha_i^{\wedge}$才取非0值。

Outline

- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- 核化SVM模型
- SVM回归（SVR）
- **Sklearn中的SVM的API**
- SVM应用案例

Scikit-Learn中的SVM实现

- sklearn.svm模块提供支持向量机算法，可用于分类、回归和异常值检测。
- 分类实现有三种方式
  - LinearSVC基于liblinear实现线性SVM ，比基于libsvm实现的线性SVC / NuSVC更快，同时可采用更多正则选择（L1/L2）和损失函数选择
    - L1正则可以得到特征系数稀疏的效果
    - 适用于样本数更多的情况
  - SVC和NuSVC类似，都是基于libsvm实现的C-SVM，二者在参数方面有细微不同（ NuSVC 有参数*nu*控制训练误差的上限和支持向量的下限）
  - SGDClassifier（不在sklearn.svm模块，在sklearn.linear_model ）实现了基于随机梯度下的线性SVM分类。
- 回归实现方式同分类类似。
- SVM还支持非监督的异常值检测： OneClassSVM

LinearSVC

class sklearn.svm.LinearSVC(*penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1000*)

- 支持两种正则：L2正则和L1正则，正则参数为：penalty、C
- 支持两种损失函数： 标准的合页损失 'hinge' 和合页损失的平方 'squared_hinge'
- 支持不同类别的样本权重设置：class_weight

参数	说明	备注
penalty	惩罚函数 / 正则函数，支持L2正则和L1正则，默认：L2	
loss	损失函数，支持标准的合页损失 ‘hinge’ 和合页损失的平方 ‘squared_hinge’ 。	
dual	原问题（primal）还是对偶问题求解。默认：True	当样本数n_samples>特征数目n_features时，原问题求解更简单。
tol	迭代终止判据的误差范围。默认:1e-4	
C	损失函数项的系数。默认：1	
multi_class	多类分类处理策略，可为 ‘ovr’，‘crammer_singer’ 。 ‘ovr’ 为1对多，将多类分类转化为多个两类分类问题， ‘crammer_singer’ 是一起优化多个分类的目标函数。缺省： ‘ovr’	‘crammer_singer’ 理论上看起来很好，实际上很少用，因为分类效果没有更好但计算量大。

参数	说明	备注
fit_intercept	是否在决策函数中加入截距项。缺省：True	如果数据已经中心化，可以不用。
intercept_scaling	截距缩放因子，当fit_intercept为True且时，输入为[x, self.intercept_scaling]，即对输入特征加入1维常数项	增加的常数项系数也受到l1/l2 正则的惩罚，所以要适当增大常数项。
class_weight	不同类别样本的权重，用户指定每类样本权重或‘balanced’（每类样本权重与该类样本出现比例成反比）。缺省：None	在损失计算中，对不同类别的样本施加相应的权重。
verbose	是否详细输出	
random_state	随机种子。如果随机种子相同，每次洗牌得到的结果一样。可设置为某个整数	
max_iter	最大迭代次数。缺省：1000	

LinearSVC的属性

属性	说明	备注
coef_	回归系数/权重，与特征维数相同。	
intercept_	截距项。	

LinearSVC的方法

方法	说明
fit(X, y[, sample_weight])	模型训练。参数X, y为训练数据，也可以通过sample_weight设置每个样本的权重。
predict(X)	返回X对应的预测值（类别标签）
decision_function(X)	预测的置信度（样本到分类超平面的带符号距离）
score(X, y[, sample_weight])	评估模型预测性能，返回模型预测的正确率。
densify()	如果之前将系数矩阵变成了稀疏模式，再将其变回稠密模式（fit函数的格式）
sparsify()	将系数矩阵变成了稀疏模式

注意：LinearSVC不能像Logistic回归那样得到每个类别的概率。

SVC

class sklearn.svm.SVC(*C=1.0*, *kernel='rbf'*, *degree=3*, *gamma='auto'*, *coef0=0.0*, *shrinking=True*, *probability=False*, *tol=0.001*, *cache_size=200*, *class_weight=None*, *verbose=False*, *max_iter= -1*, *decision_function_shape='ovr'*, *random_state=None*)

- 支持L2正则
- 支持多种核函数： 'linear'， 'poly'， 'rbf'， 'sigmoid'， 'precomputed' 和自定义核函数
- 支持多类分类实现：一对一 'ovo'
- 支持不同类别的样本权重设置：*class_weight*

参数	说明	备注
C	损失函数项的系数。默认：1	
kernel	核函数。支持 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. 默认： ''rbf' 。	
degree	多项式核的多项式阶数	
gamma	核函数（'poly', 'rbf', 'sigmoid'）系数。	默认'auto' =$1/D$
coef0	核函数（'poly'， 'sigmoid'）参数。	
shrinking	是否收缩	
probability	是否支持概率估计。	支持概率估计会降低速度，需在模型训练（fit）之前设置。
tol	迭代终止判据的误差范围。默认:1e-3	
cache_size	核的cache的大小（单位：MB）	

参数	说明	备注
class_weight	不同类别样本的权重，用户指定每类样本权重或‘balanced’（每类样本权重与该类样本出现比例成反比）。缺省：None	在损失计算中，对不同类别的样本施加相应的权重。
verbose	是否详细输出	
max_iter	最大迭代次数。缺省：-1，没有限制	
decision_function_shape	决策函数的形式，可为‘ovo’，‘ovr’。‘ovr’为1对多，将多类分类转化为多个两类分类问题。‘ovo’为libsvm原始决策函数形式，1对1，每两个类之间有一个分类面。缺省：‘ovr’	
random_state	随机种子。如果随机种子相同，每次洗牌得到的结果一样。可设置为某个整数。	用于概率估计的数据重排时的伪随机数生成器的种子

SVC的属性

属性	说明	备注
coef_	原问题的系数/权重（线性核有效），与特征维数相同。	$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b$
intercept_	截距项。	
support_	支持向量的索引	
support_vectors_	支持向量	
n_support_	每个类的支持向量的数目	
dual_coef_	对偶系数α	

注意：LinearSVC没有支持向量等属性。

SVC的常用方法

方法	说明
fit(X, y[, sample_weight])	模型训练。参数X, y为训练数据，也可以通过sample_weight设置每个样本的权重。
predict(X)	返回X对应的预测值（类别标签）
decision_function(X)	预测的置信度（样本到分类超平面的带符号距离）
score(X, y[, sample_weight])	评估模型预测性能，返回模型预测的正确率。
predict_log_proba(X)	返回X对应的预测值（每个类别对应的概率的log值）
predict_proba(X)	返回X对应的预测值（每个类别对应的概率）

注意：如果参数probability被设为True，则可以得到属于每个类别的概率估计（predict_proba和predict_log_proba方法可用）

核函数参数

- 核函数的参数有degree(M)、gamma(γ)、coef0(θ)，核函数可以有以下几种选择
  - 线性核：$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}'$
  - 多项式核，缺省为3阶多项式：$k(\boldsymbol{x}, \boldsymbol{x}') = \left(\gamma \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}' + r\right)^{M}$
  - 径向基函数（RBF）：$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2)$
  - sigmoid函数：$k(\boldsymbol{x}, \boldsymbol{x}') = \tanh\left(-\gamma \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}' + \theta\right)$

多类别分类

- SVM最初是处理二分类问题的。多类别分类通过一对一（One vs. One, ‘ovo’ ）方案实现（SVC、NuSVC）
- 一对一法：假设进行C个类别的分类任务，一对一法在任意两类样本之间设计一个SVM，构造$C\times(C-1)/2$个分类器。当对一个未知样本进行分类时，最后得票最多的类别即为该样本的类别。
- 为了提供一个和其它分类器一致的接口，参数decision_function_shape允许调用者将所有“一对一”分类器的结果聚合进一个(n_samples, n_classes)的决策函数。
- LinearSVC实现 “一对多”分类法（ ‘ovr’ ），训练C个模型。

得分与概率

- SVC中的decision_function方法对每个样本都会给出在各个类别上的分数（在两类分类问题中，对每个样本只给出一个分数）。
- 如果构造函数的参数probability被设为True，则可以得到属于每个类别的概率估计（通过predict_proba和predict_log_proba方法）。概率使用Platt缩放进行调整，通过在训练集上做额外的交叉检验来拟合一个在SVM分数上的Logistic回归。
  - Platt缩放中的交叉检验在大数据集上是一个代价很高的操作
  - 概率估计与实际得分可能会不一致，即使得分取得了最大值，概率并不一定也能取到最大值
  - Platt的方法在理论上也有一些问题
  - 如果需要拿到置信分数，而这些分数又不一定非得是概率，则建议把probability置为False，并且使用decision_function，而不是predict_proba

计算复杂度

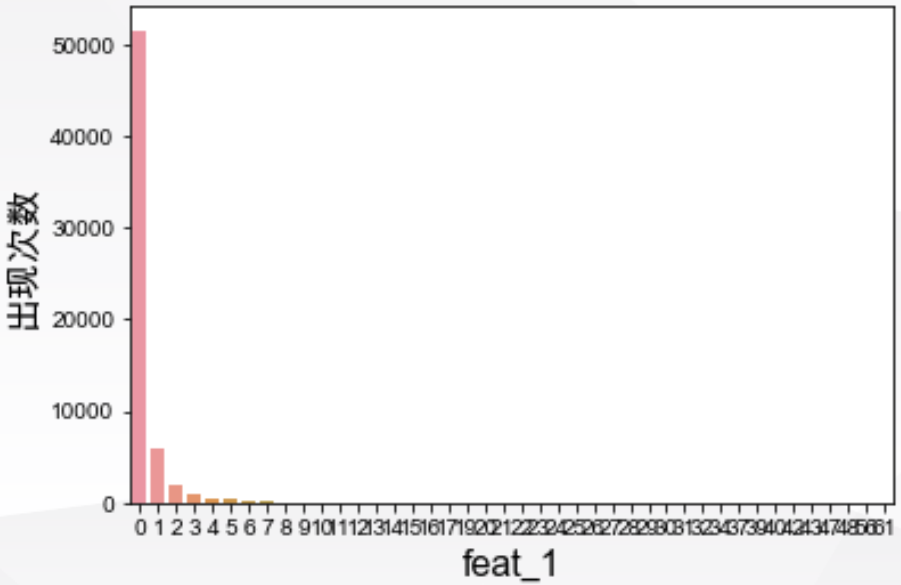
- 支持向量机很强大，一度是处理机器学习任务的首选算法。
- 但随着训练样本数目的增加，它们对计算和存储资源的需求也快速增长
  - SVM的核心是二次规划问题（QP）。令特征维数为D，训练样本数目为N，取决于libsvm缓存在实践中使用的效率（依赖于数据集），基于libsvm的实现所用的QP求解器时间复杂度在DN^2到DN^3不等。如果数据非常稀疏，D为一个样本中的平均非零特征数目。因此通常在$N < 10000$时可用。
- 对线性的情况，基于liblinear实现的LinearSVC的算法比基于libsvm实现的SVC效率要高很多，liblinear在处理以百万计的样本和/或特征时仅以线性增长。

Outline

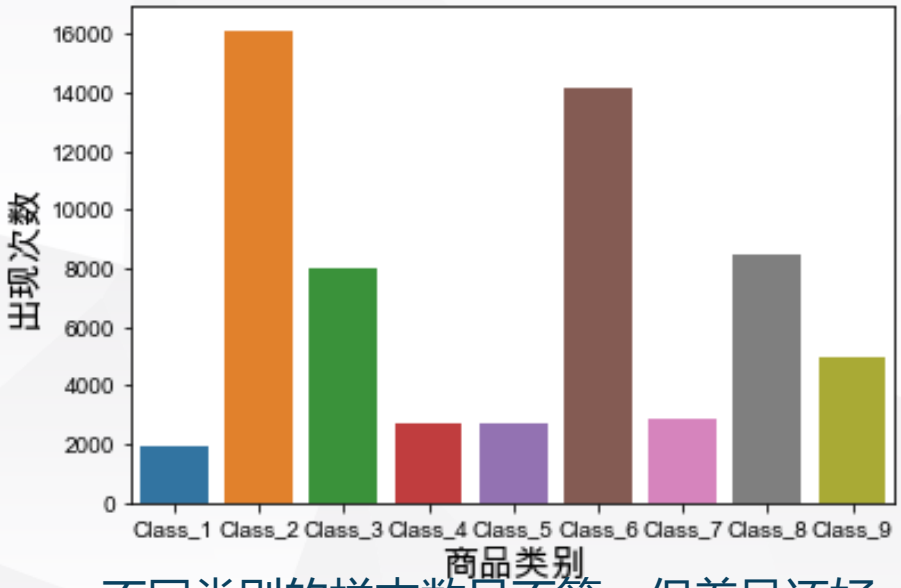
- 线性SVM分类原理
- 带松弛变量的SVM
- SVM的对偶问题
- 核化SVM模型
- SVM回归（SVR）
- Sklearn中的SVM的API
- SVM应用案例
  - SVC：Otto商品分类
  - SVR：共享单车骑行量预测

Otto商品分类——数据分析

- 训练集 61,878个样本，测试集144,368个样本
  - 每个样本有93个数值型特征： 表示某种事件发生的次数
  - 标签：9种商品类别



特征稀疏：大部分样本的特征值为0



不同类别的样本数目不等，但差异还好（不少于0.1倍）

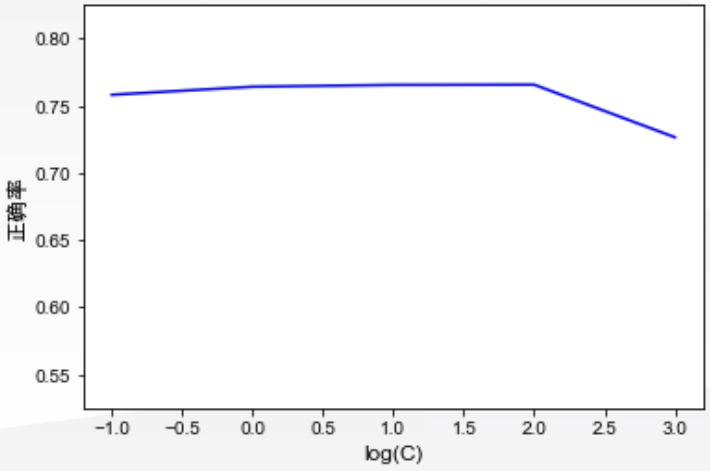
Otto商品分类——特征工程

- 特征均为数值型特征
  - 计数：可以考虑log变换
  - TF-IDF：事件计数可类比文档分类任务中的词频特征，可进一步更有判别力的（Term Frequency–Inverse Document Frequency，TF-IDF）
  - 标准化处理

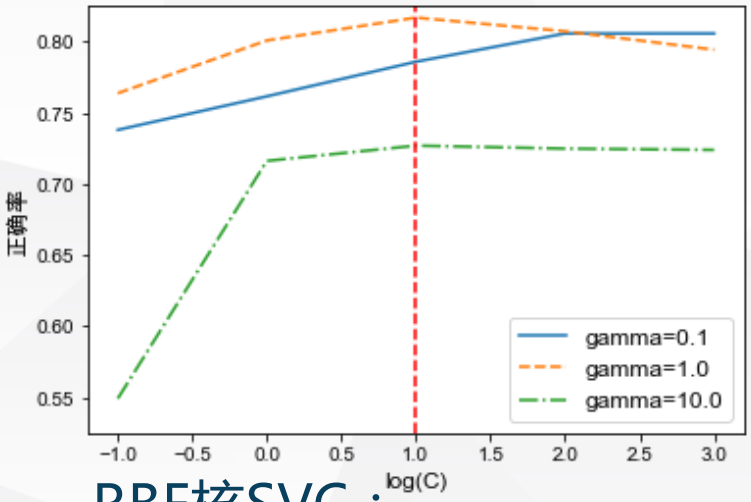
Otto商品分类——模型选择

- 正则函数：L1正则、L2正则
- 正则超参数C

- GridSearchCV （3折交叉验证）
- 评价指标：正确率（SVC默认是不支持概率输出）



线性SVC：
L2，当$C = 100$时性能最好



RBF核SVC：
当$C = 10$，$\text{gamma} = 1$时性能最好
远超线性模型

Otto商品分类——模型性能

■测试数据上的性能（Logloss）

特征编码方案	Logistic回归	RBF核SVM
TF-IDF特征	0.63319	0.48877

SVM模型的性能远比线性Logistic回归在该任务上的性能要好
当然还有提升空间

共享单车骑行量预测

- 数据分析： 1_EDA_BikeSharing.ipynb
  - 两年每天的骑行量以及当天的天气情况
  - 731个样本，没有缺失值
  - 每个样本有12个特征：

    时间信息：年、季节、月份、日期、星期、是否为工作日、是否为节假日

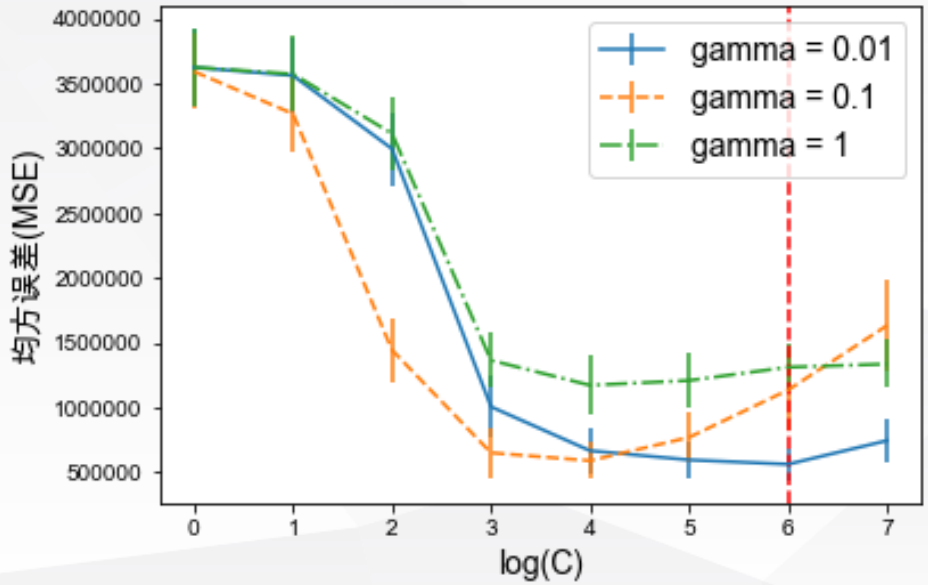
    天气情况：天气概况（晴、阴、小雨雪、大雨雪）、湿度、温度、体感温度、风速
  - 标签：骑行量
  - 有些特征相关性很高（季节与月份、温度与体感温度），冗余大

共享单车骑行量预测

- 特征工程： 2_FE_Advertising.ipynb
  - 数值型特征做标准化处理：湿度、温度、体感温度、风速特征
  - 类别型特征取值均不多，独热编码：季节、月份、星期、天气概况
  - 二值特征编码为0/1：年份、是否为工作日、是否为节假日
  - 每个日期只有2个样本，所以去掉特征“日期”

共享单车骑行量预测

- RBF核SVM
- 超参数调优评价指标：MSE（红色竖线为最佳超参数）



共享单车骑行量预测

最小二乘线性回归	岭回归	Lasso	RBF核SVM	
			$\gamma = 0.01, C = 10^6$	$\gamma = 0.01, C = 10^6$
785.595792	776.975361	784.878890	689.787410	670.161404

RBF核的SVM的性能远高于线性回归。

本章总结

- SVM
  - 函数集合：输入特征的线性组合
  - 目标函数

    损失函数： 合页损失

    正则项：L2正则、L1正则
  - 优化方法：SMO（略）
- Sklearn中的SVC/SVR回归实现
- 对偶问题 & 核方法

SVM的优点

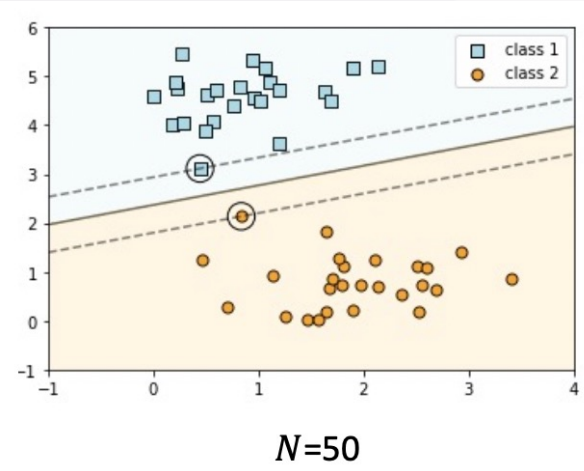
- SVM依赖于相对较少的支持向量
  - SVM是非常紧凑的模型，占用的内存非常少
- SVM适合高维数据
  - 分类器的复杂性由支持向量的数目而不是数据的维度来表征
  - SVM能够很好地处理高维数据，甚至可以处理比样本更多的维度
  - 支持向量的数目可以用于计算软间隔SVM分类器的预期错误率的上界，该阈值与数据维度无关
- 一旦对模型进行了训练，预测阶段就非常快

SVM的缺点

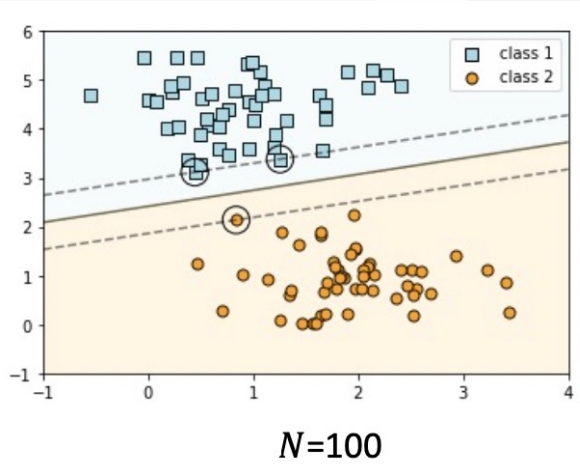
- 对于大量的训练样本，计算成本可能过高
- 结果没有直接的概率解释

小练习1: 硬间隔SVM

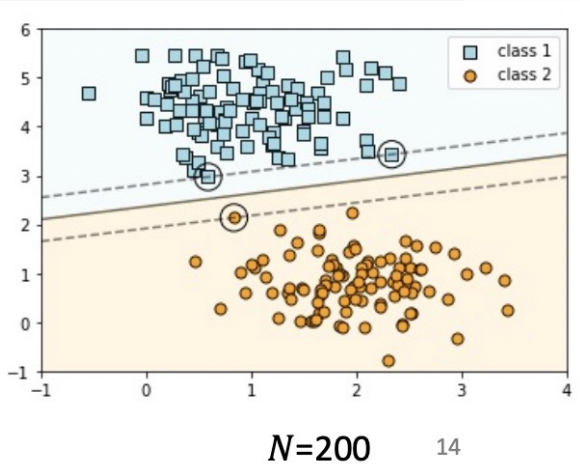
- 决策边界只与支持向量的位置有关（ √ ）
- 离间隔更远的正确一侧的其他点不会改变决策边界（ √ ）



N=50



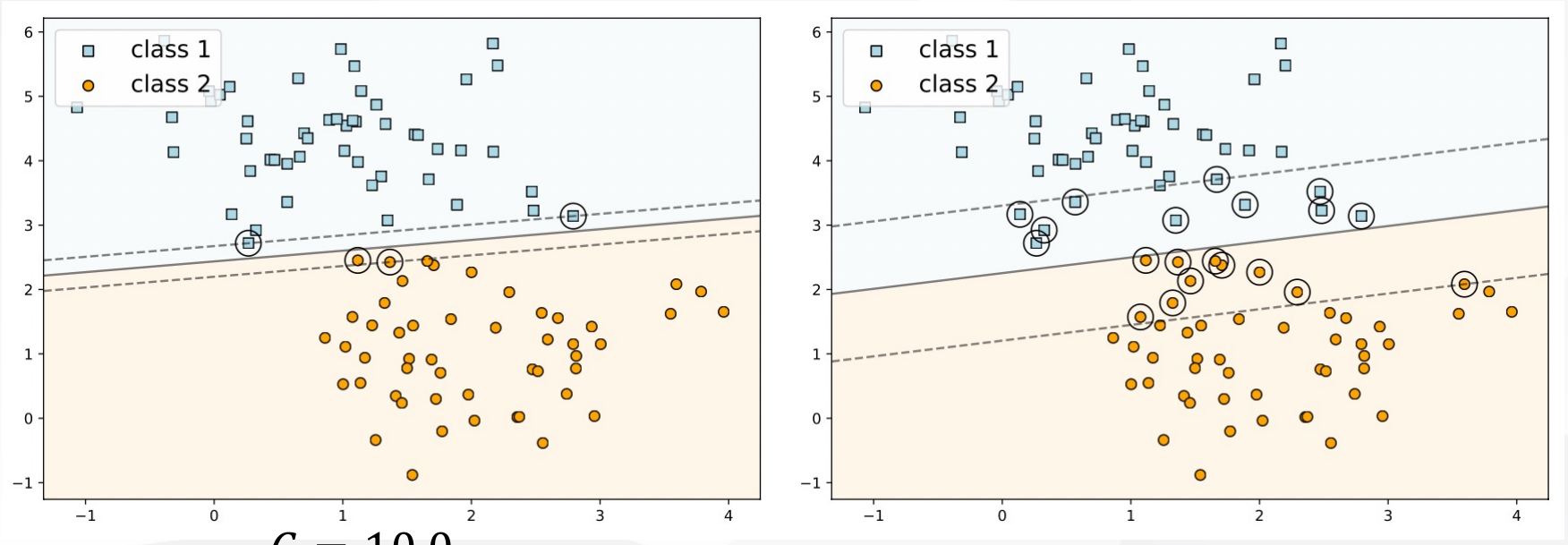
N=100



N=200

小练习2: 软间隔SVM

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



$C = 10.0$

C较大，
错误分类误差容忍低，
较小的间隔

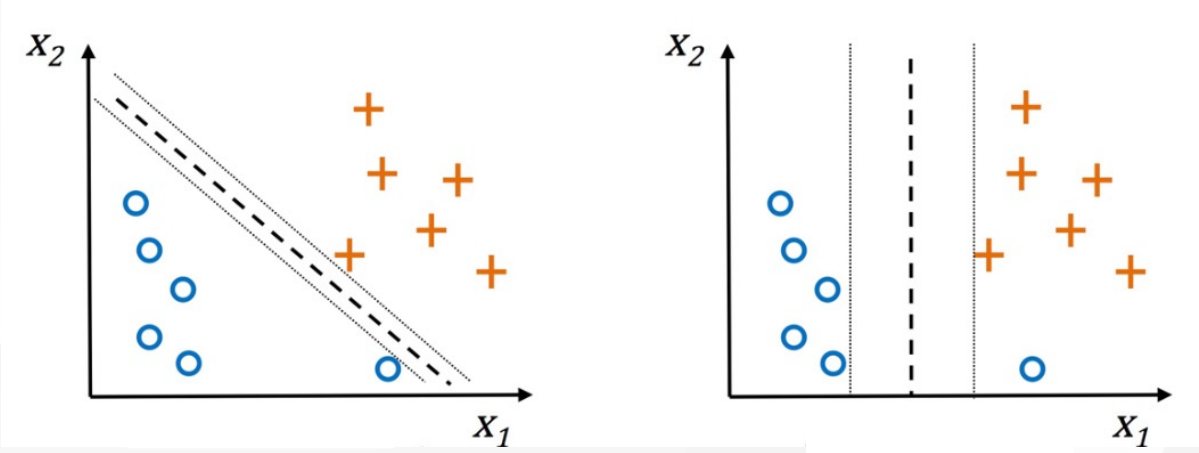
$C = 0.1$

C较小，
错误分类误差的容忍高，
较大的间隔

小练习3: : 硬间隔SVM vs. 软间隔SVM

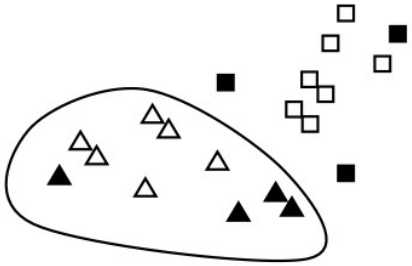
- 给定线性可分的数据集，我们训练一个硬间隔SVM，发现间隔太小，以至于模型容易过拟合。你如何解决过拟合问题？

- 我们可以通过使用软间隔SVM来选择更大的间隔，以使得模型泛化性能更好。

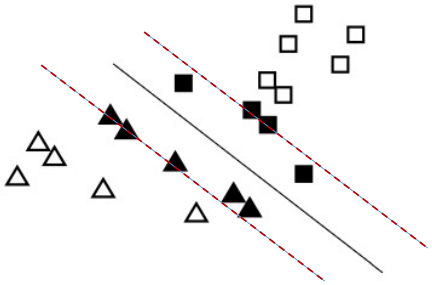


C很大（$C \to \infty$）　　C很小

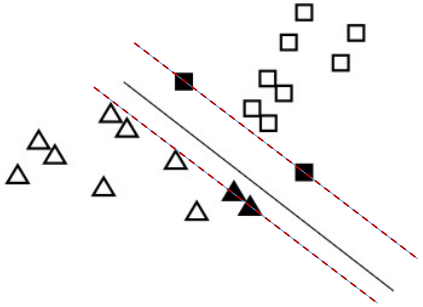
小练习4: 核SVM



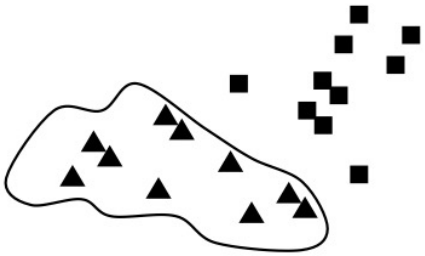
RBF核，C较小
模型更简单
决策边界更平滑



线性，C较小
间隔大
支持向量多



线性，C较大
间隔小
支持向量少



RBF核，C较大
模型更复杂
决策边界更曲折