



第6章 支持向量机



-----• 中国矿业大学 计算机科学与技术学院 •-----



主要内容

1. 间隔与支持向量
2. 对偶问题
3. 核函数
4. 软间隔与正则化
5. 支持向量回归
6. 核方法



SVM 与统计学习简史

- 1963: Vapnik 提出支持向量的概念
- 1968: Vapnik 和 Chervonenkis 提出 VC 维
- 1974: 提出结构风险最小化原则

苏联解体前一年(1990), Vapnik 来到美国

- 1995: Support Vector Network 文章发表

“Nothing is more practical than a good theory”

-- V. Vapnik

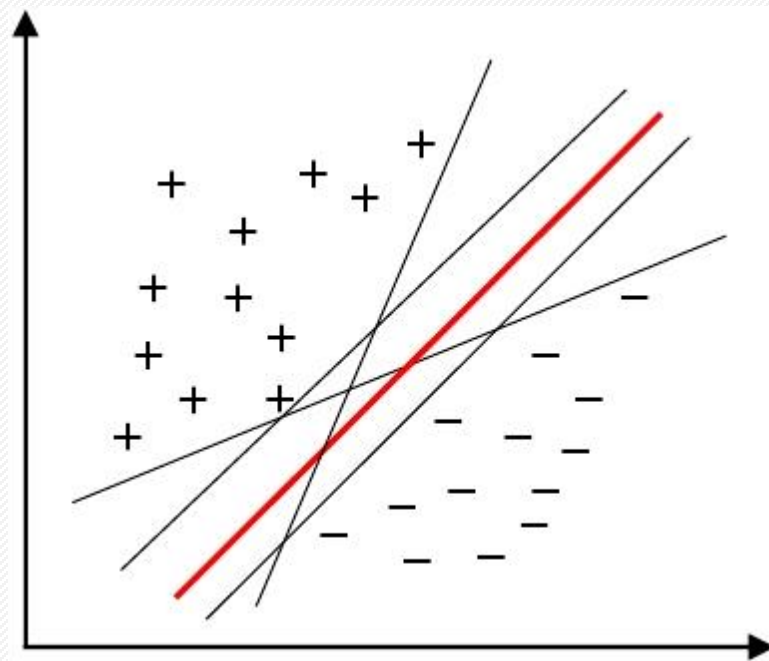
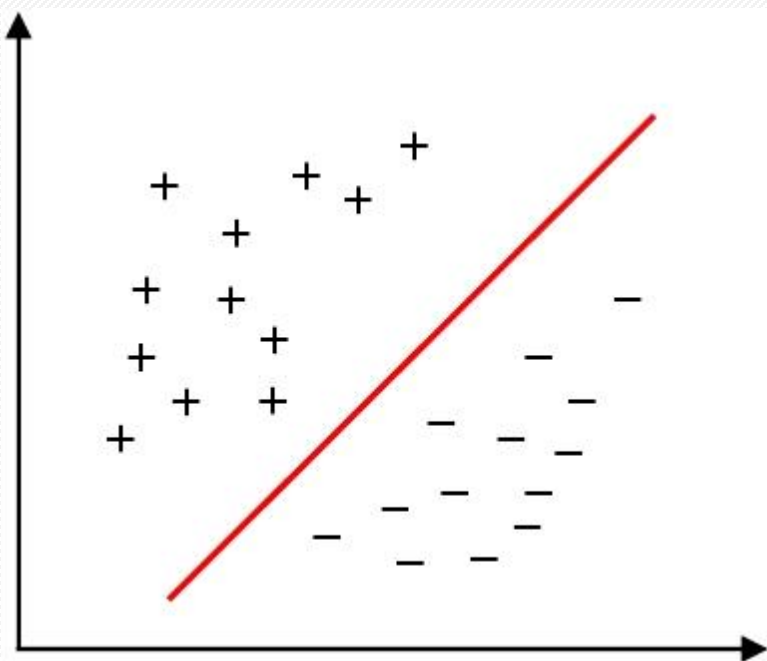
- 1995: 《The Nature of Statistical Learning》出版
- 1998: SVM 在文本分类上取得巨大成功
- 1998: 《Statistical Learning Theory》出版





6.1 线性可分支持向量机与硬间隔最大化

- 假定训练样本集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$
- 在样本空间中寻找一个超平面，将不同类别的样本分开
- 将训练样本分开的超平面可能有很多，哪一个更好呢？



“ 正中间 ” 的划分超平面：鲁棒性最好，泛化能力最强

6.1.1 线性可分支持向量机

SVM的思想：不仅让样本点被分割超平面分开，还希望那些离分割超平面最近的点到分割超平面的距离最大。

超平面方程： $w \cdot x + b = 0$

其中 $w = (w_1; w_2; \dots; w_d)$ 为法向量，决定了超平面的方向。法向量指向的一侧为正类，另一侧为负类。 b 为偏移项（截距），决定了超平面与原点之间的距离。

定义6.1 (线性可分支持向量机)给定线性可分训练数据集，通过间隔最大化或等价地求解相应的凸二次规划问题学习得到的分离超平面为

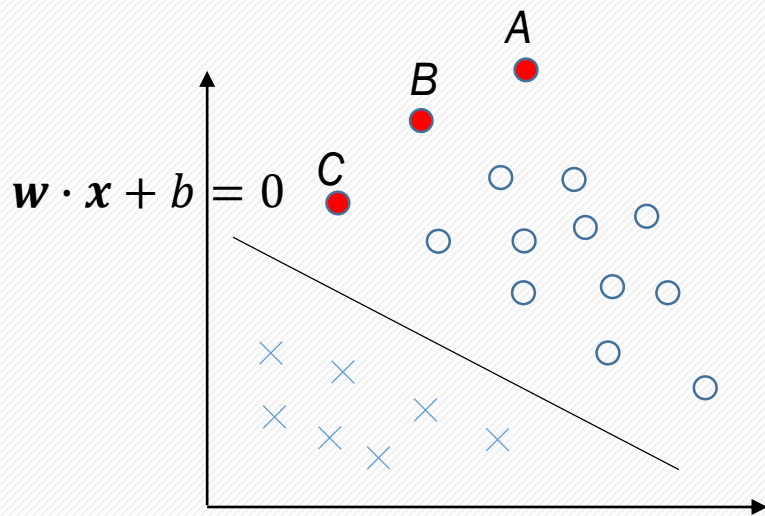
$$w^* \cdot x + b^* = 0 \quad (6.1)$$

以及相应的分类决策函数

$$f(x) = \text{sign}(w^* \cdot x + b^*) \quad (6.2)$$

称为线性可分支持向量机。

6.1.2 函数间隔与几何间隔



二类分类问题

点A距分离超平面较远，若预测该点为正类，就比较确信预测是正确的；
点C距分离超平面较近，若预测该点为正类就不那么确信；
点B介于点A与点C之间，预测其为正类的确信度也在A与C之间。

用 $y(w \cdot x + b)$ 表示分类的正确性及确信度。

6.1.2 函数间隔与几何间隔

定义6.2 (函数间隔) 对于给定的训练数据集 T 和超平面 (\mathbf{w}, b) ，定义超平面 (\mathbf{w}, b) 关于样本点 (\mathbf{x}_i, y_i) 的函数间隔为

$$\hat{\gamma}_i = y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \quad (6.3)$$

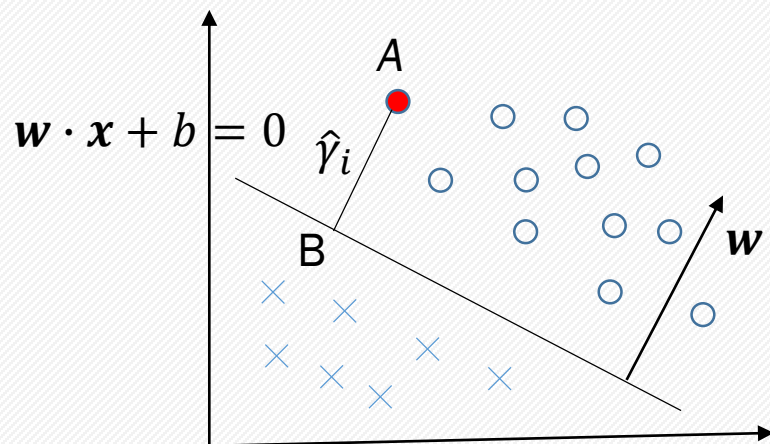
定义超平面 (\mathbf{w}, b) 关于训练数据集 T 的函数间隔为超平面 (\mathbf{w}, b) 关于 T 中所有样本点 (\mathbf{x}_i, y_i) 的函数间隔之最小值，即

$$\hat{\gamma} = \min_{i=1, \dots, N} \hat{\gamma}_i \quad (6.4)$$

只有函数间隔还不够。因为只要成比例地改变 \mathbf{w} 和 b ，例如，将它们改为 $2\mathbf{w}$ 和 $2b$ ，超平面并没有改变，但函数间隔却成为原来的两倍。

对分离超平面的法向量 \mathbf{w} 加某些约束，如规范化， $\|\mathbf{w}\| = 1$ ，使得间隔是确定的。这时函数间隔成为几何间隔(geometric margin)。

6.1.2 函数间隔与几何间隔



几何间隔

A在超平面正的一侧:
$$\hat{\gamma}_i = \frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|}$$

A在超平面负的一侧:
$$\hat{\gamma}_i = -\left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|}\right)$$

一般地, 点 \mathbf{x}_i 与超平面 (\mathbf{w}, b) 的距离:
$$\gamma_i = y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right)$$

6.1.2 函数间隔与几何间隔

定义6.3 (几何间隔) 对于给定的训练数据集 T 和超平面 (\mathbf{w}, b) , 定义超平面 (\mathbf{w}, b) 关于样本点 (\mathbf{x}_i, y_i) 的几何间隔为

$$\gamma_i = y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right) \quad (6.5)$$

定义超平面 (\mathbf{w}, b) 关于训练数据集 T 的几何间隔为超平面 (\mathbf{w}, b) 关于 T 中所有样本点 (\mathbf{x}_i, y_i) 的几何间隔之最小值, 即

$$\gamma = \min_{i=1, \dots, N} \gamma_i \quad (6.6)$$

函数间隔和几何间隔的关系

$$\gamma_i = \frac{\hat{y}_i}{\|\mathbf{w}\|} \quad (6.7)$$

$$\gamma = \frac{\hat{\gamma}}{\|\mathbf{w}\|} \quad (6.8)$$



6.1.3 间隔最大化

1. 最大间隔分离超平面

$$\max_{\mathbf{w}, b} \gamma$$

(6.9)

$$\text{s. t. } y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right) \geq \gamma, \quad i = 1, 2, \dots, N$$

(6.10)

$$\gamma = \frac{\hat{\gamma}}{\|\mathbf{w}\|}$$

(6.8)

改写为

$$\max_{\mathbf{w}, b} \frac{\hat{\gamma}}{\|\mathbf{w}\|}$$

(6.11)

$$\text{s. t. } y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq \hat{\gamma}, \quad i = 1, 2, \dots, N$$

(6.12)

函数间隔 $\hat{\gamma}$ 的取值并不影响最优化问题的解

$$\max_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|}$$

$$\text{s. t. } y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N$$



6.1.3 间隔最大化

等价于

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (6.13)$$

$$\text{s. t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N \quad (6.14)$$

凸优化问题是指约束最优化问题

$$\min_{\mathbf{w}} f(\mathbf{w}) \quad (6.15)$$

$$\text{s. t. } g_i(\mathbf{w}) \leq 0, i = 1, \dots, k, \quad (6.16)$$

$$h_i(\mathbf{w}) \leq 0, i = 1, \dots, l, \quad (6.17)$$

其中，目标函数 $f(\mathbf{w})$ 和约束函数 $g_i(\mathbf{w})$ 都是 \mathbb{R}^n 上的连续可微的凸函数。约束函数 $h_i(\mathbf{w})$ 是 \mathbb{R}^n 上的仿射函数。

当目标函数 $f(\mathbf{w})$ 是二次函数且约束函数 $g_i(\mathbf{w})$ 是仿射函数时，上述凸优化问题是凸二次规划问题。



6.1.3 间隔最大化

算法6.1 线性可分支持向量机——最大间隔法

输入：线性可分训练数据集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

输出：最大间隔超平面和分类决策函数

(1) 构造并求解约束最优化问题

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

$$\text{s. t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N$$

求得最优解 \mathbf{w}^*, b^*

(2) 由此得到分离超平面：

$$\mathbf{w}^* \cdot \mathbf{x} + b^* = 0$$

以及相应的分类决策函数

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^* \cdot \mathbf{x} + b^*)$$

定理6.1 (最大间隔分离超平面的存在唯一性)
若训练数据集 T 线性可分，则可将训练数据集中的样本点完全正确分开的最大间隔分离超平面存在且唯一。

3. 支持向量和间隔边界

在线性可分情况下，训练数据集的样本点中与分离超平面距离最近的样本点称为**支持向量**。支持向量是使约束条件是(6.14)等号成立的点，即

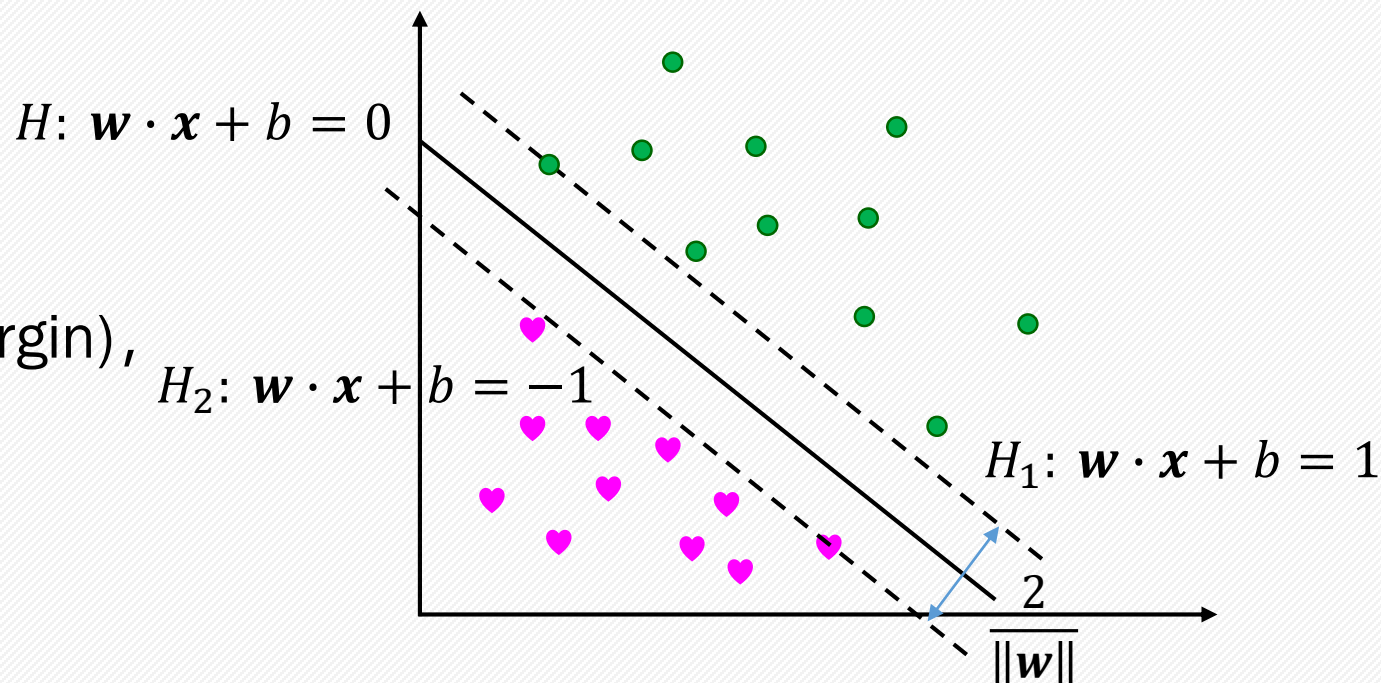
对于 $y_i = +1$ 的正例点，支持向量在超平面 H_1 上。

$$H_1: \mathbf{w} \cdot \mathbf{x} + b = 1$$

对于 $y_i = -1$ 的负例点，支持向量在超平面 H_2 上。

$$H_2: \mathbf{w} \cdot \mathbf{x} + b = -1$$

H_1 与 H_2 之间的距离 $\frac{2}{\|\mathbf{w}\|}$ 称为**间隔**(margin),
 H_1 和 H_2 称为**间隔边界**。



3. 支持向量和间隔边界

例6.1 已知一个如图所示的训练数据集，其正例点是 $x_1 = (3,3)^T$ ， $x_2 = (4,3)^T$ ，负例点是 $x_3 = (1,1)^T$ ，试求最大间隔分离超平面。

解 按照算法6.1，根据训练数据集构造约束最优化问题：

$$\min_w \frac{1}{2} (w_1^2 + w_2^2)$$

$$\text{s.t. } 3w_1 + 3w_2 + b \geq 1,$$

$$4w_1 + 3w_2 + b \geq 1,$$

$$-w_1 - w_2 + b \geq 1,$$

求得此最优化问题的解 $w_1 = w_2 = \frac{1}{2}$ ， $b = -2$ 。于是最大间隔分离超平面为

$$\frac{1}{2}x_1 + \frac{1}{2}x_2 - 2 = 0$$

其中 $x_1 = (3,3)^T$ 和 $x_3 = (1,1)^T$ 是支持向量。



6.1.4 学习的对偶算法

引入拉格朗日乘子 $\alpha_i \geq 0, i = 1, 2, \dots, N$ 得到拉格朗日函数

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)), \quad (6.18)$$

其中, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ 为拉格朗日乘子向量。

根据拉格朗日对偶性, 原始问题的对偶问题是极大极小问题:

$$\max_{\boldsymbol{\alpha}} \min_{\mathbf{w}, b} L(\mathbf{w}, b, \boldsymbol{\alpha})$$

(1) 求 $\min_{\mathbf{w}, b} L(\mathbf{w}, b, \boldsymbol{\alpha})$

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (6.19)$$

$$\nabla_b L(\mathbf{w}, b, \boldsymbol{\alpha}) = - \sum_{i=1}^N \alpha_i y_i = 0 \quad \Rightarrow \quad 0 = \sum_{i=1}^N \alpha_i y_i \quad (6.20)$$



公式推导

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

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

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

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

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

$$= -\frac{1}{2} \left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \right)^T \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i y_i b$$

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

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

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



6.1.4 学习的对偶算法

(2) 求 $\min_{w,b} L(w, b, \alpha)$ 对 α 的极大, 即对偶问题

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_j \cdot x_j)$$

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

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

(6.21)

将上述目标函数由求极大转换为求极小, 得到下面与之等价的对偶优化问题

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_j \cdot x_j) - \sum_{i=1}^N \alpha_i$$

(6.22)

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

(6.23)

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

(6.24)



6.1.4 学习的对偶算法

定理6.2 设 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$ 是对偶最优化问题(6.22) ~ (6.24)的解, 则存在下标 j , 使得 $\alpha_j^* > 0$, 并可按下式求得原始最优化问题 (6.13)~(6.14)的解 w^*, b^* :

$$w^* = \sum_{i=1}^N \alpha_i^* y_i x_i \quad (6.25)$$

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j) \quad (6.26)$$

因此, 分离超平面可以写成

$$\sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x) + b^* = 0 \quad (6.29)$$

分类决策函数为:

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x) + b^*\right) \quad (6.30)$$



6.1.4 学习的对偶算法

算法 6.2 (线性可分支持向量机器学习算法)

输入：线性可分训练数据集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

输出：最大间隔超平面和分类决策函数

(1) 构造并求解约束最优化问题

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_j \cdot \mathbf{x}_j) - \sum_{i=1}^N \alpha_i$$

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

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

求得最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$

(2) 计算

$$\mathbf{w}^* = \sum_{i=1}^N \alpha_i^* y_i \mathbf{x}_i$$

选择 α^* 的一个正分量 $\alpha_j^* > 0$, 计算

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}_j)$$

(3) 求得分离超平面:

$$\sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^* = 0$$

分类决策函数为:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^*\right)$$

例6.2

例6.2 训练数据同例6.1, 其正例点是 $\mathbf{x}_1 = (3,3)^T$, $\mathbf{x}_2 = (4,3)^T$, 负例点是 $\mathbf{x}_3 = (1,1)^T$, 试求最大间隔分离超平面。

解 根据所给数据, 对偶问题是

$$\begin{aligned} \min_{\alpha} & \frac{1}{2} \sum_{i=1}^3 \sum_{j=1}^3 \alpha_i \alpha_j y_i y_j (\mathbf{x}_j \cdot \mathbf{x}_j) - \sum_{i=1}^3 \alpha_i \\ & = \frac{1}{2} (18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - (\alpha_1 + \alpha_2 + \alpha_3) \\ \text{s. t. } & \sum_{i=1}^3 \alpha_i y_i = \alpha_1 + \alpha_2 - \alpha_3 = 0, \\ & \alpha_i \geq 0, \quad i = 1, 2, 3 \end{aligned}$$

将 $\alpha_3 = \alpha_1 + \alpha_2$ 代入目标函数并记为

$$s(\alpha_1, \alpha_2) = 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$$

例6.2 (续)

对 α_1, α_2 求偏导数, 并令其为0, 易知 $s(\alpha_1, \alpha_2)$ 在 $(\frac{3}{2}, -1)^T$ 取极值, 但该点不满足约束条件 $\alpha_2 \geq 0$, 所以最小值应在边界上达到。

$$\text{当 } \alpha_1=0 \text{ 时, 最小值 } s\left(0, \frac{3}{2}\right) = -\frac{2}{13}$$

$$\text{当 } \alpha_2=0 \text{ 时, 最小值 } s\left(\frac{1}{4}, 0\right) = -\frac{1}{4}$$

$$\text{于是 } s(\alpha_1, \alpha_2) \text{ 在 } \alpha_1 = \frac{1}{4}, \alpha_2 = 0 \text{ 获得极小, } \alpha_3 = \alpha_1 + \alpha_2 = \frac{1}{4}$$

$$\text{因此, } \alpha_1^* = \alpha_3^* = \frac{1}{4} \text{ 对应的实例向量 } x_1, x_2 \text{ 为支持向量。}$$

$$\text{得: } \omega_1^* = \omega_2^* = \frac{1}{2}$$

$$b^* = -2$$

$$\text{分离超平面: } \frac{1}{2}x_1 + \frac{1}{2}x_2 - 2 = 0$$

$$\text{分类决策函数: } f(x) = \text{sign}\left(\frac{1}{2}x_1 + \frac{1}{2}x_2 - 2\right)$$

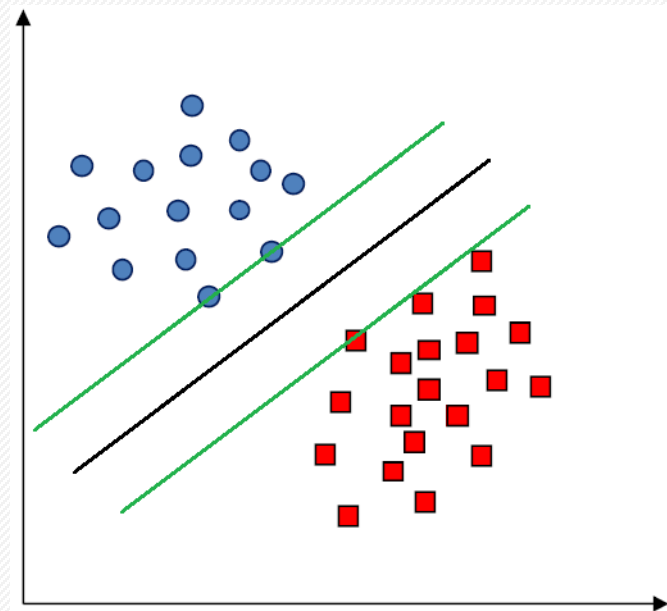
解的稀疏性

KKT条件:

$$\begin{cases} \alpha_i \geq 0 \\ y_i f(\mathbf{x}_i) - 1 \geq 0 \\ \alpha_i (y_i f(\mathbf{x}_i) - 1) = 0 \end{cases}$$



必有 $\alpha_i = 0$ 或 $y_i f(\mathbf{x}_i) = 1$



解的稀疏性: 训练完成后, 最终模型仅与支持向量有关

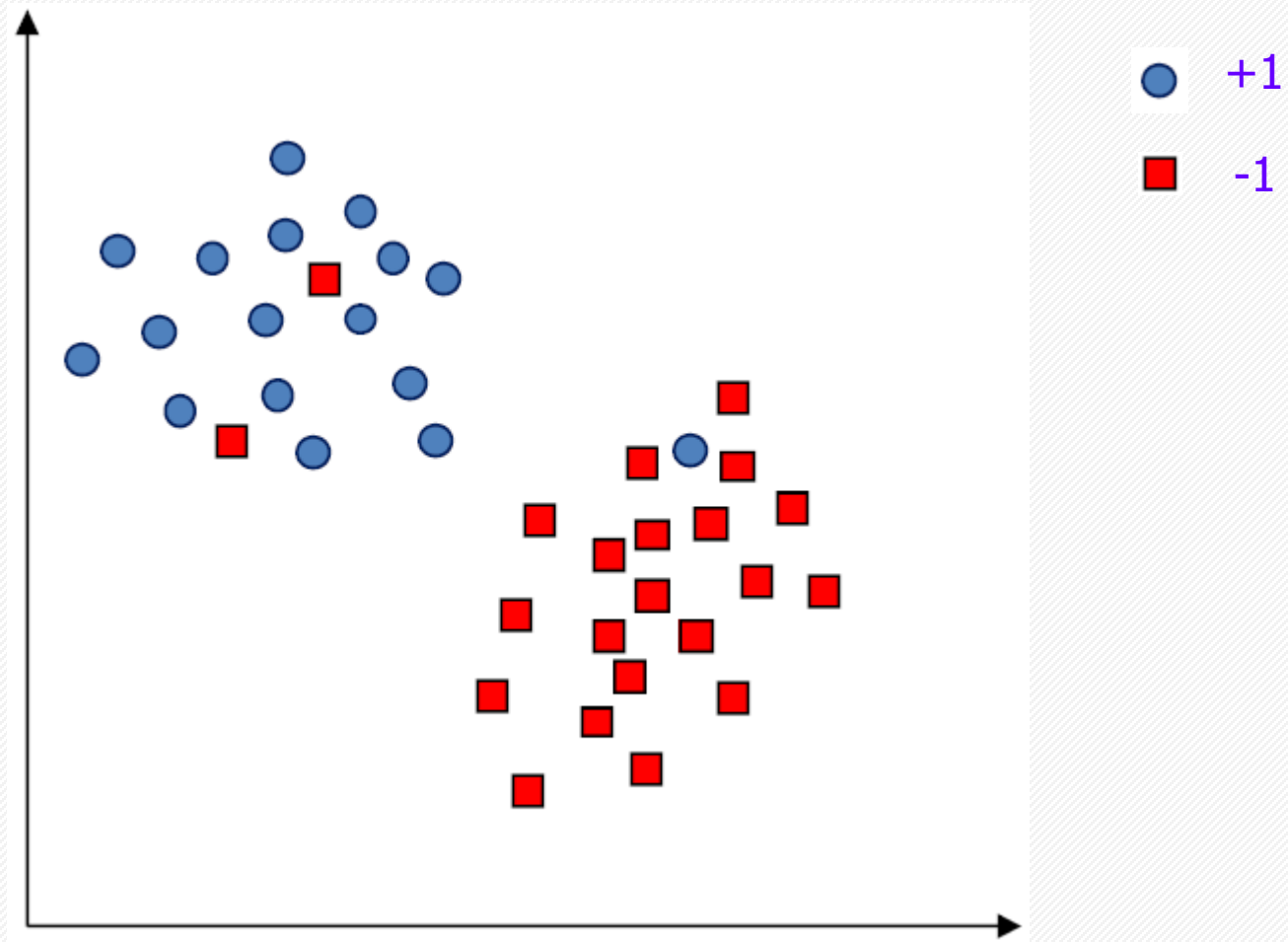
正例支持向量: $\mathbf{w}^T \mathbf{x} + b = 1$

反例支持向量: $\mathbf{w}^T \mathbf{x} + b = -1$

支持向量机(Support Vector Machine, SVM) 因此而得名



6.2 线性支持向量机与软间隔最大化





6.2 线性支持向量机与软间隔最大化

6.2.1 线性支持向量机

假定线性不可分训练集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

基本思路：对每个样本点引入松弛变量 $\xi_i \geq 0$ ，使函数间隔加上松弛变量大于等于1。

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) + \xi_i \geq 1$$

目标函数变为：

$$\frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^N \xi_i \quad (6.31)$$

其中， c 是惩罚参数，一般由应用问题决定。

使 $\frac{1}{2} \|\mathbf{w}\|^2$ 尽量小即间隔尽量大，同时使误分类的样本数尽量少。



6.2.1 线性支持向量机

线性支持向量学习问题变成凸二次规划问题

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (6.32)$$

$$\text{s. t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1, 2, \dots, N \quad (6.33)$$

$$\xi_i \geq 0, i = 1, 2, \dots, N \quad (6.34)$$

定义6.5 (线性支持向量机) 对于给定的线性不可分训练数据集，通过求解凸二次规划问题，即软间隔最大化问题(6.32)-(6.34)，得到分离超平面为

$$\mathbf{w}^* \cdot \mathbf{x} + b^* = 0 \quad (6.35)$$

以及相应的分类决策函数

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^* \cdot \mathbf{x} + b^*) \quad (6.36)$$

称为线性支持向量机。



6.2.2 学习的对偶问题

原始问题的对偶问题：

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_j \cdot \mathbf{x}_j) - \sum_{i=1}^N \alpha_i \quad (6.37)$$

$$\text{s. t. } \sum_{i=1}^N \alpha_i y_i = 0, \quad (6.38)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \quad (6.39)$$

6.2.2 学习的对偶问题

原始最优化问题的拉格朗日函数为

$$L(\mathbf{w}, b, \xi, \alpha, \mu) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^N \mu_i \xi_i, \quad (6.40)$$

其中, $\alpha_i \geq 0, \mu_i \geq 0$ 。

对偶问题是拉格朗日函数的极大极小问题。

$$\max_{\alpha, \mu} \min_{\mathbf{w}, b, \xi} L(\mathbf{w}, b, \xi, \alpha, \mu)$$

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \xi, \alpha, \mu) = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i = 0$$



$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (6.41)$$

$$\nabla_b L(\mathbf{w}, b, \xi, \alpha, \mu) = - \sum_{i=1}^N \alpha_i y_i = 0$$



$$0 = \sum_{i=1}^N \alpha_i y_i \quad (6.42)$$

$$\nabla_{\xi_i} L(\mathbf{w}, b, \xi, \alpha, \mu) = C - \alpha_i - \mu_i = 0$$



$$C - \alpha_i - \mu_i = 0 \quad (6.43)$$



6.2.2 学习的对偶问题

得

$$\min_{\mathbf{w}, b, \xi} L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_j \cdot \mathbf{x}_j) + \sum_{i=1}^N \alpha_i$$

再对 $\min_{\mathbf{w}, b, \xi} L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\mu})$ 求 $\boldsymbol{\alpha}$ 的极大, 即

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_j \cdot \mathbf{x}_j) + \sum_{i=1}^N \alpha_i \quad (6.44)$$

$$\text{s. t. } \sum_{i=1}^N \alpha_i y_i = 0, \quad (6.45)$$

$$C - \alpha_i - \mu_i = 0 \quad (6.46)$$

$$\alpha_i \geq 0, \quad (6.47)$$

$$\mu_i \geq 0, \quad i = 1, 2, \dots, N \quad (6.48)$$



6.2.2 学习的对偶问题

定理6.3 设 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$ 是对偶最优化问题(6.37) ~ (6.39)的解, 则存在下标 j , 使得 $0 < \alpha_j^* < C$, 并可按下式求得原始最优化问题 (6.32)~(6.34)的解 w^*, b^* :

$$w^* = \sum_{i=1}^N \alpha_i^* y_i x_i \quad (6.25)$$

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j) \quad (6.26)$$

因此, 分离超平面可以写成

$$\sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x) + b^* = 0 \quad (6.29)$$

分类决策函数为:

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x) + b^*\right) \quad (6.30)$$



6.2.2 学习的对偶问题

算法 6.3 (线性支持向量机学习算法)

输入：线性可分训练数据集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

输出：分离超平面和分类决策函数

(1) 选择惩罚参数 $C > 0$ ，并构造凸二次规划问题

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_j \cdot \mathbf{x}_j) - \sum_{i=1}^N \alpha_i$$

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

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N$$

求得最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$

(2) 计算

$$\mathbf{w}^* = \sum_{i=1}^N \alpha_i^* y_i \mathbf{x}_i$$

选择 α^* 的一个分量 $0 < \alpha_j^* < C$ ，计算

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}_j)$$

(3) 求得分离超平面：

$$\mathbf{w}^* \cdot \mathbf{x} + b^* = 0$$

得分类决策函数：

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^* \cdot \mathbf{x} + b^*)$$



6.2.3 支持向量

- 当 $0 < \alpha_i^* < C$ 时, $\xi_i = 0$, (x_i, y_i) 是支持向量, 在间隔边界上;
- 当 $\alpha_i^* = 0$ 时, (x_i, y_i) 在间隔边界之外;
- 当 $\alpha_i^* = C$, $0 < \xi_i < 1$ 时, (x_i, y_i) 在间隔边界与分离超平面之间;
- 当 $\alpha_i^* = C$, $\xi_i = 1$ 时, (x_i, y_i) 在分离超平面上;
- 当 $\alpha_i^* = C$, $\xi_i > 1$ 时, (x_i, y_i) 在分离超平面误分类一侧。

$$\alpha_i(y_i f(x_i) - 1 + \xi_i) = 0$$

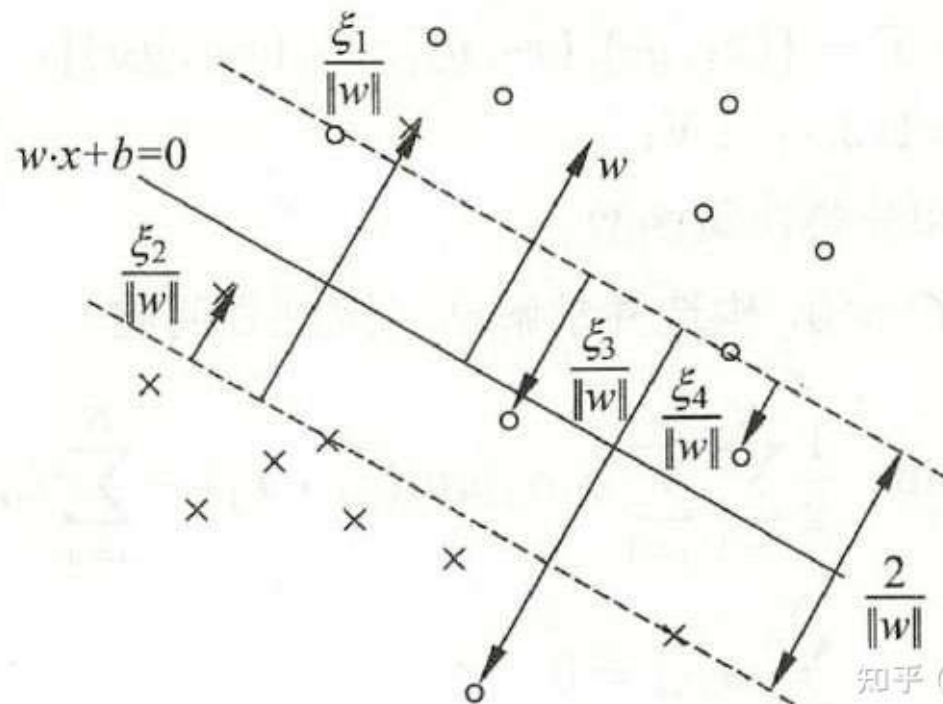
$$\mu_i \xi_i = 0$$

$$y_i f(x_i) - 1 + \xi_i \geq 0$$

$$C - \alpha_i - \mu_i = 0$$

$$\alpha_i \geq 0$$

$$\mu_i \geq 0$$





6.2.4 合页损失函数

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

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

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

$$\ell_{0/1} = \begin{cases} 1, & z < 0 \\ 0, & z \geq 0 \end{cases}$$

障碍： 0/1 损失函数非凸、不连续，不易优化！

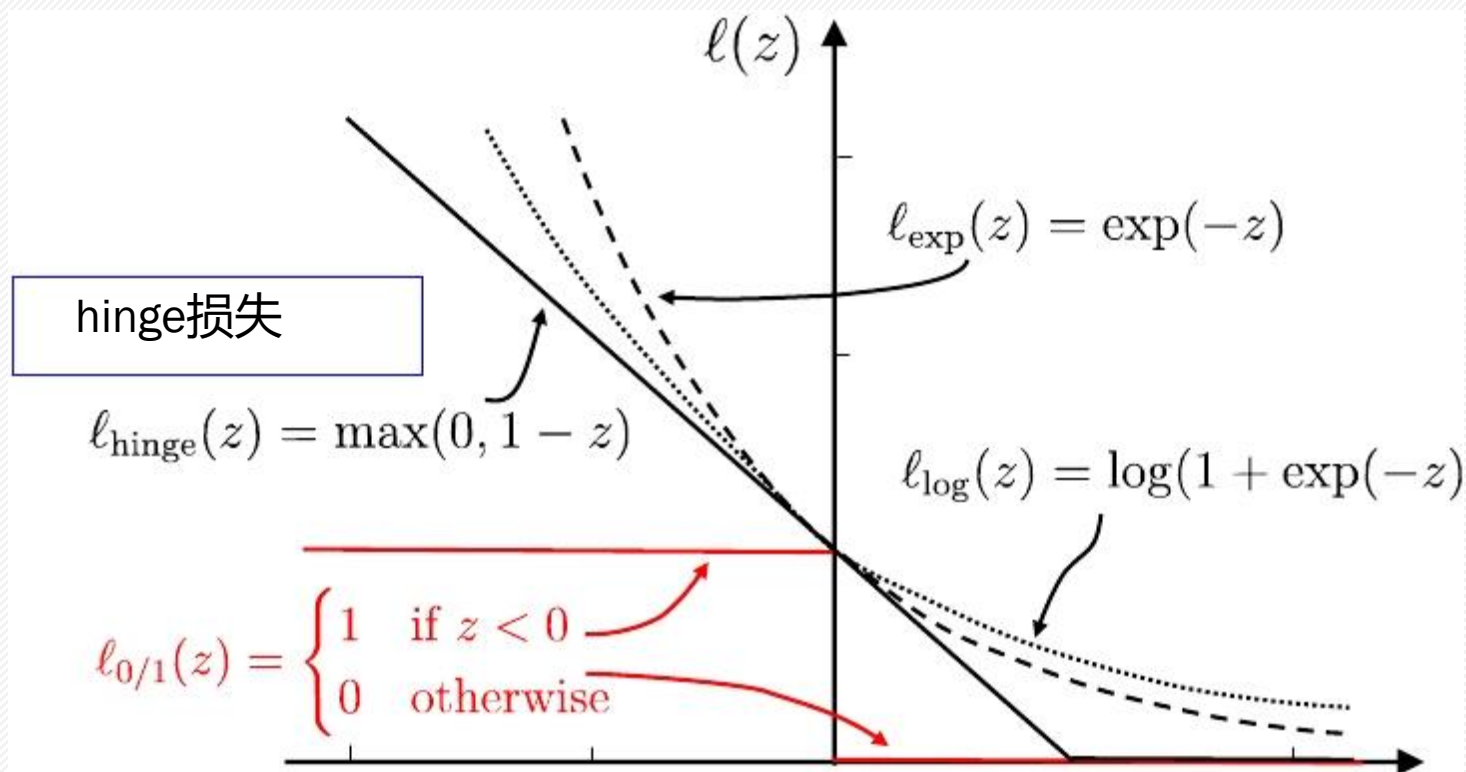
hinge 损失： $\ell_{hinge}(z) = \max(0, 1 - z)$

替代损失(surrogate loss)

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

指数损失: $\ell_{\text{exp}}(z) = \exp(-z)$

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



替代损失(surrogate loss)

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

引入 “松弛变量” (slack variables)

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad \text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, i = 1, 2, \dots, N \quad (6.36)$$

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

其中 $\alpha_i \geq 0$, $\mu_i \geq 0$ 是拉格朗日乘子。令 $L(\mathbf{w}, b, \alpha, \xi, \mu)$ 对 \mathbf{w}, b, ξ_i 的偏导数为零, 可得

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad 0 = \sum_{i=1}^N \alpha_i y_i \quad C = \alpha_i + \mu_i \quad (6.39)$$

替代损失(surrogate loss)

得到对应的对偶问题

$$C = \alpha_i + \mu_i \quad (6.39)$$

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (6.40)$$

$$\text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0, \quad C \geq \alpha_i \geq 0, \quad i = 1, 2, \dots, N$$

对软间隔支持向量机, KKT条件要求

$$\begin{cases} \alpha_i \geq 0, \mu_i \geq 0 \\ y_i f(\mathbf{x}_i) - 1 + \xi_i \geq 0 \\ \alpha_i (y_i f(\mathbf{x}_i) - 1 + \xi_i) = 0 \\ \xi_i \geq 0, \mu_i \xi_i = 0 \end{cases} \quad (6.41)$$



6.2.4 合页损失函数

定理6.4 线性支持向量机原始最优化问题

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (6.60)$$

$$\text{s. t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1, 2, \dots, N \quad (6.61)$$

$$\xi_i \geq 0, i = 1, 2, \dots, N \quad (6.62)$$

等价于最优化问题

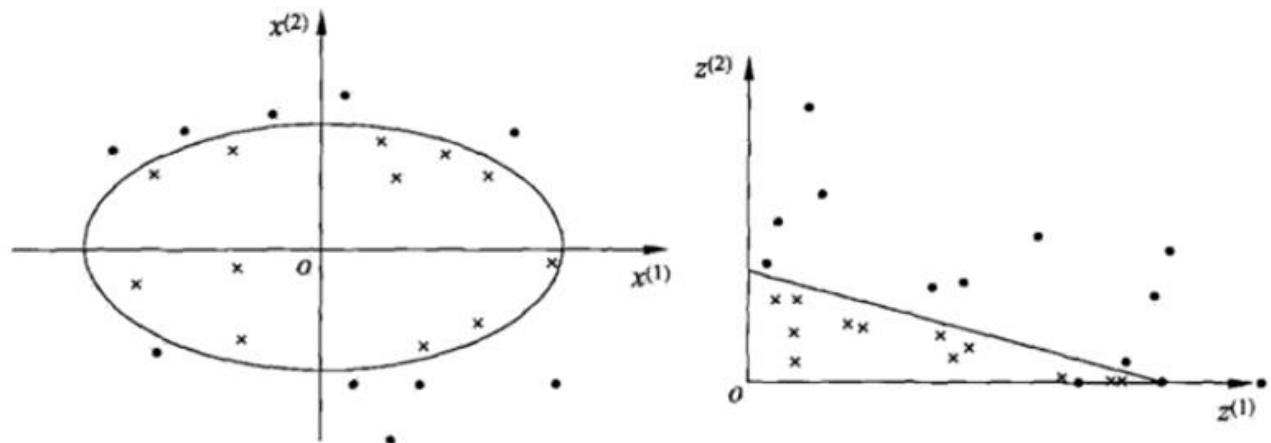
$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \quad (6.63)$$

6.3 非线性支持向量机与核函数

6.3.1 核技巧

1. 非线性分类问题

采用方法是进行一个非线性变换，将非线性问题转换成线性问题。



非线性支持向量机与核技巧示例

设原空间为 $\mathcal{X} \subset \mathbb{R}^2$, $\mathbf{x} = (x^{(1)}, x^{(2)})^T \in \mathcal{X}$, 新空间为 $\mathcal{Z} \subset \mathbb{R}^2$, $\mathbf{z} = (z^{(1)}, z^{(2)})^T \in \mathcal{Z}$, 定义从原空间到新空间的变换（映射）：

$$\mathbf{z} = \phi(\mathbf{x}) = ((x^{(1)})^2, (x^{(2)})^2)^T$$

原空间的椭圆 $w_1(x^{(1)})^2 + w_2(x^{(2)})^2 + b = 0$

变为新空间中的直线 $w_1 z^{(1)} + w_2 z^{(2)} + b = 0$

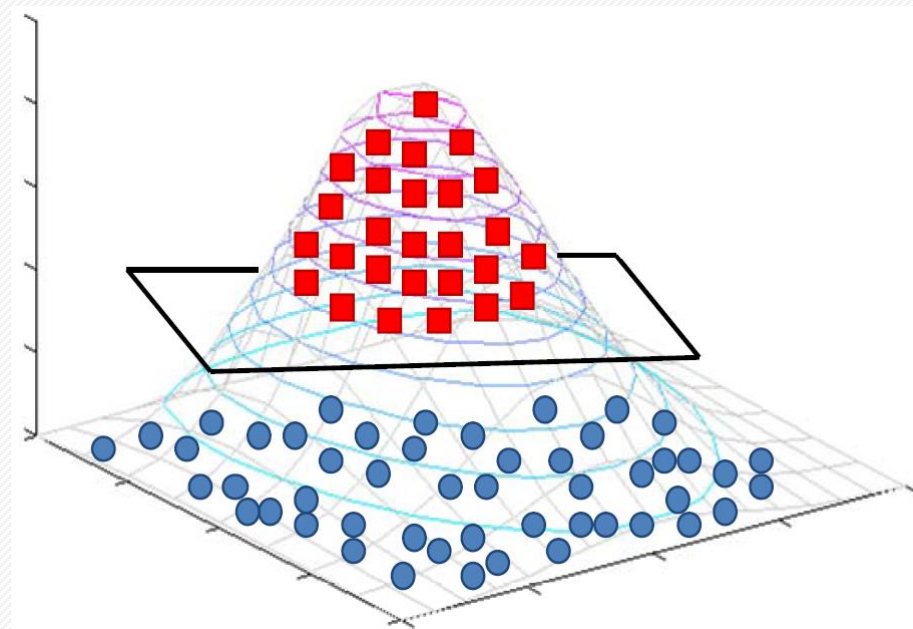
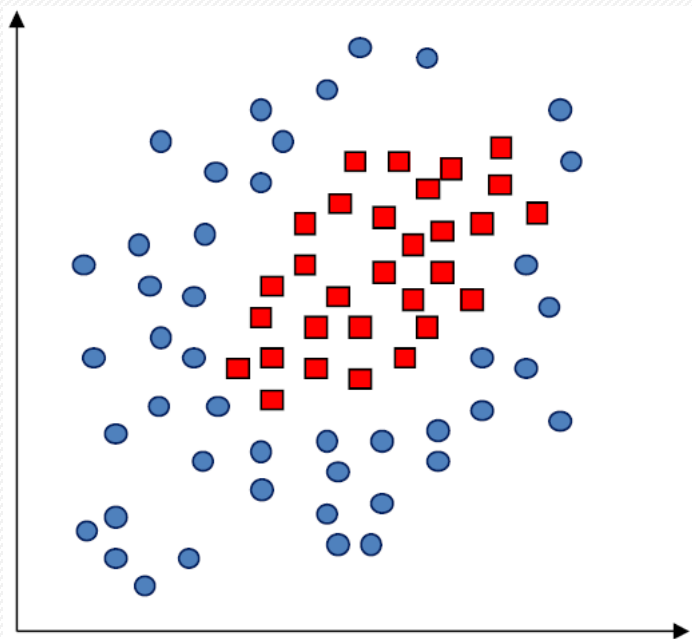
核技巧方法的两个步骤：

- 使用一个变换将原空间的数据映射到新空间。
- 在新空间里用线性分类学习方法从训练集学习模型。



6.3 核函数

将样本从原始空间映射到一个高维特征空间，使样本在该特征空间内线性可分



如果原始空间是有限维 (属性有限), 则一定存在一个高维特征空间使样本可分



6.3.1 核技巧

2.核函数的定义

定义： 设 \mathcal{X} 是输入空间， \mathcal{H} 为特征空间，如果存在从 \mathcal{X} 到 \mathcal{H} 的映射

$$\phi(x): \mathcal{X} \rightarrow \mathcal{H}$$

使得对所有 $x, z \in \mathcal{X}$ ，函数 $K(x, z)$ 满足条件

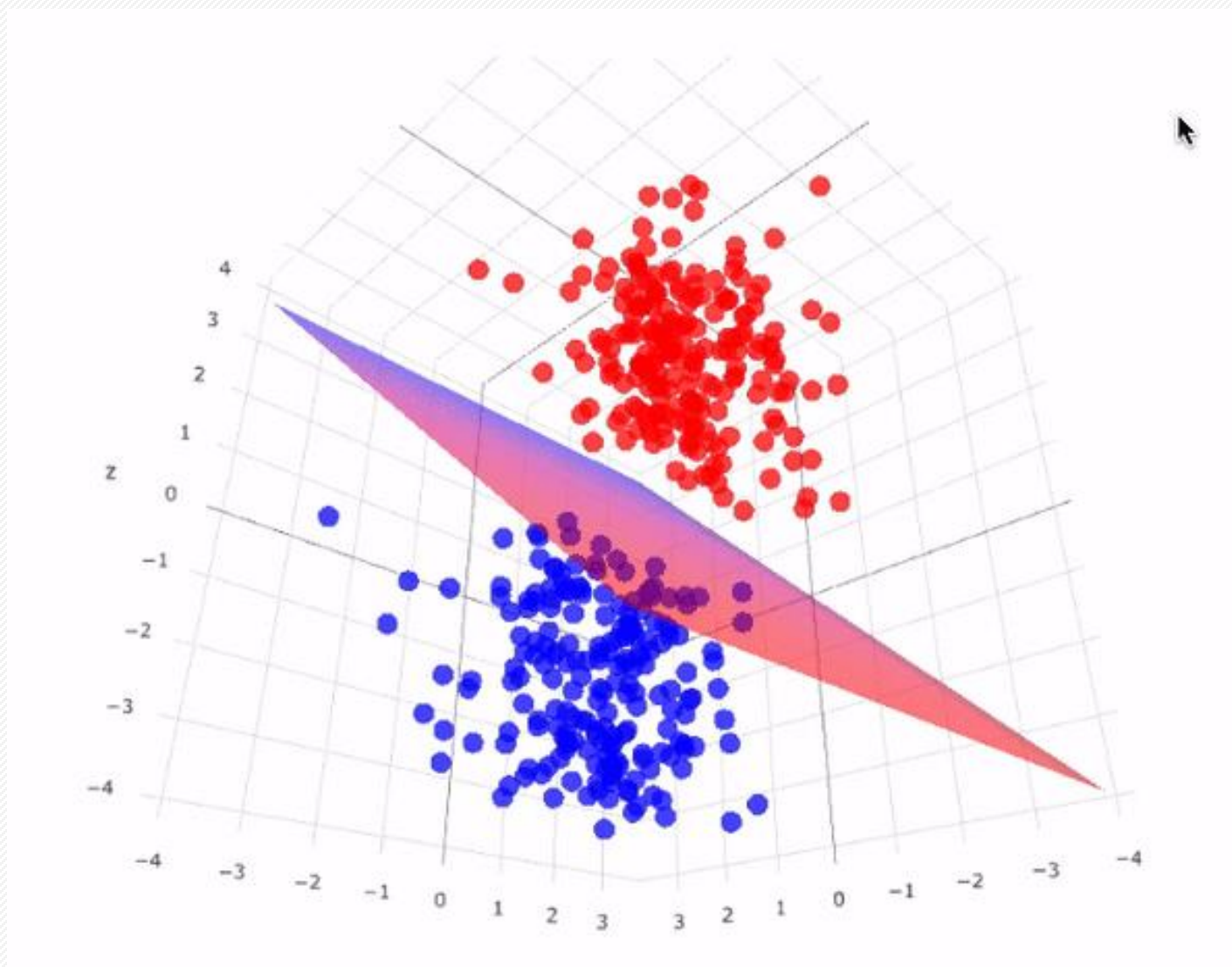
$$K(x, z) = \phi(x) \cdot \phi(z)$$

则称 $K(x, z)$ 为核函数， $\phi(x)$ 为映射函数。

核技巧的思想：在学习与预测中只定义核函数 $K(x, z)$ ，而不显示地定义映射函数 $\phi(x)$ 。通常直接计算 $K(x, z)$ 比较容易，通过 $\phi(x)$ 计算并不容易。

对于给定的核函数 $K(x, z)$ ，特征空间 \mathcal{H} 和映射函数 ϕ 取法不唯一，可以取不同的特征空间，即使同一特征空间也可以取不同的映射。

6.3 核函数



6.3.1 核技巧

例6.3 假设输入空间是 \mathbb{R}^2 ，核函数是 $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^2$ ，试找出其相关的特征空间 \mathcal{H} 和映射 $\phi(\mathbf{x}): \mathbb{R}^2 \rightarrow \mathcal{H}$ 。

解：取特征空间 $\mathcal{H} = \mathbb{R}^3$ ，记 $\mathbf{x} = (x^{(1)}, x^{(2)})^T$ ， $\mathbf{z} = (z^{(1)}, z^{(2)})^T$ 容易验证：

$$(\mathbf{x} \cdot \mathbf{z})^2 = (x^{(1)}z^{(1)} + x^{(2)}z^{(2)})^2$$

$$= (x^{(1)}z^{(1)})^2 + 2x^{(1)}z^{(1)}x^{(2)}z^{(2)} + (x^{(2)}z^{(2)})^2$$

$$\phi(\mathbf{x}) = ((x^{(1)})^2, \sqrt{2}x^{(1)}x^{(2)}, (x^{(2)})^2)^T$$

$$\phi(\mathbf{z}) = ((z^{(1)})^2, \sqrt{2}z^{(1)}z^{(2)}, (z^{(2)})^2)^T$$

$$\phi(\mathbf{x})^T \phi(\mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^2 = K(\mathbf{x}, \mathbf{z})$$

同样，以下两个映射也都满足条件：

$$\mathcal{H} = \mathbb{R}^3 \quad \phi(\mathbf{x}) = \frac{1}{\sqrt{2}} ((x^{(1)})^2 - (x^{(2)})^2, 2x^{(1)}x^{(2)}, (x^{(1)})^2 + (x^{(2)})^2)^T$$

$$\mathcal{H} = \mathbb{R}^4 \quad \phi(\mathbf{x}) = ((x^{(1)})^2, x^{(1)}x^{(2)}, x^{(1)}x^{(2)}, (x^{(2)})^2)^T$$



6.3.1 核技巧

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

原始问题

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

$$\text{s.t. } y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1, i = 1, 2, \dots, N$$

对偶问题

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, N$$

预测

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$



6.3.2 正定核

假设 \mathcal{X} 为输入空间, $K(x, z)$ 是定义在 $\mathcal{X} \times \mathcal{X}$ 上的对称函数, 并且对于任意 $x_1, x_2, \dots, x_m \in \mathcal{X}$, $K(x, z)$ 关于 x_1, x_2, \dots, x_m 的Gram矩阵是半正定的。可以依据函数 $K(x, z)$ 构成一个希尔伯特空间 (Hilbert space)。

1. 定义映射, 构成向量空间 S

定义映射 $\phi: x \rightarrow K(\cdot, x)$ (6.69)

根据该映射, 对任意的 $x_i \in \mathcal{X}$, $\alpha_i \in \mathbb{R}$, $i = 1, 2, \dots, m$, 定义线性组合

$$f(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, x_i) \quad (6.70)$$

考虑由线性组合为元素的集合 S 。可以证明, S 对乘法和加法封闭, 所以 S 构成了一个向量空间。

$$kf(\cdot) = k \sum_{i=1}^m \alpha_i K(\cdot, x_i) = \sum_{i=1}^m k\alpha_i K(\cdot, x_i) \in S$$

$$\text{设 } g(\cdot) = \sum_{j=1}^l \beta_j K(\cdot, z_j) \text{ 则 } f(\cdot) + g(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, x_i) + \sum_{j=1}^l \beta_j K(\cdot, z_j) = \sum_{k=1}^{m+l} \gamma_k K(\cdot, v_k)$$



6.3.2 正定核

2. 在 S 上定义内积，使其成为内积空间

在 S 上定义一个运算 $*$ ，对任意的 $f, g \in S$ 有

$$f(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, \mathbf{x}_i) \quad (6.71)$$

$$g(\cdot) = \sum_{j=1}^l \beta_j K(\cdot, \mathbf{z}_j) \quad (6.72)$$

定义运算 $*$

$$f * g = \sum_{i=1}^m \sum_{j=1}^l \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{z}_j) \quad (6.73)$$

可以证明运算 $*$ 是空间 S 的内积。

$$(1) (cf) * g = c(f * g), c \in \mathbb{R} \quad (6.74)$$

$$(2) (f + g) * h = f * h + g * h, h \in S \quad (6.75)$$

$$(3) f * g = g * f \quad (6.76)$$

$$(4) f * f \geq 0 \quad (6.77)$$

$$f * f = 0 \Leftrightarrow f = \mathbf{0} \quad (6.78)$$

证明： (1) $(cf) * g = c(f * g), c \in \mathbb{R}$

$$\begin{aligned} (cf) * g &= \sum_{i=1}^m c \alpha_i K(\cdot, \mathbf{x}_i) * \sum_{j=1}^l \beta_j K(\cdot, \mathbf{z}_j) \\ &= \sum_{i=1}^m c \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{z}_j) \\ &= c \sum_{i=1}^m \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{z}_j) \\ &= c(f * g) \end{aligned}$$



6.3.2 正定核

证明: (2) $(f + g) * h = f * h + g * h, h \in S$

$$\begin{aligned}(f + g) * h &= \left(\sum_{i=1}^m \alpha_i K(\cdot, \mathbf{x}_i) + \sum_{j=1}^l \beta_j K(\cdot, \mathbf{z}_j) \right) * \left(\sum_{k=1}^n \gamma_k K(\cdot, \mathbf{v}_k) \right) \\&= \sum_{i=1}^m \alpha_i K(\cdot, \mathbf{x}_i) * \sum_{k=1}^n \gamma_k K(\cdot, \mathbf{v}_k) + \sum_{j=1}^l \beta_j K(\cdot, \mathbf{z}_j) * \sum_{k=1}^n \gamma_k K(\cdot, \mathbf{v}_k) \\&= f * h + g * h\end{aligned}$$

证明: (3) $f * g = g * f$

$$\left. \begin{aligned}f * g &= \sum_{i=1}^m \sum_{j=1}^l \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{z}_j) \\g * f &= \sum_{i=1}^m \sum_{j=1}^l \beta_j \alpha_i K(\mathbf{z}_j, \mathbf{x}_i)\end{aligned} \right\} f * g = g * f$$



6.3.2 正定核

证明: (4) $f * f \geq 0$

$$\begin{aligned} f * f &= \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i,j=1}^m \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &= [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_m] \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \cdots & K(\mathbf{x}_1, \mathbf{x}_m) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & \cdots & K(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_m, \mathbf{x}_1) & K(\mathbf{x}_m, \mathbf{x}_2) & \cdots & K(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} \\ \therefore f * f &\geq 0 \end{aligned}$$

证明: (5) $f * f = 0 \Leftrightarrow f = 0$

充分性显然。为证明必要性, 首先证明不等式:

$$|f * g|^2 \leq (f * f)(g * g)$$

设 $f, g \in S, \lambda \in \mathbb{R}$, 则 $f + \lambda g \in S$, 于是

$$(f + \lambda g) * (f + \lambda g) \geq 0$$

$$f * f + 2\lambda(f * g) + \lambda^2(g * g) \geq 0$$

$$(g * g)\lambda^2 + 2(f * g)\lambda + (f * f) \geq 0$$

$$\Delta^2 = 4(f * g)^2 - 4(f * f)(g * g) \leq 0$$

$$(f * g)^2 - (f * f)(g * g) \leq 0$$

$$\therefore |f * g|^2 \leq (f * f)(g * g)$$



6.3.2 正定核

若
$$f(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, \mathbf{x}_i)$$

由于上述等式全部得证，所以运算*是空间 S 的内积。
赋予内积的向量空间为内积空间，所以 S 是一个内积空间。

对于任意的 $\mathbf{x} \in \mathcal{X}$ ，有

$$K(\cdot, \mathbf{x}) * f = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i) = f(\mathbf{x})$$

于是

$$|f(\mathbf{x})|^2 = |K(\cdot, \mathbf{x}) * f|^2$$

则

$$\begin{aligned} |K(\cdot, \mathbf{x}) * f|^2 &\leq (K(\cdot, \mathbf{x}) * K(\cdot, \mathbf{x}))(f * f) \\ &= K(\mathbf{x}, \mathbf{x})(f * f) \end{aligned}$$

因此

$$|f(\mathbf{x})|^2 \leq K(\mathbf{x}, \mathbf{x})(f * f)$$

当 $f * f = 0$ 时，对任意的 \mathbf{x} 都有 $|f(\mathbf{x})| = 0$



6.3.2 正定核

3. 将内积空间 S 完备化为希尔伯特空间

由(6.81)定义的内积可以得到范数

$$\|f\| = \sqrt{f \cdot f}$$

因此 S 是一个**赋范向量空间**。

依据泛函分析理论，对于不完备的赋范向量空间 S ，一定可以使之完备化，得到完备的赋范向量空间 \mathcal{H} 。

对于一个内积空间，当作为一个赋范向量空间是完备的时候，就是**希尔伯特空间**。这样就得到了希尔伯特空间 \mathcal{H} 。该空间称为**再生核希尔伯特空间** (Reproducing Kernel Hilbert Space, RKHS)

$$K(\cdot, \mathbf{x}) * f = f(\mathbf{x})$$

$$K(\cdot, \mathbf{x}) * K(\cdot, \mathbf{z}) = K(\mathbf{x}, \mathbf{z})$$

称为**再生核**。



6.3.2 正定核

定理6.5 (正定核的充要条件) 设 \mathcal{X} 为输入空间, $K(\cdot, \cdot)$ 是 $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ 上的对称函数, 则 $K(x, z)$ 是正定核函数的充要条件是对任意数据 $x_i, i = 1, 2, \dots, N$, $K(x, z)$ 对应的Gram矩阵 K 是半正定的矩阵:

$$K = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_j) & \cdots & K(x_1, x_N) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ K(x_i, x_1) & \cdots & K(x_i, x_j) & \cdots & K(x_i, x_N) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ K(x_N, x_1) & \cdots & K(x_N, x_j) & \cdots & K(x_N, x_N) \end{bmatrix}$$

“核函数选择” 成为决定支持向量机性能的关键!



6.3.3 常用核函数

表6.1 常用核函数

名称	表达式	参数
线性核	$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$	
多项式核	$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^T \boldsymbol{x}_j + 1)^d$	$d \geq 1$ 为多项式的次数
高斯核	$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\ \boldsymbol{x}_i - \boldsymbol{x}_j\ ^2}{2\sigma^2})$	$\sigma > 0$ 为高斯核的带宽(width)
拉普拉斯核	$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\ \boldsymbol{x}_i - \boldsymbol{x}_j\ }{2\sigma^2})$	$\sigma > 0$
Sigmoid核	$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^T \boldsymbol{x}_j + \theta)$	\tanh 为双曲正切函数, $\beta > 0, \theta < 0$

经验：文本数据常用线性核，情况不明时可先尝试高斯核。

若 K_1 和 K_2 是核函数，则对任意正数 γ_1, γ_2 和任意函数 $g(x)$

$\gamma_1 K_1 + \gamma_2 K_2$

$K_1 \otimes K_2(\boldsymbol{x}, \boldsymbol{z}) = K_1(\boldsymbol{x}, \boldsymbol{z})K_2(\boldsymbol{x}, \boldsymbol{z})$

$K(\boldsymbol{x}, \boldsymbol{z}) = g(\boldsymbol{x})^T K_1(\boldsymbol{x}, \boldsymbol{z})g(\boldsymbol{z})$

}

均为核函数

6.3.4 非线性支持向量机

定义6.8 (非线性支持向量机) 从非线性分类训练集, 通过核函数与软间隔最大化, 或凸二次规划 (6.95)~(6.97), 学习得到的分类决策函数

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^m \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (6.94)$$

算法6.4 非线性支持向量机学习算法

输入: 训练样本集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

输出: 分类决策函数

① 选取核函数 $K(\mathbf{x}, \mathbf{z})$ 和参数 C , 构造并求解最优化问题

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i \quad (6.95)$$

$$\text{s. t. } \sum_{i=1}^N \alpha_i y_i = 0, \quad (6.96)$$

6.3.4 非线性支持向量机

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \quad (6.97)$$

求最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$ 。

② 选择 α^* 的一个正分量 $0 < \alpha_j^* < C$ ，计算

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}_i \cdot \mathbf{x}_j)$$

③ 构造决策函数：

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*\right)$$

当 $K(\mathbf{x}, \mathbf{z})$ 是正定核函数时，(6.95)~(6.97) 是凸二次规划问题，解存在。



补充：坐标下降法

如何解决如下凸二次优化问题？

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i \quad (6.98)$$

$$\text{s. t. } \sum_{i=1}^N \alpha_i y_i = 0, \quad (6.99)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \quad (6.100)$$

其中，变量是拉格朗日乘子，一个变量 α_i 对应一个样本点 (\mathbf{x}_i, y_i) ，变量的总数等于训练样本集容量 N 。

记变量向量为 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$



补充：坐标下降法

思路： 每次只更新一个参数，逐次迭代更新所有参数。

$$\arg \min_{\alpha} f(\alpha_1, \alpha_2) = \alpha_1^2 + 2\alpha_2^2 - \alpha_1\alpha_2 + 1, \text{ 其中 } \alpha = (\alpha_1, \alpha_2)^T$$

① 给定初始值 $\alpha^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)})^T$

② 固定 $\alpha_2 = \alpha_2^{(0)}$ ，求解 α_1 ，使得目标函数 $\arg \min_{\alpha_1} f(\alpha_1, \alpha_2^{(0)})$ 达到最小。

$$\frac{df(\alpha_1, \alpha_2^{(0)})}{d\alpha_1} = 2\alpha_1 - \alpha_2^{(0)} = 0 \quad \Rightarrow \quad \alpha_1^{(1)} = \frac{\alpha_2^{(0)}}{2}$$

③ 固定 $\alpha_1 = \alpha_1^{(1)}$ ，求解 α_2 ，使得目标函数 $\arg \min_{\alpha_2} f(\alpha_1^{(1)}, \alpha_2)$ 达到最小。

$$\frac{df(\alpha_1^{(1)}, \alpha_2)}{d\alpha_2} = 4\alpha_2 - \alpha_1^{(1)} = 0 \quad \Rightarrow \quad \alpha_2^{(1)} = \frac{\alpha_1^{(1)}}{4}$$

④ 重复上面的①~③，直到收敛。



补充：坐标下降法

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i \quad (6.98)$$

$$\text{s. t. } \sum_{i=1}^N \alpha_i y_i = 0, \quad (6.99)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \quad (6.100)$$

不失一般性，不妨选 α_1 ，固定 $\alpha_2, \alpha_3, \dots, \alpha_N$

① 给定初始值 $\alpha^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)}, \dots, \alpha_N^{(0)})^T$

② 固定 $\alpha_2^{(0)}, \dots, \alpha_N^{(0)}$ ，求解 α_1 ，使得目标函数 $\arg \min_{\alpha_1} f(\alpha_1, \alpha_2^{(0)}, \dots, \alpha_N^{(0)})$ 达到最小。

失败！

$$\text{s. t. } \alpha_1 y_1 = - \sum_{i=2}^N \alpha_i^{(0)} y_i$$

6.4 序列最小最优化方法

1998, 由Platt提出, SMO(sequential minimal optimization)

不失一般性, 不妨选 α_1, α_2 , 固定 $\alpha_3, \dots, \alpha_N$

但凡确定一个参数, 另一个参数也就确定了。比如确定了 α_2 , 则根据约束条件得

$$\alpha_1 y_1 + \alpha_2 y_2 = - \sum_{i=3}^N \alpha_i y_i$$

$$\Rightarrow \alpha_1 = y_1 \left(- \sum_{i=3}^N \alpha_i y_i - \alpha_2 y_2 \right)$$
$$\left. \begin{array}{l} \text{记 } \zeta = - \sum_{i=3}^N \alpha_i y_i \\ \alpha_1 = y_1 (\zeta - \alpha_2 y_2) \end{array} \right\}$$

$$\text{记 } V_1 = \sum_{i=3}^N \alpha_i y_i K_{i1}$$

$$V_2 = \sum_{i=3}^N \alpha_i y_i K_{i2}$$

$$\min_{\alpha_1, \alpha_2} W(\alpha_1, \alpha_2) = -(\alpha_1 + \alpha_2) + \frac{1}{2} K_{11} \alpha_1^2 + \frac{1}{2} K_{22} \alpha_2^2 + y_1 y_2 K_{12} \alpha_1 \alpha_2 + \frac{1}{2} \times 2 \alpha_1 y_1 \sum_{i=3}^N \alpha_i y_i K_{i1} + \frac{1}{2} \times 2 \alpha_2 y_2 \sum_{i=3}^N \alpha_i y_i K_{i2}$$

$$\min_{\alpha_1, \alpha_2} W(\alpha_1, \alpha_2) = \frac{1}{2} K_{11} \alpha_1^2 + \frac{1}{2} K_{22} \alpha_2^2 + y_1 y_2 K_{12} \alpha_1 \alpha_2 - (\alpha_1 + \alpha_2) + y_1 \alpha_1 \sum_{i=3}^N \alpha_i y_i K_{i1} + y_2 \alpha_2 \sum_{i=3}^N \alpha_i y_i K_{i2}$$

$$\min_{\alpha_1, \alpha_2} W(\alpha_1, \alpha_2) = \frac{1}{2} K_{11} \alpha_1^2 + \frac{1}{2} K_{22} \alpha_2^2 + y_1 y_2 K_{12} \alpha_1 \alpha_2 - (\alpha_1 + \alpha_2) + y_1 \alpha_1 V_1 + y_2 \alpha_2 V_2$$

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

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

$$0 \leq \alpha_i \leq C, i = 1, \dots, N$$

6.4.1 两个变量二次规划的求解方法

$$\min_{\alpha_1, \alpha_2} W(\alpha_1, \alpha_2) = \frac{1}{2} K_{11} \alpha_1^2 + \frac{1}{2} K_{22} \alpha_2^2 + K_{12} y_1 y_2 \alpha_1 \alpha_2 - (\alpha_1 + \alpha_2) + y_1 \alpha_1 V_1 + y_2 \alpha_2 V_2$$

$$\alpha_1 = y_1(\zeta - \alpha_2 y_2)$$

$$\begin{aligned} \min_{\alpha_2} W(\alpha_2) &= \frac{1}{2} K_{11} y_1^2 (\zeta - \alpha_2 y_2)^2 + \frac{1}{2} K_{22} \alpha_2^2 + K_{12} y_1 y_2 y_1 (\zeta - \alpha_2 y_2) \alpha_2 - (y_1(\zeta - \alpha_2 y_2) + \alpha_2) \\ &\quad + y_1 y_1 (\zeta - \alpha_2 y_2) V_1 + y_2 \alpha_2 V_2 \end{aligned}$$

$$\begin{aligned} \min_{\alpha_2} W(\alpha_2) &= \frac{1}{2} K_{11} (\zeta - \alpha_2 y_2)^2 + \frac{1}{2} K_{22} \alpha_2^2 + K_{12} y_2 (\zeta - \alpha_2 y_2) \alpha_2 - (y_1(\zeta - \alpha_2 y_2) + \alpha_2) \\ &\quad + (\zeta - \alpha_2 y_2) V_1 + y_2 \alpha_2 V_2 \\ &= \left(\frac{1}{2} K_{11} + \frac{1}{2} K_{22} - K_{12} \right) \alpha_2^2 + (-K_{11} \zeta y_2 + K_{12} \zeta y_2 + y_1 y_2 - 1 - V_1 y_2 + V_2 y_2) \alpha_2 \\ &\quad + \left(\frac{1}{2} K_{11} \zeta^2 - y_1 \zeta + V_1 \zeta \right) \end{aligned}$$

$$\frac{dW(\alpha_2)}{d\alpha_2} = 2 \left(\frac{1}{2} K_{11} + \frac{1}{2} K_{22} - K_{12} \right) \alpha_2 + y_2 (y_1 - y_2 + (K_{12} - K_{11}) \zeta + V_2 - V_1) = 0$$

$$\Rightarrow \alpha_2 = \frac{y_2}{K_{11} + K_{22} - 2K_{12}} (y_2 - y_1 + (K_{11} - K_{12}) \zeta + V_1 - V_2)$$



6.4.1 两个变量二次规划的求解方法

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

$$V_1 = \sum_{i=3}^N \alpha_i y_i K_{i1}$$

$$V_2 = \sum_{i=3}^N \alpha_i y_i K_{i2}$$

$$\alpha_2 = \frac{1}{K_{11} + K_{22} - 2K_{12}} (y_2 - y_1 + (K_{11} - K_{12})\zeta + V_1 - V_2) y_2$$

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

$$\begin{aligned} K_{11}\zeta + V_1 &= (\alpha_1^{old} y_1 + \alpha_2^{old} y_2) K_{11} + \sum_{i=3}^N \alpha_i^{old} y_i K_{i1} \\ &= \alpha_1^{old} y_1 K_{11} + \alpha_2^{old} y_2 K_{21} + \alpha_2^{old} y_2 K_{11} + \sum_{i=3}^N \alpha_i^{old} y_i K_{i1} - \alpha_2^{old} y_2 K_{21} \\ &= \sum_{i=1}^N \alpha_i^{old} y_i K_{i1} + \alpha_2^{old} y_2 K_{11} - \alpha_2^{old} y_2 K_{21} \end{aligned}$$

$$\begin{aligned} K_{12}\zeta + V_2 &= (\alpha_1^{old} y_1 + \alpha_2^{old} y_2) K_{12} + \sum_{i=3}^N \alpha_i y_i K_{i2} \\ &= \alpha_1^{old} y_1 K_{12} + \alpha_2^{old} y_2 K_{22} + \alpha_2^{old} y_2 K_{12} + \sum_{i=3}^N \alpha_i y_i K_{i2} - \alpha_2^{old} y_2 K_{22} \\ &= \sum_{i=1}^N \alpha_i y_i K_{i2} + \alpha_2^{old} y_2 K_{12} - \alpha_2^{old} y_2 K_{22} \end{aligned}$$

$$K_{11}\zeta + V_1 - (K_{12}\zeta + V_2) = \sum_{i=1}^N \alpha_i^{old} y_i K_{i1} - \sum_{i=1}^N \alpha_i^{old} y_i K_{i2} + \alpha_2^{old} y_2 (K_{11} + K_{22} - 2K_{12})$$

$$\text{分子 } (\sum_{i=1}^N \alpha_i^{old} y_i K_{i1} - y_1) y_2 - (\sum_{i=1}^N \alpha_i^{old} y_i K_{i2} - y_2) y_2 + \alpha_2^{old} \eta$$

$$\text{分母 } \eta$$



6.4.1 两个变量二次规划的求解方法

分母 $(\sum_{i=1}^N \alpha_i^{old} y_i K_{i1} - y_1)y_2 - (\sum_{i=1}^N \alpha_i^{old} y_i K_{i2} - y_2)y_2 + \alpha_2^{old}\eta$

决策函数 $f(\mathbf{x}_i) = \text{sign}(\sum_{i=1}^m \alpha_i^* y_j K_{ij} + b^*)$

令 $g(\mathbf{x}_1, \boldsymbol{\alpha}^{old}, b^{old}) = \sum_{i=1}^N \alpha_i^{old} y_i K_{i1} + b^{old}$

$$g(\mathbf{x}_2, \boldsymbol{\alpha}^{old}, b^{old}) = \sum_{i=1}^N \alpha_i^{old} y_i K_{i2} + b^{old}$$

令 $E_1 = g(\mathbf{x}_1, \boldsymbol{\alpha}^{old}, b^{old}) - y_1$

$$E_2 = g(\mathbf{x}_2, \boldsymbol{\alpha}^{old}, b^{old}) - y_2$$

分子 $(E_1 - b^{old})y_2 - (E_2 - b^{old})y_2 + \alpha_2^{old}\eta$

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

$$\alpha_1^{new,unc} = y_1(\zeta - \alpha_2^{new}y_2)$$



6.4.1 两个变量二次规划的求解方法

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

$$L \leq \alpha_2^{new} \leq H$$

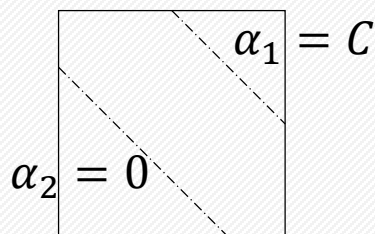
约束条件

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

$$0 \leq \alpha_1 \leq C$$

$$0 \leq \alpha_2 \leq C$$

①当 $y_1 = y_2$ 时, 有 $\alpha_1 + \alpha_2 = y_1 \zeta = k$



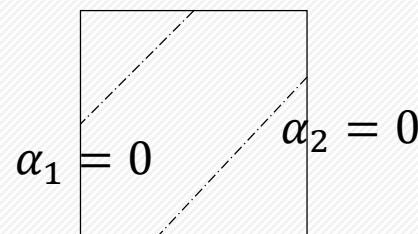
$$L = \max(0, k - C)$$

$$k = y_1 \zeta = y_1 (\alpha_1^{old} y_1 + \alpha_2^{old} y_2) = \alpha_1^{old} + \alpha_2^{old}$$

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

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

②当 $y_1 \neq y_2$ 时, 有 $\alpha_1 - \alpha_2 = y_1 \zeta = k$



$$L = \max(0, -k)$$

$$k = y_1 \zeta = y_1 (\alpha_1^{old} y_1 + \alpha_2^{old} y_2) = \alpha_1^{old} - \alpha_2^{old}$$

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

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



6.4.1 两个变量二次规划的求解方法

未经剪辑的解:

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

$$\alpha_1^{new,unc} = y_1(\zeta - \alpha_2^{new} y_2)$$

经剪辑后 α_2 的解:

$$\alpha_2^{new} = \begin{cases} H, & \alpha_2^{new,unc} > H \\ \alpha_2^{new,unc}, & L \leq \alpha_2^{new,unc} \leq H \\ L, & \alpha_2^{new,unc} < L \end{cases}$$

再由 α_2^{new} 求得剪辑后的 α_1^{new}

$$\alpha_1^{new} = y_1(\zeta - \alpha_2^{new} y_2)$$

$$= y_1 \zeta - y_1 y_2 \alpha_2^{new}$$

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

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

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



6.4.1 两个变量二次规划的求解方法

非线性支持向量机

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i$$

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

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N$$

求解优化问题，得到最优解

$$\mathbf{w}^* = \sum_{i=1}^m \alpha_i^* y_i K(\cdot, \mathbf{x}_i)$$

任选符合 $0 < \alpha_i^* < C$ 的点 (\mathbf{x}_j, y_j) ，计算

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}_i \cdot \mathbf{x}_j)$$

得决策函数

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*\right)$$



6.4.1 两个变量二次规划的求解方法

$$\alpha_1^{old}, \alpha_2^{old}, \dots, \alpha_N^{old}$$

$$\alpha_1^{new}, \alpha_2^{new}, \dots, \alpha_N^{old}$$

1) 若 $0 < \alpha_1^{new} < C$, 则 (x_1, y_1) 是支持向量

$$b_1^{new} = y_1 - \sum_{i=1}^N \alpha_i^* y_i K(x_i \cdot x_1)$$

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

$$\because E_1 = \sum_{i=1}^N \alpha_i^{old} y_i K_{i1} + b^{old} - y_1$$

$$= (\sum_{i=3}^N \alpha_i^{old} y_i K_{i1} - y_1) + \alpha_1^{old} y_1 K_{11} + \alpha_2^{old} y_2 K_{21} + b^{old}$$

$$\begin{aligned} \therefore b_1^{new} &= -E_1 - y_1 K_{11} (\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{21} (\alpha_2^{new} - \alpha_2^{old}) \\ &\quad + b^{old} \end{aligned}$$

2) 若 $0 < \alpha_2^{new} < C$, 则 (x_2, y_2) 是支持向量

$$b_2^{new} = -E_2 - y_1 K_{12} (\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{22} (\alpha_2^{new} - \alpha_2^{old})$$

3) 若 $0 < \alpha_1^{new} < C$ 且 $0 < \alpha_2^{new} < C$, 则 (x_1, y_1) 和 (x_2, y_2) 均为支持向量, 可以任取一个来更新 b 的值, 且计算结果一定有 $b_1^{new} = b_2^{new}$ 。

4) 若 $\alpha_1^{new}, \alpha_2^{new}$ 的值为 0 或 C , 那么 b_1^{new}, b_2^{new} 以及它们之间的数都是符合 KKT 条件的阈值, 这时可以选择它们的中点作为 b^{new} 。

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

$$\text{更新 } E_i^{new} = \sum_{x_j \in S} y_j \alpha_j K_{ij} + b^{new} - y_i$$



6.4.2 变量的选择方法

1. 第一个变量的选择

$$y_i g(x_i) = \begin{cases} > 1 & \alpha_i = 0 \\ = 1 & 0 < \alpha_i < C \\ < 1 & \alpha_i = C \end{cases}$$

选择违反KKT条件最厉害的变量作为 α_1 。

1. 第2个变量的选择

假设外循环中选择的变量为 α_1

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

j 从2变到 N , 寻找差值 $|E_1 - E_2|$ 最大的那个 i 作为 α_2



6.4.3 SMO算法

算法6.5 SMO算法

输入：训练数据集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$, 精度 ε 。

输出：近似解 $\hat{\alpha}$ 。

- (1) 取初始值 $\alpha^{(0)} = \mathbf{0}$, 令 $k = 0$;
- (2) 选取变量 $\alpha_1^{(k)}, \alpha_2^{(k)}$, 求解两个变量的最优化问题, 得最优解 $\alpha_1^{(k+1)}, \alpha_2^{(k+1)}$, 更新 α 为 $\alpha^{(k+1)}$;
- (3) 若在精度 ε 范围内满足停机条件

$$\sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N$$

$$y_i g(\mathbf{x}_i) = \begin{cases} > 1 & \alpha_i = 0 \\ = 1 & 0 < \alpha_i < C \\ < 1 & \alpha_i = C \end{cases}$$

则转(4); 否则令 $k = k + 1$, 转(2);

- (4) 取 $\hat{\alpha} = \alpha^{(k+1)}$ 。

正则化(regularization)

统计学习模型（例如 SVM）的更一般形式

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

归纳偏好

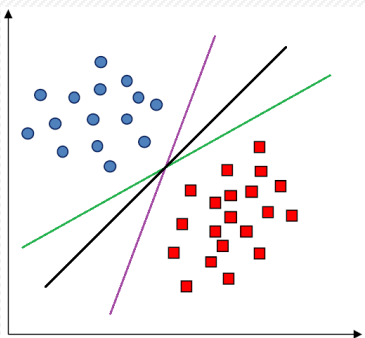
结构风险
(structural risk)
描述模型本身的某些性质

经验风险
(empirical risk)
描述模型与训练数据的契合程度

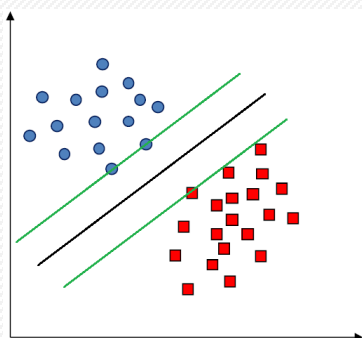
- 正则化可理解为“**罚函数法**”：通过对不希望的结果施以惩罚，使得优化过程趋向于希望目标。
- 从贝叶斯估计的角度，则可认为是提供了模型的先验概率。



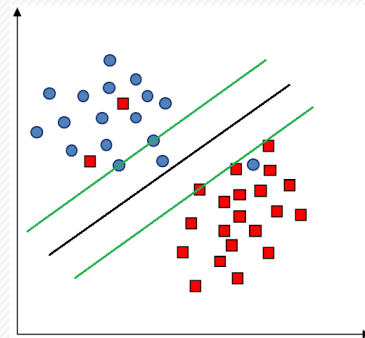
小结



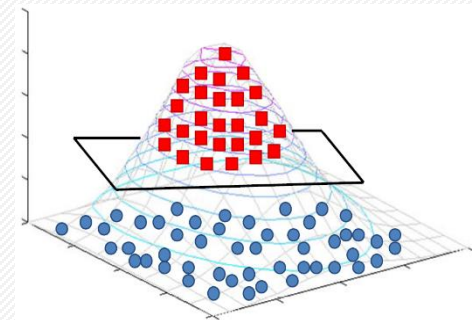
$$f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$



$$f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$



$$f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$



$$f(x) = \text{sign}(\mathbf{w} \cdot \phi(\mathbf{x}) + b)$$

感知机

线性可分
支持向量机

线性不可分
支持向量机

非线性
支持向量机

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \\ & i = 1, 2, \dots, N \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \dots, N \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \dots, N \end{aligned}$$

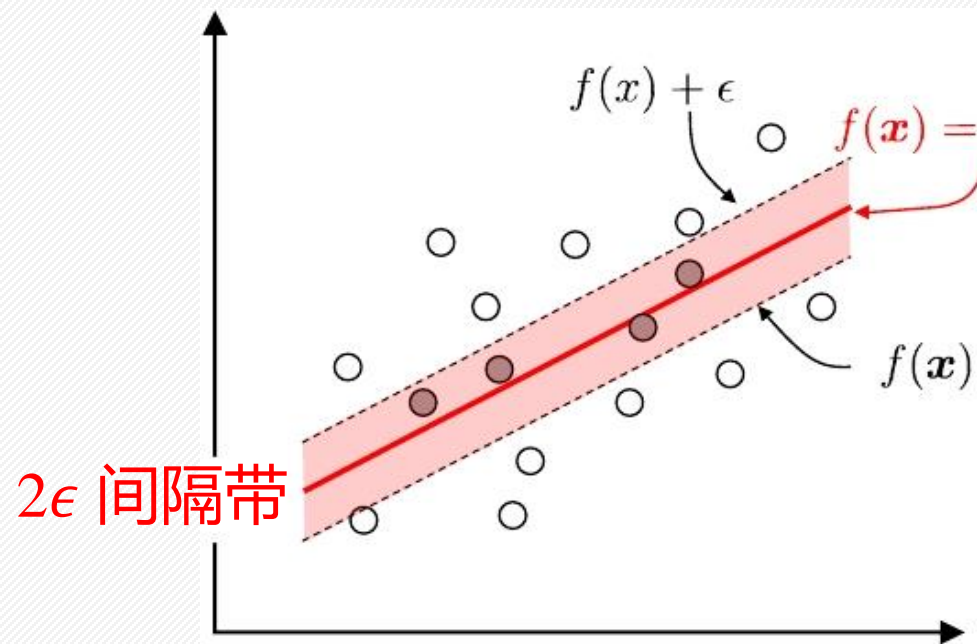
6.5 支持向量回归(SVR)

训练样本集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, $\mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathbb{R}$

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

基本思路：支持向量回归 (Support Vector Regression, SVR) 假设能容忍 $f(\mathbf{x})$ 与 y 之间最多有 ϵ 的偏差，即仅当 $f(\mathbf{x})$ 与 y 之间的差别绝对值大于 ϵ 时才计算损失。

相当于以 $f(\mathbf{x})$ 为中心，构建了一个宽度为 2ϵ 的间隔带，若训练样本落入此间隔带，则认为预测正确。



6.5 支持向量回归(SVR)

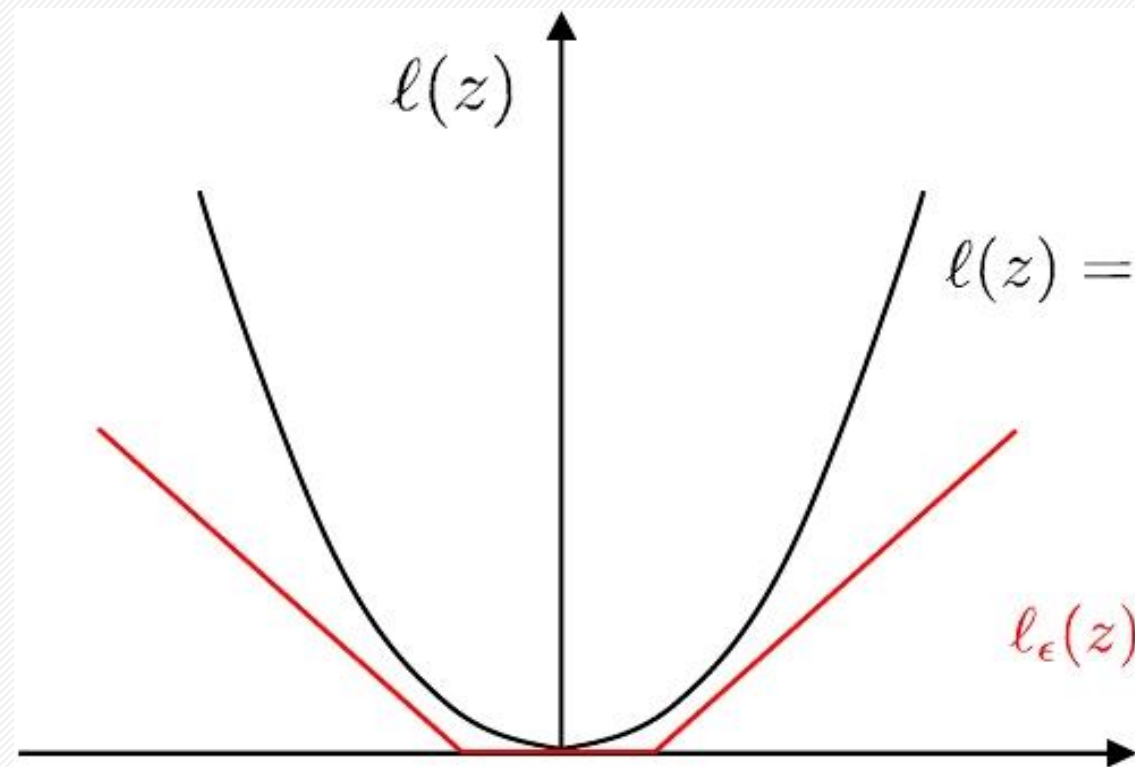
$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \ell_{\epsilon}(f(\mathbf{x}_i) - y_i)$$

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

引入松弛向量 ξ_i 和 $\hat{\xi}_i$, 则有

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

$$\begin{aligned} \text{s.t. } & f(\mathbf{x}_i) - y_i \leq \epsilon + \xi_i \\ & y_i - f(\mathbf{x}_i) \leq \epsilon + \hat{\xi}_i \\ & \xi_i \geq 0, \hat{\xi}_i \geq 0, i = 1, 2, \dots, N \end{aligned}$$



ϵ -不敏感损失函数

6.5 支持向量回归(SVR)

通过引入拉格朗日乘子 μ_i 和 $\hat{\mu}_i$ ，得到拉格朗日函数：

$$L(\mathbf{w}, b, \alpha, \hat{\alpha}, \mu, \hat{\mu}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \hat{\xi}_i) - \sum_{i=1}^N \mu_i \xi_i - \sum_{i=1}^N \hat{\mu}_i \hat{\xi}_i + \sum_{i=1}^m \alpha_i (f(\mathbf{x}_i) - y_i - \epsilon - \xi_i) + \sum_{i=1}^N \hat{\alpha}_i (y_i - f(\mathbf{x}_i) - \epsilon - \hat{\xi}_i) \quad (6.46)$$

令 $L(\mathbf{w}, b, \alpha, \hat{\alpha}, \mu, \hat{\mu})$ 对 \mathbf{w} , b , ξ_i 和 $\hat{\xi}_i$ 的偏导数等于零，可得

$$\mathbf{w} = \sum_{i=1}^N (\hat{\alpha}_i - \alpha_i) \mathbf{x}_i \quad (6.47)$$

$$0 = \sum_{i=1}^N (\hat{\alpha}_i - \alpha_i) \quad (6.48)$$

$$C = \alpha_i + \xi_i \quad (6.49)$$

$$C = \hat{\alpha}_i + \hat{\xi}_i \quad (6.50)$$

6.5 支持向量回归(SVR)

对偶问题

$$\max_{\alpha, \hat{\alpha}} \sum_{i=1}^N y_i (\hat{\alpha}_i - \alpha_i) - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\hat{\alpha}_i - \alpha_i) \mathbf{x}_i^T \mathbf{x}_j \quad (6.51)$$

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

满足KKT条件，即要求

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

预测

$$f(\mathbf{x}) = \sum_{i=1}^N (\hat{\alpha}_i - \alpha_i) \mathbf{x}_i^T \mathbf{x} + b \quad (6.53)$$

$$f(\mathbf{x}) = \sum_{i=1}^N (\hat{\alpha}_i - \alpha_i) k(\mathbf{x}_i, \mathbf{x}) + b$$



6.6 核方法

核SVM: $f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{i=1}^m \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b$

核SVR: $f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{i=1}^m (\hat{\alpha}_i - \alpha_i) k(\mathbf{x}, \mathbf{x}_i) + b$

无论 SVM 还是 SVR, 学得模型总能表示为核函数的线性组合

表示定理: 令 \mathbb{H} 为核函数 k 对应的再生核希尔伯特空间, 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

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

的解总可写为

$$h^*(\mathbf{x}) = \sum_{i=1}^m \alpha_i k(\mathbf{x}, \mathbf{x}_i)$$

6.6 核方法(Kernel methods)

基于表示定理能得到很多线性模型的“核化”(kernelized)版本

例如 KLDA (Kernelized LDA):

通过某种映射 $\phi: \mathcal{X} \rightarrow \mathbb{F}$ 将样本映射到高维特征空间 \mathbb{F} , 然后在此特征空间 \mathbb{F} 中执行线性判别分析, 求得

$$h(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

类似地, KLDA的学习目标为

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{\mathbf{w}^T S_b^{\phi} \mathbf{w}}{\mathbf{w}^T S_w^{\phi} \mathbf{w}}$$

其中 S_b^{ϕ} 和 S_w^{ϕ} 分别为训练样本在特征空间 \mathbb{F} 中的类间散度矩阵和类内散度矩阵。

6.6 核方法(Kernel methods)

设第 i 类样本在特征空间 \mathbb{F} 中的均值为:

$$\mu_i^\phi = \frac{1}{m_i} \sum_{x \in X_i} \phi(x)$$

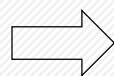
两个散度矩阵分别为:

$$S_b^\phi = (\mu_1^\phi - \mu_0^\phi)(\mu_1^\phi - \mu_0^\phi)^T$$

$$S_w^\phi = \sum_{i=0}^1 \sum_{x \in X_i} (\phi(x) - \mu_i^\phi)(\phi(x) - \mu_i^\phi)^T$$

使用核函数 $k(x, x_i) = \phi(x_i)^T \phi(x)$ 来隐式地表示这个映射和特征空间 \mathbb{F} , 使用

$$h(x) = \mathbf{w}^T \phi(x) = \sum_{i=1}^m \alpha_i K(x, x_i)$$



$$\mathbf{w} = \sum_{i=1}^m \alpha_i \phi(x_i)$$

6.6 核方法(Kernel methods)

设 $K \in \mathbb{R}^{m \times m}$ 为核函数对应的核矩阵, $\mathbf{1}_i \in \{1,0\}^{m \times 1}$ 为第 i 类样本的指示向量, 令

$$\hat{\mu}_0 = \frac{1}{m_0} \mathbf{K} \mathbf{1}_0$$

$$\hat{\mu}_1 = \frac{1}{m_1} \mathbf{K} \mathbf{1}_1$$

$$\mathbf{M} = (\hat{\mu}_0 - \hat{\mu}_1)(\hat{\mu}_0 - \hat{\mu}_1)^T$$

$$\mathbf{N} = \mathbf{K} \mathbf{K}^T - \sum_{i=0}^1 m_i \hat{\mu}_i \hat{\mu}_i^T$$

于是,

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

$$h(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

“核技巧” (kernel trick) 是机器学习中处理非线性问题的基本技术之一。

|| SVM与神经网络（NN）的对比

- ◆ SVM的理论基础比NN更坚实，更像一门严谨的“科学”（三要素：问题的表示、问题的解决、证明）
 - ◆ SVM —— 严格的数学推理
 - ◆ NN —— 强烈依赖于工程技巧
- ◆ 推广能力取决于“经验风险值”和“置信范围值”，NN不能控制两者中的任何一个。
- ◆ NN设计者用高超的工程技巧弥补了数学上的缺陷——设计特殊的结构，利用启发式算法，有时能得到出人意料的好结果。

|| SVM与神经网络（NN）的对比

“我们必须从一开始就澄清一个观点，就是如果某事不是科学，它并不一定不好。比如说，爱情就不是科学。因此，如果我们说某事不是科学，并不是说它有什么不对，而只是说它不是科学。”

—— *by R. Feynman*

from *The Feynman Lectures on Physics*, Addison-Wesley

同理，与SVM相比，NN不像一门科学，更像一门工程技巧，但并不意味着它就一定不好！



SVM小结

■优点

- 具有较好的泛化能力，特别是在小样本训练集上能够得到比其他算法好很多的结果。这是因为其优化目标是结构风险最小，而不是经验风险最小，因此通过margin，可以得到对数据分布的结构化描述，从而降低了数据规模和数据分布的要求。
- 具有较强的数学理论支撑，基本不涉及概率测度和大数定律等。
- 引入核函数可以解决非线性分类问题。

■缺点

- 不方便解决多分类问题。
- 存在对结果影响较大的超参数，比如采用rbf核函数时，惩罚项 C 和核参数 γ ，只能通过穷举或经验推测获得。

LIBSVM <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

`sklearn.svm`: Support Vector Machines

The `sklearn.svm` module includes Support Vector Machine algorithms.

User guide: See the [Support Vector Machines](#) section for further details.

Estimators ¶

`svm.LinearSVC`([penalty, loss, dual, tol, C, ...]) Linear Support Vector Classification.

`svm.LinearSVR`(*[, epsilon, tol, C, loss, ...]) Linear Support Vector Regression.

`svm.NuSVC`(*
[, nu, kernel, degree, gamma, ...]) Nu-Support Vector Classification.

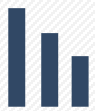
`svm.NuSVR`(*
[, nu, C, kernel, degree, gamma, ...]) Nu Support Vector Regression.

`svm.OneClassSVM`(*
[, kernel, degree, gamma, ...]) Unsupervised Outlier Detection.


`svm.SVC`(*[, C, kernel, degree, gamma, ...]) C-Support Vector Classification.

`svm.SVR`(*
[, kernel, degree, gamma, coef0, ...]) Epsilon-Support Vector Regression.

`svm.l1_min_c`(X, y, *
[, loss, fit_intercept, ...]) Return the lowest bound for C such that for C in (l1_min_C, infinity) the model is guaranteed not to be empty.



sklearn.svm.SVC

```
class sklearn.svm.SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale', 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', break_ties=False,
random_state=None) 
```

[\[source\]](#)

Methods

<code>decision_function(X)</code>	Evaluates the decision function for the samples in X.
<code>fit(X, y[, sample_weight])</code>	Fit the SVM model according to the given training data.
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>predict(X)</code>	Perform classification on samples in X.
<code>score(X, y[, sample_weight])</code>	Return the mean accuracy on the given test data and labels.
<code>set_params(**params)</code>	Set the parameters of this estimator.



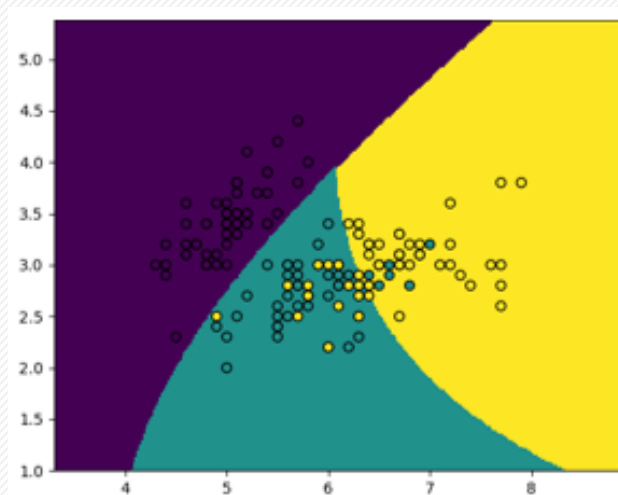
例6.6

```
>>> import numpy as np
>>> from sklearn.pipeline import make_pipeline
>>> from sklearn.preprocessing import StandardScaler
>>> X = np.array([[-1, -1], [-2, -1], [1, 1], [2, 1]])
>>> y = np.array([1, 1, 2, 2])
>>> from sklearn.svm import SVC
>>> clf = make_pipeline(StandardScaler(), SVC(gamma='auto'))
>>> clf.fit(X, y)
Pipeline(steps=[('standardscaler', StandardScaler()),
                 ('svc', SVC(gamma='auto'))])
```

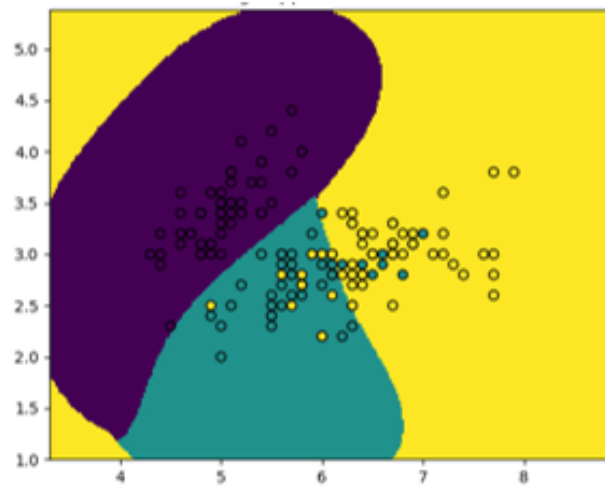
```
>>> print(clf.predict([[-0.8, -1]]))
[1]
```



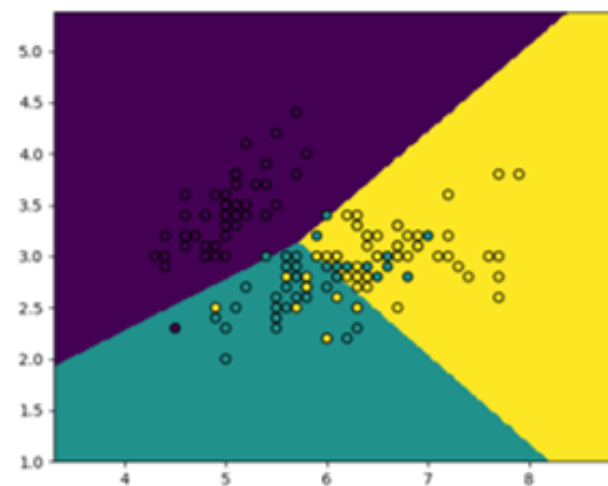
鸢尾花数据集分类



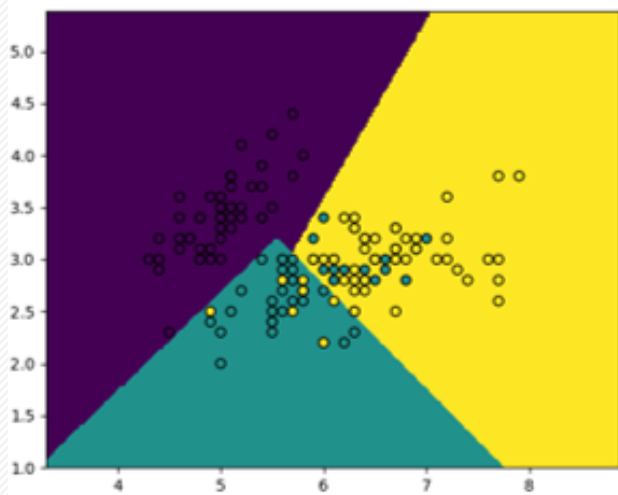
SVC(多项式poly)



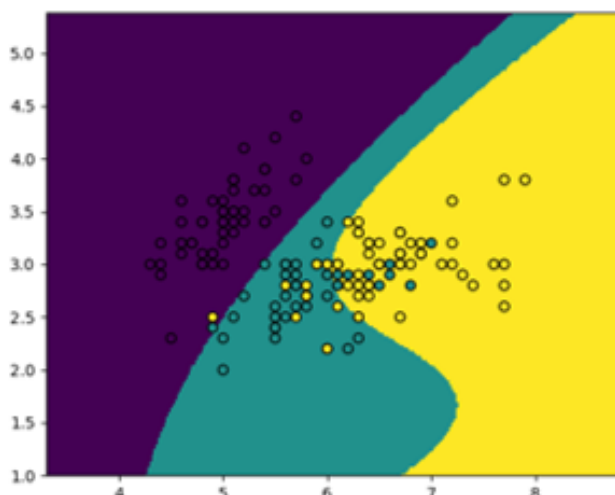
SVC(径向基rbf,gamma=1)



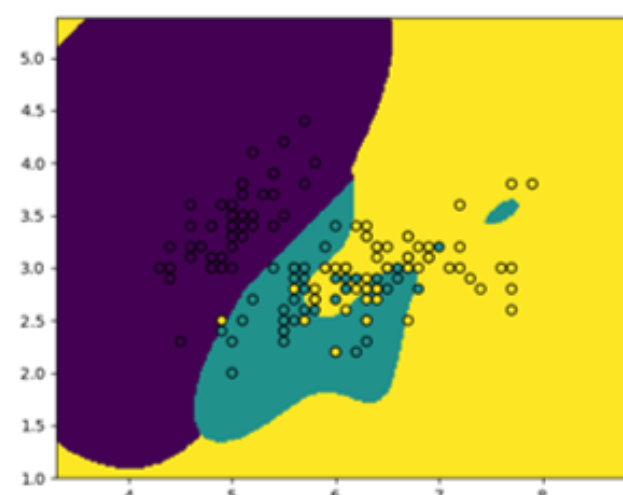
LinearSVC



NuSVC(线性linear)



NuSVC(多项式poly,gamma=1)



NuSVC(径向基rbf)



Faces recognition example using eigenfaces and SVMs

predicted: Bush
true: Bush



predicted: Bush
true: Bush



predicted: Blair
true: Blair



predicted: Bush
true: Bush



predicted: Bush
true: Bush



predicted: Bush
true: Bush



predicted: Schroeder
true: Schroeder



predicted: Powell
true: Powell



predicted: Bush
true: Bush



predicted: Bush
true: Bush



predicted: Bush
true: Bush



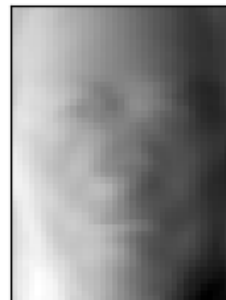
predicted: Bush
true: Bush



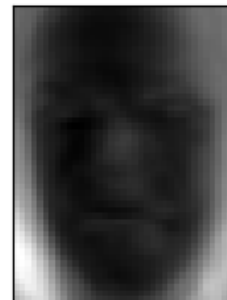
eigenface 0



eigenface 1



eigenface 2



eigenface 3



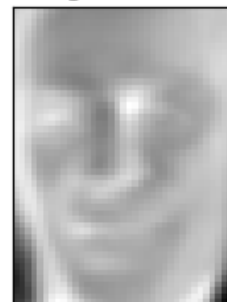
eigenface 4



eigenface 5



eigenface 6



eigenface 7



eigenface 8



eigenface 9



eigenface 10



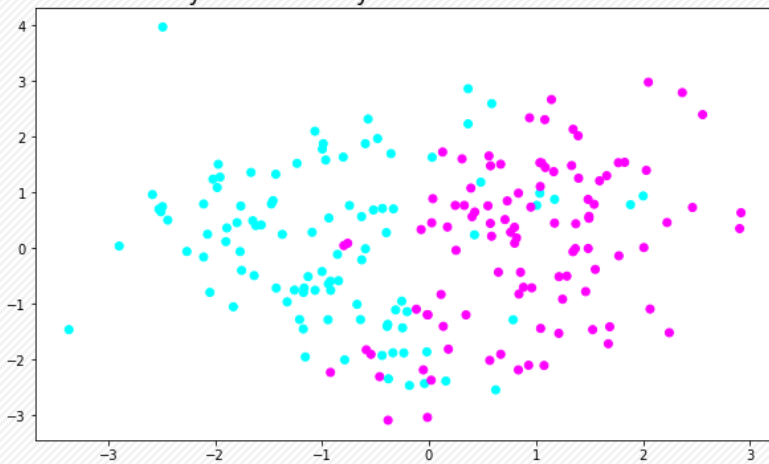
eigenface 11



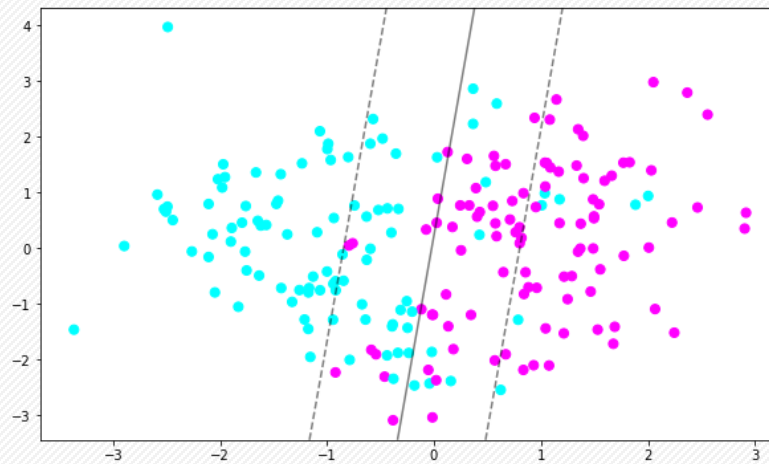


超参数C和gamma的影响

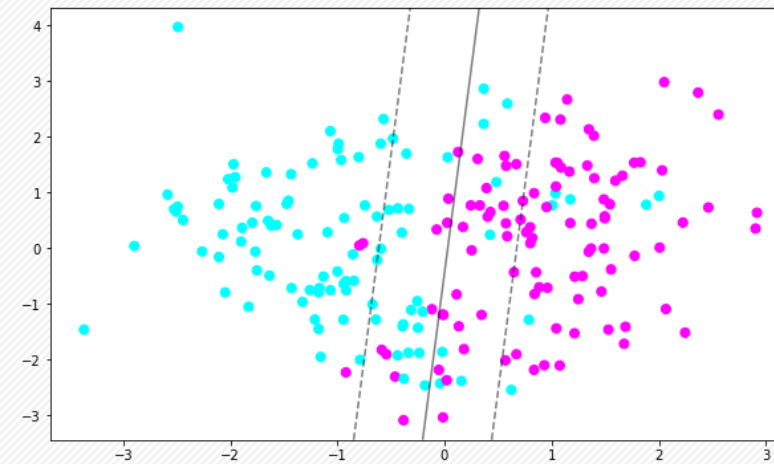
Synthetic Binary Classification Dataset



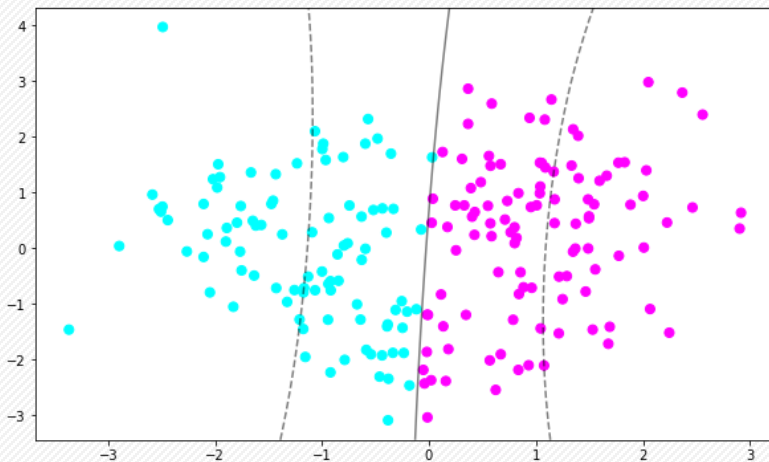
Linear kernel with $C=0.1$



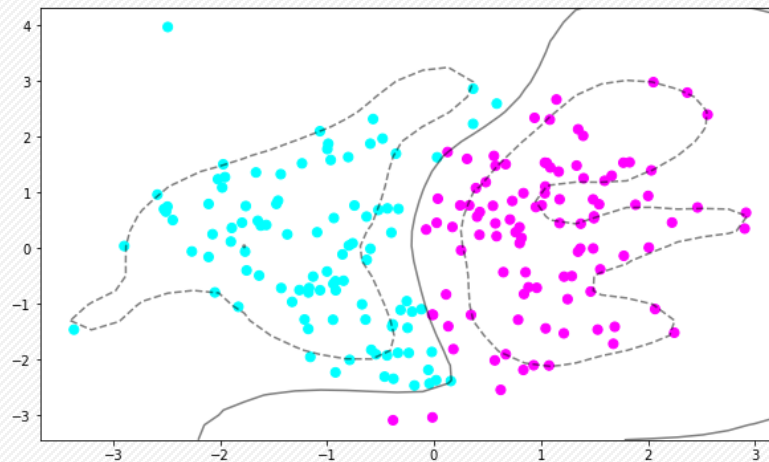
Linear kernel with $C=0.1$



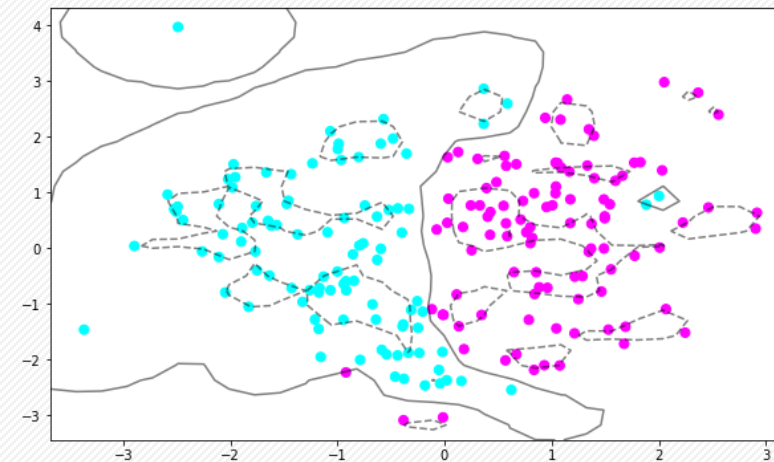
Predictions of RBF kernel with $C=1$ and $\Gamma=0.01$

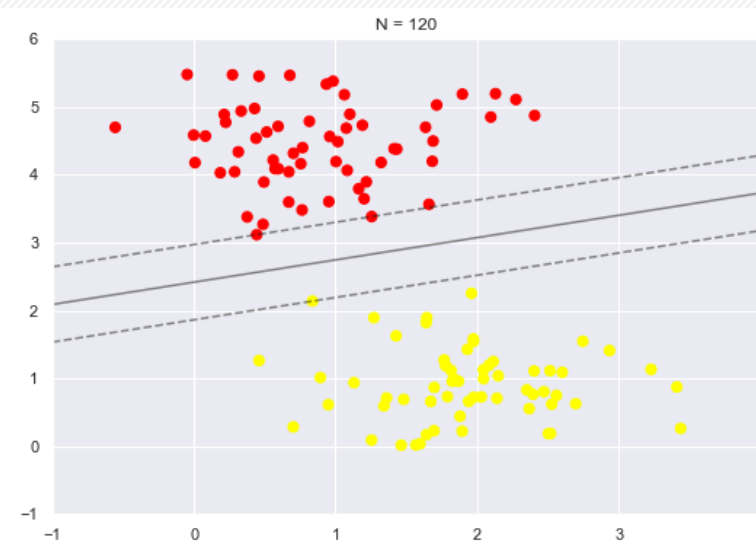
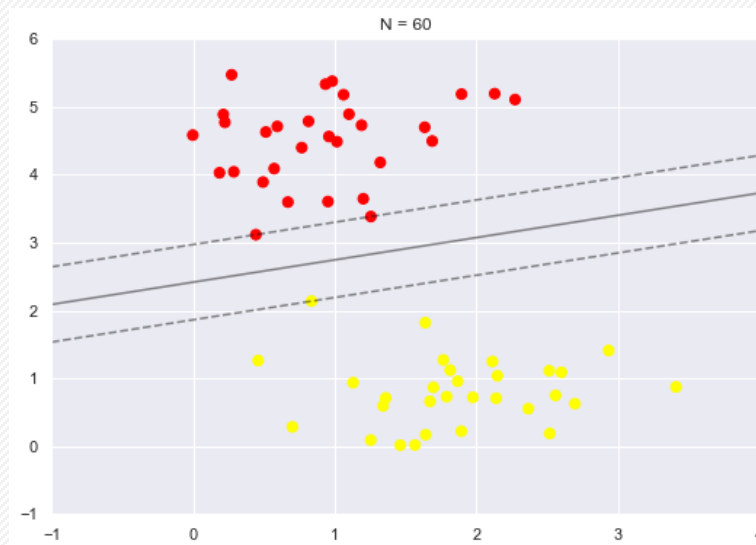
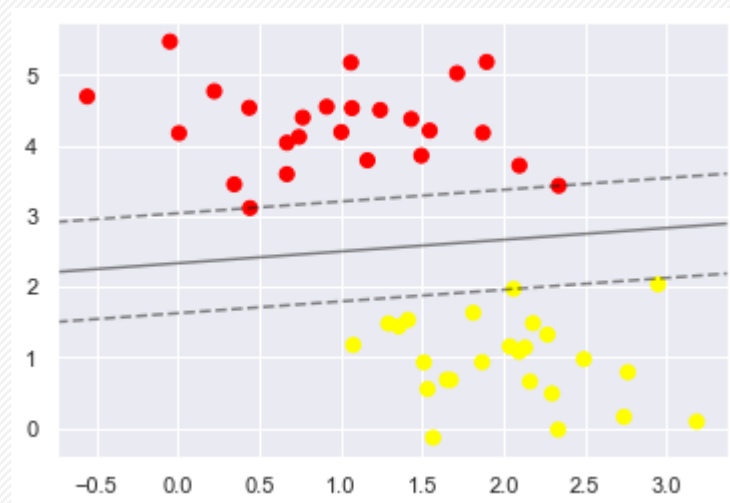
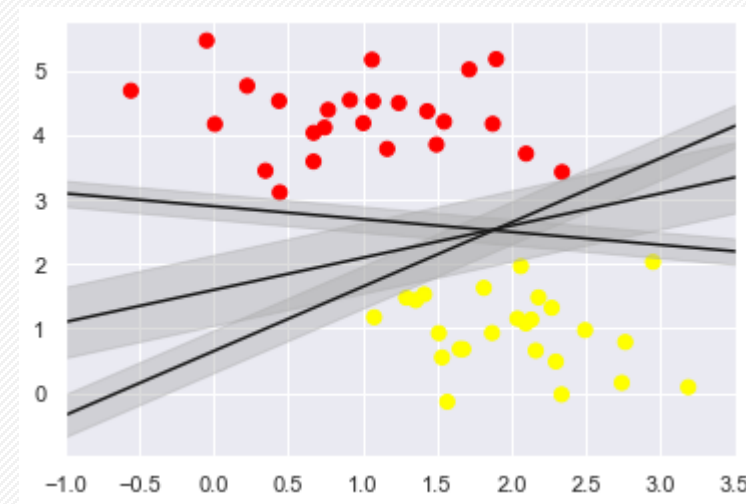
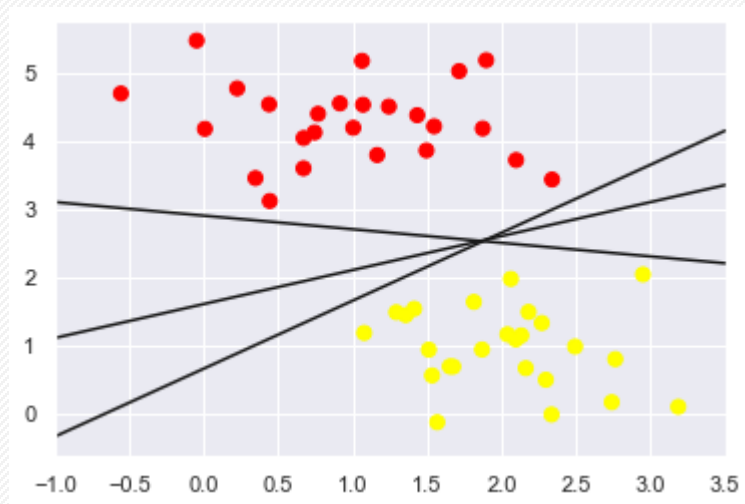
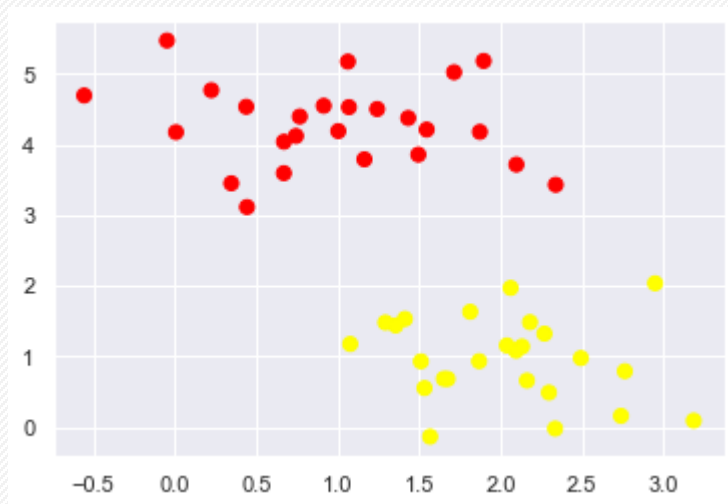


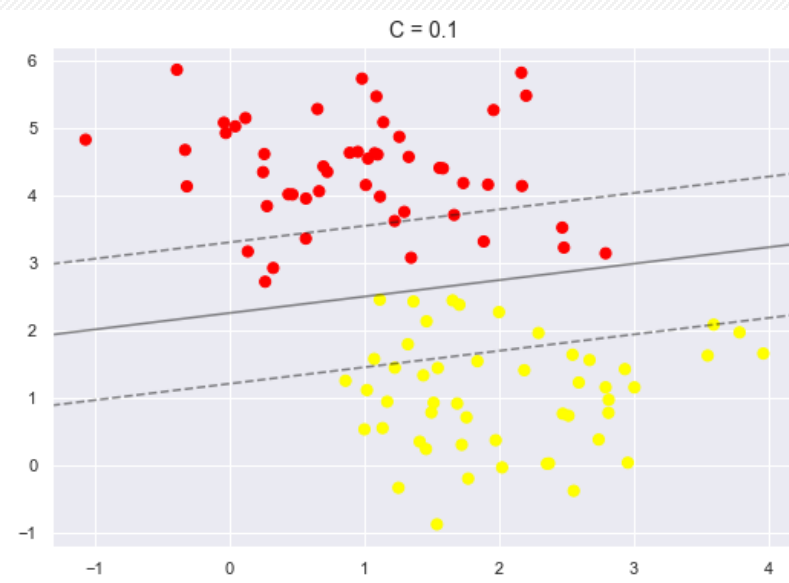
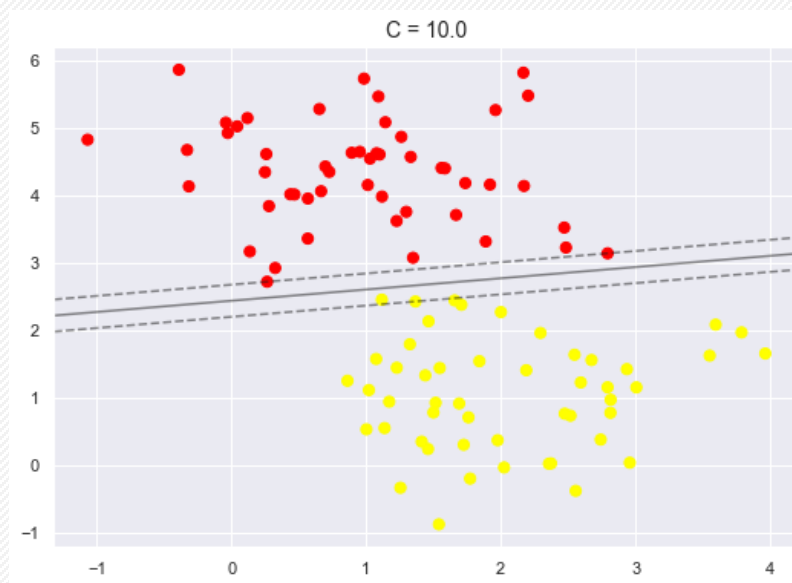
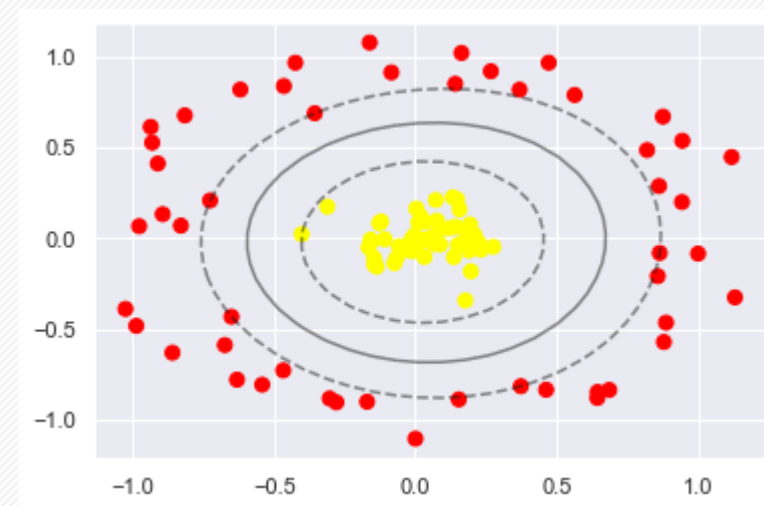
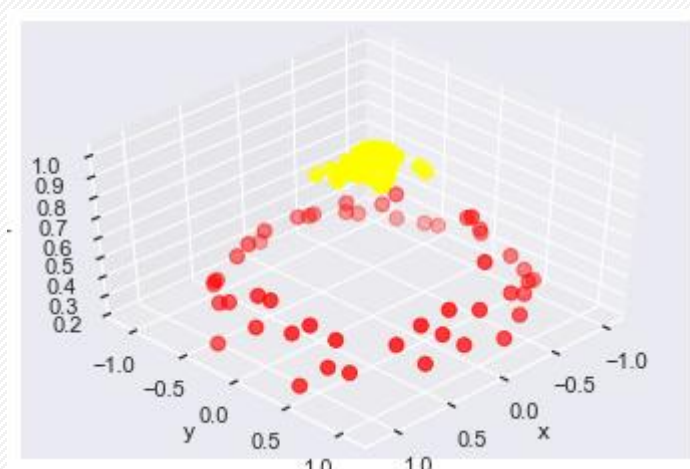
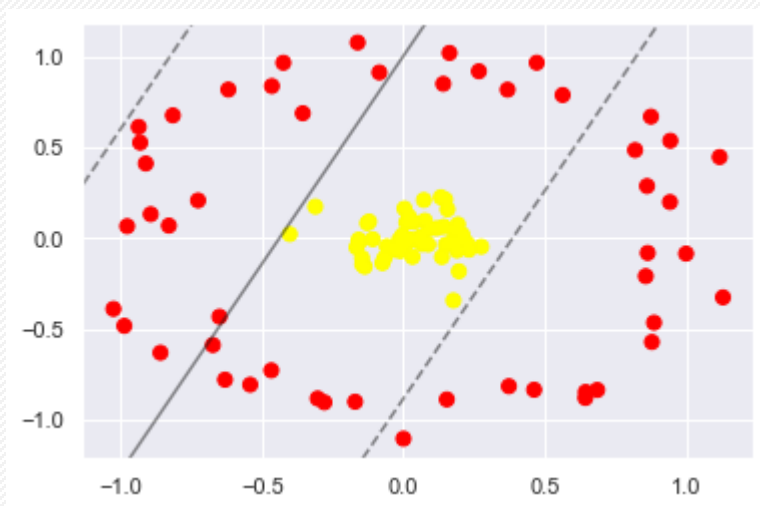
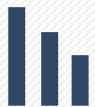
Predictions of RBF kernel with $C=1$ and $\Gamma=0.01$

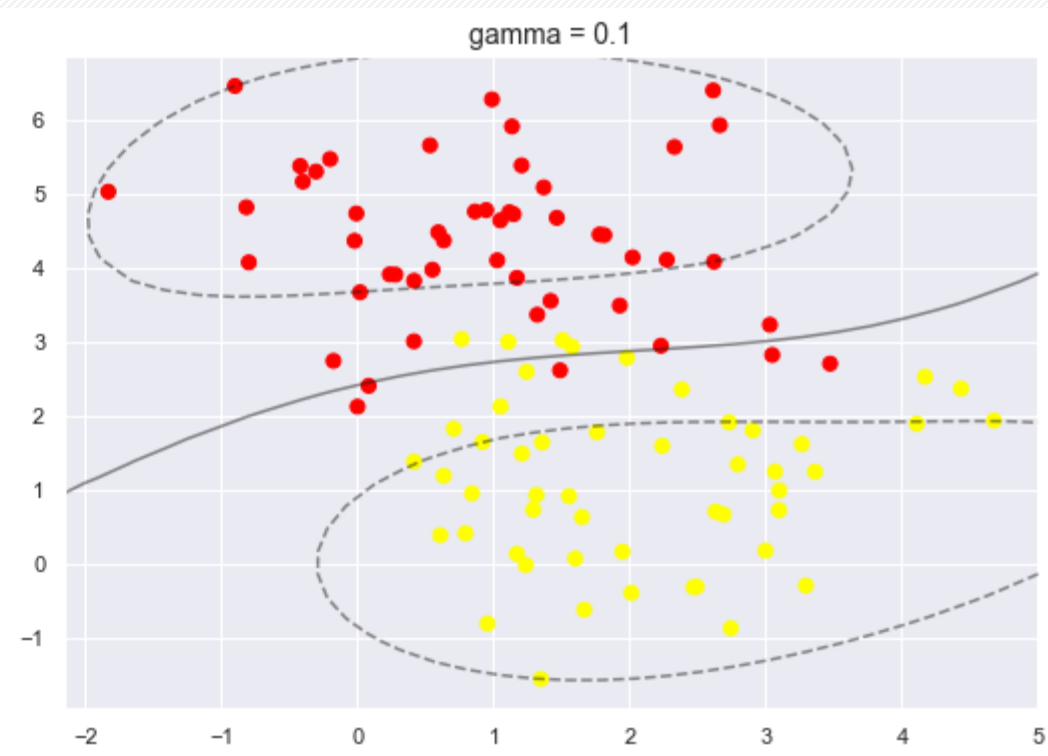
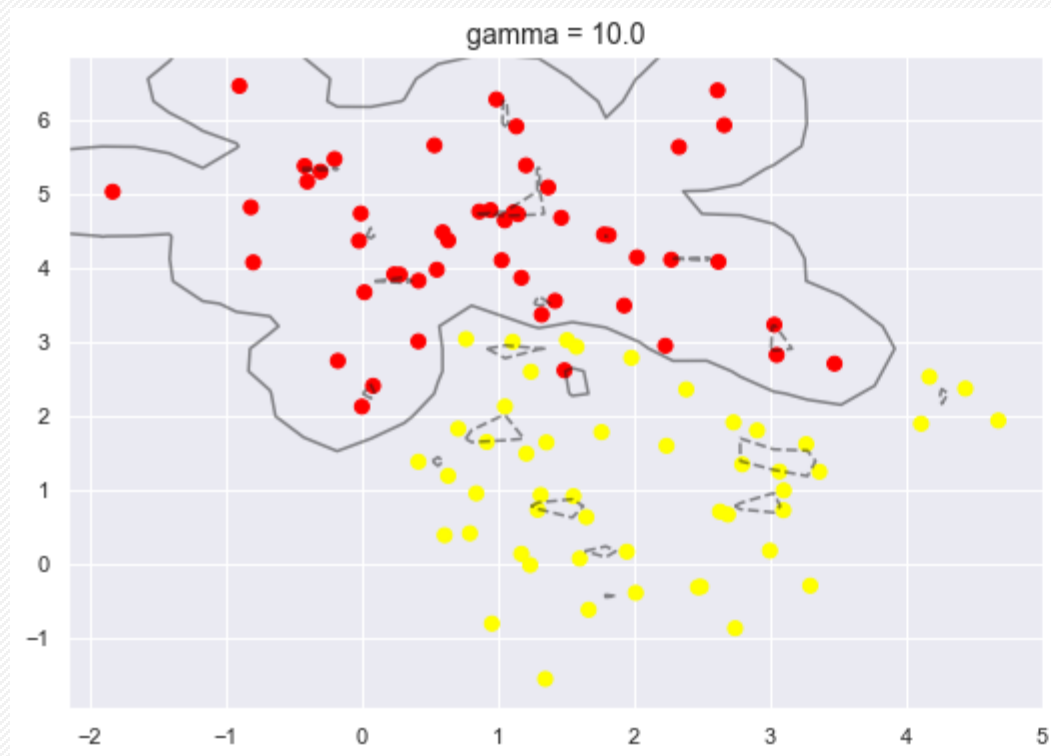
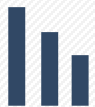


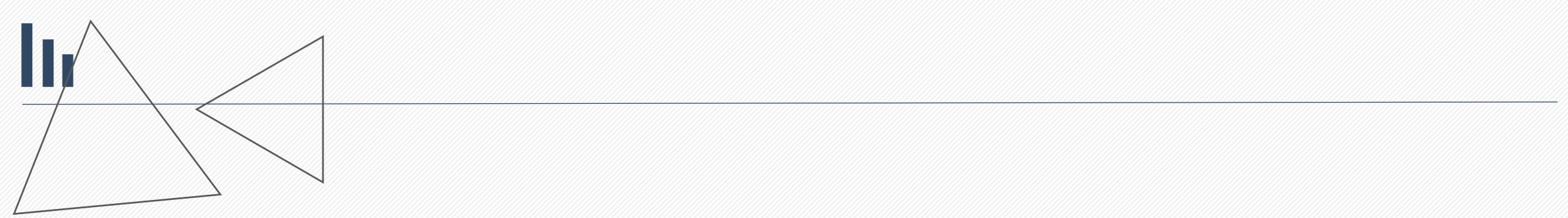
Predictions of RBF kernel with $C=1$ and $\Gamma=0.01$











The end

