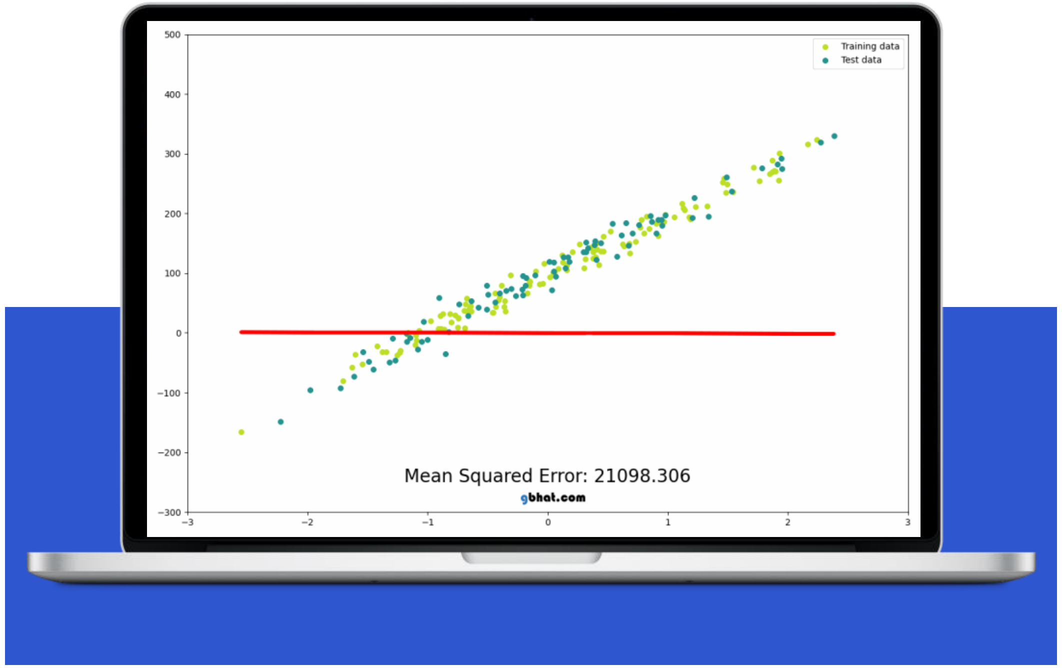
《机器学习》

第九章 支持向量机

机器学习就是对计算机一部分数据进行学习，然后对另外一些数据进行预测与判断。
核心是“使用算法解析数据，从中学习，然后对新数据做出决定或预测”。

黄亮 2023年11月

上节回顾



监督学习

回归 Regression

分类 Classification

线性回归

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

逻辑回归

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

1.分类问题

分类预测 离散标签

(a) 公共安全

(b) 金融风控

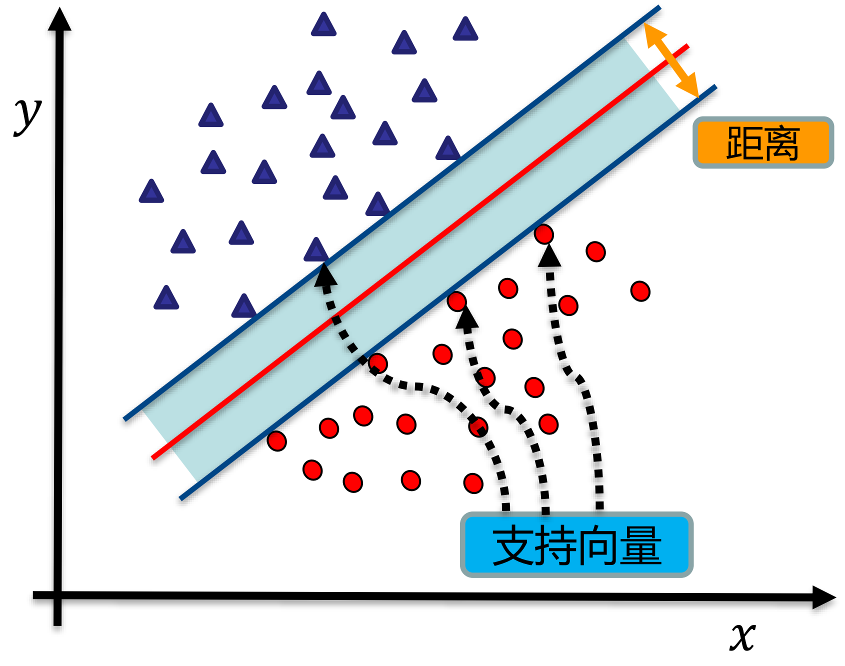
(c) 媒体监管

目录/CONTENTS

目录/CONTENTS

支持向量机概述

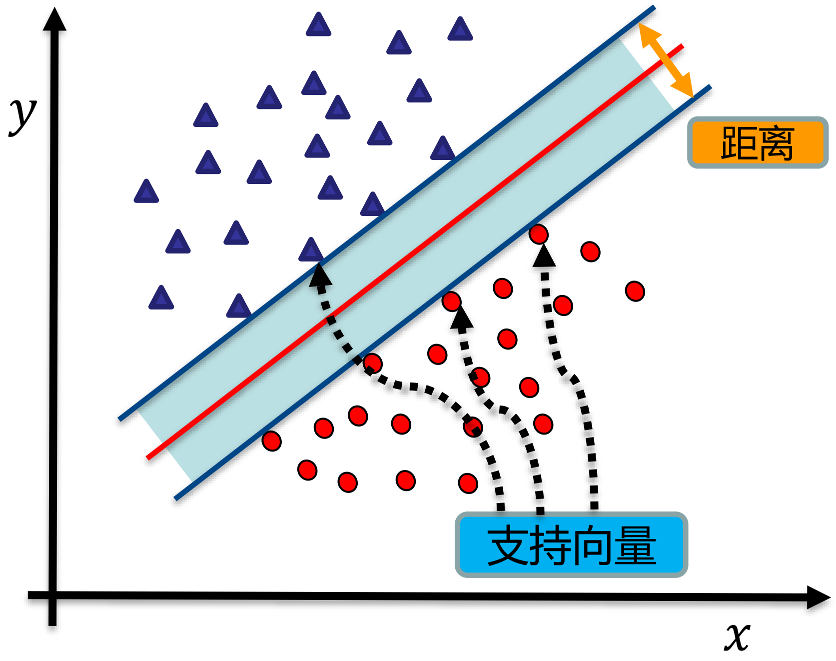
- **支持向量机（Support Vector Machine, SVM）**是一类按监督学习方式对数据进行二元分类的广义线性分类器。
  - 其决策边界是对学习样本求解的最大边距超平面（maximum-margin hyperplane）。
  - 与逻辑回归和神经网络相比，支持向量机，在学习复杂的非线性方程时提供了一种更为清晰，更加强大的方式。



支持向量机概述

算法思想

- 找到集合边缘上的若干数据，称为**支持向量**（Support Vector）。
- 用这些点找出一个**超平面**。
- 使支持向量到该平面的距离最大。



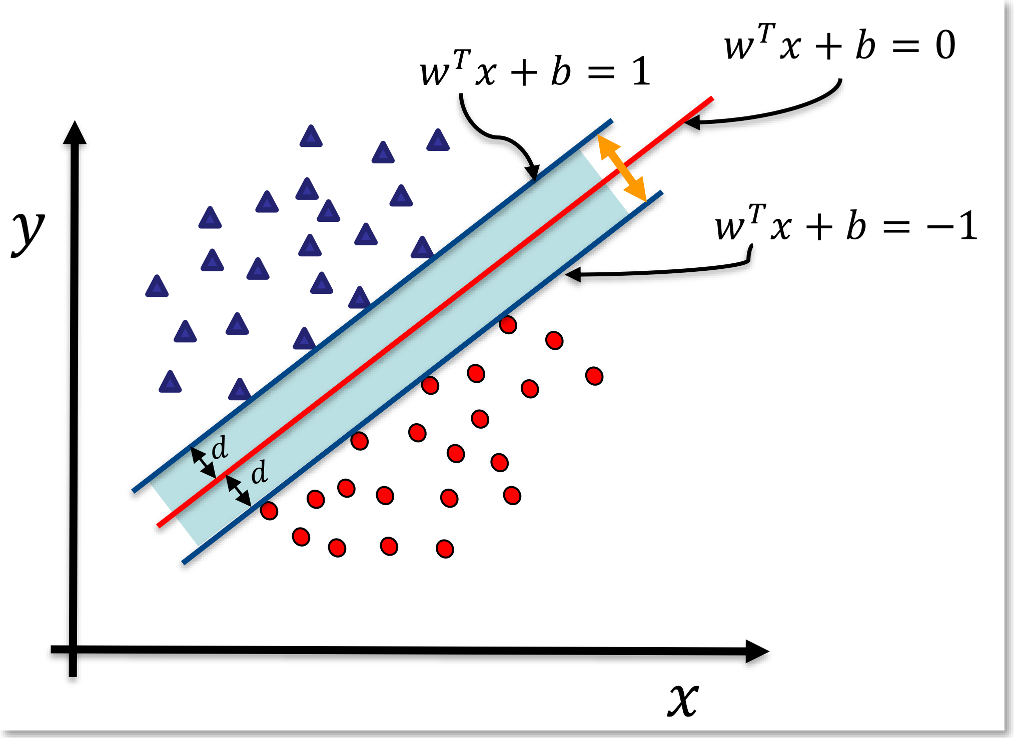
目录/CONTENTS

线性可分支持向量机

背景知识

支持向量点到超平面的距离公式

$$d = \frac{|w^T x + b|}{||w||}$$



如图所示，其他点到超平面的距离大于 d。

$$\begin{cases} \dfrac{w^T x + b}{\| w \|} \geq d, & y = 1 \\ \dfrac{w^T x + b}{\| w \|} \leq -d, & y = -1 \end{cases}$$

$$y(w^T x + b) = |w^T x + b|$$

因此，最终目的是最大化 $d = \frac{1}{||w||} y(w^T x + b)$

支持向量机求解

函数间隔：

$$d^* = y_i(w^T x + b)$$

几何间隔（点到超平面的距离）：

$$d = \frac{y(w^T x + b)}{||w||} = \frac{d^*}{||w||}$$

为了求几何间隔最大，SVM基本问题可以转化为求解

$$\max_{w,b} \frac{d^*}{||w||}$$

约束条件：$y_i(w^T x_i + b) \geq d^*,\ i = 1,2,..,m$

支持向量机求解

转化为凸函数

先令 $d^*=1$，方便计算（参照衡量，不影响评价结果）

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

$$s.t.\ y_i(w^T x_i + b) \geq 1,\ i = 1,2,..,m$$

再将 $\max\limits_{w,b} \frac{1}{||w||}$ 转化成 $\min\limits_{w,b} \frac{1}{2}||w||^2$ 求解凸函数，1/2是为了求导之后方便计算。

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

$$s.t.\ y_i(w^T x_i + b) \geq 1,\ i = 1,2,..,m$$

支持向量机求解

求解凸优化问题

用拉格朗日乘子法：

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

其中 α_i 为拉格朗日乘子。最优解满足对 w,b 求导等于零。

$$\frac{\partial}{\partial w}L(w,b,\alpha)=w-\sum_{i=1}^{m}\alpha_i\,y_ix_i=0$$

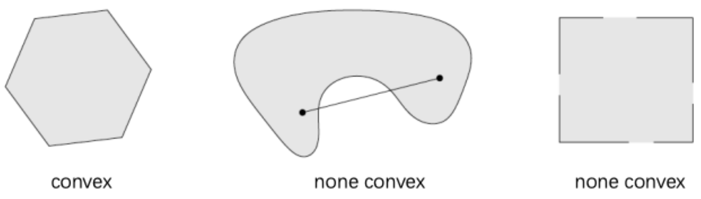
KKT条件求解最优值

$$w^*=\sum_{i=1}^{m}\alpha_i^*\,y_ix_i$$

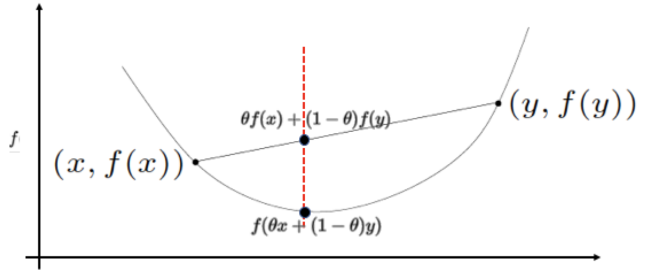
其中$\alpha^*=(\alpha_1^*,\alpha_2^*,\dots,\alpha_m^*)^T$，**支持向量$\alpha_i^*>0$，其它向量$\alpha_i^*=0$**

凸函数

- 凸集 Convex set



- 凸函数 Convex function



- 优点： 求导=0，得到全局最优解

凸优化及对偶问题

- 原问题

$$\begin{cases} \min f(x) \\ s.t. \quad g_i(x) \le 0, \quad i=1,\dots,m \\ \qquad h_i(x)=0, \quad i=1,\dots,n \end{cases}$$

- 拉格朗日表达式

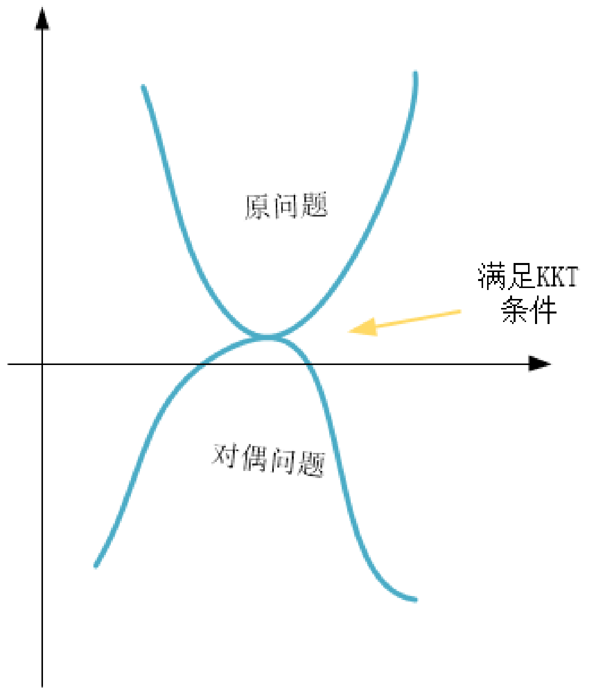
$$L(x,\alpha,\beta) = f(x) + \sum_i \alpha_i g_i(x) + \sum_i \beta_i h_i(x) \quad s.t.\ \alpha_i \ge 0$$

- 对偶问题转换

$$\min_x \max_{\alpha,\beta \ \alpha\ge 0} L(x,\alpha,\beta) \to \max_{\alpha,\beta \ \alpha\ge 0} \min_x L(x,\alpha,\beta)$$

- 转换依据

$$\min_x \max_{\alpha,\beta \ \alpha\ge 0} L(x,\alpha,\beta) \ge \max_{\alpha,\beta \ \alpha\ge 0} \min_x L(x,\alpha,\beta)$$



对偶问题

$\boxed{w = \sum_{i=1}^{m} \alpha_i y_i x_i}$ $\boxed{\sum_{i=1}^{m} \alpha_i y_i = 0}$

代入$L(w, b, \alpha)$

$$\min_{w,b} L(w,b,\alpha) = \frac{1}{2}||w||^2 + \sum_{i=1}^{m} \alpha_i \left(-y_i(w^T x_i + b) + 1\right)$$

$$= \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i y_i w^T x_i - b \boxed{\sum_{i=1}^{m} \alpha_i y_i} + \sum_{i=1}^{m} \alpha_i$$

$$= \frac{1}{2} w^T \boxed{\sum_{i=1}^{m} \alpha_i y_i x_i} - \sum_{i=1}^{m} \alpha_i y_i w^T x_i + \sum_{i=1}^{m} \alpha_i$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i w^T x_i$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j x_j^T x_i$$

$\boxed{w^T = \sum_{j=1}^{m} \alpha_j y_j x_j^T}$

对偶问题

$$f(x) = w^T x + b,\quad w = \sum_{i=1}^{m} \alpha_i\, y_i x_i,\quad \sum_{i=1}^{m} \alpha_i\, y_i = 0$$

对偶问题:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i\, \alpha_j y_i y_j x_j^T x_i$$

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

$$\alpha_i \geq 0, i = 1,2,\ldots,m$$

最优解需满足Karush-Kuhn-Tucker（KKT）条件:

$$\begin{cases} \alpha_i \geq 0 \\ y_i f(x_i) \geq 1 \\ \alpha_i\big(y_i f(x_i) - 1\big) = 0 \end{cases}$$

得到最优解$\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_m^*)^T$

支持向量$\alpha_i^* > 0$，其它向量$\alpha_i^* = 0$

$w^* = \sum_{i=1}^{m} \alpha_i^*\, y_i x_i$

求解b^*。任取一个支持向量(x, y)，代入超平面模型:

$y = {w^*}^T x + b^*$,

可得$b^* = y - \sum_{i=1}^{m} \alpha_i^*\, y_i (x_i \cdot x)$

支持向量机求解

求解凸优化问题

代码实践的时候：调用凸优化求解器CVXOPT求解。

python

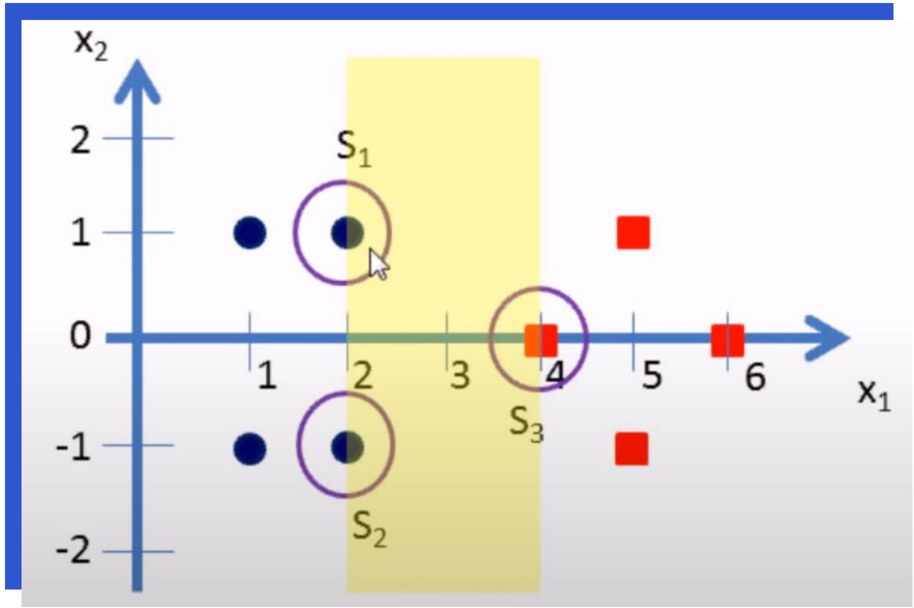
```python
from cvxopt import matrix, solvers
```

目录/CONTENTS

SVM举例

举例：在已求解得到三个支持向量后，计算权重 w^*和b^*

$$s_1 = \binom{2}{1},\ s_2 = \binom{2}{-1},\ s_3 = \binom{4}{0}$$



支持向量满足

$$y = {w^*}^T x + b^*$$

$$w^* = \sum_{i=1}^{m} \alpha_i^* y_i x_i$$

引入$\tilde{w} = \binom{w}{b}$，$\tilde{x} = \binom{x}{1}$, $\tilde{\alpha}_i = \alpha_i y_i$则满足

$$w^* = \sum_{i=1}^{m} \tilde{\alpha}_i^* \tilde{x}_i$$

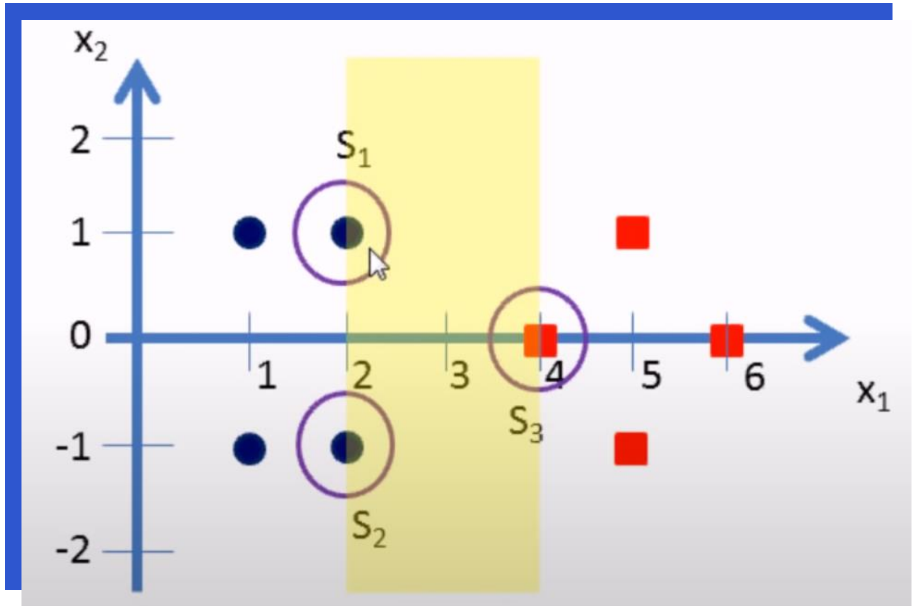
$$y = {\tilde{w}^*}^T \tilde{x}$$

SVM举例

引入$\tilde{w} = \begin{pmatrix} w \\ b \end{pmatrix}$, $\tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}$, $\tilde{\alpha}_i = \alpha_i y_i$则满足

$$w^* = \sum_{i=1}^{m} \tilde{\alpha}_i^* \tilde{x}_i$$

$$y = \tilde{w}^{*T} \tilde{x}$$



$$\tilde{w} = \alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = +1$$

$$\tilde{w} = (-3.25) . \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3.25) . \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3.5) . \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

SVM举例

引入$\tilde{w} = \begin{pmatrix} w \\ b \end{pmatrix}$，$\tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}$, $\tilde{\alpha}_i = \alpha_i y_i$则满足

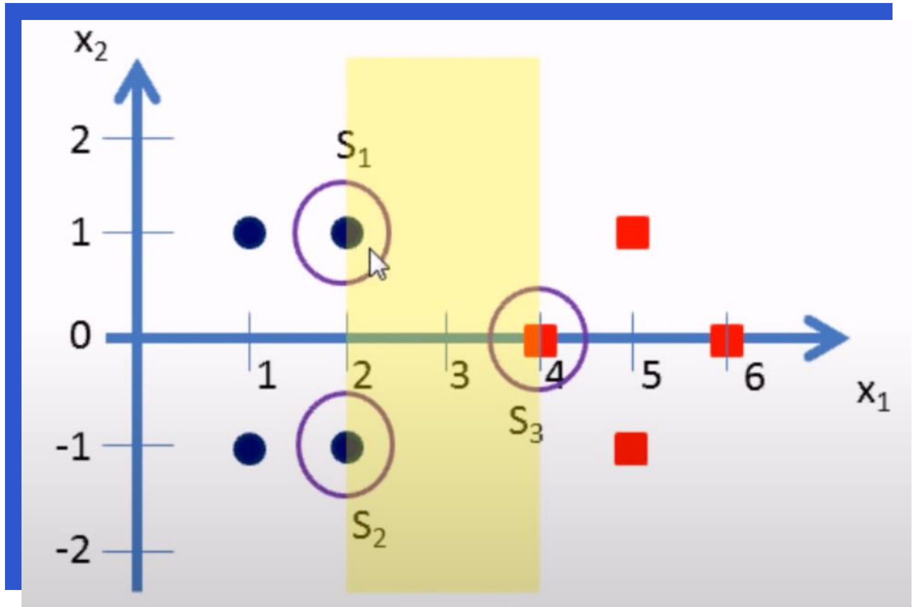
$$w^* = \sum_{i=1}^{m} \tilde{\alpha}_i^* \tilde{x}_i$$

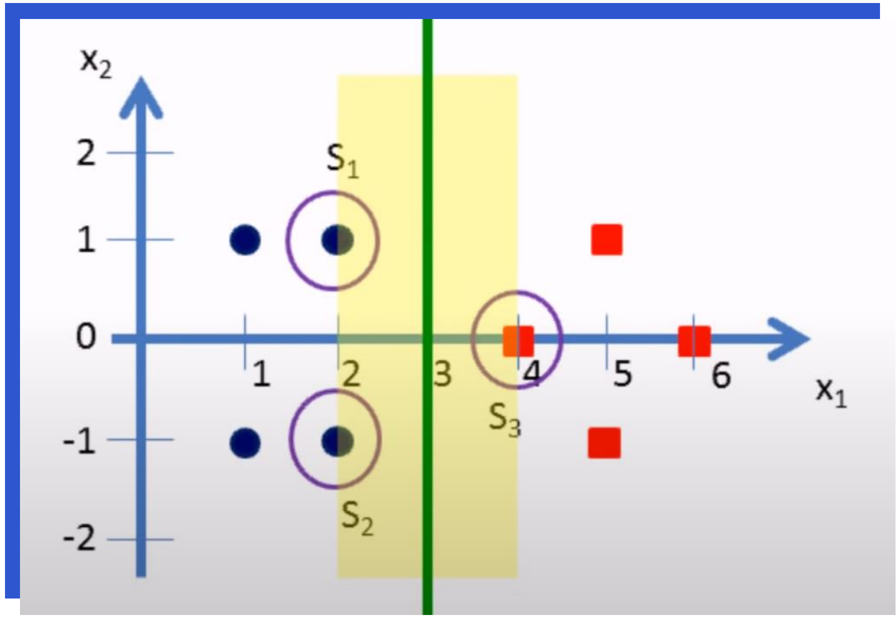
$$y = \tilde{\mathrm{w}}^{*T} \tilde{x}$$

$$\tilde{w} = \alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

$$\tilde{w} = (-3.25).\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3.25).\begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3.5).\begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

最终得到$w^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$和$b^* = -3$





目录/CONTENTS

支持向量机代码实现

支持向量机代码

```
import numpy as np
from cvxopt import matrix, solvers


class SVM:
    def __init__(self):
        self.weights = None
        self.bias = None
    def predict(self, X):
        return np.sign(np.dot(X, self.weights) + self.bias)
```

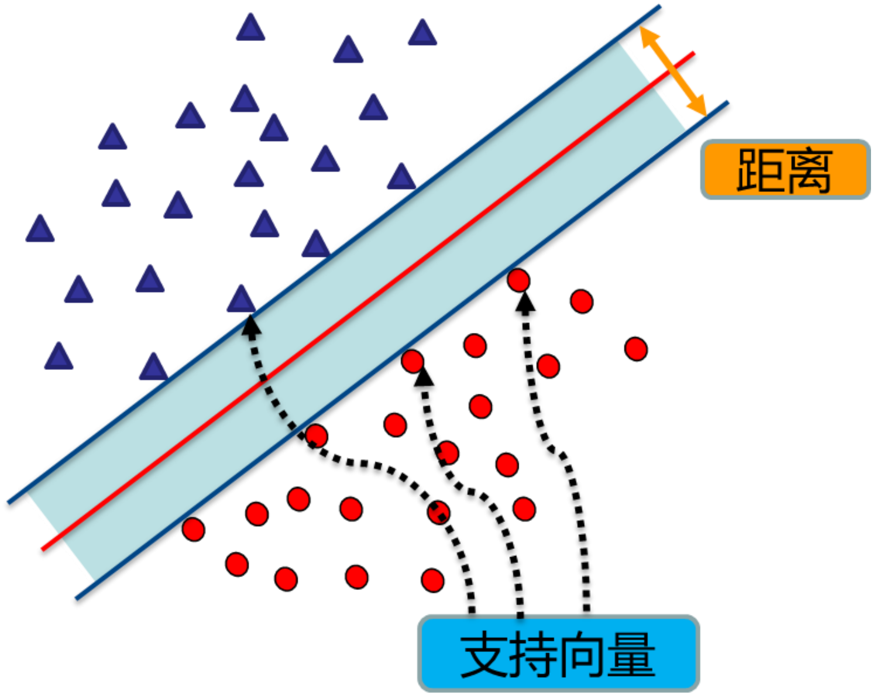
```
    def fit(self, X, y):
        [redacted]

        [redacted]
```

小组PK

V.S

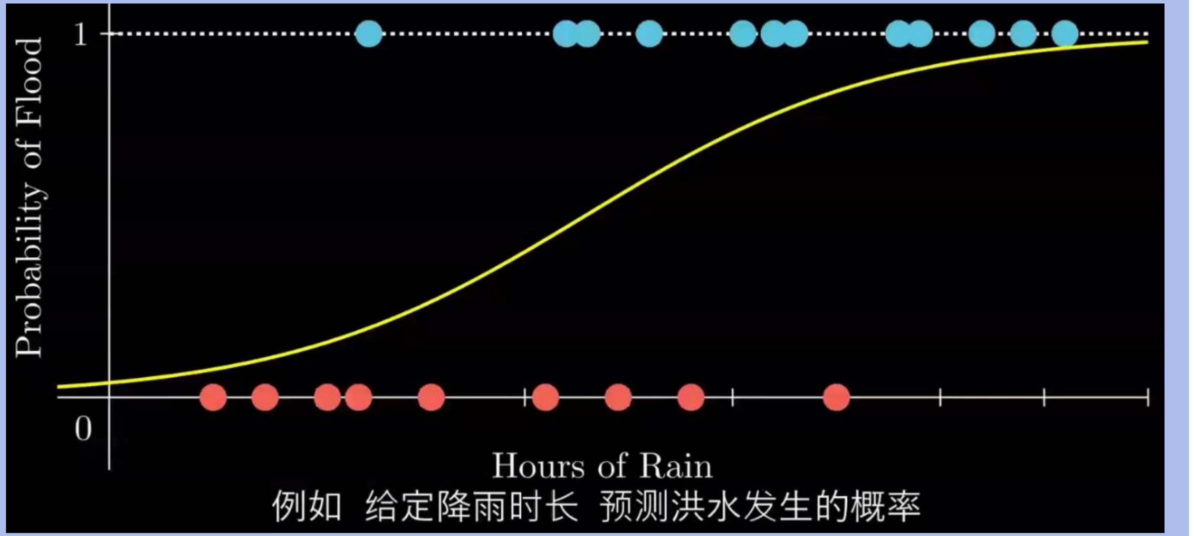
知识小结

1. 间隔与支持向量
2. 支持向量机求解
   - 支持向量机目标函数的基本型
   - 对偶问题和求解
3. 支持向量机代码实现



作业与拓展

1. 课后练习：实现完整的支持向量机代码，解决洪水预测问题。
   训练数据集见学习通APP。



2. 拓展思考：如何使用KKT条件推导得到支持向量？

目录/CONTENTS

线性松弛

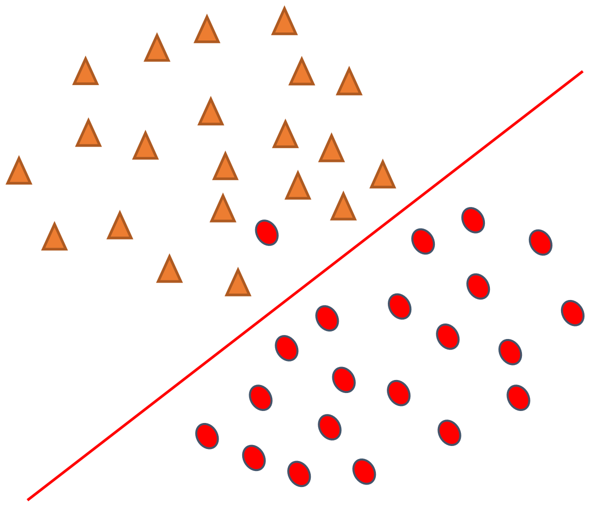
若数据线性不可分，则可以引入松弛变量$\xi \geq 0$，使函数间隔加上松弛变量大于等于1，则目标函数：

$$\min_{w,b,\xi} \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \quad s.t.\ y_i(w^T x_i + b) \geq 1 - \xi_i$$

对偶问题：

$$\max_{\alpha} \ \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\,\alpha_j y_i y_j\left(x_i \cdot x_j\right) = \min_{\alpha} \ \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\,\alpha_j y_i y_j(x_i \cdot x_j) - \sum_{i=1}^{m}\alpha_i$$

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



软间隔

C为惩罚参数，C 值越大，对分类的惩罚越大。跟线性可分求解的思路一致，同样这里先用拉格朗日乘子法得到拉格朗日函数，再求其对偶问题。

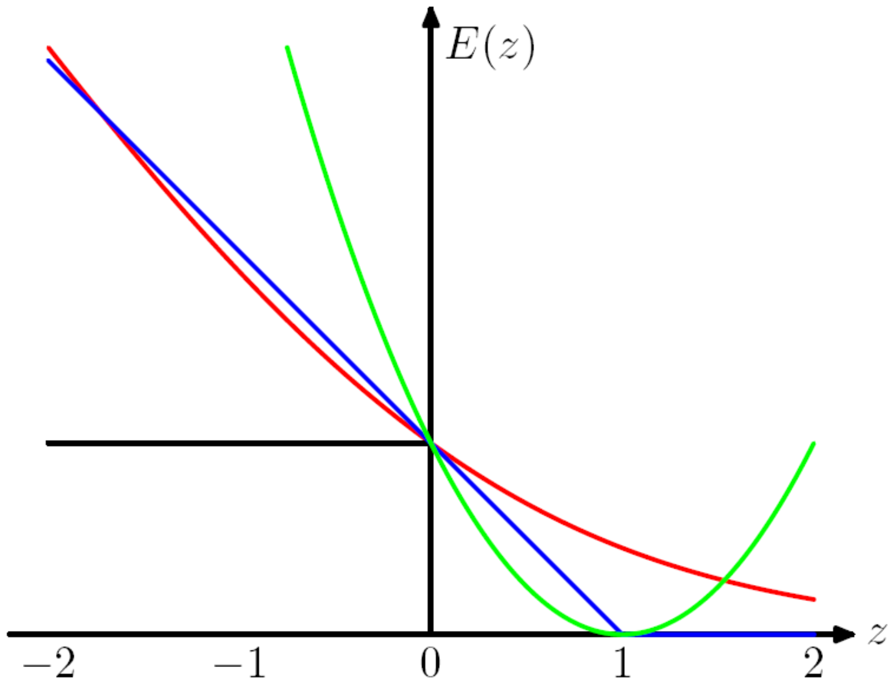
松弛变量

绿色的线为 square loss
蓝色的线为 hinge loss
红的的线为负 log loss

ξ 为"松弛变量"

$\xi_i = \max(0,1 - y_i(w \cdot x_i + b))$

即hinge损失函数。每一个样本都有一个对应的松弛变量，表征该样本不满足约束的程度。



损失函数hinge loss

线性支持向量机

求解原始最优化问题的解w^*和b^*，得到线性支持向量机，其分离超平面为

$$w^{*T}x + b^* = 0$$

分类决策函数为：$f(x) = \mathrm{sign}(w^{*T}x + b^*)$

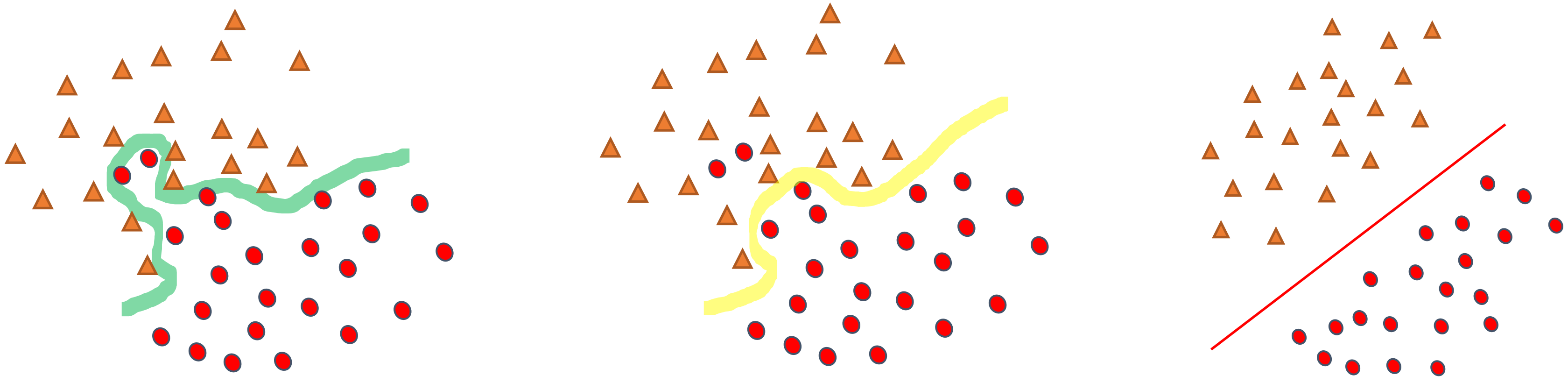
线性可分支持向量机的解w^*唯一，但b^*不唯一，取决于代入的样本(x_j, y_j)。

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

$$w^* = \sum_{i=1}^{m} \alpha_i^* y_i x_i$$

多种支持向量机

硬间隔、软间隔和非线性 SVM



硬间隔　　软间隔　　线性可分　　非线性

硬间隔：完全分类准确，不能存在分类错误的情况。

软间隔：就是允许一定量的样本分类错误。

假如数据是完全的线性可分的，学习到的模型可以称为硬间隔支持向量机。

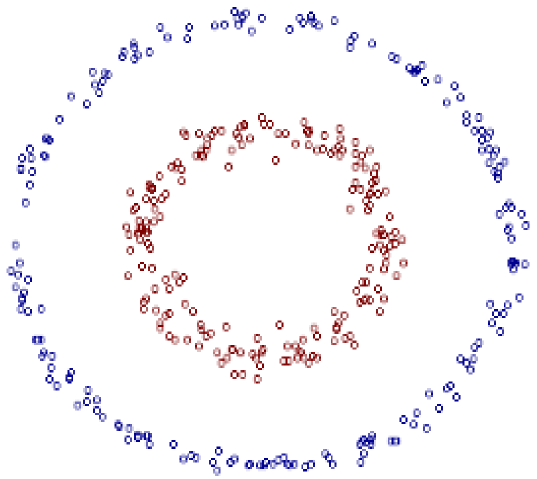
目录/CONTENTS

非线性支持向量机

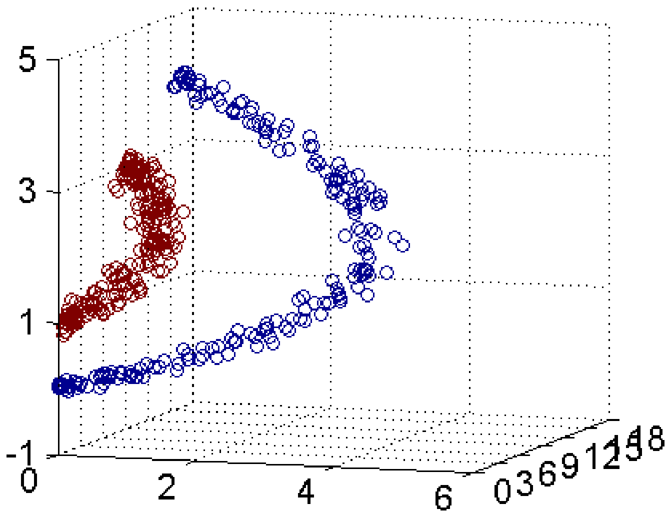
核技巧

核函数:它可以将样本从原始空间映射到一个更高维的特质空间中，使得样本在新的空间中线性可分。

使用原来的推导来进行计算，只是所有的推导是在新的空间，而不是在原来的空间中进行，即用核函数来替换当中的内积。



线性不可分

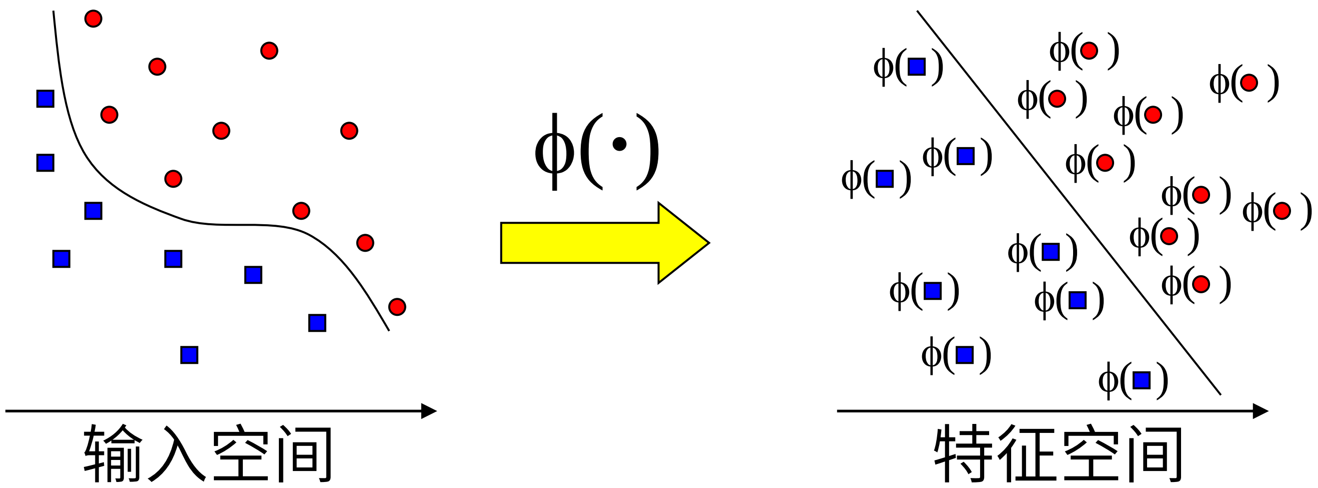


高维下线性可分

核函数

核技巧

用核函数来替换原来的内积。



即通过一个非线性转换后的两个样本间的内积。具体地，$K(x,z)$是一个核函数，或正定核，意味着存在一个从输入空间到特征空间的映射，对于任意空间输入的x,z 有：

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

非线性支持向量机

在线性支持向量机学习的对偶问题中，用核函数$K(x,z)$替代内积，求解得到的就是非线性支持向量机

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

非线性支持向量机

常用核函数有:

线性核函数

$$K(x_i, x_j) = x_i^T x_j$$

多项式核函数

$$K(x_i, x_j) = (x_i^T x_j)^d$$

高斯核函数

$$K(x_i, x_j) = exp(-\frac{||x_i - x_j||}{2\delta^2})$$

这三个常用的核函数中,只有高斯核函数是需要调参的。

非线性支持向量机

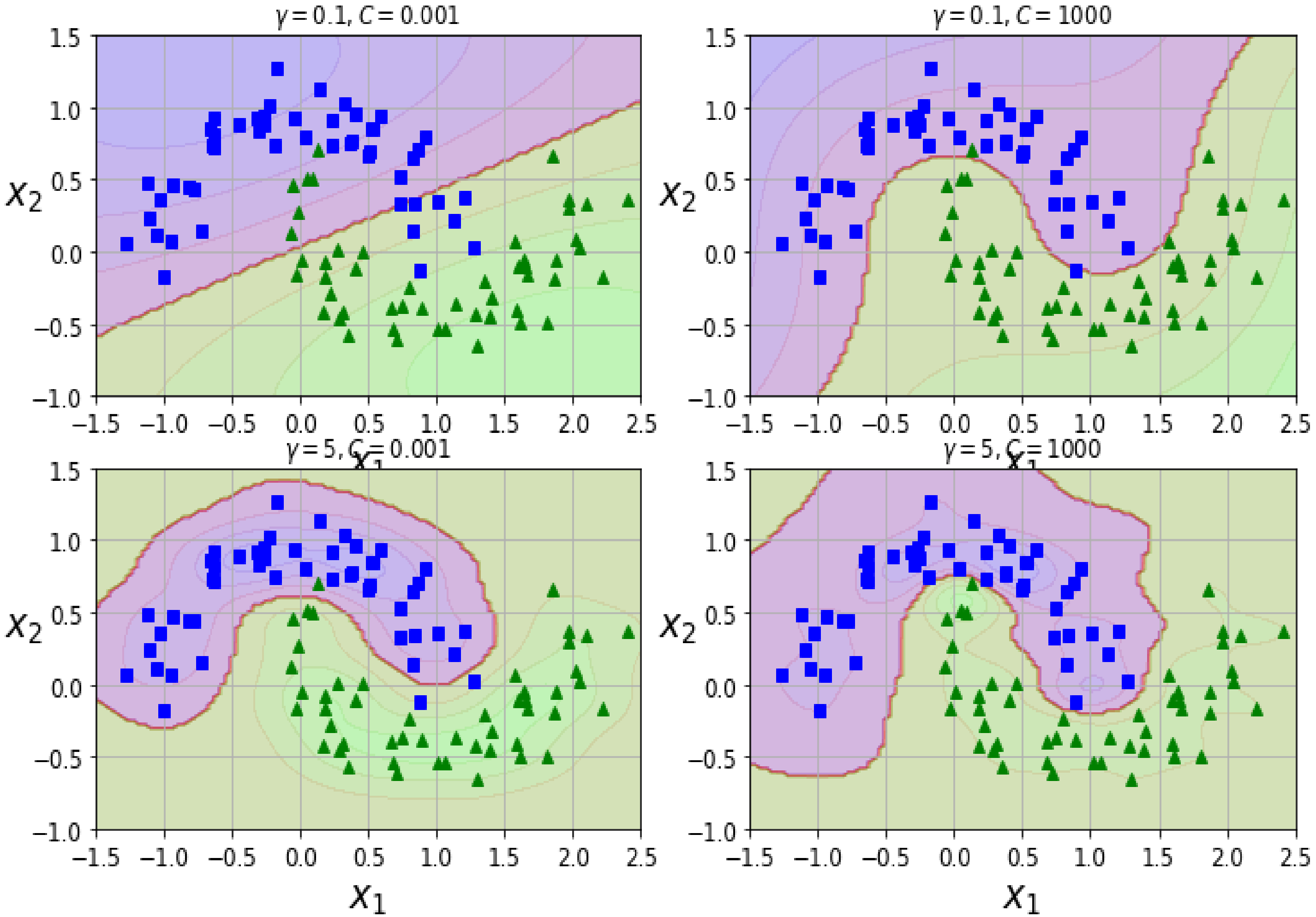
SVM的超参数

γ越大，支持向量越少，

γ越小，支持向量越多。

C是惩罚系数，即对误差的宽容度。

- C越高，说明越不能容忍出现误差,容易过拟合。
- C越小，容易欠拟合。



总结

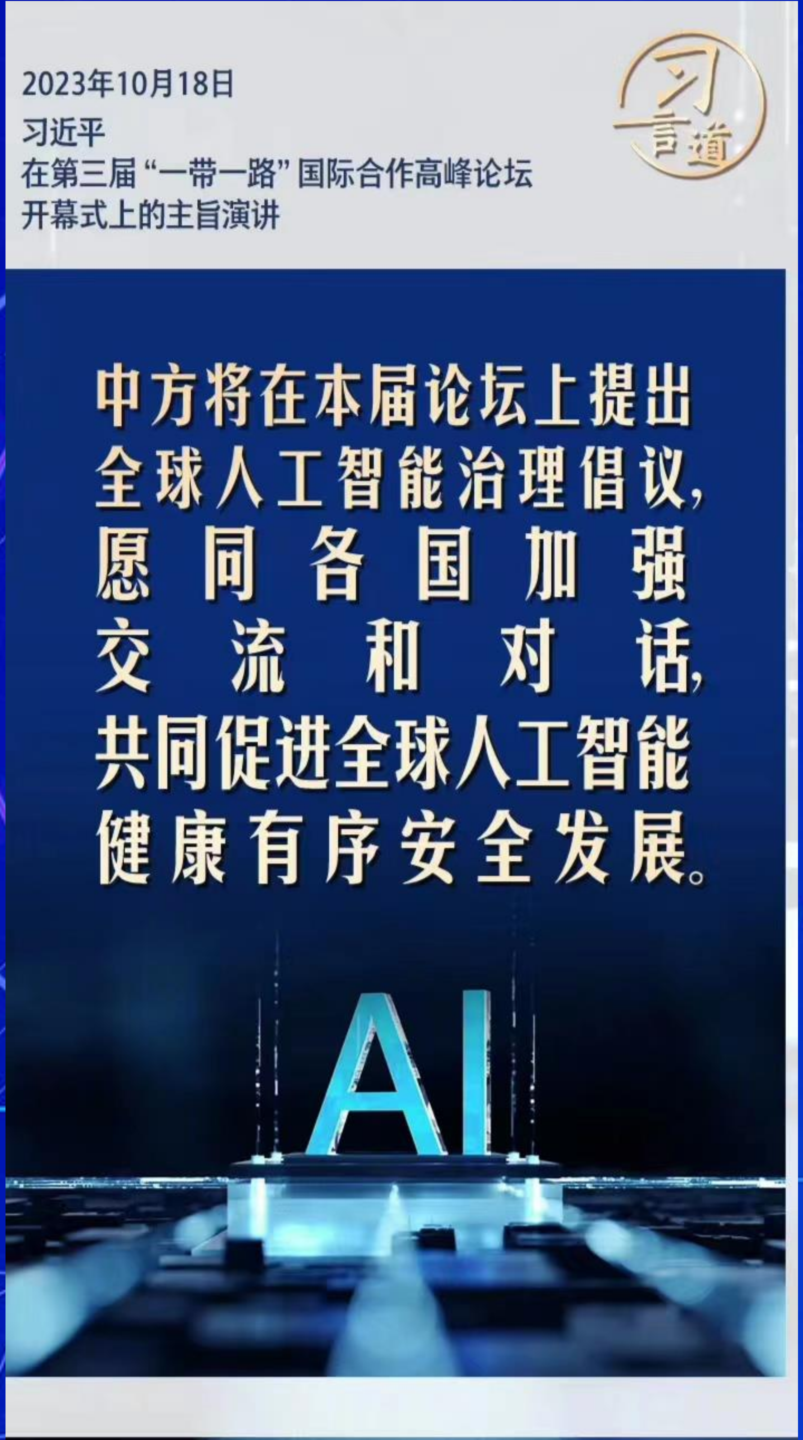
下面是一些SVM普遍使用的准则：

n为特征数，m为训练样本数。

(1)如果相较于m而言，n要大许多，即训练集数据量不够支持我们训练一个复杂的非线性模型，我们选用逻辑回归模型或者不带核函数的支持向量机。

(2)如果n较小，而且m大小中等，例如n在 1-1000 之间，而m在10-10000之间，使用高斯核函数的支持向量机。

(3)如果n较小，而m较大，例如n在1-1000之间，而m大于50000，则使用支持向量机会非常慢，解决方案是创造、增加更多的特征，然后使用逻辑回归或不带核函数的支持向量机。



《机器学习》

THANKS

感谢您的观看

下一课：实验课