支持向量机

支持向量机

百度智能云

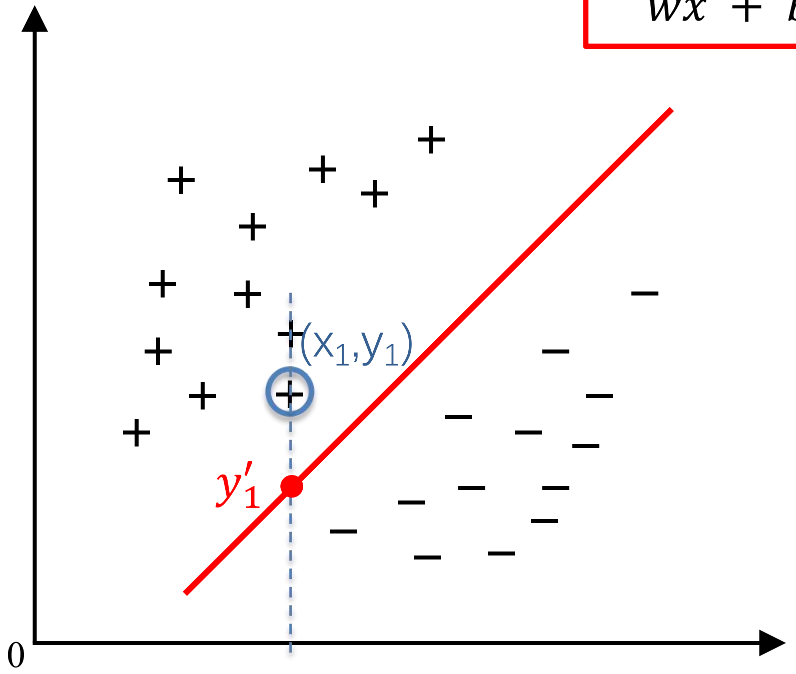
- 支持向量机，SVM的全称是Support Vector Machine
- 支持向量机算法是一种二分类模型
- 核心思想：
  - 找到空间中的一个能够将所有数据样本划开的超平面，并且使得所有数据到这个超平面的距离最短

前导基本概念

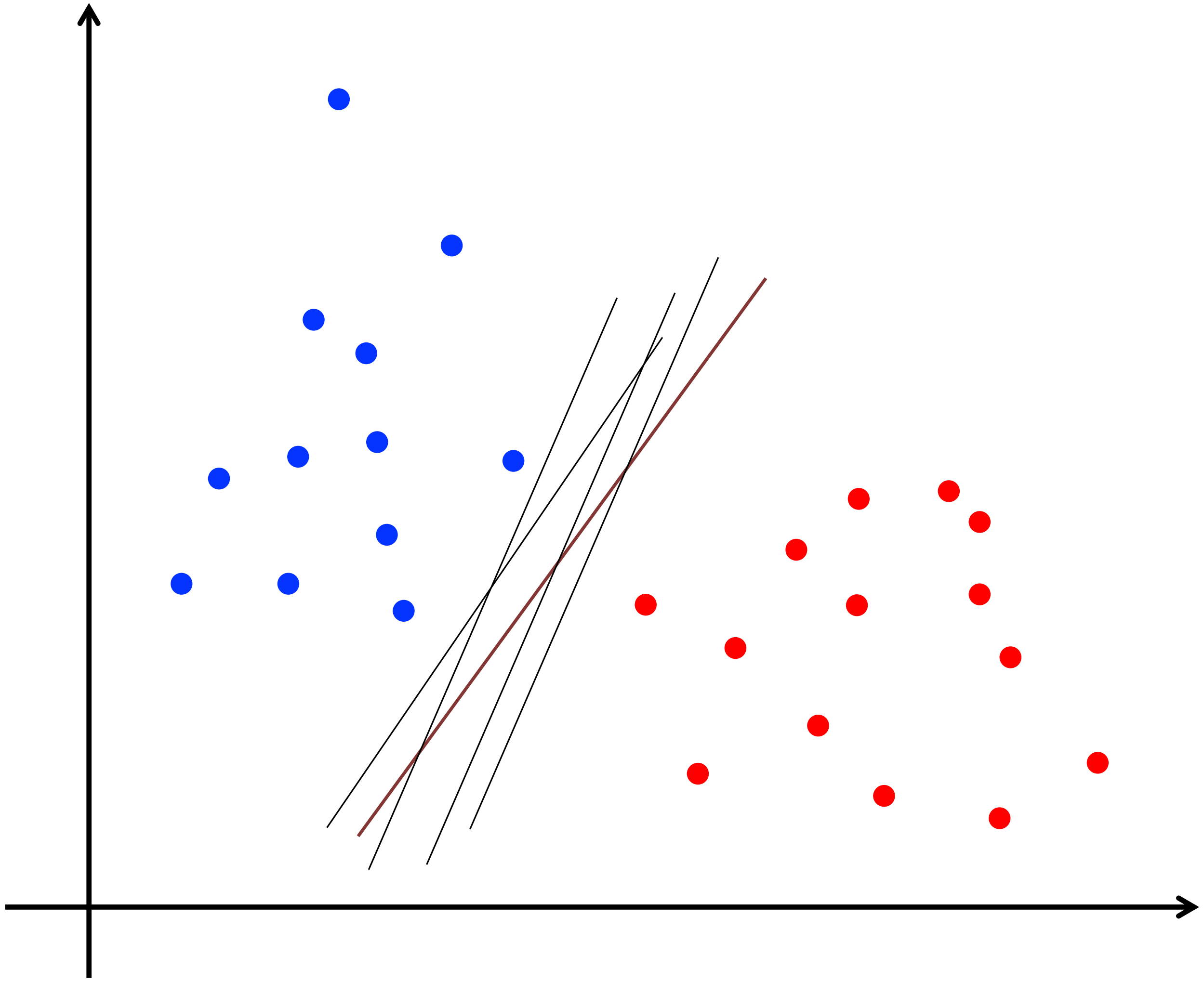
线性可分

- D_0 和 D_1 是 n 维欧氏空间中的两个点集
- 如果存在 n 维向量 w 和实数 b
- 使得所有属于 D_0 的点 x_i 都有 $wx_i+b>0$，
- 而对于所有属于 D_1 的点 x_j 则有 $wx_j+b<0$，
- 则称 D_0 和 D_1 线性可分

$\boxed{wx + b = y}$

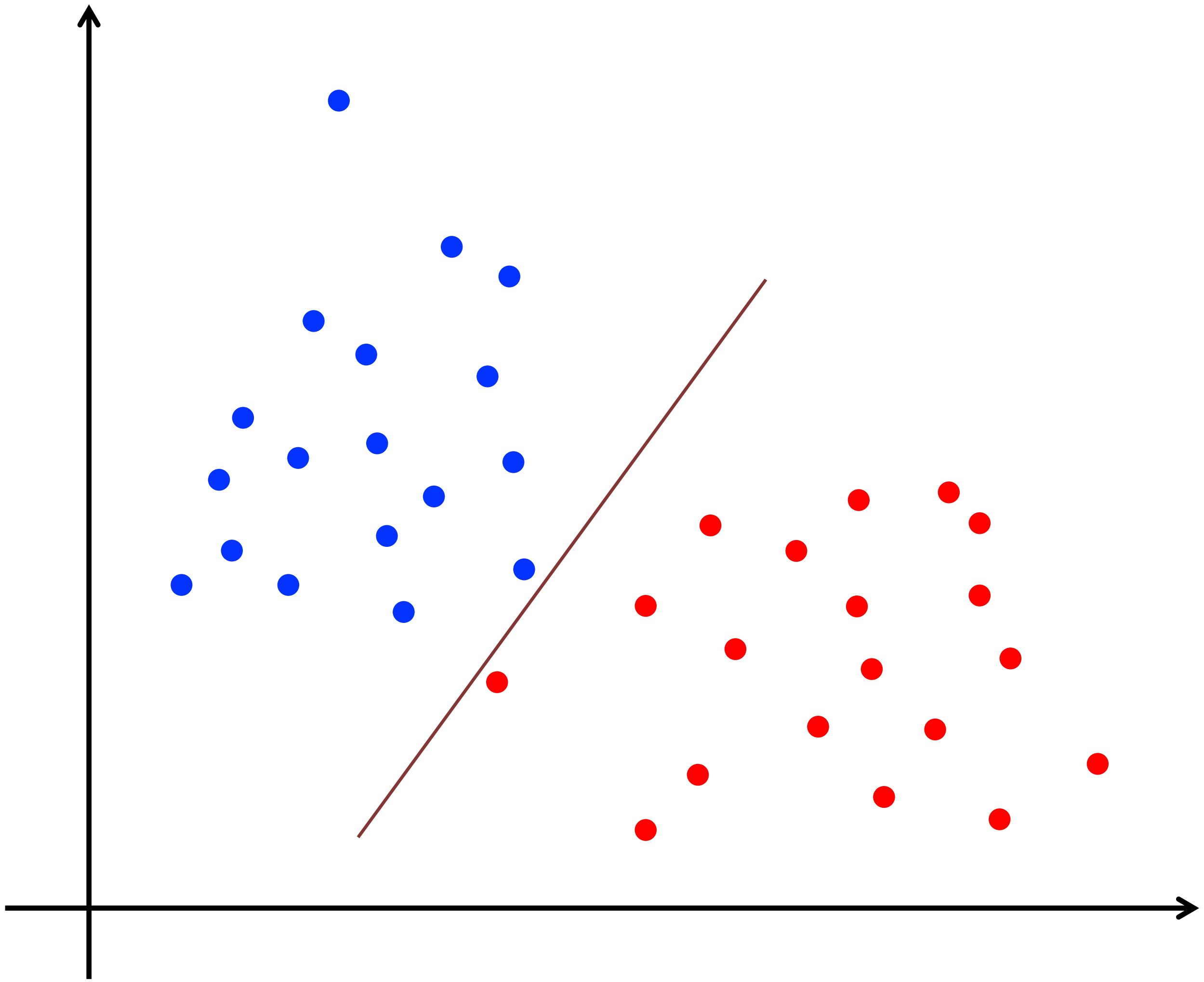


多个可用分类器



分类器的泛化性能

- 应选择”正中间”，容忍性好，鲁棒性高，泛化能力最强.



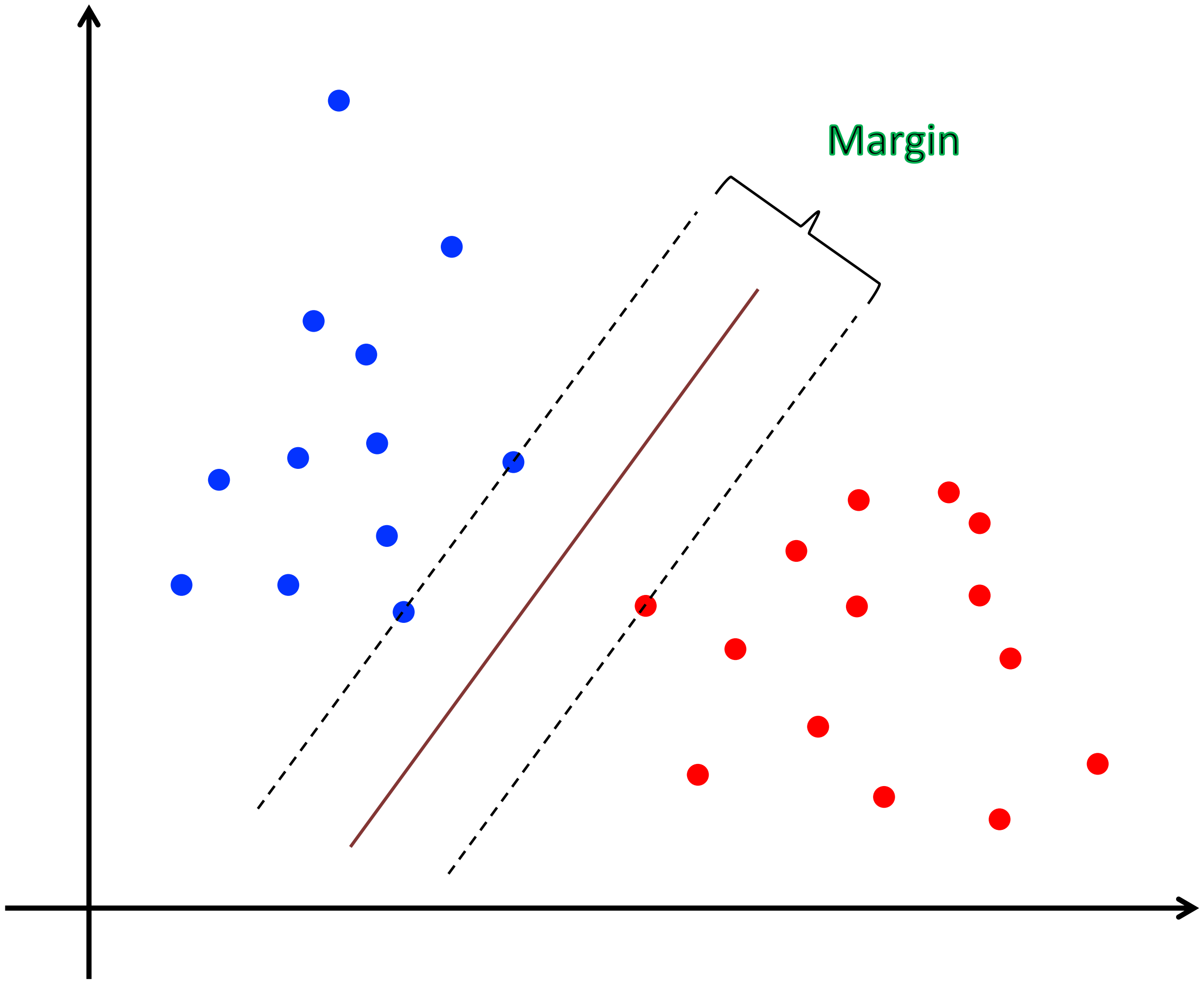
新数据的出现，将使许多其他的划分超平面出现错误，而绿色的超平面受影响最小。换言之，这个划分超平面所产生的分类结果是最鲁棒的，对未见样本的泛化能力最强

最大间隔超平面

百度智能云

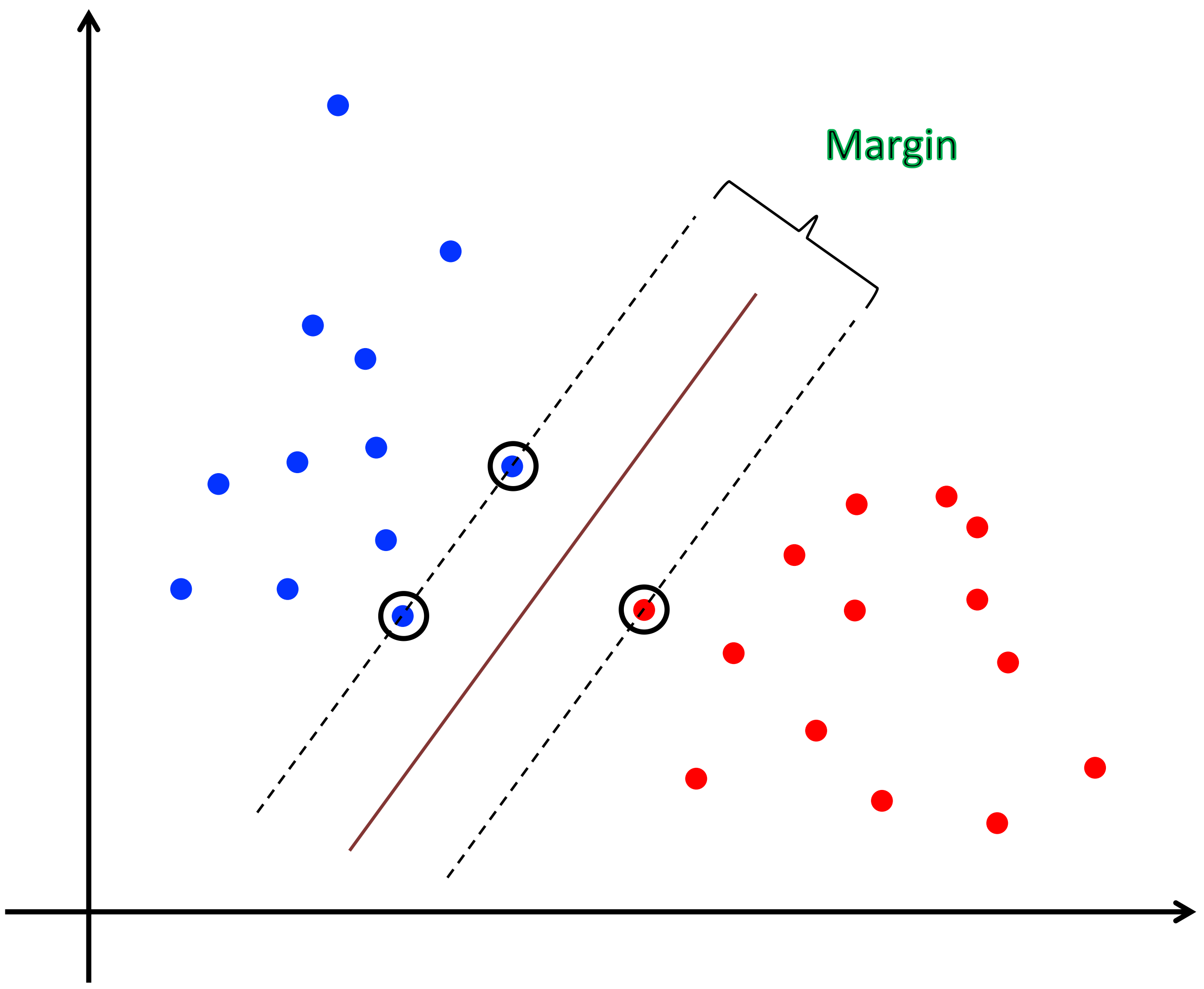
- 两个极限位置(虚线)
- 两个因素决定：
  - 一个是决策面的方向，
  - 一个是距离原决策面最近的几个样本的位置。

- 而这两条平行虚线正中间的绿色分界线就是在保持当前决策面方向不变的前提下的**最优决策面**。



支持向量与超平面

- 两个极限位置(虚线)
- 两个因素决定:
  - 一个是决策面的方向,
  - 一个是距离原决策面最近的几个样本的位置。
- 而这两条平行虚线正中间的绿色分界线就是在保持当前决策面方向不变的前提下的**最优决策面**。
- 两条虚线之间的垂直距离就是这个最优决策面对应的分类间隔。显然每一个可能把数据集正确分开的方向都有一个最优决策面，而不同方向的最优决策面的分类间隔通常是不同的，那个具有**"最大间隔"**的决策面就是SVM要寻找的最优解。



构造SVM 最优化问题

- SVM算法要解决的是一个最优分类器的设计问题。
- 既然叫作最优分类器，其本质必然是个**最优化问题**。
- 接下来我们要讨论的就是如何把SVM变成用数学语言描述的最优化问题，这就是我们接下来要讲的“线性SVM算法的数学建模”。

- 一个最优化问题通常有两个最基本的因素：1.目标函数；2.优化对象
  - 1）目标函数，也就是你希望什么东西的什么指标达到最好；
  - 2）优化对象，你期望通过改变哪些因素来使你的目标函数达到最优。

- 在线性SVM算法中
  - 目标函数显然就是那个“分类间隔”，
  - 优化对象则是超平面。

- 所以要对SVM问题进行数学建模，首先要对 “分类间隔”和我们关心的“超平面”进行数学描述。
  - 超平面
  - 分类间隔

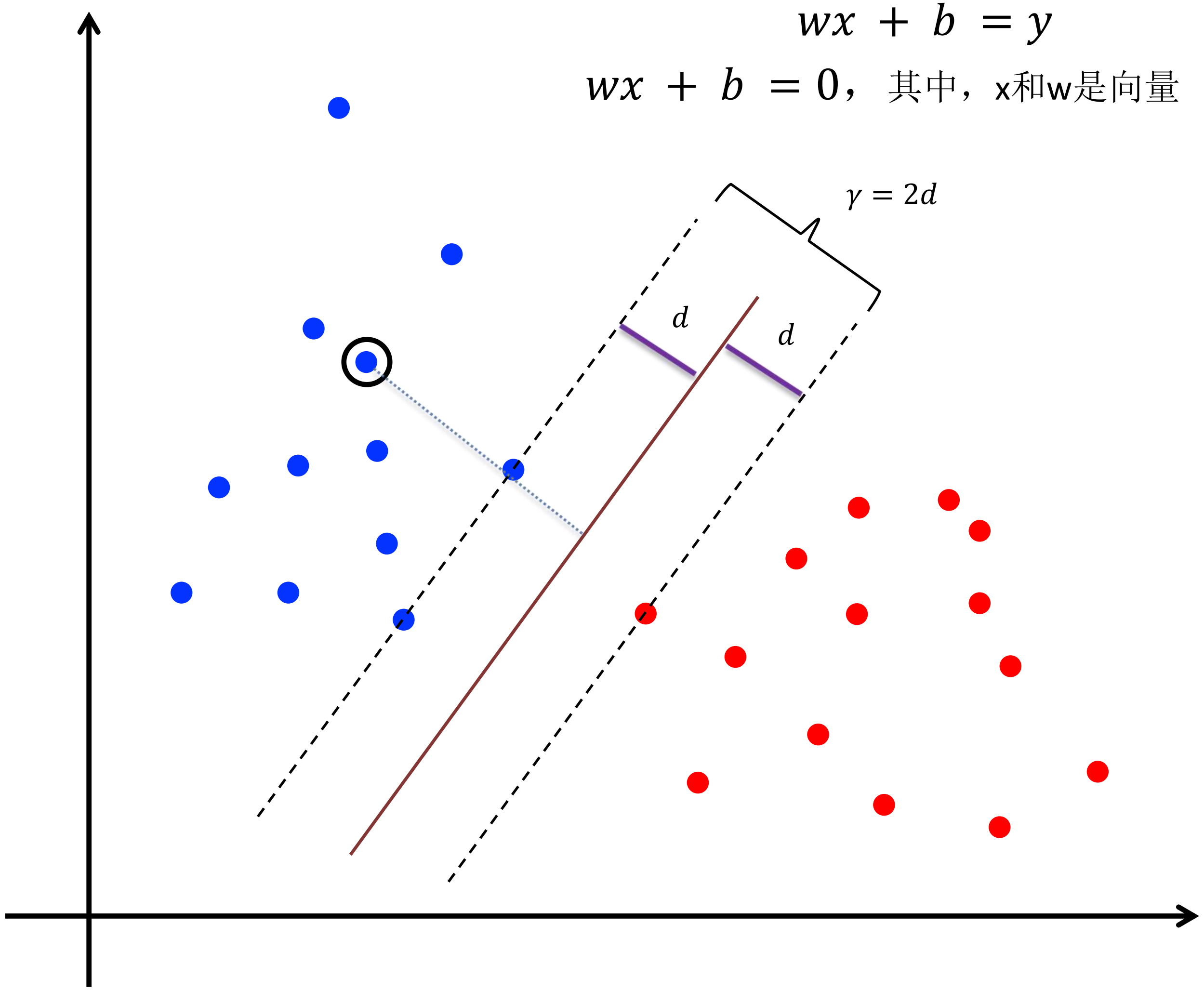
点到线的距离

- 点：　(x,y)
- 线：　$Ax + By + C = 0$
- 点到直线的距离：$d = \frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}$
- 高维空间的点：　$x = \{x_1, x_2 ... x_n\}$
- 高维空间的线：　$wx + b = 0$，其中，x为高维向量
- 高维空间的点到直线的距离：

$d = \frac{|wx+b|}{||w||}$，　其中，$||w|| = \sqrt{w_1{}^2 + w_2{}^2 + ... + w_n{}^2}$

$wx + b = y$

$wx + b = 0$，其中，x和w是向量



超平面公式演化

$$\begin{cases} \dfrac{wx+b}{\|w\|} \geq d, \ y=1 & \text{蓝色样本} \\ \dfrac{wx+b}{\|w\|} \leq d, \ y=-1 & \text{红色样本} \end{cases}$$

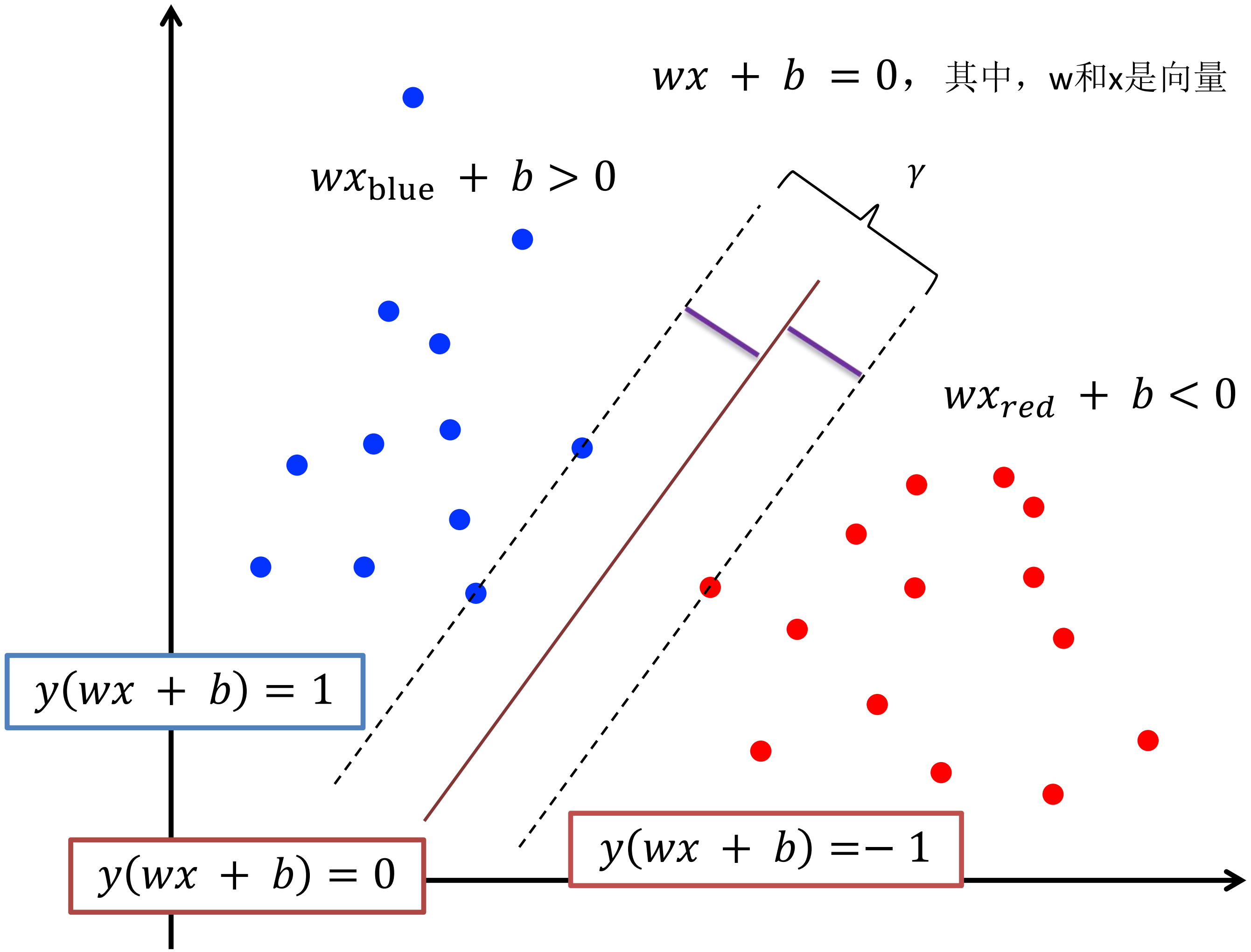
$$\begin{cases} \dfrac{wx+b}{\|w\|d} \geq 1, \ y=1 \\ \dfrac{wx+b}{\|w\|d} \leq 1, \ y=-1 \end{cases}$$

$\|w\|d > 0$

令 $\|w\|\mathrm{d} = 1$

$$\begin{cases} wx+b \geq 1, \ y=1 \\ wx+b \leq 1, \ y=-1 \end{cases}$$

$$y(wx+b) \geq 1$$



超平面公式演化

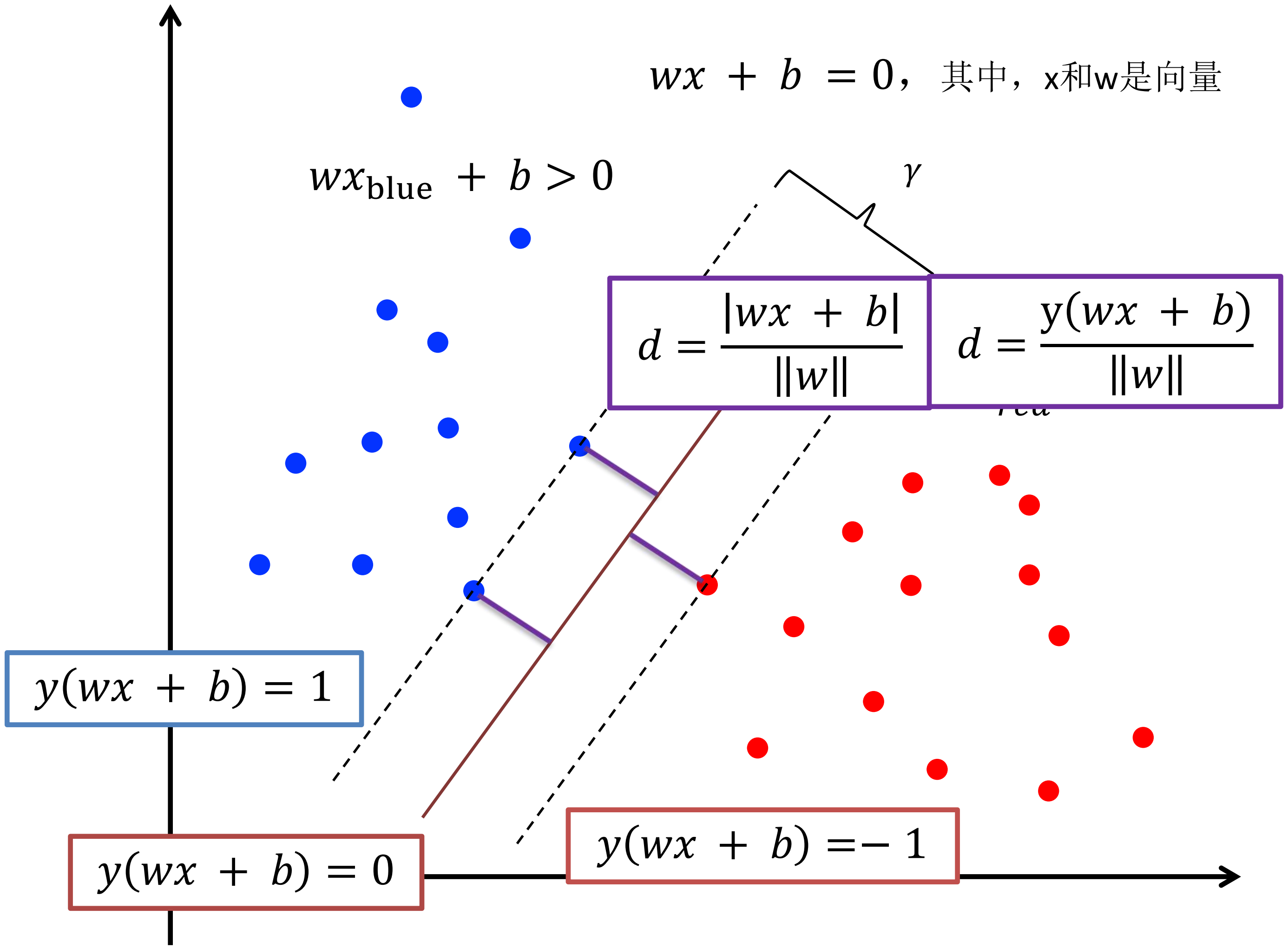
$$\begin{cases} \dfrac{wx + b}{\|w\|} \geq d, & y = 1 \\ \dfrac{wx + b}{\|w\|} \leq d, & y = -1 \end{cases}$$

$$\begin{cases} \dfrac{wx + b}{\|w\|\mathrm{d}} \geq 1, & y = 1 \\ \dfrac{wx + b}{\|w\|d} \leq 1, & y = -1 \end{cases} \qquad \|w\|d > 0$$

$$令\ \|w\|\mathrm{d} = 1$$

$$\begin{cases} wx + b \geq 1, & y = 1 \\ wx + b \leq 1, & y = -1 \end{cases}$$

$$y(wx + b) \geq 1 \qquad y(wx + b) = |wx + b|$$



距离最大化

$$d = \frac{y(wx + b)}{\|w\|}$$

⇓ 最大化

$$\max_{w,b} 2 * \frac{y(wx + b)}{\|w\|}$$

这个式子的几何意义就是，支持向量样本点到决策面方程的距离就是 $\frac{1}{\|w\|}$

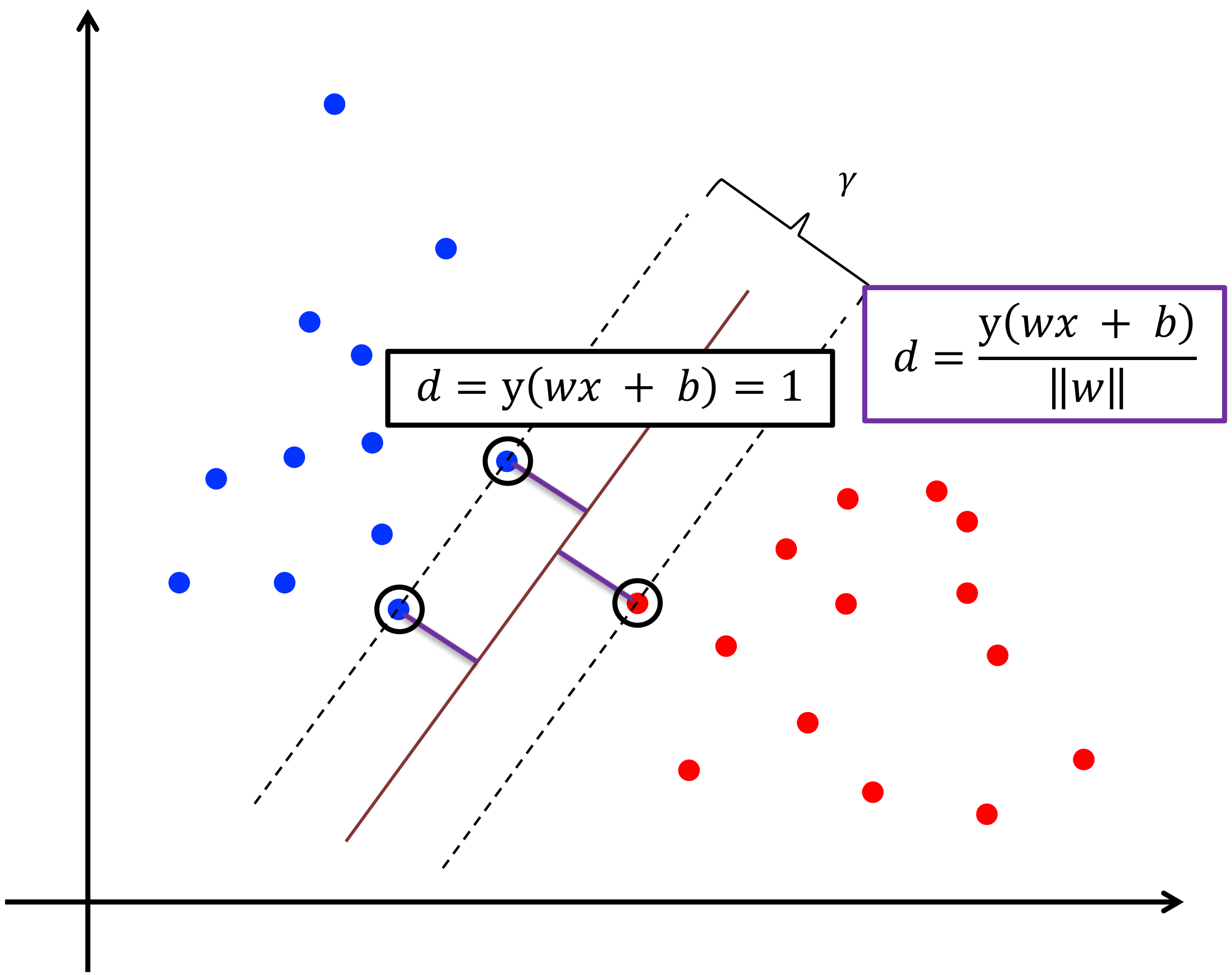
$$\max_{w,b} 2 * \frac{1}{\|w\|} \xrightarrow{\text{转化}} \min_{w,b} \frac{1}{2}\|w\|$$

其中，$\|w\| = \sqrt{w_1{}^2 + w_2{}^2 + \dots + w_n{}^2}$

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

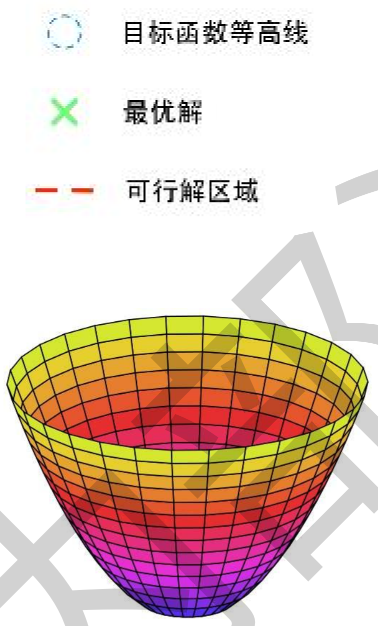
$s.t.\ \ y_i(wx_i + b) \geq 1$　其中，i =1,2,3...m

- **间隔最大化：** 几何间隔最大



拉格朗日乘数法

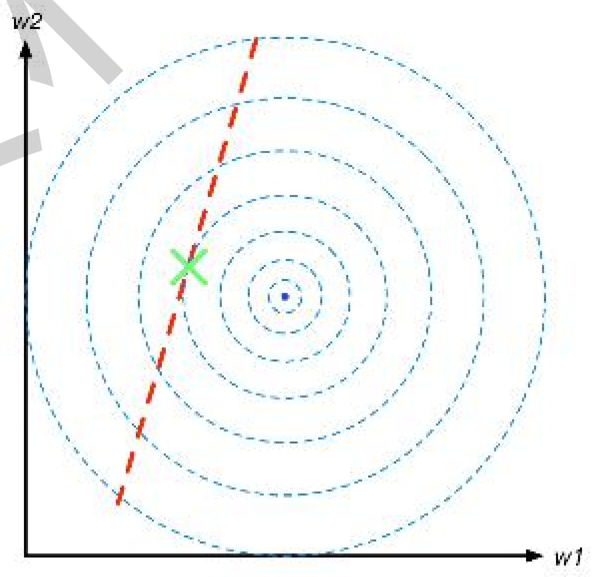
有约束优化问题的几何意象



目标函数

- 等式约束

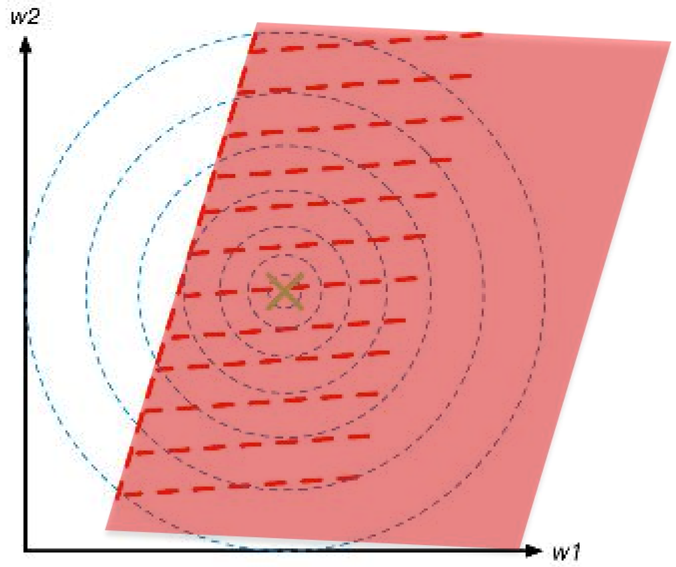
$g(x) = 0$



等式约束情况

- 不等式约束

$g(x) \leq 0$



不等式约束情况

- 假设 $x \in \mathbb{R}^d$，如果函数 $g(x)$ 是线性的，则 $g(x) = 0$ 是个d-1维的超平面
- 假设 $x \in \mathbb{R}^d$，如果函数 $g(x)$ 是非线性的，则 $g(x) \leq 0$ 是个d-1维的区域

拉格朗日乘数法定义——等式约束优化问题

百度智能云

- 拉格朗日乘数法(Lagrange multipliers，也译作拉格朗日乘子法)
  - 是一种寻找多元函数在一组约束下的极值的方法
- 多元函数微积分，以二元函数为例
  - **目标函数：** $z = f(x, y)$
  - **约束条件：** $g(x, y) = 0$
    - 拉格朗日辅助函数：$L(x, y, \lambda) = f(x, y) + \lambda g(x, y)$ 其中，实数 λ 为拉格朗日乘数
    - 求偏导数，令其为零，等到方程组：

$$\begin{cases} L_x(x, y, \lambda) = f_x(x, y) + \lambda g_x(x, y) = 0 \\ L_y(x, y, \lambda) = f_y(x, y) + \lambda g_y(x, y) = 0 \\ L_\lambda(x, y, \lambda) = g_x(x, y) = 0 \end{cases}$$

    - 解此方程组，得到结果：(x, y)

拉格朗日乘数法运算示例

设实数 x, y 满足 $4x^2 + y^2 + xy = 1$ ，求 $2x + y$ 的最大值

第一步，明确目标函数和约束条件

目标函数为： $z = f(x,y) = 2x + y$

约束条件为： $4x^2 + y^2 + xy = 1$ $\qquad$ $4x^2 + y^2 + xy - 1 = 0$

第二步，构造拉格朗日辅助函数

辅助函数为：$L(x,y,\lambda) = f(x,y) + \lambda g(x,y)$

$$L(x,y,\lambda) = 2x + y + \lambda(4x^2 + y^2 + xy - 1 = 0)$$

第三步，辅助函数分别对未知量 x, y, λ 求偏导数，

得到方程组为：
$$\begin{cases} L_x = 2 + \lambda(8x + y) = 0 \\ L_y = 1 + \lambda(x + 2y) = 0 \\ L_\lambda = 4x^2 + y^2 + xy - 1 = 0 \end{cases}$$

解方程组，得：
$$\begin{cases} x = \pm \frac{\sqrt{10}}{10} \\ y = \pm \frac{\sqrt{10}}{5} \end{cases}$$

最大值为$\frac{2\sqrt{10}}{5}$

SVM的最优化问题

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

$s.t.\ \ y_i(wx_i + b) \geq 1$ 其中，i =1,2,3...m

SVM最优化问题的计算

SVM的计算步骤

- 求解 SVM 总共分为五步：
- 1. 构造拉格朗日函数
- 2. 构造拉格朗日对偶
- 3. 使用SMO 算法
- 4. 计算参数值
- 5. 预测

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

$$s.t.\ \ y_i(wx_i\ +\ b) \geq 1$$ 其中，i =1,2,3...m

SVM的最优化问题

第一步

百度智能云

- 第一步
  - 构造拉格朗日函数

$$L(x,\alpha,\beta)=f(x)+\sum_{i=1}^{m}\alpha_i h_i(x)+\sum_{i=1}^{m}\beta_j g_j(x)$$

$$L(\mathrm{w},\mathrm{b},\lambda)=\frac{1}{2}\|w\|^2+\sum_{i=1}^{m}\lambda_i\left(1-y_i(wx_i+b)\right)$$

其中， $\lambda_i \geq 0$

带入，并展开

$$\min_{w,b}\left[\max_{\lambda} L(\mathrm{w},\mathrm{b},\lambda)\right]$$

$$\min_{w,b}\left[\max_{\lambda}\left(\frac{1}{2}\|w\|^2+\sum_{i=1}^{m}\lambda_i\left(1-y_i(wx_i+b)\right)\right)\right]$$

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

$$s.t.\ \ y_i(wx_i+b)\geq 1$$ 其中， i =1,2,3…m

SVM的最优化问题

$$\min_{x}\left[\max_{\alpha_i,\beta_j;\beta_j\geq 0} L(x,\alpha,\beta)\right]$$

第二步

- 第二步
  - 使用拉格朗日对偶
  - 求内层的 min 函数的最小值

$$\max_{\alpha_i,\beta_j;\beta_j\geq 0}\left[\min_x L(x,\alpha,\beta)\right]$$

$$\min_x L(x,\alpha,\beta) \qquad \min_{w,b} L(\mathrm{w},\mathrm{b},\lambda)$$

$$\min_{w,b}\left(\frac{1}{2}\|w\|^2+\sum_{i=1}^{m}\lambda_i\left(1-y_i(wx_i\ +\ b)\right)\right)$$

$$\frac{\partial L}{\partial w}=\mathrm{w}-\sum_{i=1}^{m}\lambda_i y_i x_i=0$$

$$\frac{\partial L}{\partial \mathrm{b}}=\sum_{i=1}^{m}\lambda_i y_i=0$$

求解

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

$$\sum_{i=1}^{m}\lambda_i y_i=0$$

带回

$$\min_{w,b} L(\mathrm{w},\mathrm{b},\lambda)=\sum_{i=1}^{m}\lambda_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\lambda_i\lambda_j y_i y_j\left(x_i\cdot x_j\right)$$

$$\min_x\left[\max_{\alpha_i,\beta_j;\beta_j\geq 0} L(x,\alpha,\beta)\right]$$

$$\max_{\alpha_i,\beta_j;\beta_j\geq 0}\left[\min_x L(x,\alpha,\beta)\right]$$

对偶

第三步

- 第三步
  - 将第二步的结果带入 max 层
  - 使用 SMO 算法求解

$$\min_{w,b} L(\mathrm{w},\mathrm{b},\lambda) = \sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\lambda_i\lambda_j y_i y_j (x_i \cdot x_j)$$

$$\max_{\alpha_i, \beta_j; \beta_j \geq 0}\left[\min_x L(x,\alpha,\beta)\right] \qquad \max_{\lambda}\left[\min_{w,b} L(\mathrm{w},\mathrm{b},\lambda)\right]$$

带入，并展开

$$\max_{\lambda}\left[\sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\lambda_i\lambda_j y_i y_j (x_i \cdot x_j)\right]$$

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

$$\lambda_i \geq 0$$

SMO 算法

- SMO(Sequential Minimal Optimization)算法
  - **序列最小优化算法**
- 核心思想
  - 先固定住所有参数，然后放开一个参数
  - 每次只优化这一个参数
  - 仅求当前这个优化参数的极值

$$\max_{\lambda}\left[\sum_{i=1}^{m}\lambda_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\lambda_i\lambda_j y_i y_j(x_i\cdot x_j)\right]$$

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

$$\lambda_i\geq 0$$

对偶优化问题

$$\sum_{i=1}^{m}\lambda_i y_i=0 \xrightarrow{\text{改写}} \lambda_m y_m-\sum_{i=1}^{m-1}\lambda_i y_i=0 \xrightarrow{\text{改写}} \lambda_m y_m=\sum_{i=1}^{m-1}\lambda_i y_i$$

放开两个变量 $\lambda_i y_j+\lambda_j y_j=\sum_{k\neq i,j}^{m-2}\lambda_k y_k=c$ 其中，$\lambda_i\geq 0,\lambda_j\geq 0$ $\xrightarrow{\text{改写}}$ $\lambda_j=\dfrac{c-\lambda_i y_i}{y_j}$

第四步

- 第四步
  - 计算参数值
    - 计算 b 的值
    - 计算 w 的值

$$y_s(wx_s + b) = 1$$

$\downarrow$ 等式两侧同时乘以y_s

$$y_s^2(wx_s + b) = y_s$$

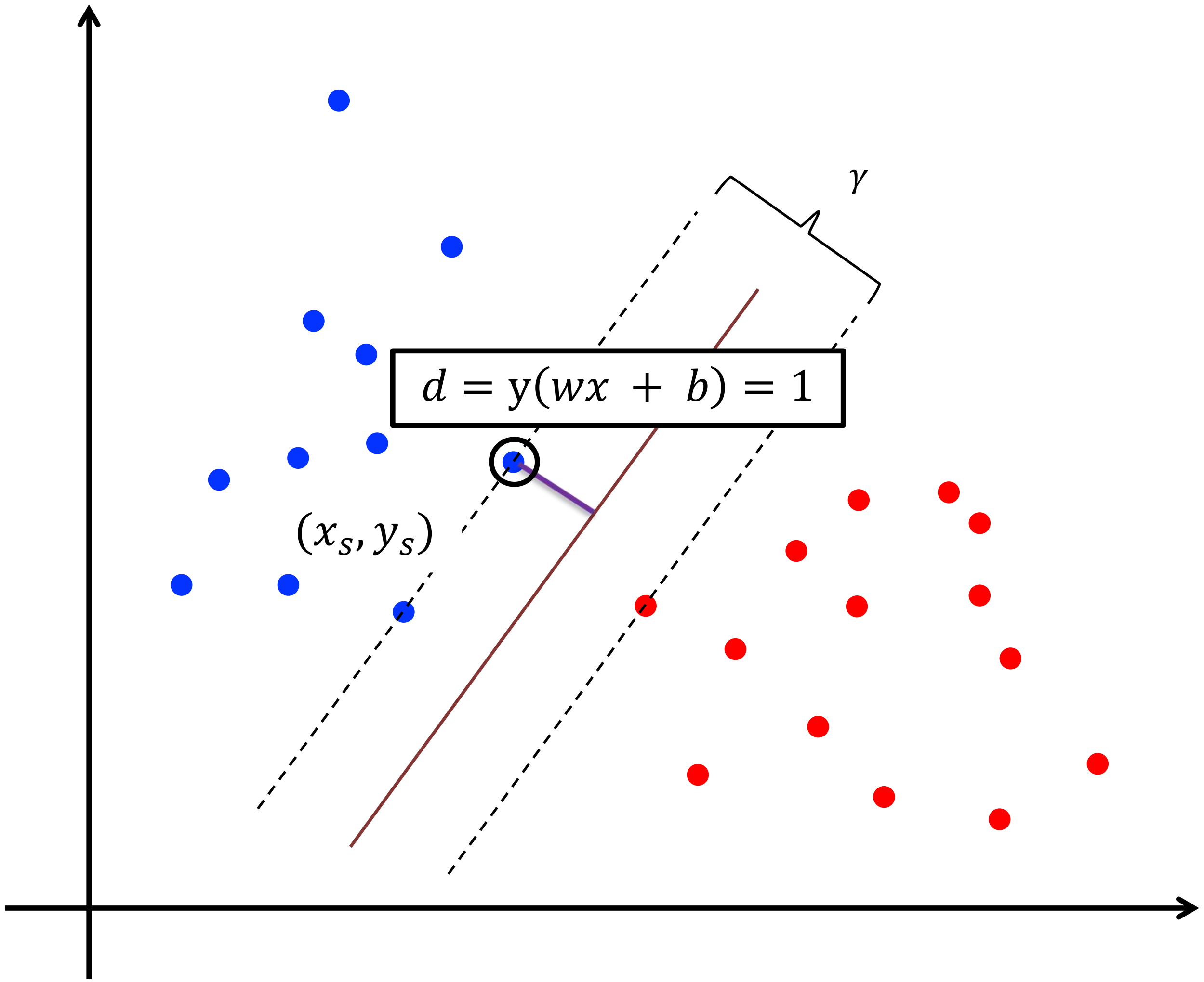
$\downarrow$ y_s的取值为 1 或 −1

$$wx_s + b = y_s$$

$\downarrow$ 改写

$$b = y_s - wx_s \xrightarrow{\text{考虑所有的支持向量}} b = \frac{1}{|s|}\sum_{s\in S} y_s - wx_s$$

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

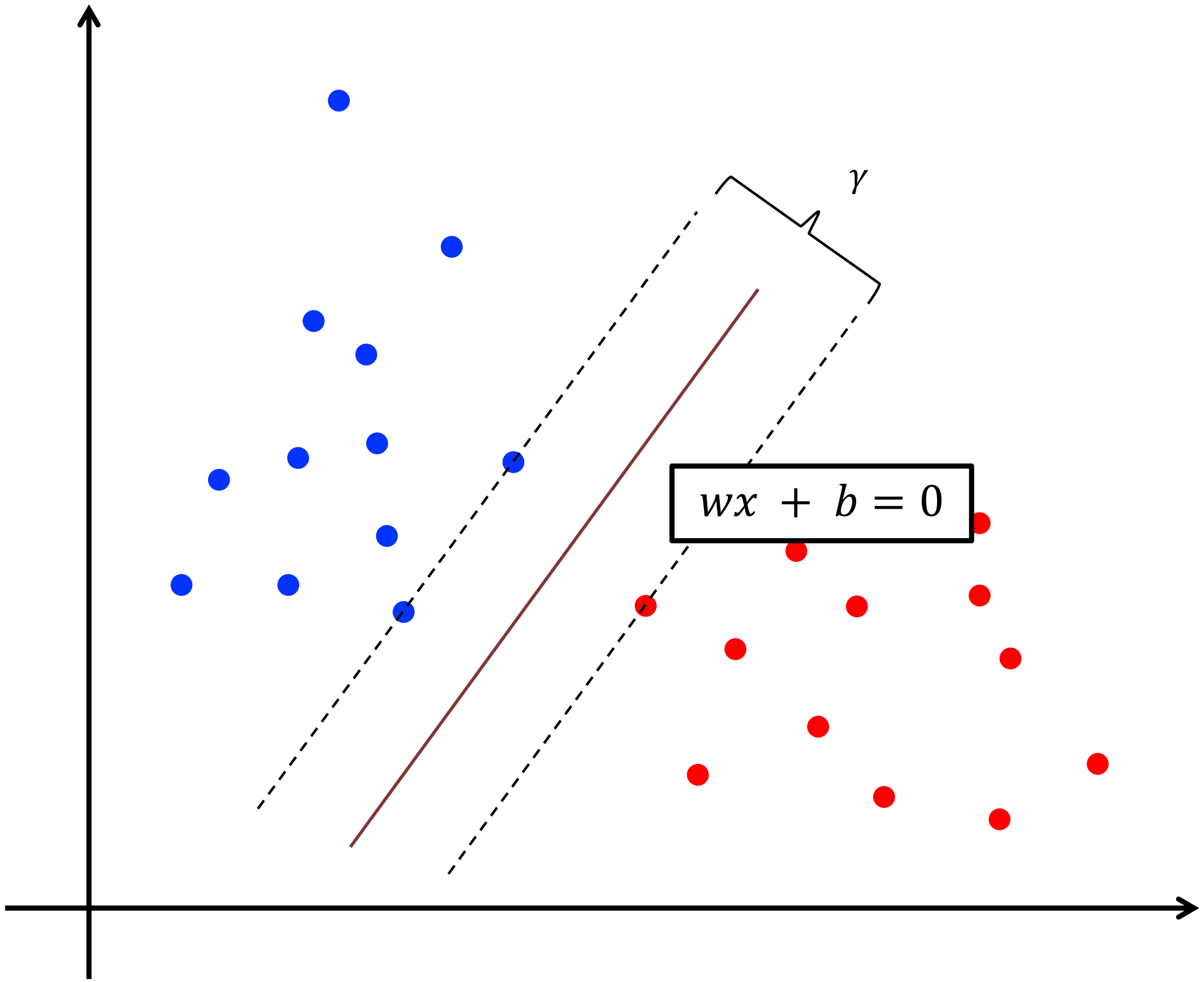


第五步

- 第五步
  - 构造分割超平面
  - 构造分类决策函数
  - 预测

$$f(x) = \text{sign}(wx + b)$$

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

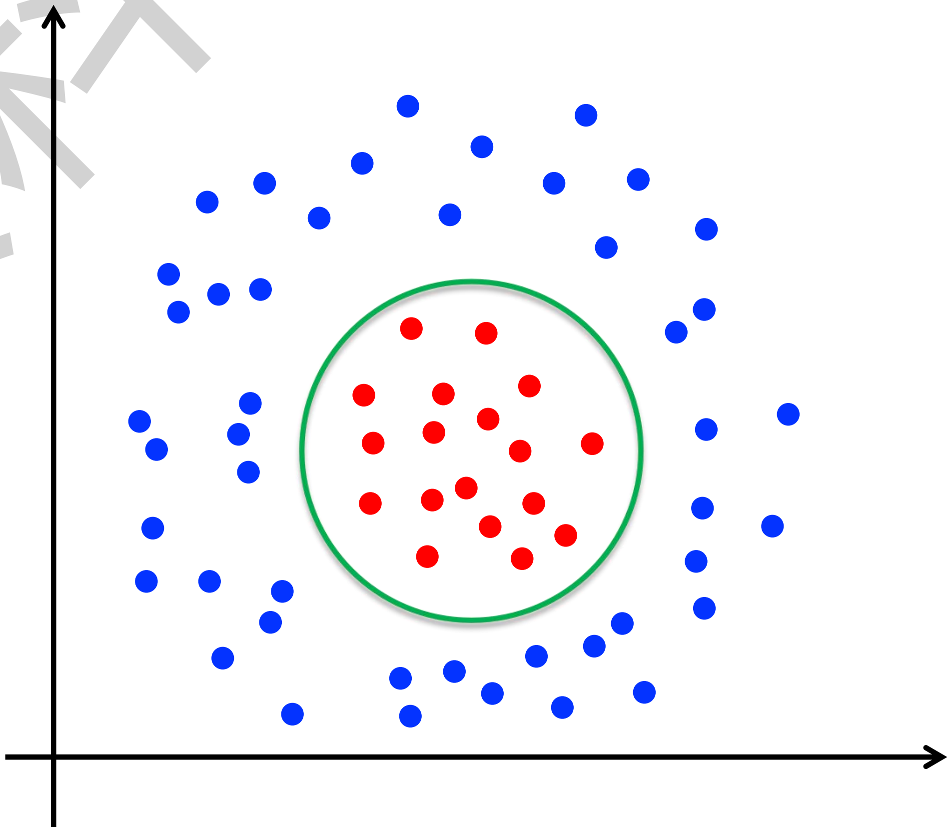


核函数

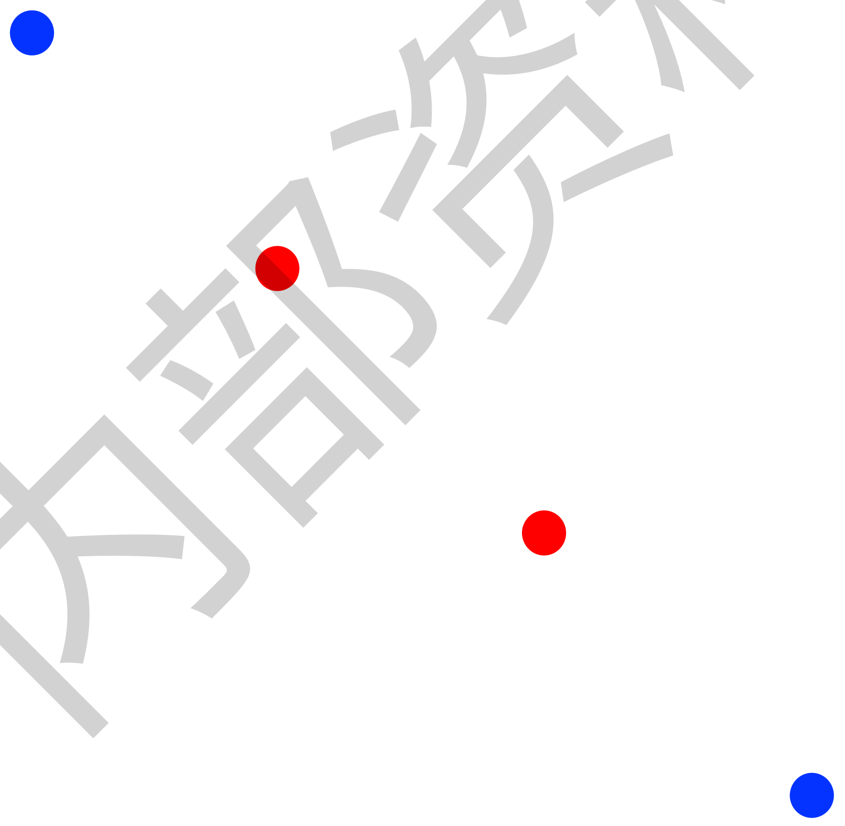
线性不可分

百度智能云

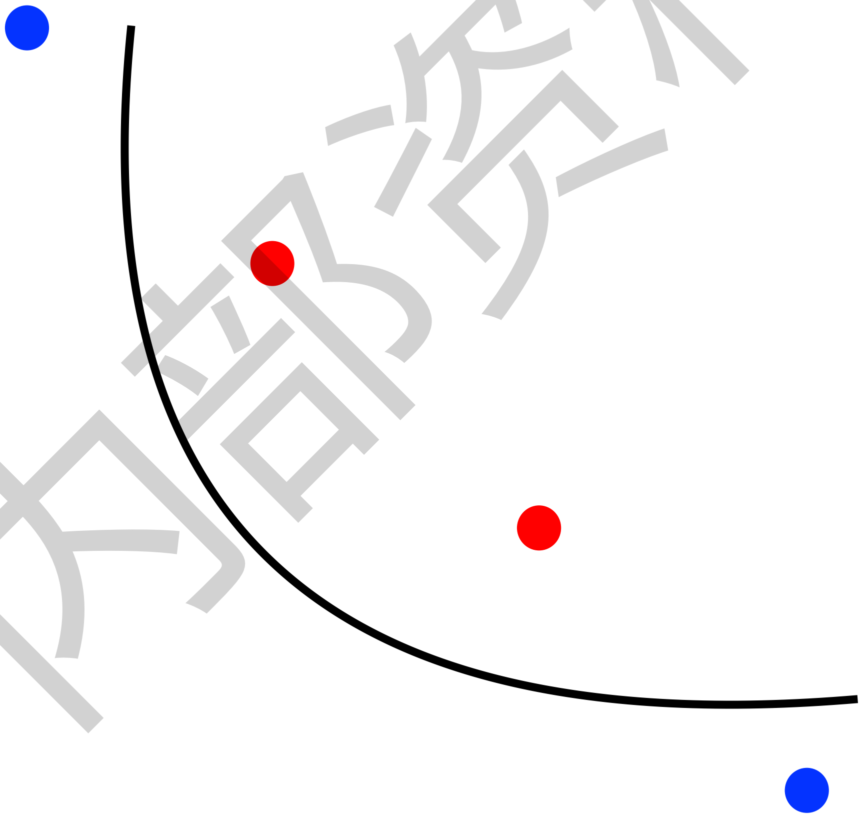
- 线性 SVM 核心都是用线性分类器的方法进行分类。然而，现实生活中很多数据仅用线性的方法是无法分类的



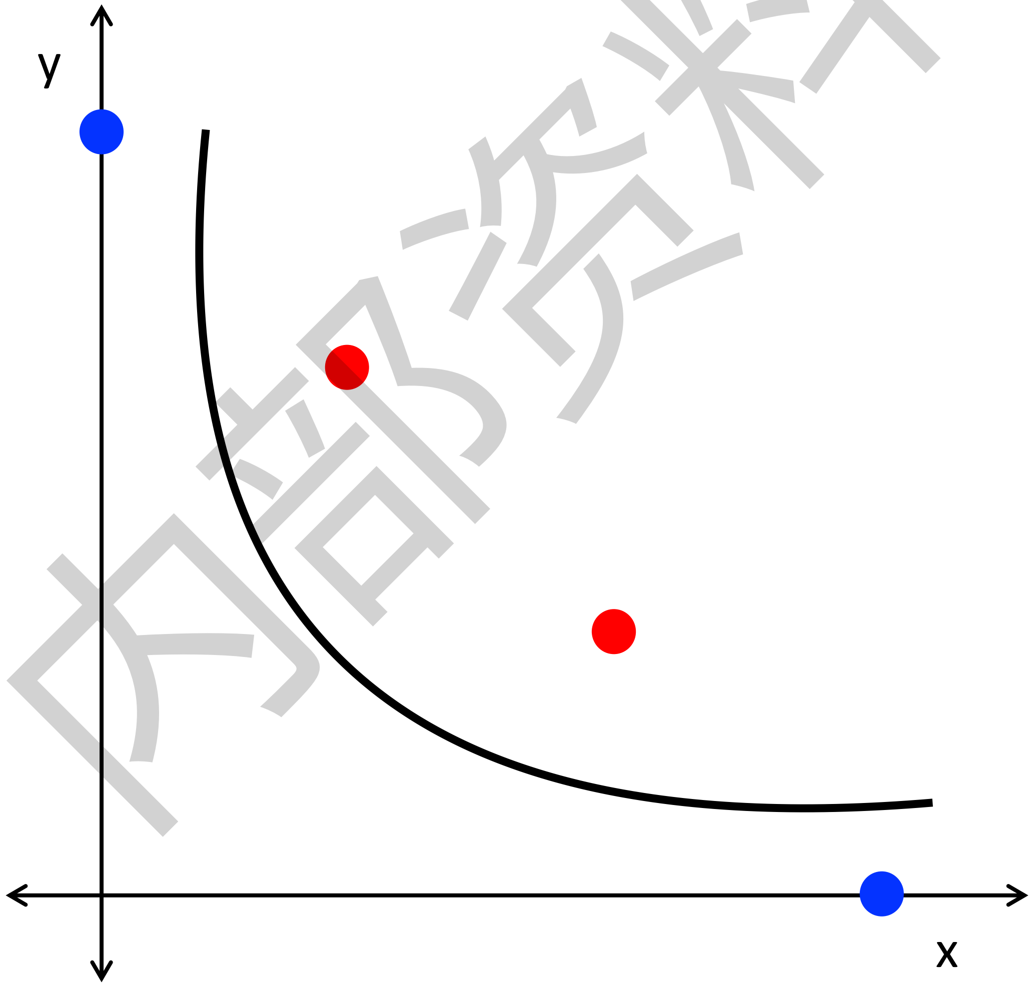
线性不可分



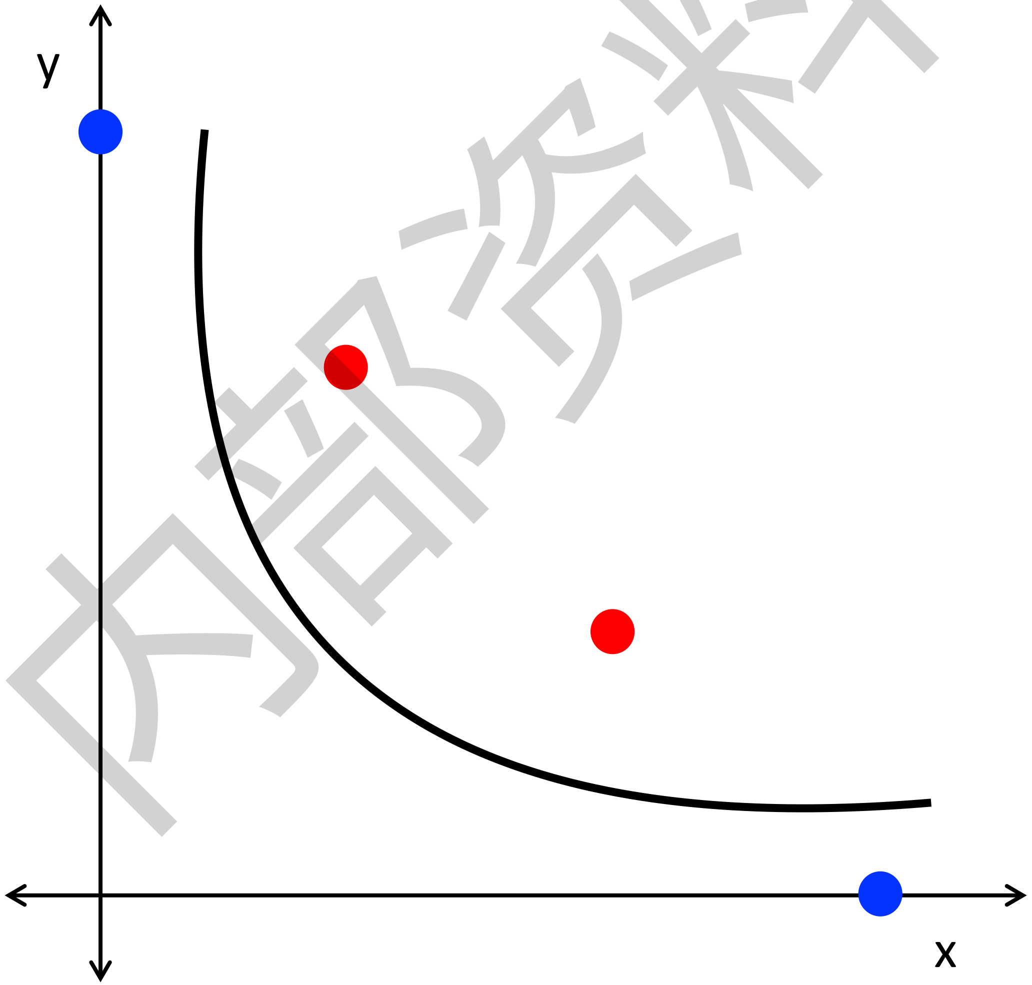
非线性可分

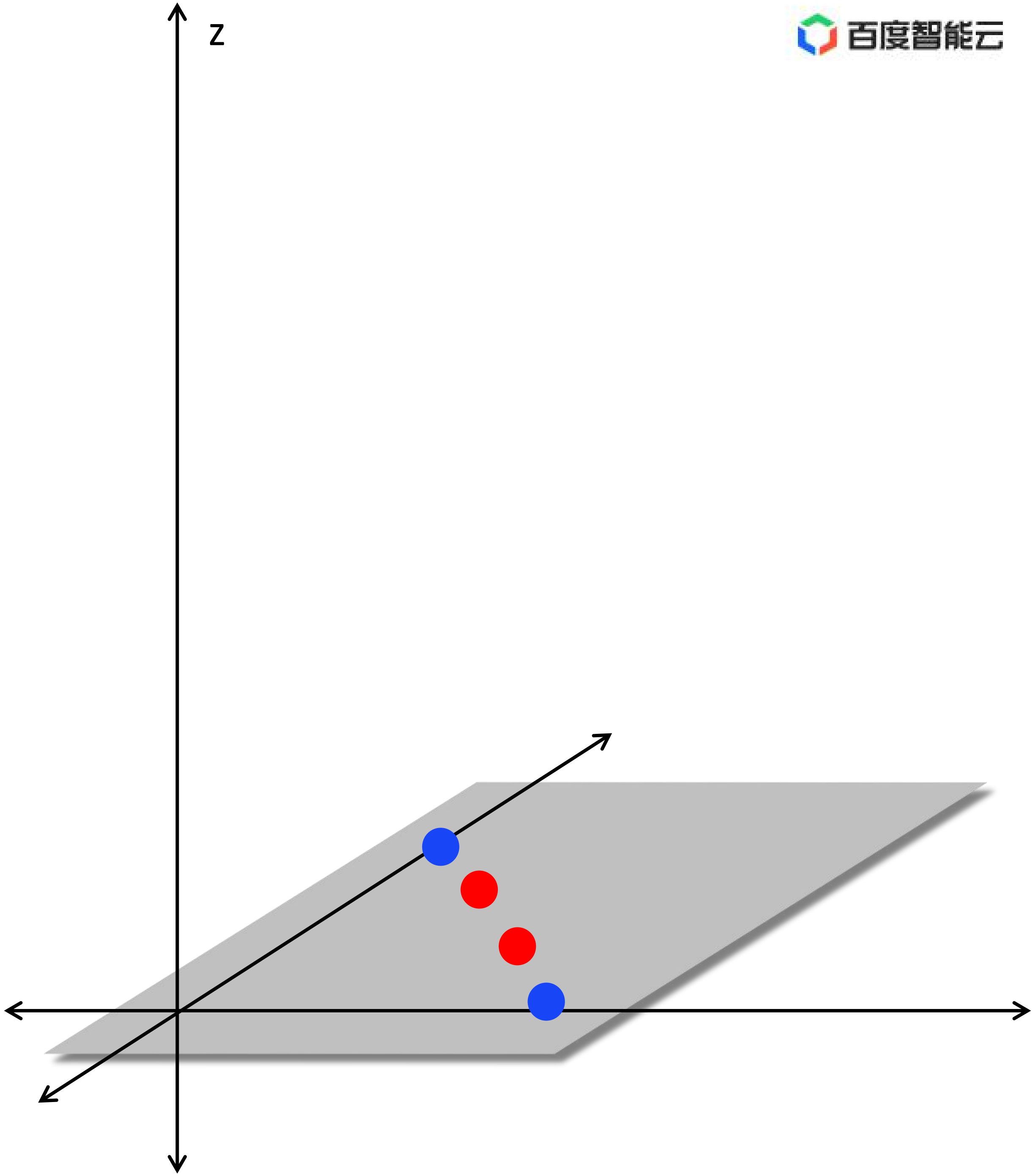


将线与点放到平面直角坐标系中

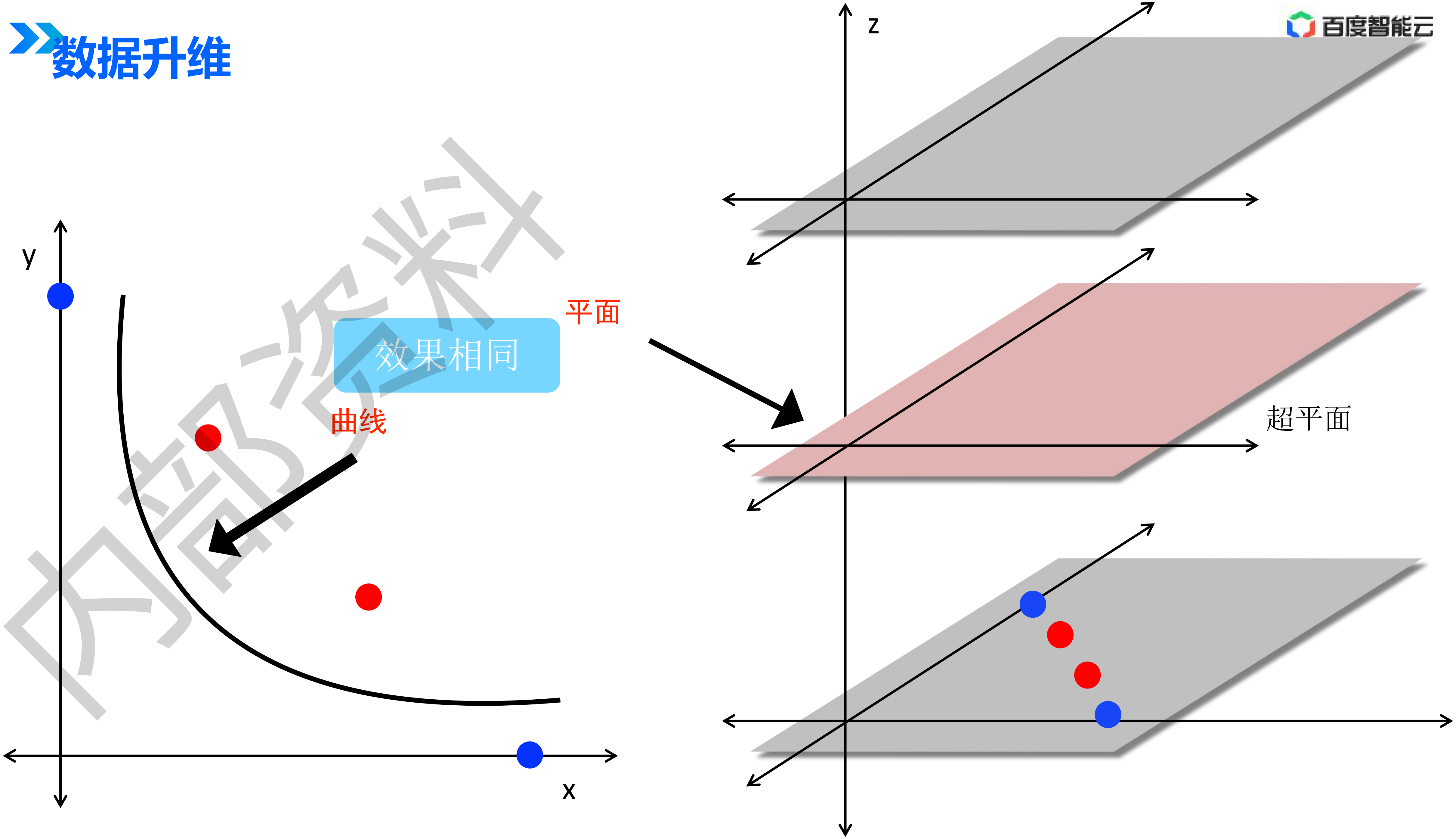


建立空间直角坐标系

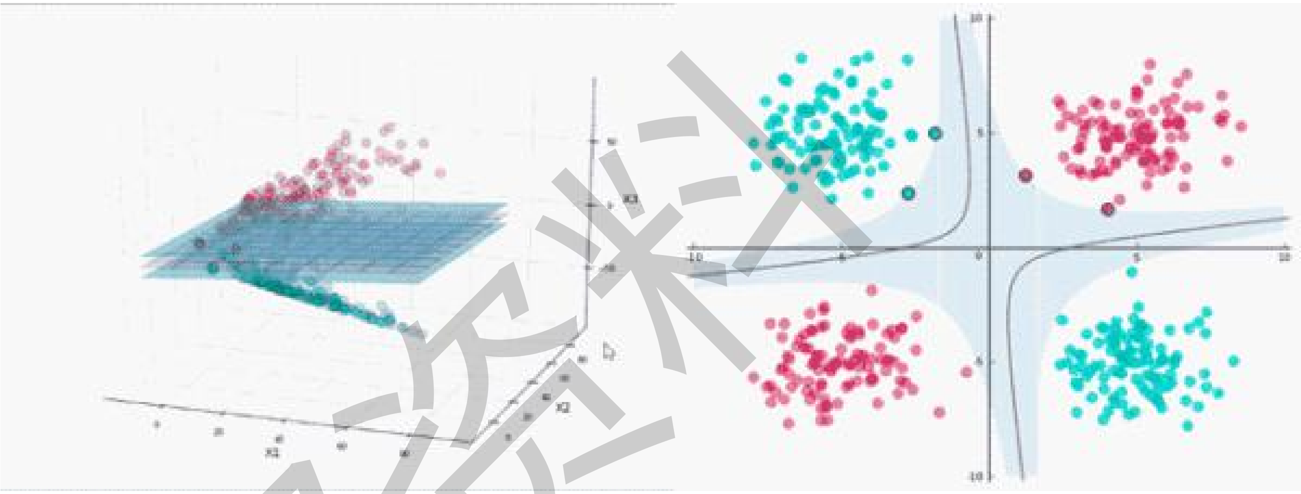


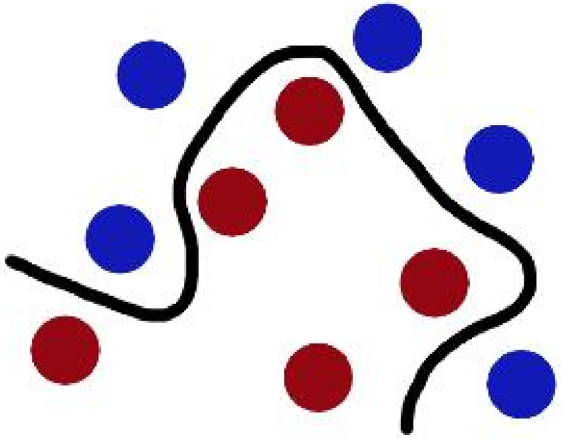


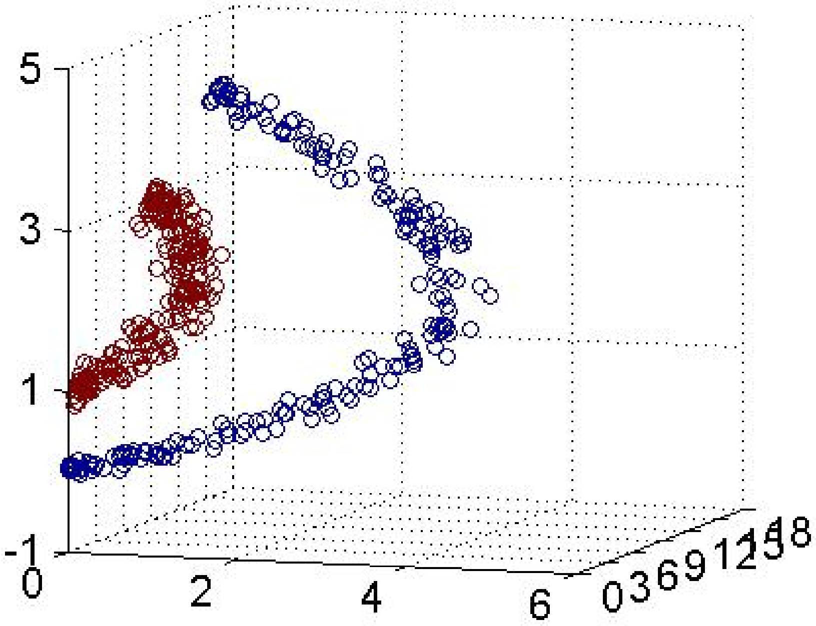
数据升维



数据升维的示例







核函数

- 通过核函数来升维
- 旧目标函数：$f(x) = wx + b$
- 新目标函数：$f(x) = w\,\phi(x) + b$

$$\max_{\lambda}\left[\sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\lambda_i\lambda_j y_i y_j (x_i\cdot x_j)\right]$$

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

$$\lambda_i \geq 0$$

SVM线性对偶优化问题

$\xrightarrow{\text{升维}}$

$$\max_{\lambda}\left[\sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}\lambda_i\lambda_j y_i y_j \left(\phi(x_i)\cdot\phi(x_j)\right)\right]$$

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

$$\lambda_i \geq 0$$

SVM非线性对偶优化问题

常见的核函数

线性核函数

$$K(x_i, x_j) = x_i \cdot x_j$$

多项式核函数

$$K(x_i, x_j) = (x_i \cdot x_j)^d$$

高斯核函数

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

SVM的优缺点

SVM的优缺点

- 优点
  - 有严格的数学理论支持，可解释性强，不依靠统计方法，从而简化了通常的分类和回归问题；
  - 能找出对任务至关重要的关键样本（即：支持向量）；
  - 采用核技巧之后，可以处理非线性分类/回归任务；
  - 最终决策函数只由少数的支持向量所确定，计算的复杂性取决于支持向量的数目，而不是样本空间的维数，这在某种意义上避免了“维数灾难”。

- 缺点
  - 训练时间长。当采用 SMO 算法时，由于每次都需要挑选一对参数，因此时间复杂度为$O(N^2)$，其中 N 为训练样本的数量；
  - 当采用核技巧时，如果需要存储核矩阵，则空间复杂度为$O(N^2)$
  - 模型预测时，预测时间与支持向量的个数成正比。当支持向量的数量较大时，预测计算复杂度较高。

百度智能云

THANK YOU

CLOUD.BAIDU.COM ABCXUEYUAN.BAIDU.COM