

支持向量机

SVM(Support Vector Machines)

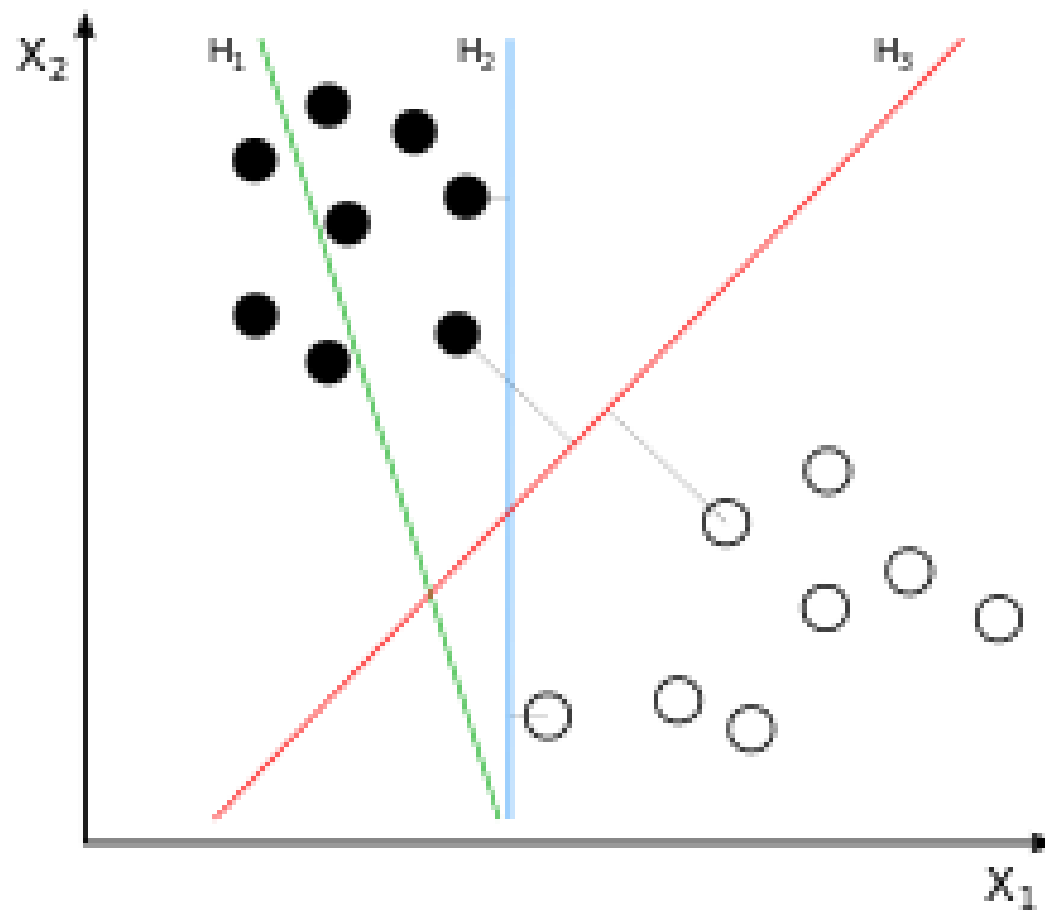


最早是由 Vladimir N. Vapnik 和 Alexey Ya. Chervonenkis 在 1963年提出

目前的版本(soft margin)是由Corinna Cortes 和 Vapnik在1993年提出，并在1995年发表

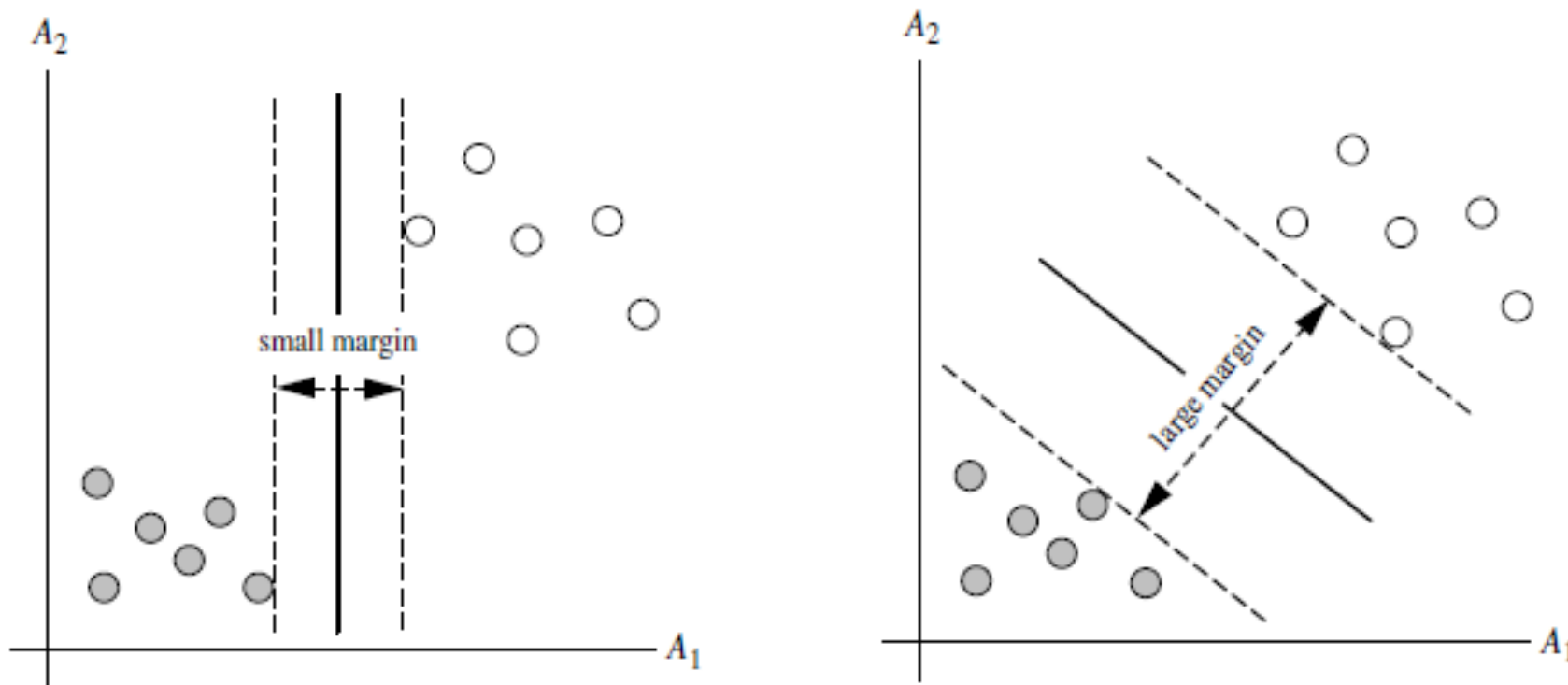
深度学习（2012）出现之前，SVM被认为机器学习中近十几年来最成功，表现最好的算法

SVM





SVM寻找区分两类的超平面 (hyper plane),
使边际(margin)最大





$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{Bmatrix} \quad y = \begin{Bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{Bmatrix}$$

向量内积： $x \cdot y = x_1y_1 + x_2y_2 + \dots + x_ny_n$

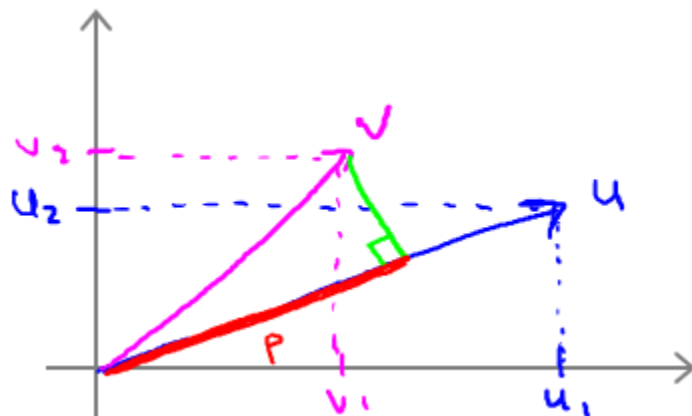
向量内积： $x \cdot y = \|x\| \|y\| \cos(\theta)$

范数： $\|x\| = \sqrt{x \cdot x} = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$

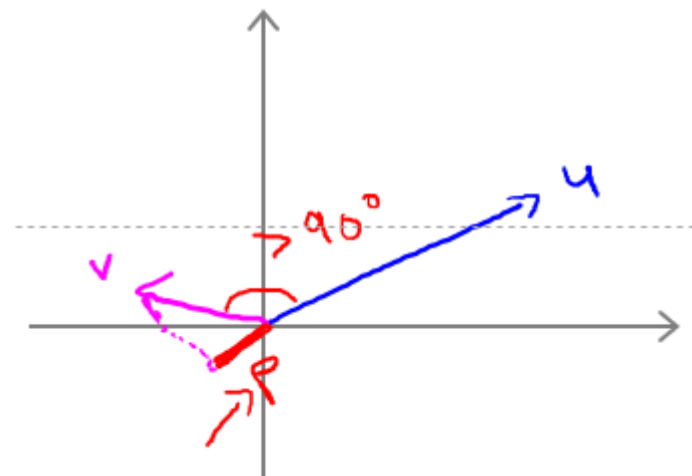
当 $\|x\| \neq 0, \|y\| \neq 0$ 时，可以求余弦相似度：

$$\cos\theta = \frac{x \cdot y}{\|x\| \|y\|}$$

向量内积

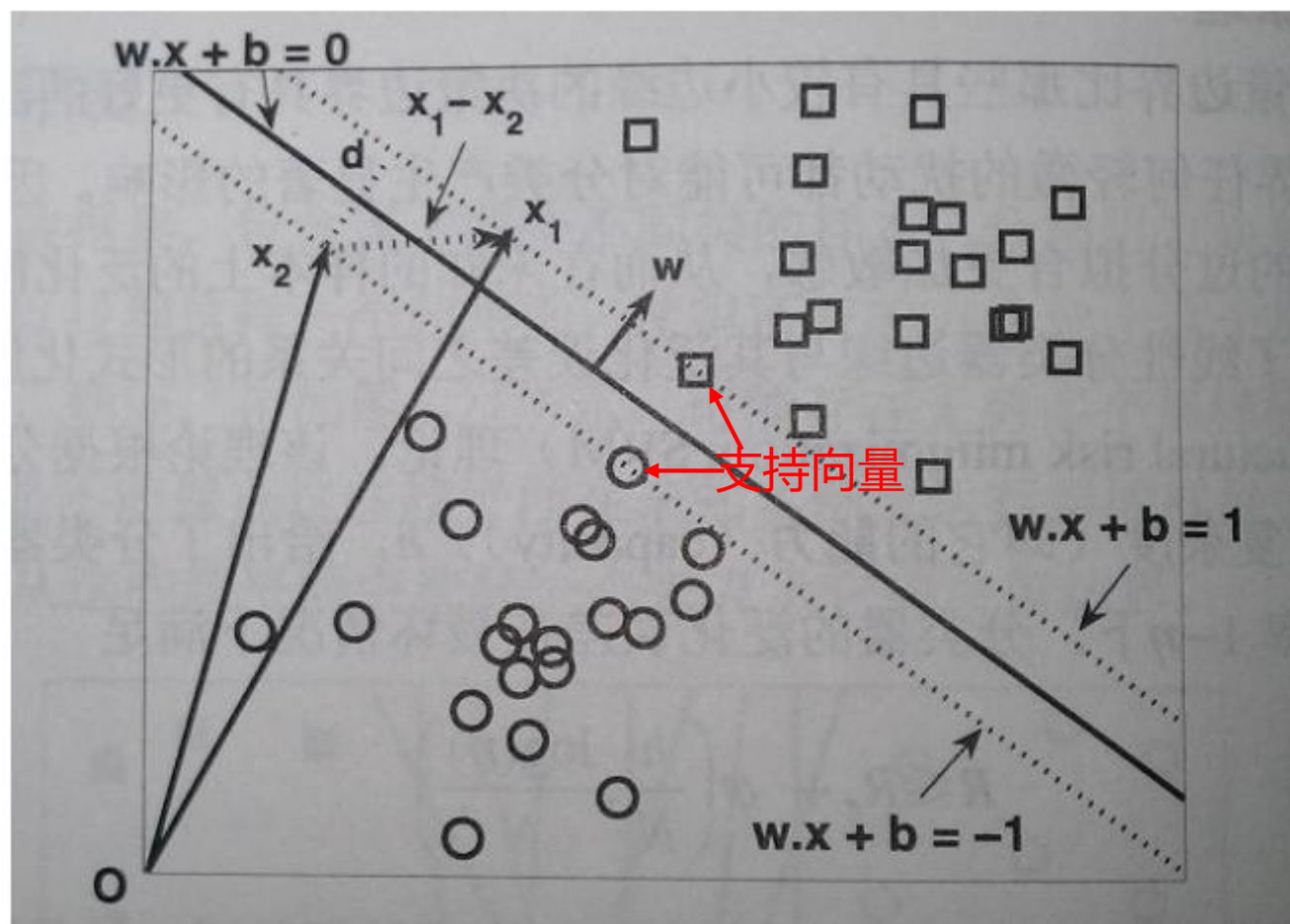


向量内积： $v \cdot u > 0$



向量内积： $v \cdot u < 0$

SVM分类





$$\begin{aligned}w \cdot x + b &= 1 \\w \cdot x + b &= -1\end{aligned}$$

$$\begin{aligned}w \cdot x + b &= 2 \\w \cdot x + b &= -3\end{aligned}$$

$$\begin{aligned}2 \cdot u + v &= 1 \\-3 \cdot u + v &= -1 \\u &= -2/5 \\v &= 1/5\end{aligned}$$

$$\begin{aligned}w \cdot x_1 + b &= 1 \\w \cdot x_2 + b &= -1\end{aligned}$$

$$\begin{aligned}w \cdot (x_1 - x_2) &= 2 \\||w|| ||(x_1 - x_2)|| \cos(\theta) &= 2 \\||w|| \cdot d &= 2 \\d &= \frac{2}{||w||}\end{aligned}$$

SVM简单例子



转化为凸优化问题



$$\left. \begin{array}{l} w \cdot x + b \geq 1, \text{ 则分类 } y=1 \\ w \cdot x + b \leq -1, \text{ 则分类 } y=-1 \end{array} \right\} y(w \cdot x + b) \geq 1$$

求 $d = \frac{2}{\|w\|}$ 最大值 ,

也就是求 $\min \frac{\|w\|^2}{2}$



1. 无约束优化问题：

$$\min f(\mathbf{x})$$

-费马定理

2. 带等式约束的优化问题：

$$\min f(\mathbf{x})$$

-拉格朗日乘子法：

$$s.t. h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, n$$

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^n \lambda_i h_i(\mathbf{x})$$

3. 带不等式约束的优化问题：

$$\min f(\mathbf{x})$$

-KKT条件

$$s.t. h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, n$$

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, k$$

$$\mathcal{L}(\mathbf{x}, \lambda, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^k \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^n v_i h_i(\mathbf{x})$$

广义拉格朗日乘子法



目标函数 α 拉格朗日乘子 约束条件

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1)$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0$$

跟岭回归和LASSO类似



岭回归代价函数：

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

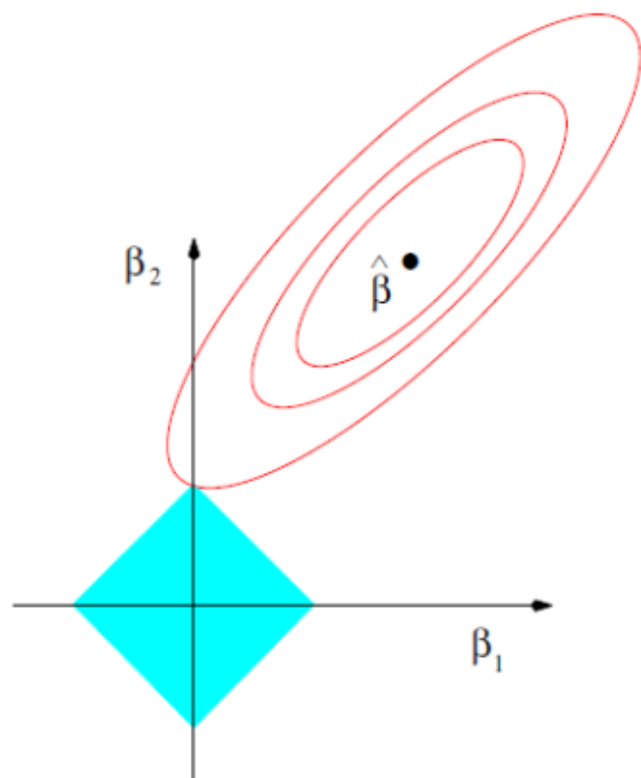
λ 的值可以用于限制 $\sum_{j=1}^n \theta_j^2 \leq t$

LASSO代价函数：

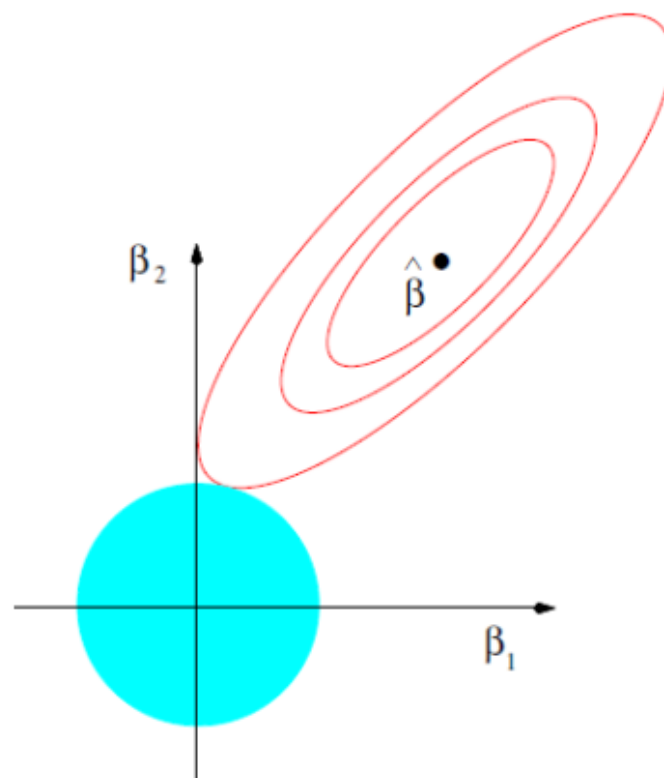
$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n |\theta_j| \right]$$

λ 的值可以用于限制 $\sum_{j=1}^n |\theta_j| \leq t$

跟岭回归和LASSO类似



LASSO



岭回归

Karush-Kuhn-Tucker最优化条件(KKT条件)



拉格朗日乘子法的一种推广，可以处理有不等号的约束条件。

$$\begin{aligned} \min f(\mathbf{x}) \\ \text{s.t. } h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, n \\ g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, k \end{aligned}$$

$$\mathcal{L}(x, \lambda, v) = f(x) + \sum_{i=1}^k \lambda_i g_i(x) + \sum_{i=1}^n v_i h_i(x)$$

进一步简化为对偶问题



$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1)$$

上述问题可以改写成：

$$\min_{w, b} \max_{\alpha_i \geq 0} \mathcal{L}(w, b, \alpha) = p^*$$

可以等价于下列对偶问题：

$$\max_{\alpha_i \geq 0} \min_{w, b} \mathcal{L}(w, b, \alpha) = d^*$$

进一步简化为对偶问题



$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1)$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0$$

把w和b消除了

→
$$L(w, b, a) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j$$

进一步简化为对偶问题



$$\max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} \left[\sum_{i=1}^k \alpha_i - \frac{1}{2} \sum_{i,j=1}^k \alpha_i \alpha_j y_i y_j (x_i)^T x_j \right]$$

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

$$\min_{\boldsymbol{\alpha}} \left[\frac{1}{2} \sum_{i,j=1}^k \alpha_i \alpha_j y_i y_j (x_i)^T x_j - \sum_{i=1}^k \alpha_i \right] = \min_{\boldsymbol{\alpha}} \left[\frac{1}{2} \sum_{i,j=1}^k \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^k \alpha_i \right]$$

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

进一步简化为对偶问题



由此可以求出最优解 α^* ，求出该值后将其带入可以得到：

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

$$b^* = y_i - (w^*)^T x_i$$



Microsoft Research的John C. Platt在1998年提出
针对线性SVM和数据稀疏时性能更优

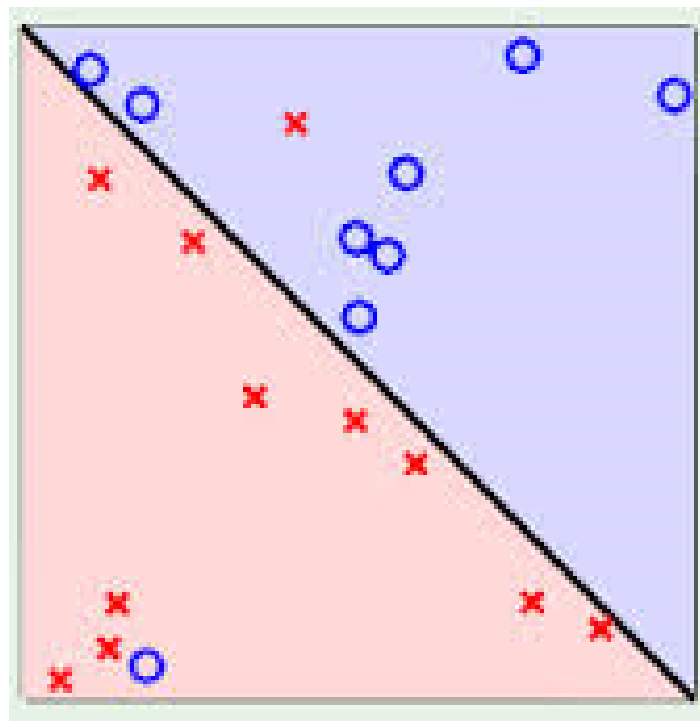
$$\min_{\alpha} \left[\frac{1}{2} \sum_{i,j=1}^k \alpha_i \alpha_j y_i y_j (x_i)^T x_j - \sum_{i=1}^k \alpha_i \right] = \min_{\alpha} \left[\frac{1}{2} \sum_{i,j=1}^k \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^k \alpha_i \right]$$

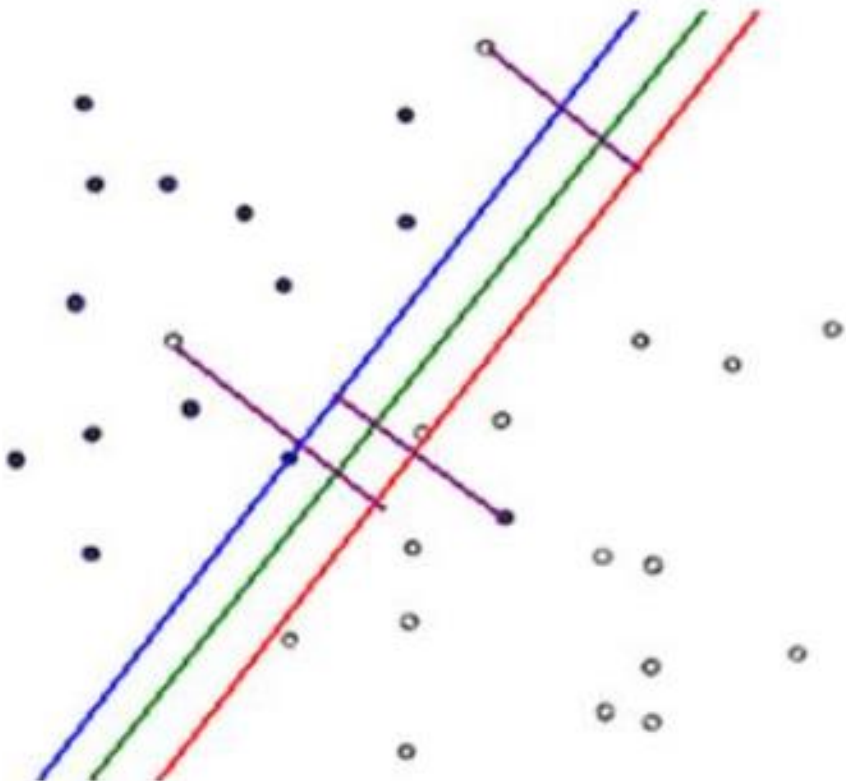
$$s.t. \sum_{i=1}^k \alpha_i y_i = 0, \quad \alpha_i \geq 0, i = 1, 2, \dots, n$$

$$s.t., C \geq \alpha_i \geq 0, i=1, \dots, n$$

基本思路是先根据约束条件随机给 α 赋值。然后每次选取两个 α ，调节这两个 α 使得目标函数最小。然后再选取两个 α ，调节 α 使得目标函数最小。以此类推

线性不可分的情况





$$y_i(w_i \cdot x_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0$$

约束条件没有体现错误分类
的点要尽量接近分类边界

$$\min \frac{\|w\|^2}{2} + C \sum_{i=1}^n \varepsilon_i$$

使得分错的点越少越好，距
离分类边界越近越好

线性不可分情形下的对偶问题

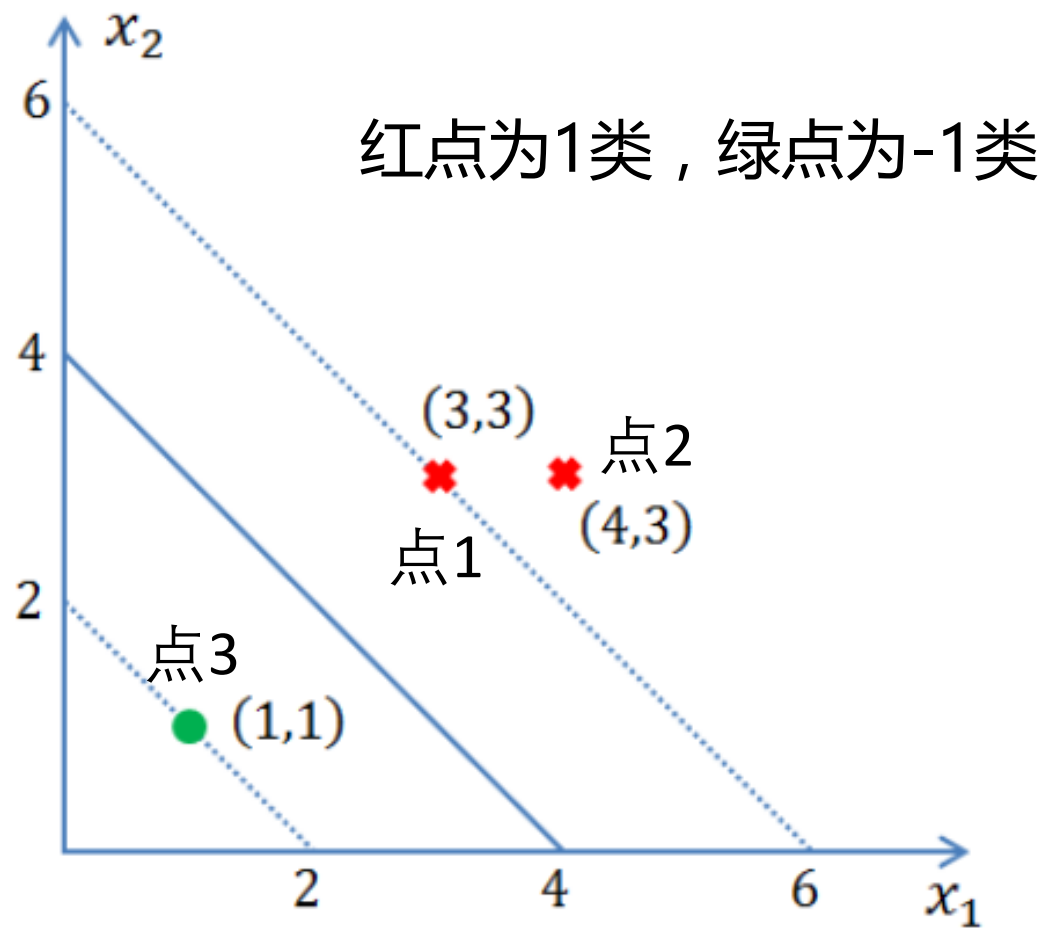


$$\min_{\alpha} \left[\frac{1}{2} \sum_{i,j=1}^k \alpha_i \alpha_j y_i y_j (x_i)^T x_j - \sum_{i=1}^k \alpha_i \right] = \min_{\alpha} \left[\frac{1}{2} \sum_{i,j=1}^k \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^k \alpha_i \right]$$

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

$$\text{s.t., } C \geq \alpha_i \geq 0, i=1, \dots, n$$

SVM例子





可知目标函数为

$$\min_{\alpha} f(\alpha), \quad s.t. \alpha_1 + \alpha_2 - \alpha_3 = 0, \quad \alpha_i \geq 0, i = 1, 2, 3$$

其中

$$f(\alpha) = \frac{1}{2} \sum_{i,j=1}^3 \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^3 \alpha_i$$

求内积

$$= \frac{1}{2} (18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

然后，将 $\alpha_3 = \alpha_1 + \alpha_2$ 带入目标函数，得到一个关于 α_1 和 α_2 的函数

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



对 α_1 和 α_2 求偏导数并令其为0，易知 $s(\alpha_1, \alpha_2)$ 在点 $(1.5, -1)$ 处取极值。而该点不满足 $a_i \geq 0$ 的约束条件，于是可以推断最小值在边界上达到。经计算当 $\alpha_1 = 0$ 时， $s(\alpha_1 = 0, \alpha_2 = 2/13) = -0.1538$ ；当 $\alpha_2 = 0$ 时， $s(\alpha_1 = 1/4, \alpha_2 = 0) = -0.25$ 。于是 $s(\alpha_1, \alpha_2)$ 在 $\alpha_1 = 1/4, \alpha_2 = 0$ 时取得最小值，此时亦可算出 $\alpha_3 = \alpha_1 + \alpha_2 = 1/4$ 。因为 α_1 和 α_3 不等于0，所以对应的点 x_1 和 x_3 就应该是支持向量。



进而可以求得

$$\mathbf{w}^* = \sum_{i=1}^3 \alpha_i^* y_i x_i = \frac{1}{4} \times (3, 3) - \frac{1}{4} \times (1, 1) = \left(\frac{1}{2}, \frac{1}{2}\right)$$

即 $w_1 = w_2 = 0.5$ 。进而有

$$b^* = 1 - (w_1, w_2) \cdot (3, 3) = -2$$

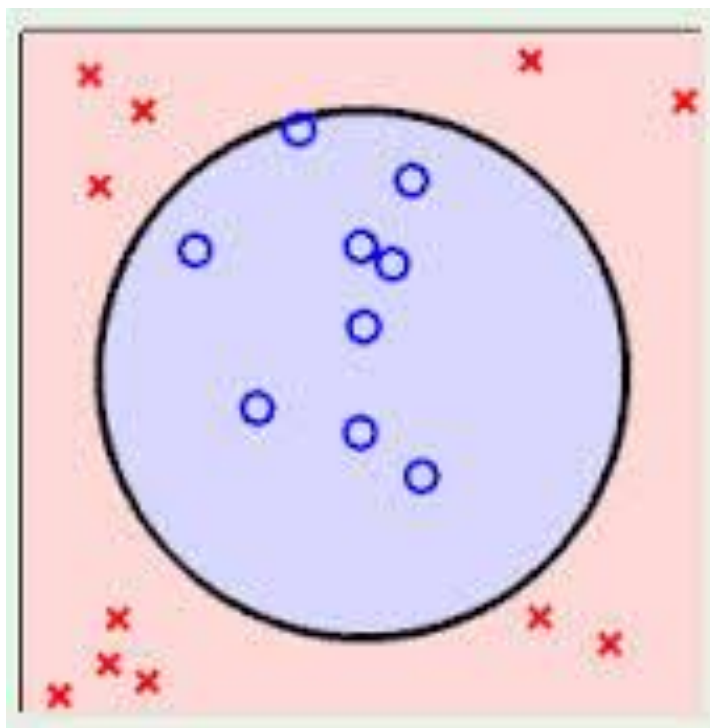
因此最大间隔分类超平面为

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

分类决策函数为

$$f(\mathbf{x}) = \text{sign}\left(\frac{1}{2}x_1 + \frac{1}{2}x_2 - 2\right)$$

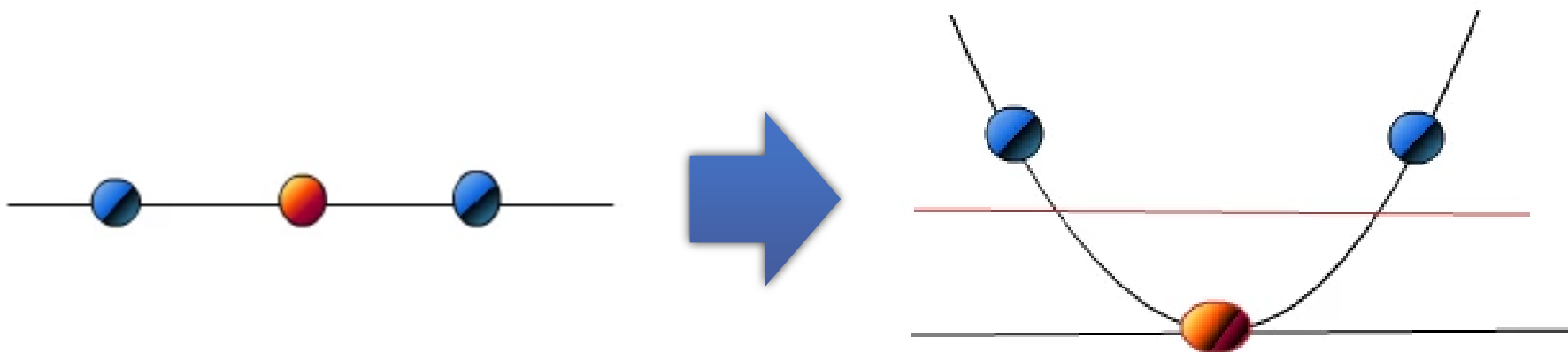
非线性的情况



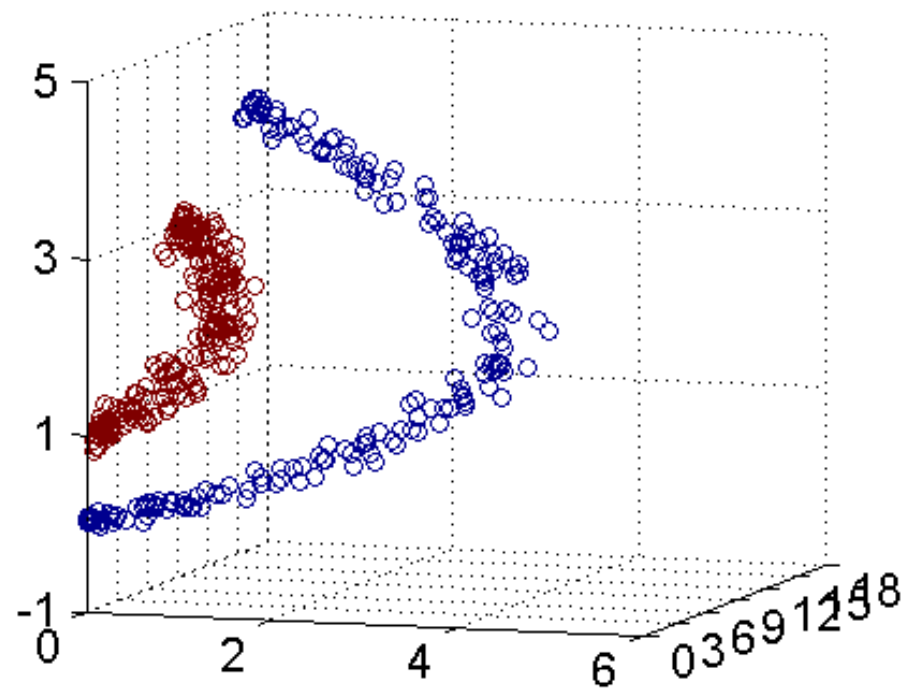
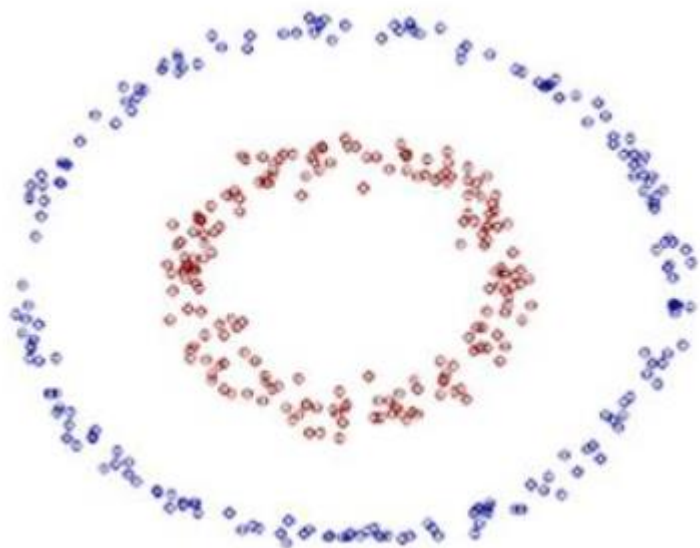
非线性的情况



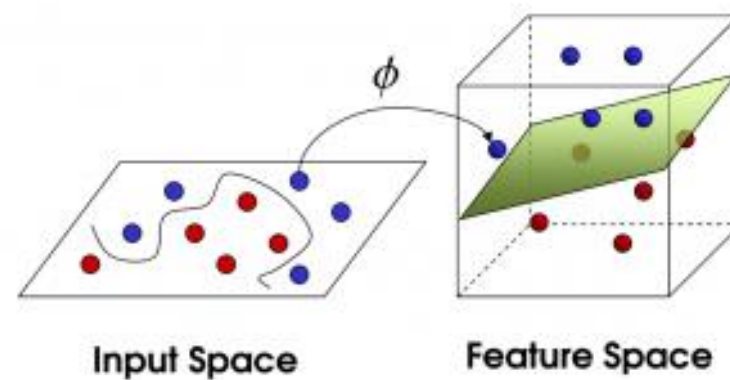
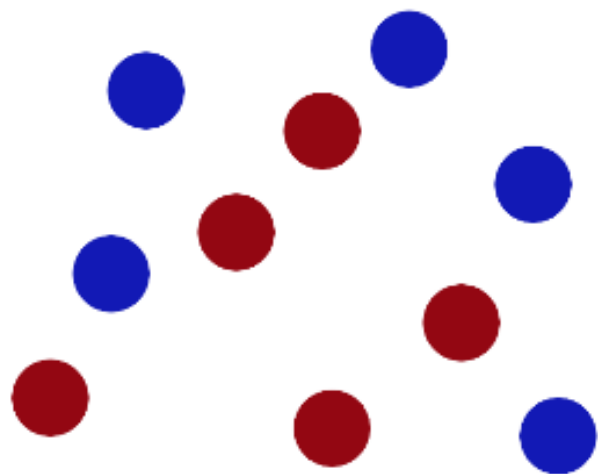
把低维空间的非线性问题映射到
高维空间，变成求解线性问题



非线性的情况



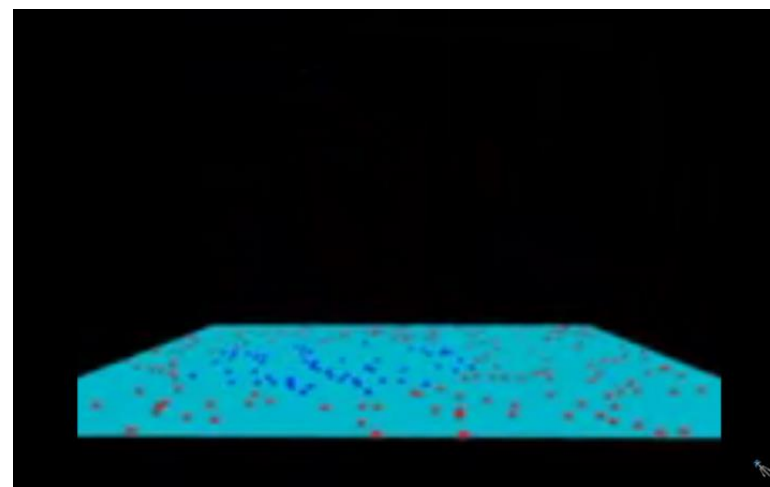
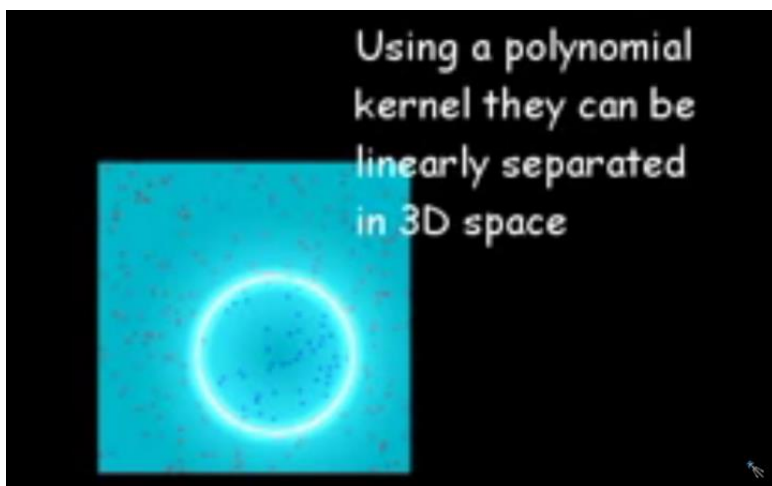
非线性的情况



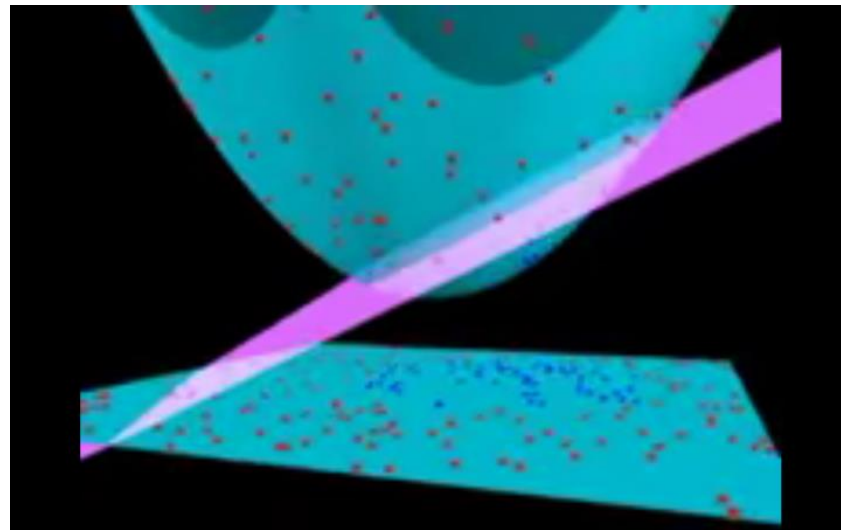
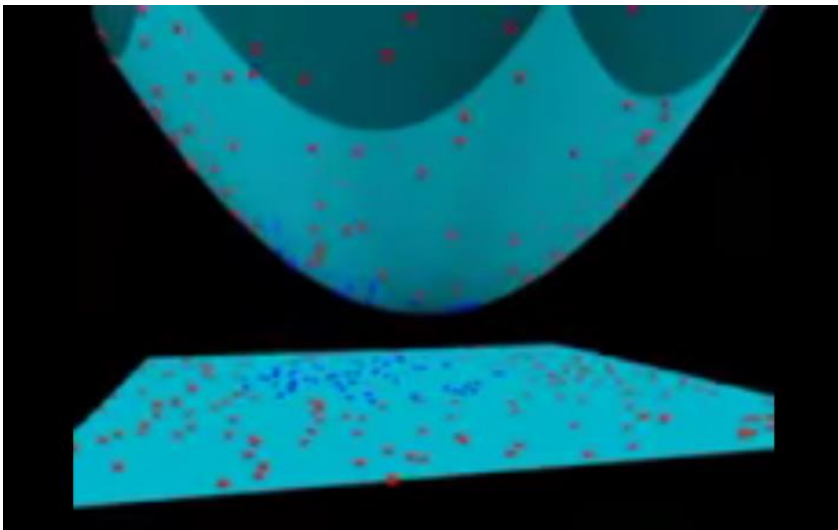
非线性的情况



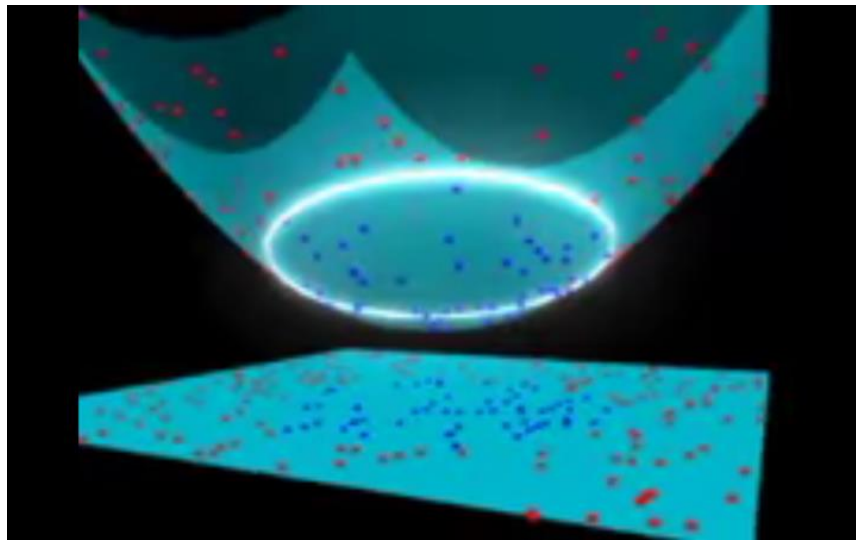
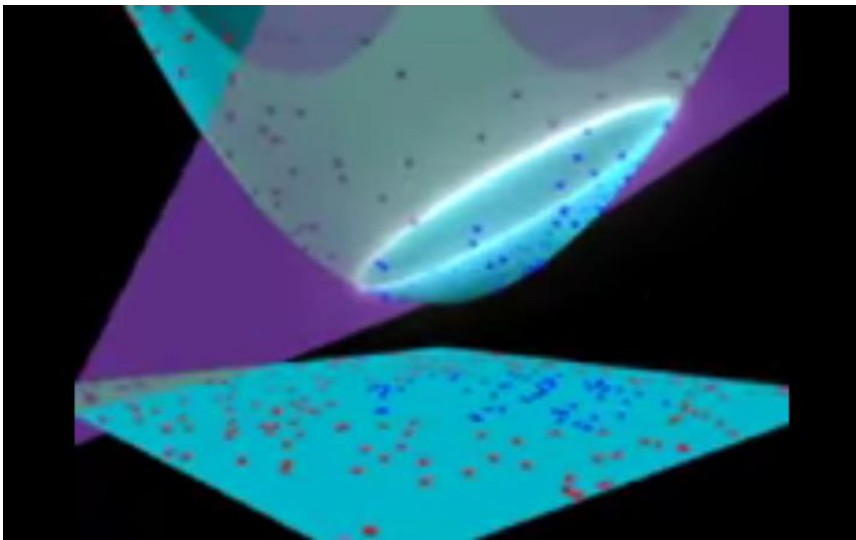
<https://v.qq.com/x/page/k05170ntgzc.html>



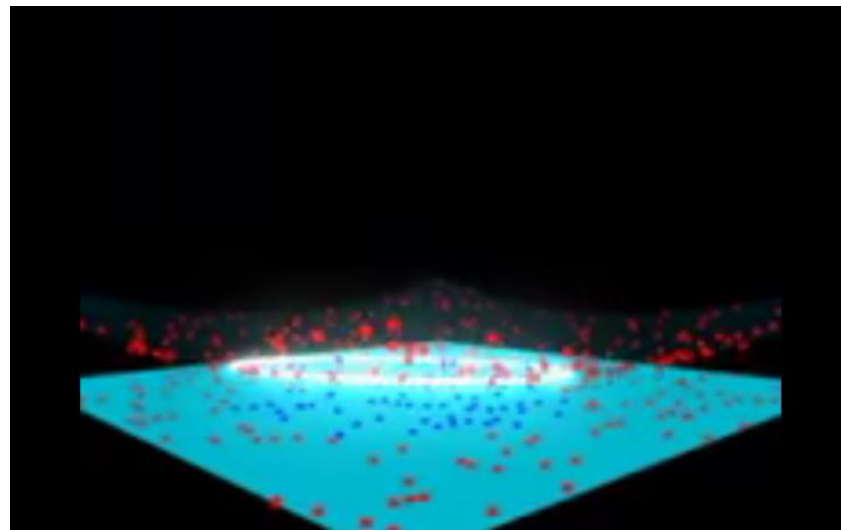
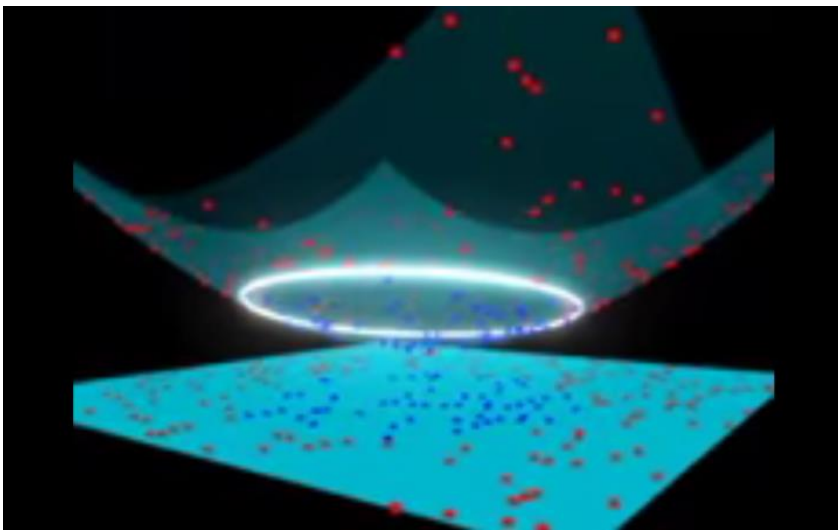
非线性的情况



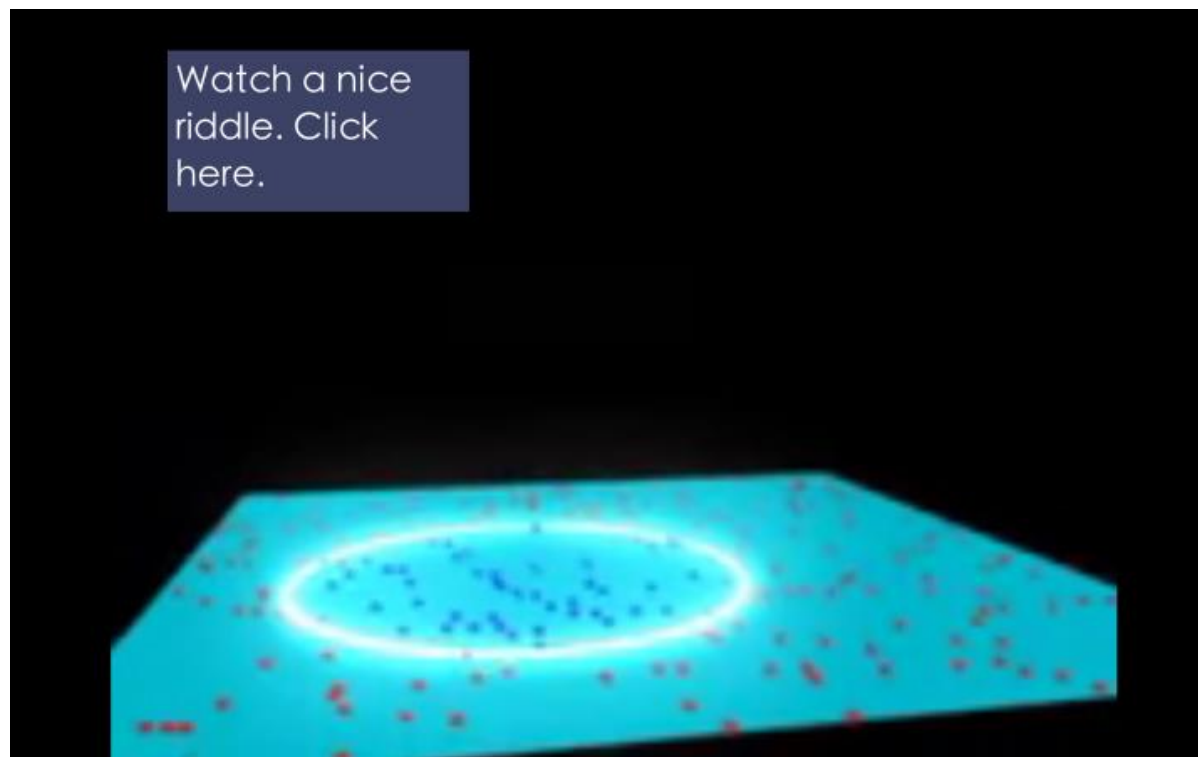
非线性的情况



非线性的情况



非线性的情况



SVM-低维映射高维





3维输入向量： $X = (x_1, x_2, x_3)$

转化到6维空间 Z 中去：

$$\phi_1(X) = x_1, \phi_2(X) = x_2, \phi_3(X) = x_3, \phi_4(X) = (x_1)^2, \phi_5(X) = x_1x_2, \text{ and } \phi_6(X) = x_1x_3.$$

新的决策超平面： $d(Z) = WZ + b$ ，其中 W 和 Z 是向量，这个超平面是线性的，解出 W 和 b 之后，并且带回原方程：

$$\begin{aligned} d(Z) &= w_1x_1 + w_2x_2 + w_3x_3 + w_4(x_1)^2 + w_5x_1x_2 + w_6x_1x_3 + b \\ &= w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6z_6 + b \end{aligned}$$



$$\min_{\alpha} \left[\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \right], \sum_{i=1}^k \alpha_i y_i = 0, C \geq \alpha_i \geq 0$$

$$\min_{\alpha} \left[\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \right], \sum_{i=1}^k \alpha_i y_i = 0, C \geq \alpha_i \geq 0$$

1. 维度灾难

红色的地方要使用映射后的样本向量做内积

假如最初的特征是n维的，我们把它映射到n²维，然后再计算。这样需要的时间从原来的O(n)，变成了O(n²)

2. 如何选择合理的非线性转换？



我们可以构造核函数使得运算结果等同于非线性映射，同时运算量要远远小于非线性映射。

$$K(X_i, X_j) = \phi(X_i) \cdot \phi(X_j)$$

h 次多项式核函数: $K(X_i, X_j) = (X_i \cdot X_j + 1)^h$

高斯径向基函数核函数: $K(X_i, X_j) = e^{-\|X_i - X_j\|^2 / 2\sigma^2}$

S 型核函数: $K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$

核函数举例



假设定义两个向量： $x = (x_1, x_2, x_3)$; $y = (y_1, y_2, y_3)$

定义高维映射方程： $f(x) = (x_1x_1, x_1x_2, x_1x_3, x_2x_1, x_2x_2, x_2x_3, x_3x_1, x_3x_2, x_3x_3)$

假设 $x = (1, 2, 3)$, $y = (4, 5, 6)$.

$f(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9)$

$f(y) = (16, 20, 24, 20, 25, 36, 24, 30, 36)$

求内积 $\langle f(x), f(y) \rangle = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024$

定义核函数： $K(x,y) = (\langle f(x), f(y) \rangle)^{1/2}$

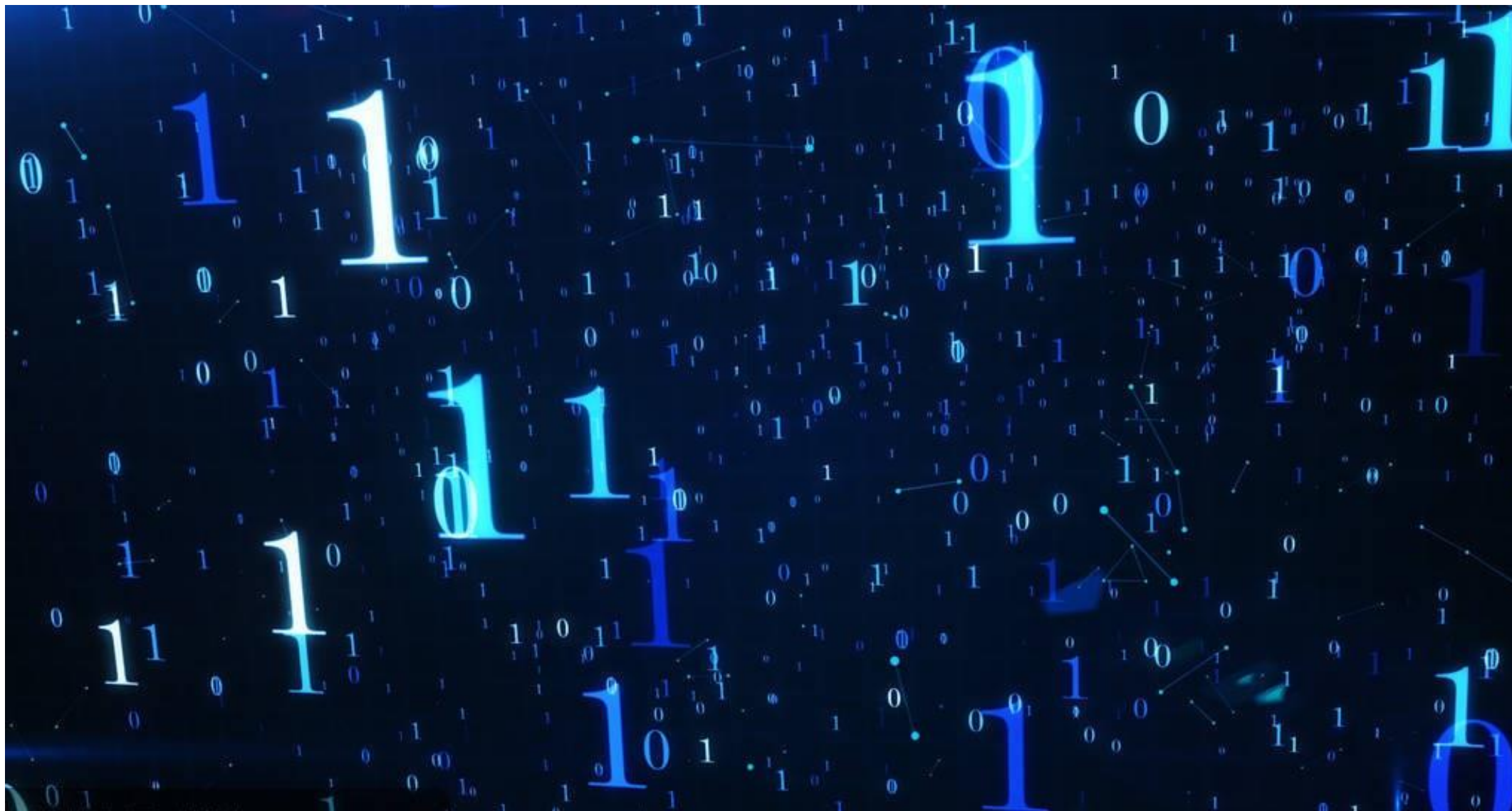
$K(x,y) = (4 + 10 + 18)^{1/2} = 1024$

同样的结果，使用核方法计算容易得多。



- 训练好的模型的算法复杂度是由支持向量的个数决定的，而不是由数据的维度决定的。所以SVM不太容易产生 overfitting
- SVM训练出来的模型完全依赖于支持向量(Support Vectors), 即使训练集里面所有非支持向量的点都被去除，重复训练过程，结果仍然会得到完全一样的模型。
- 一个SVM如果训练得出的支持向量个数比较小，SVM训练出的模型比较容易被泛化。

SVM-线性分类



SVM-非线性分类





<http://vis-www.cs.umass.edu/lfw/>

Labeled Faces in the Wild



Menu

- LFW Home
 - Mailing
 - Explore
 - Download
 - Train/Test
 - Results
 - Information
 - Errata
 - Reference
 - Resources
 - Contact
 - Support
 - Changes
- Part Labels
- UMass Vision

Labeled Faces in the Wild Home



NEW SURVEY PAPER:

Erik Learned-Miller, Gary B. Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua.

Labeled Faces in the Wild: A Survey.

In *Advances in Face Detection and Facial Image Analysis*, edited by Michal Kawulok, M. Emre Celebi, and Bogdan Smolka, Springer, pages 189-248, 2016.

[\[Springer Page\]](#) [\[Draft pdf\]](#)

SVM-人脸识别





「关注 AI MOOC 官方公众号」

这里持续分享
Python、机器学习和深度学习的
各种资源干货及
有趣又实用的硬知识



AI MOOC创始人

网易计算机视觉微专业核心讲师

机器学习、深度学习多年开发经验

受邀为中国移动、国家电网、华夏银行、
太平洋保险等世界五百强企业总部做AI内训

