



# Support Vector Machine

张长水

清华大学自动化系

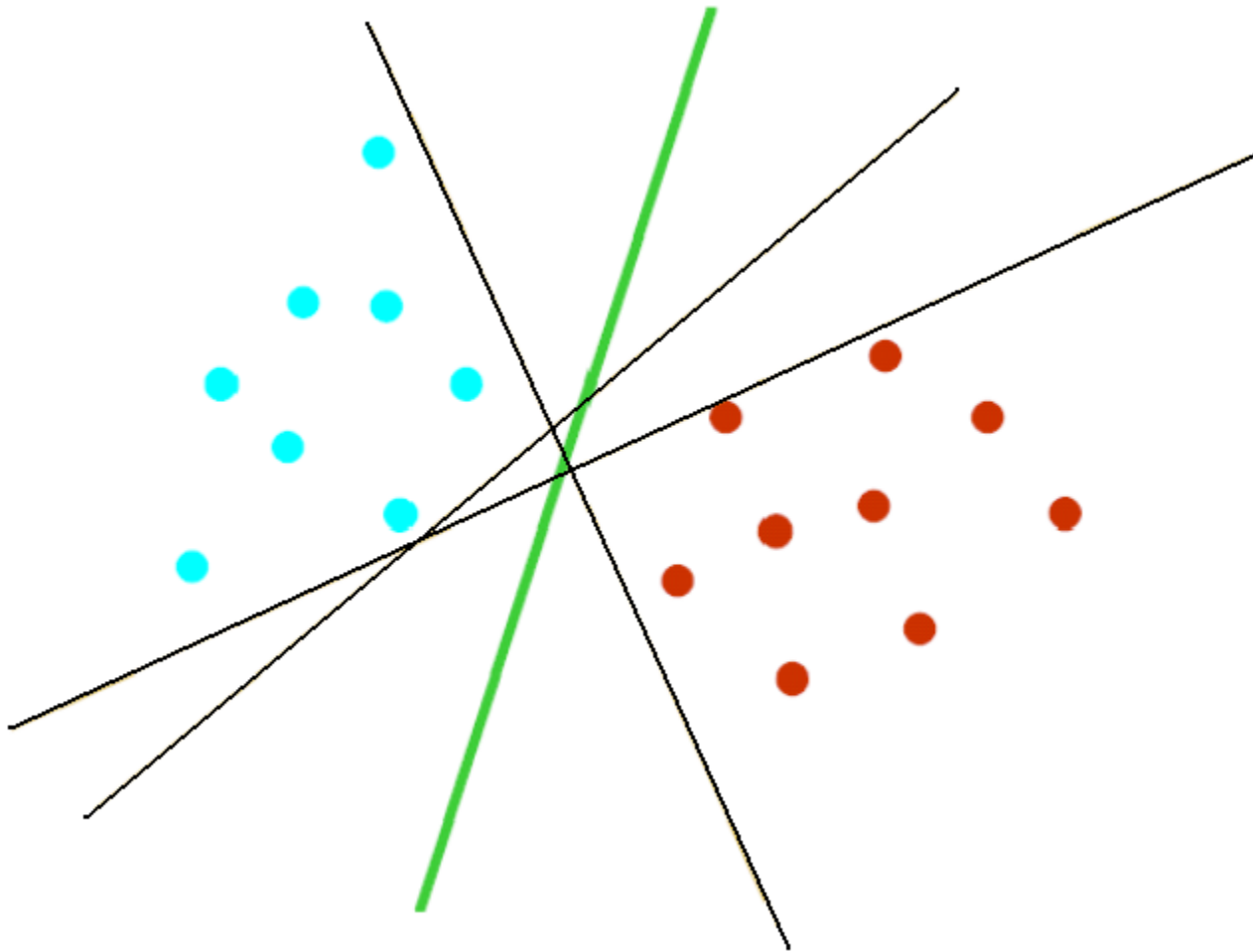


# Outline

---

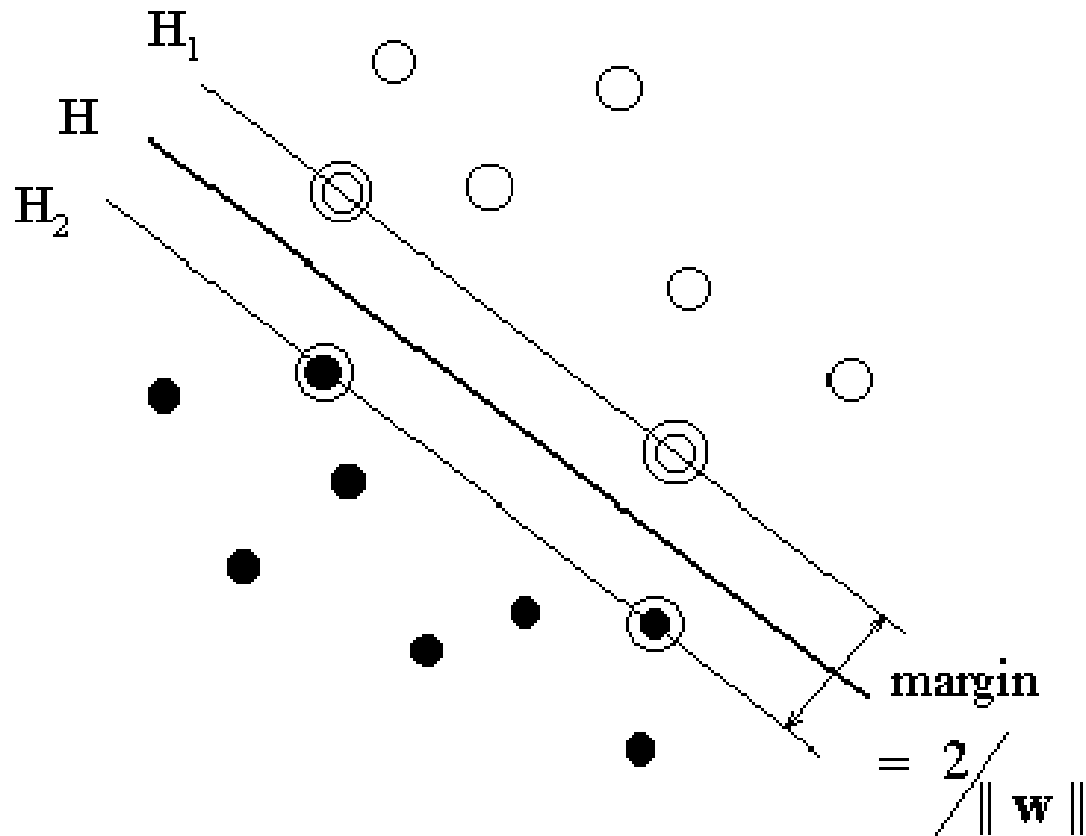
- Linearly separable patterns
- Linearly non-separable patterns
- Nonlinear case
- Some examples

# Linearly separable case



***Optimal Separating hyperplane***

# Optimal Hyperplane





# Linear classification

---

Training sample set  $\mathcal{T} = \{(\mathbf{x}_i, d_i)\}_{i=1}^N$

$$\begin{cases} d_i = +1, \text{positive patterns} \\ d_i = -1, \text{negative patterns} \end{cases}$$

Decision surface:  $w^T x + b = 0$

$$w^T x_i + b \geq 0 \quad \text{for } d_i = +1$$

$$w^T x_i + b < 0 \quad \text{for } d_i = -1$$

# Decision surface (line)

Decomposition of  $x$ :

$$x = x_p + r \frac{w_o}{\|w_o\|}$$

$$g(x) = w_o^T x + b_o = r \|w_o\|$$

$$\Rightarrow r = \frac{g(x)}{\|w_o\|}$$

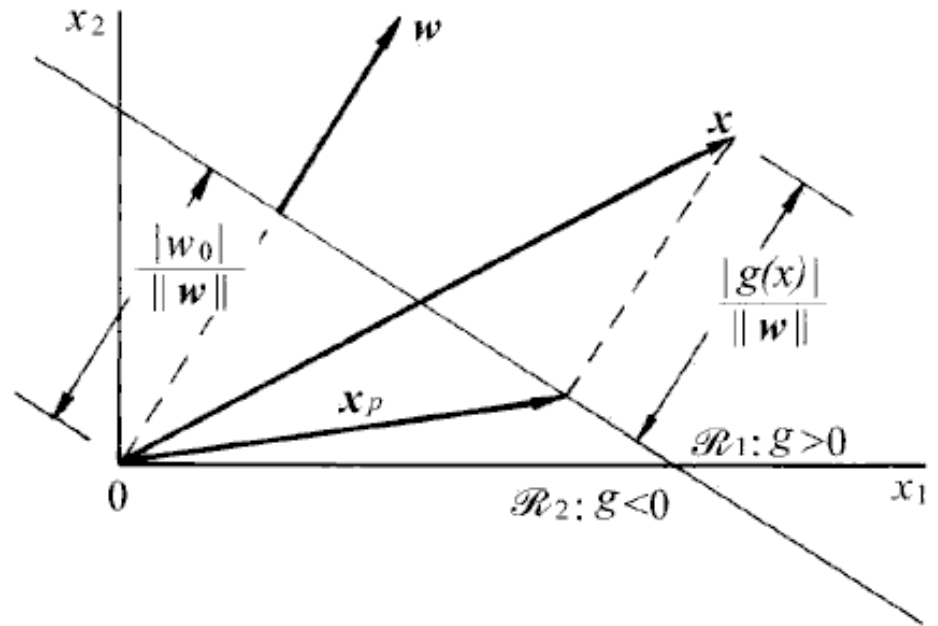


figure copied from book of Bian  $H: g=0$



# Linear classification

---

Training sample set  $\mathcal{T} = \{(\mathbf{x}_i, d_i)\}_{i=1}^N$

$$\begin{cases} d_i = +1, \text{positive patterns} \\ d_i = -1, \text{negative patterns} \end{cases}$$

Decision surface:  $w_o^T x + b_o = 0$

$$\begin{array}{ll} w_o^T x_i + b_o \geq 0 & \text{for } d_i = +1 \\ w_o^T x_i + b_o < 0 & \text{for } d_i = -1 \end{array} \quad \Rightarrow \quad \begin{array}{ll} w_o^T x_i + b_o \geq +1 & \text{for } d_i = +1 \\ w_o^T x_i + b_o \leq -1 & \text{for } d_i = -1 \end{array}$$



# Margin of separation

---

Consider a support vector  $x^{(s)}$

$$g(x^{(s)}) = w_o^T x^{(s)} + b_o = \pm 1 \quad \text{for } d^{(s)} = \pm 1$$

then

$$r = \frac{g(x^{(s)})}{\|w_o\|} = \begin{cases} \frac{1}{\|w_o\|} & \text{if } d^{(s)} = +1 \\ \frac{-1}{\|w_o\|} & \text{if } d^{(s)} = -1 \end{cases}$$

Margin of separation

$$\rho = 2r = \frac{2}{\|w_o\|}$$





# Optimization problem

---

Training sample set  $\mathcal{T} = \{(\mathbf{x}_i, d_i)\}_{i=1}^N$

$$\min f(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (\text{P})$$

$$\text{subject to } \begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq +1 & \text{for } d_i = +1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1 & \text{for } d_i = -1 \end{cases}$$

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, 2, \dots, N$$

$$g_i(\mathbf{w}) = d_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0 \quad \text{for } i = 1, 2, \dots, N$$

# Lagrange function

$\Phi(w, \alpha)$

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^N \alpha_i [d_i (w^T x_i + b) - 1]$$

$f(w)$

Lagrange multipliers

$g_i(w)$

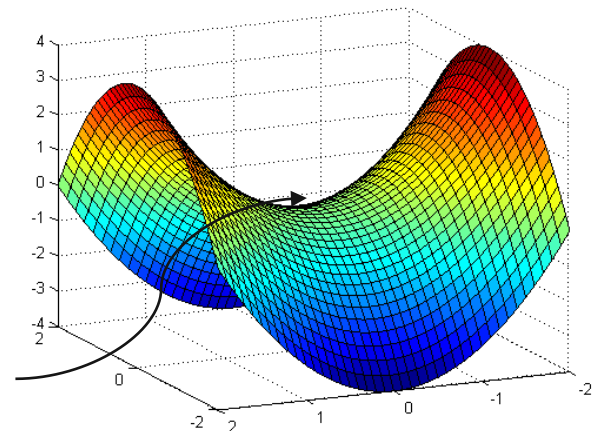
find saddle point of  $J(w, b, \alpha)$

what?

why?

exist?

saddle point





# What is a saddle point

---

Definition:

variables  $w \in D \subset R^n$        $\alpha \in E \subset R^m$

function  $\Phi: D \times E \rightarrow R$

saddle point  $(w', \alpha')$

$$\Rightarrow \Phi(w', \alpha) \leq \Phi(w', \alpha') \leq \Phi(w, \alpha') \quad \forall w \in D, \alpha \in E$$



# why we find a saddle point

Theorem: if  $(w', \alpha')$  is a saddle point of

$$\Phi(w, \alpha) = f(w) - \sum_{i=1}^N \alpha_i g_i(w)$$

then  $w'$  is a solution of (P)

Proof: 
$$f(w') - \sum_{i=1}^N \alpha_i g_i(w') \leq f(w') - \sum_{i=1}^N \alpha'_i g_i(w') \leq f(w) - \sum_{i=1}^N \alpha'_i g_i(w)$$

$$f(w') - \sum_{i=1}^N \alpha_i g_i(w') - f(w') + \sum_{i=1}^N \alpha'_i g_i(w') \leq 0 \Rightarrow \sum_{i=1}^N (\alpha'_i - \alpha_i) g_i(w') \leq 0$$

let  $\alpha_1 = \alpha'_1 + 1, \alpha_2 = \alpha'_2, \dots, \alpha_N = \alpha'_N \quad \longrightarrow \quad g_i(w') \geq 0$

$\longrightarrow \quad w'$  is a feasible solution of (P)

# why we find a saddle point (ctd.)

$$f(w') - \sum_{i=1}^N \alpha_i g_i(w') \leq f(w') - \sum_{i=1}^N \alpha'_i g_i(w') \leq f(w) - \sum_{i=1}^N \alpha'_i g_i(w)$$

$$\text{let } \alpha = 0 \quad \longrightarrow \quad \sum_{i=1}^N \alpha'_i g_i(w') \leq 0 \quad \longrightarrow \quad \sum_{i=1}^N \alpha'_i g_i(w') = 0$$

consider the second inequality  $g_i(w) \geq 0$  if  $w$  is feasible

$$\longrightarrow f(w') \leq f(w) - \sum_{i=1}^N \alpha'_i g_i(w)$$

$$\longrightarrow f(w') \leq f(w) \quad w' \text{ is optimal solution of (P)}$$



# Strong Duality

---

Strong Duality: the condition

$$\max_{\alpha} \min_w \Phi(w, \alpha) = \min_w \max_{\alpha} \Phi(w, \alpha)$$

holds if and only if there exists a pair  $(w', \alpha')$

satisfies the saddle-point condition for  $\Phi$

Proof: (omitted)

“Stephen G. Nash & Ariela Sofer Linear and Nonlinear Programming” pp468

$$\max_{\alpha} \min_w \Phi(w, \alpha) = \Phi(w', \alpha') = \min_w \max_{\alpha} \Phi(w, \alpha)$$



# Dual Problem

---

primal function  $L(w) = \max_{\alpha \geq 0} \Phi(w, \alpha) = \max_{\alpha \geq 0} [f(w) - \sum_{i=1}^N \alpha_i g_i(w)]$

primal problem  $\min_w L(w) \quad \longrightarrow \quad \min_w \max_{\alpha \geq 0} \Phi(w, \alpha)$

dual function  $Q(\alpha) = \min_w \Phi(w, \alpha)$

dual problem  $\max_{\alpha \geq 0} Q(\alpha) \quad \longrightarrow \quad \max_{\alpha \geq 0} \min_w \Phi(w, \alpha)$

we prefer to solve the dual problem!



# Solve the dual problem

---

$$\Phi(w, \alpha) = J(w, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i [d_i (\mathbf{w}^T \mathbf{x}_i + b) - 1]$$

dual function:  $Q(\alpha) = \min_{w, b} J(w, b, \alpha)$

$$\rightarrow \frac{\partial J(w, b, \alpha)}{\partial w} = 0 \quad \rightarrow w = \sum_{i=1}^N \alpha_i d_i x_i$$

$$\rightarrow \frac{\partial J(w, b, \alpha)}{\partial b} = 0 \quad \rightarrow \sum_{i=1}^N \alpha_i d_i = 0$$



# Solve the dual problem (ctd.)

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i [d_i (\mathbf{w}^T \mathbf{x}_i + b) - 1] \quad \sum_{i=1}^N \alpha_i d_i = 0$$

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

$$\mathbf{w}^T \mathbf{w} = \left( \sum_{i=1}^N \alpha_i d_i \mathbf{x}_i \right)^T \left( \sum_{i=1}^N \alpha_i d_i \mathbf{x}_i \right) = \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\sum_{i=1}^N \alpha_i d_i \mathbf{w}^T \mathbf{x}_i = \sum_{i=1}^N \alpha_i d_i \left( \sum_{j=1}^N \alpha_j d_j \mathbf{x}_j \right)^T \mathbf{x}_i = \sum_{i=1}^N \sum_{j=1}^N \alpha_i d_i \alpha_j d_j \mathbf{x}_i^T \mathbf{x}_j$$



# Dual Problem

---

We may now state the dual problem:


Given the training sample  $\{(x_i, d_i)\}_{i=1}^N$ , find the Lagrange multipliers  $\{\alpha_i\}_{i=1}^N$  that maximize the objective function

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j x_i^T x_j$$

subject to the constraints

$$(1) \sum_{i=1}^N \alpha_i d_i = 0$$

$$(2) \alpha_i \geq 0 \quad \text{for } i = 1, 2, \dots, N$$



$H(i, j)$




# Some discussions

---

1.  $Q(\alpha)$  depends only on the input patterns in the form of a set of dot products,  $\{x_i^T x_j\}_{(i,j)=1}^N$
2. support vectors determine the hyperplane

$$\sum_{i=1}^N \bar{\alpha}_i g_i(\bar{w}) = 0 \quad \longrightarrow \quad \bar{\alpha}_i g_i(\bar{w}) = 0, \forall i$$

$$\bar{\alpha}_i (d_i (\bar{w}^T x_i + \bar{b}) - 1) = 0, \forall i$$


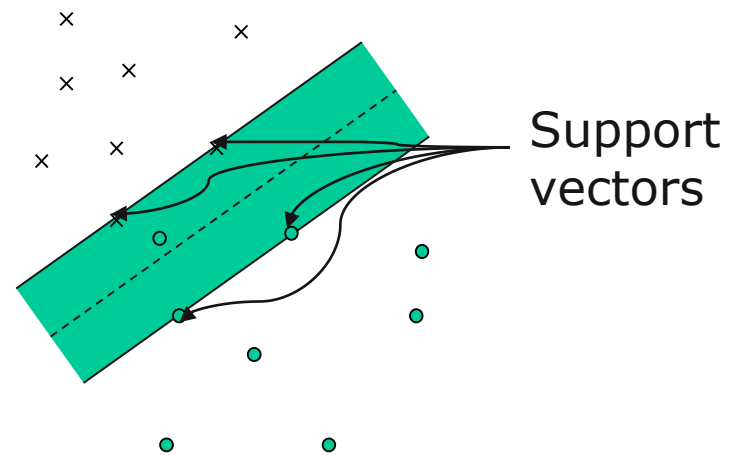
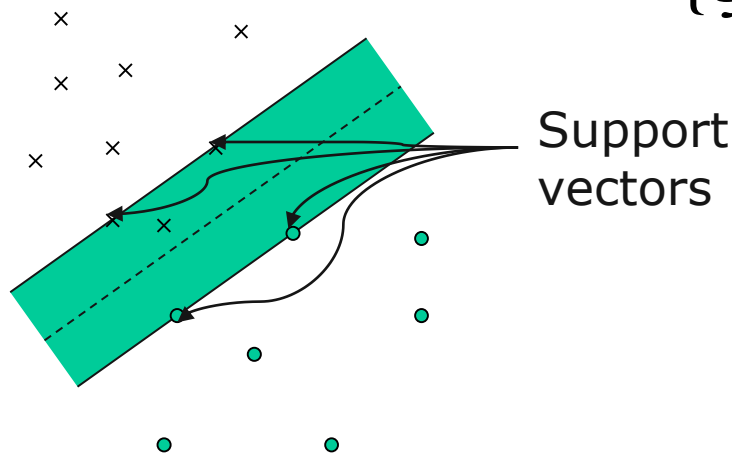
$$\bar{w} = \sum_{i=1}^N \bar{\alpha}_i d_i x_i \quad \longrightarrow \quad \bar{w} = \sum_{i=1}^{N_s} \bar{\alpha}_{i,s} d_{i,s} x_i^s$$

# Linearly non-separable patterns

hard margin  $d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$  for  $i = 1, 2, \dots, N$

introduce slack variables  $d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$  for  $i = 1, 2, \dots, N$

$\{\xi_i\}_{i=1}^N$



slack variables  $\begin{cases} 0 \leq \xi_i \leq 1, & \text{violate, but on the right side of decision surface} \\ \xi_i > 1, & \text{misclassification} \end{cases}$



# Optimization Problem

---

$$\min_{w, \xi} \Phi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i$$

subject to  $d_i(w^T x_i + b) \geq 1 - \xi_i$  for  $i = 1, 2, \dots, N$

$\xi_i \geq 0$ , for all  $i$

$\sum_{i=1}^N \xi_i$  : upper bound of misclassification error

$C$  : tradeoff between complexity of the machine  
and the number of nonseparable points



# Dual Problem

---

Given the training sample  $\{(x_i, d_i)\}_{i=1}^N$ , find the Lagrange multipliers  $\{\alpha_i\}_{i=1}^N$  that maximize the objective function

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j x_i^T x_j$$

subject to the constraints

$$(1) \sum_{i=1}^N \alpha_i d_i = 0$$

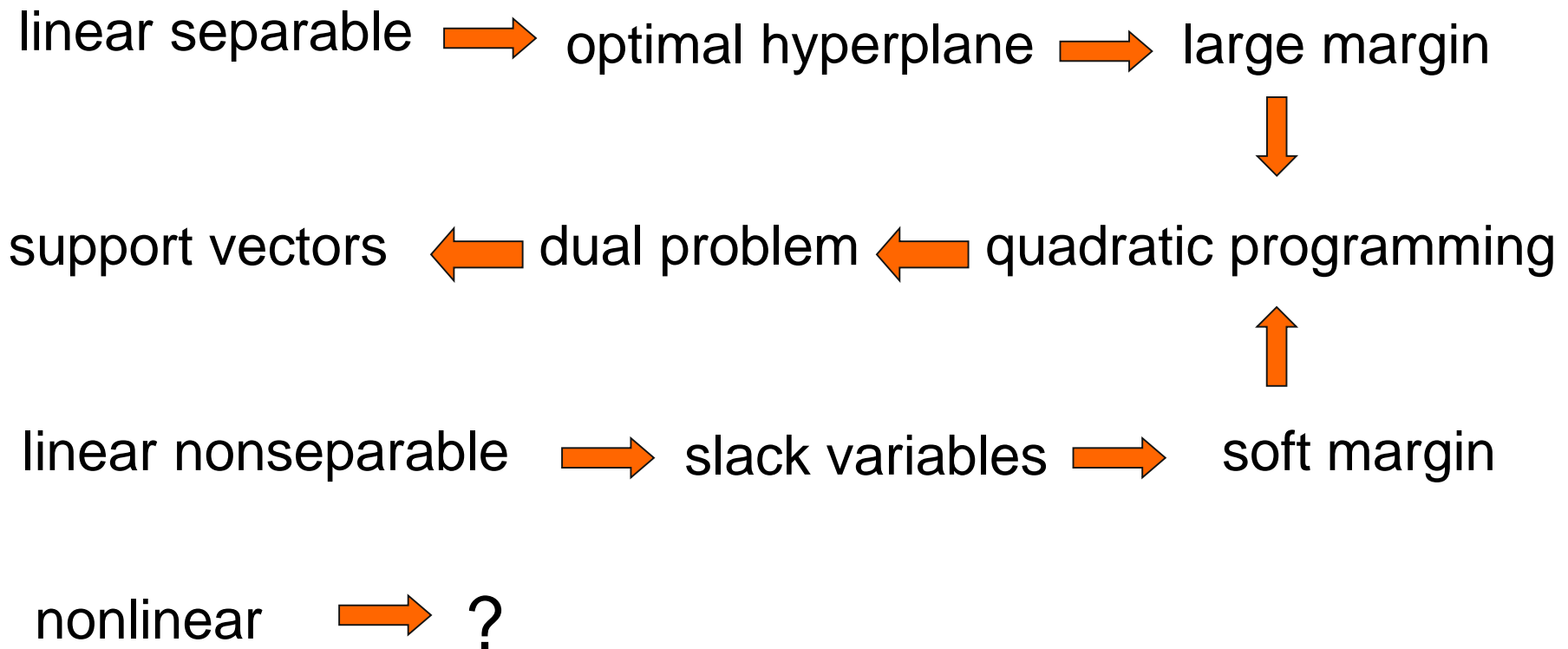
$$(2) 0 \leq \alpha_i \leq C \quad \text{for } i = 1, 2, \dots, N$$

where  $C$  is a user-specified positive parameter

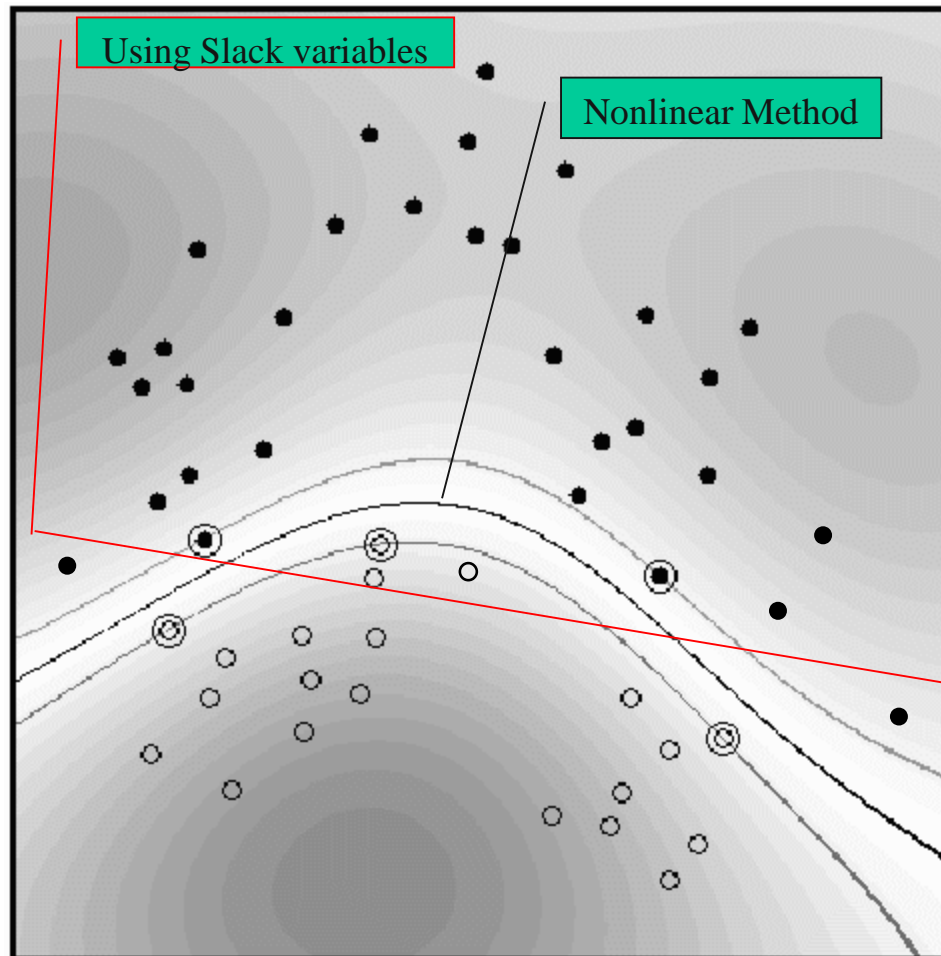


# Summary

---



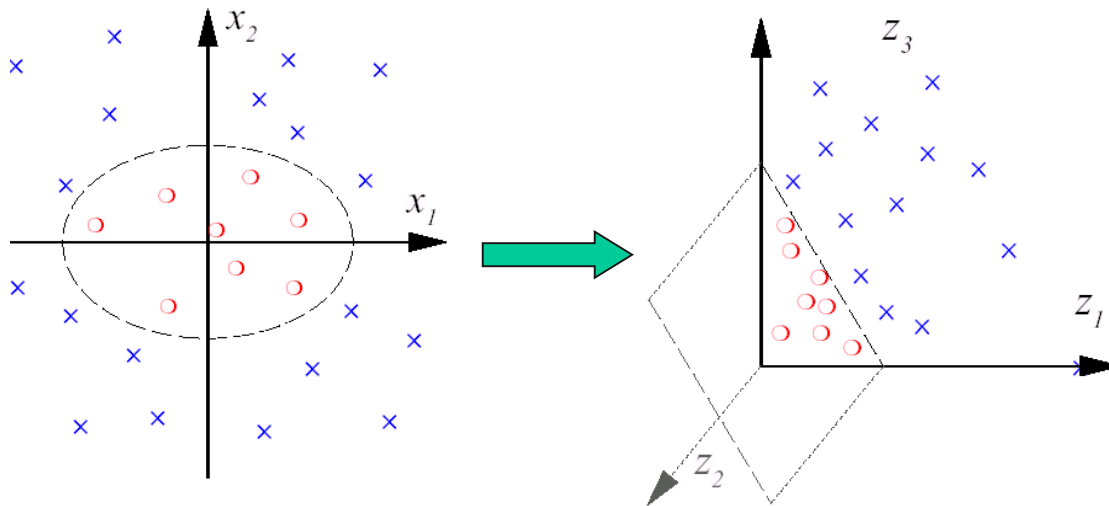
# The Non-separable Case





# From Linear to Nonlinear

- Increase Dimension and Use Linear Method in High Dimensional Space——Generalized Linearity
- An Example



$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$$



# From Linear to Nonlinear (Cont.)

---

- Why ?
- Input Space is not proper Feature Space



# Computation

---

- For  $d_L$ -dimensional input space and polynomial degree of  $p$ , there exist  $C_{d_L+p-1}^p$  different polynomials (Corresponding to at least dimensional feature space)
- So,  $16 \times 16$  input image and polynomial degree of 4 yield a dimensionality of 183,181,376!!!



# Computation

- Maximize the Wolfe dual problem (training):

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j d_i d_j \underbrace{x_i \bullet x_j}_{\Phi(x_i) \bullet \Phi(x_j)}$$

- The Classifier (testing)

$$f(x) = \text{sgn}\left(\sum_{i=0}^{N_s} \alpha_i d_i \underbrace{s_i \bullet x}_{\Phi(s_i) \bullet \Phi(x)} + b\right)$$

- Only dot product is needed!

- 
- 
- If there were a “kernel function”:

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

$$\Phi : L \mapsto H$$

- We would only need to use  $K$  in the training algorithm, and would never need to explicitly even know what  $\Phi$  is!



# An Example


---

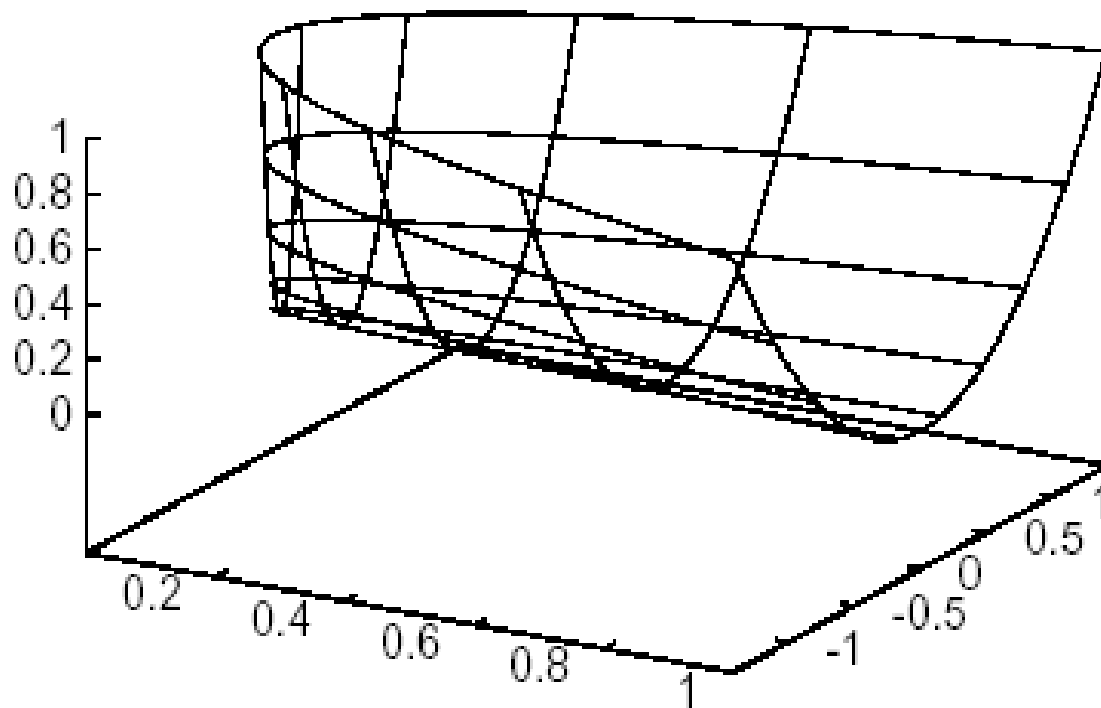
- Suppose  $L = R^2$  and  $K(x_i, x_j) = (x_i \cdot x_j)^2$ , easy to find a space  $H$  and a mapping  $\Phi$  from  $R^2$  to  $H$ , such that


$$K(x_i, x_j) = (x_i \cdot x_j)^2 = \Phi(x_i) \cdot \Phi(x_j)$$

- We choose  $H = R^3$  and

$$\Phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

- 
- For data in  $L$  defined on the square  $[-1,1] \times [-1,1]$ , the entire image of  $\Phi$  is just a surface whose intrinsic dimension is just that of  $L$



- 
- Neither the mapping  $\Phi$  nor the space  $H$  are unique for a given kernel.
  - For  $L = R^2$  and  $K(x_i, x_j) = (x_i \cdot x_j)^2$  we could equally well chose

$$H = R^3$$

- $$\Phi(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} (x_1^2 - x_2^2) \\ 2x_1x_2 \\ (x_1^2 + x_2^2) \end{pmatrix}$$

or

$$H = R^4$$

$$\Phi(x) = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{pmatrix}$$





# Mercer's Condition


---

- There exists a mapping  $\Phi$  and an expansion

$$K(x, y) = \sum_i \Phi(x)_i \Phi(y)_i$$

- if and only if, for any  $g(x)$  such that  $\int g(x)^2 dx$  ( $L_2$  norm) is finite, then

$$\int K(x, y) g(x) g(y) dx dy \geq 0$$

- 
- 
- What happens if one use a kernel which does not satisfy Mercer's condition?
  - the Hessian(  $H_{ij} = y_i y_j K(x_i, x_j)$  ) may be indefinite



- The dual objective function is not a convex function, thus can become arbitrarily large



- The quadratic programming problem have no solution!



# Kernel Examples

---

Polynomial kernel:

$$K(x, y) = (x \cdot y + 1)^p$$

RBF kernel:

$$K(x, y) = e^{-\|x-y\|^2/2\sigma^2}$$

Sigmoid kernel:

$$K(x, y) = \tanh(\kappa x \cdot y - \delta)$$



# Polynomial Kernel

---

- $K(x, y) = (x \cdot y)^p$
- Always satisfy Mercer's Condition
- VC dimension for homogeneous polynomial kernels of degree  $p$  is: ( Reference 3)

$$C_{d_L+p-1}^p + 1$$

# Radial Basis Function Kernel

- $K(x, y) = e^{-\|x-y\|^2 / 2\sigma^2}$  depends on  $\|x - y\|$
- Always satisfy Mercer's Condition
- VC dimension: infinite!!


$$f(s_j) = \text{sgn}\left(\sum_{i=1}^{N_s} \alpha_i d_i e^{-\|s_i - s_j\|^2 / 2\sigma^2} + b\right)$$



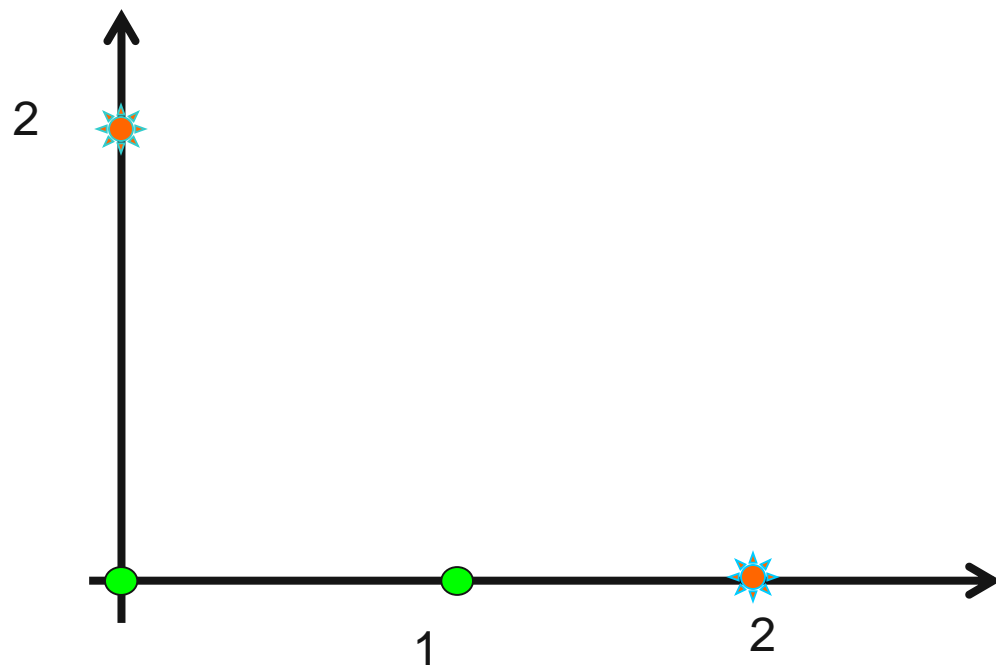
# Sigmoid Kernel

---

- $K(x, y) = \tanh(\kappa x \cdot y - \delta)$
- two-layer perceptron:  
first layer:  $N_s$  sets of weights, each consisting of  $d_L$  weights  
second layer:  $N_s$  weights( the  $\alpha_i$  )
- Only satisfy Mercer's condition for certain values of the  $K$  and  $\mathcal{S}$  (and of the data  $\|x\|^2$ )

- 
- 
- ❖ With  $K(x,y)$ , we could compute dot product in  $H$  without doing  $\Phi$  explicitly.
  - ❖ While one can equally well turn things around and start with  $\Phi$  .

# 简单例子

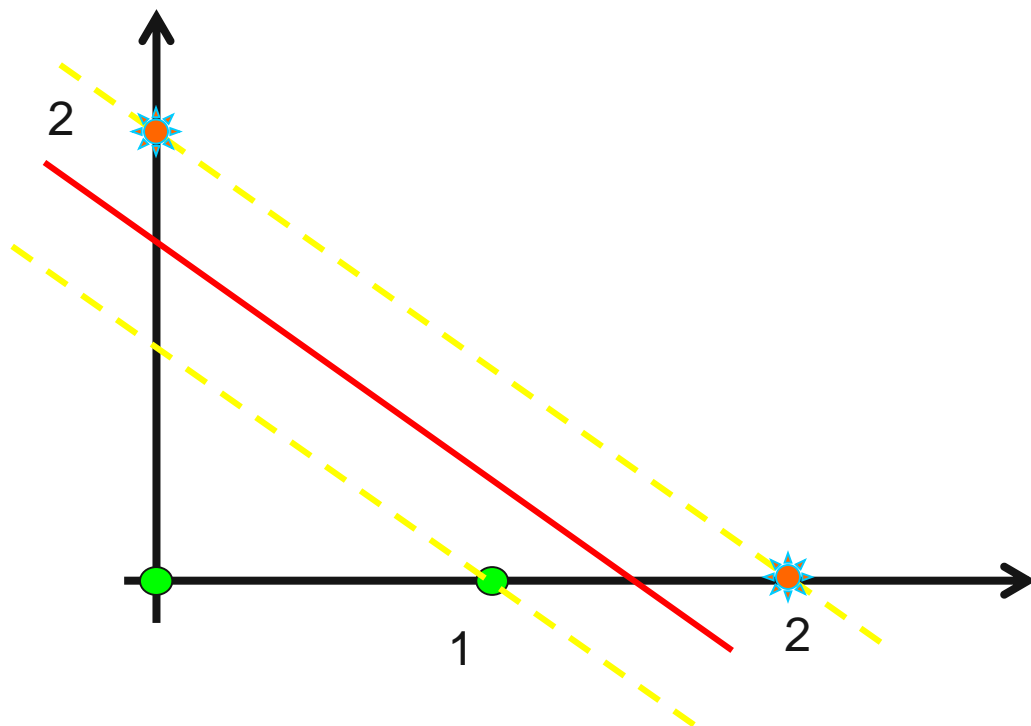


+1类 =  $x_1$   $x_2$

-1类 =  $x_3$   $x_4$



# 解答 图示

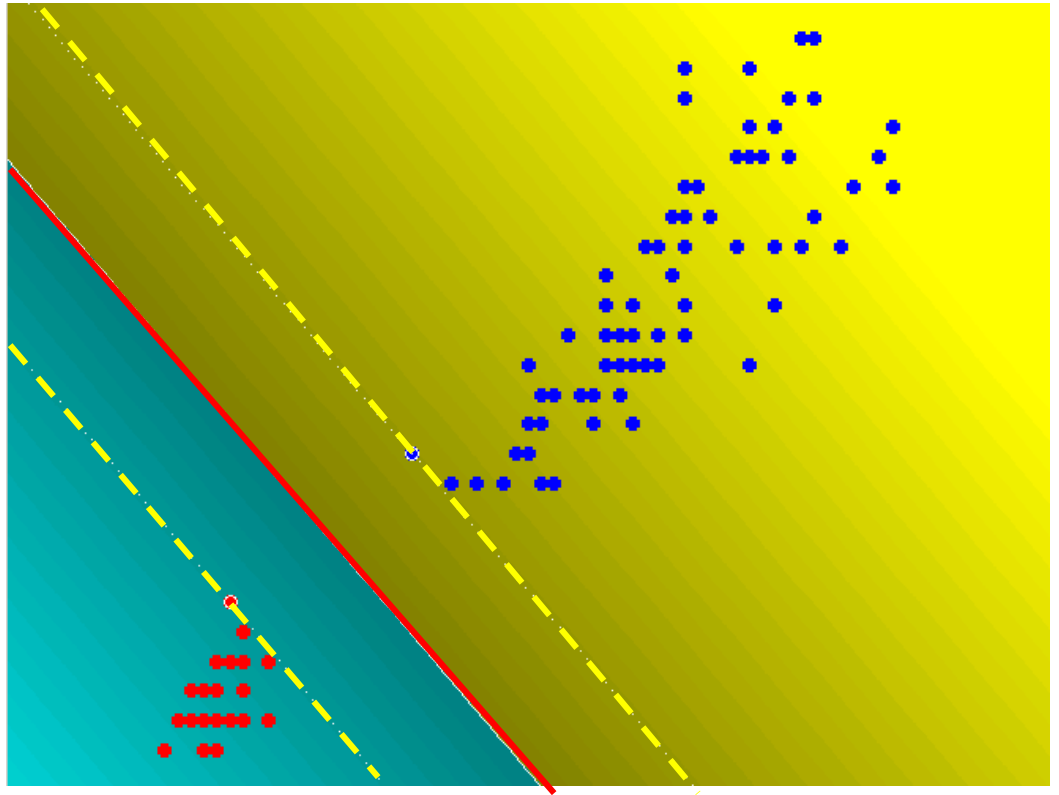


# 线性可分

样本数=120

核=一次多项式

支持向量/总样本数的  
=1.7%



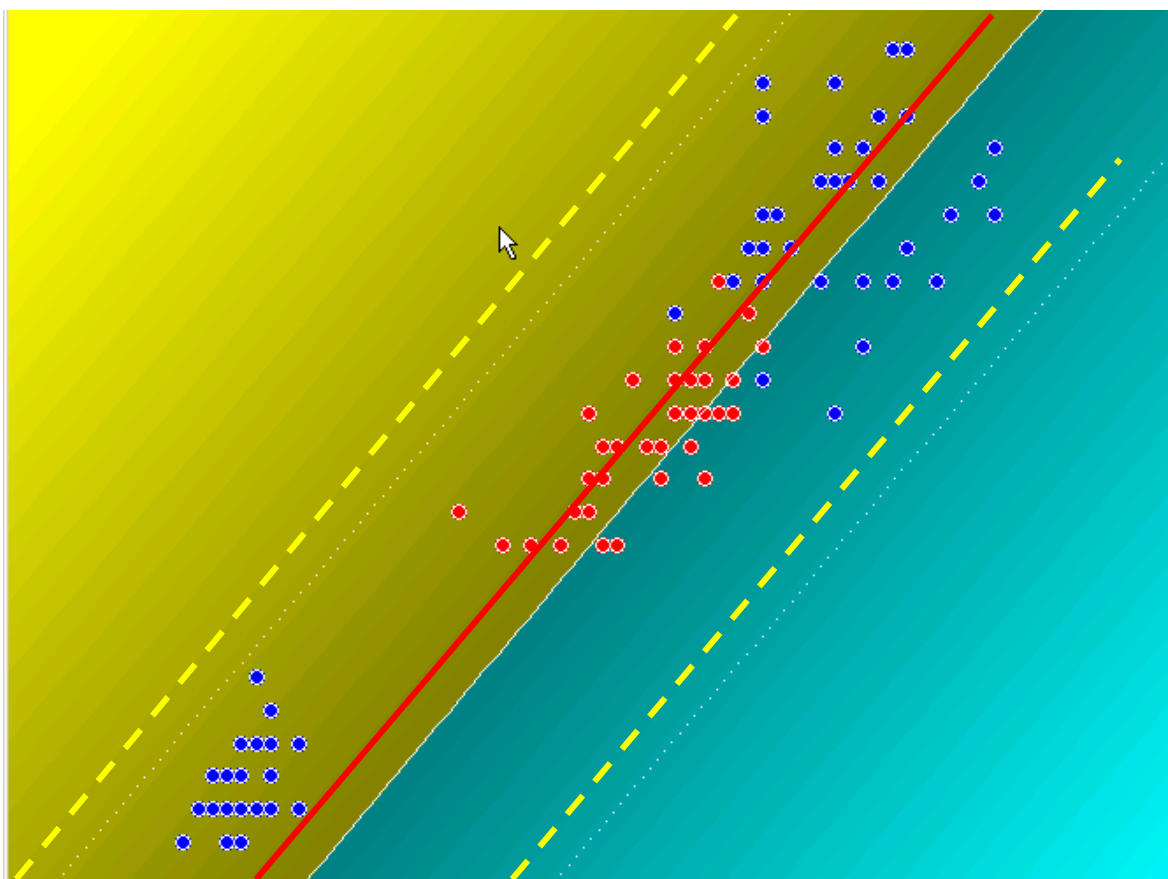
# 线性不可分

样本数=120

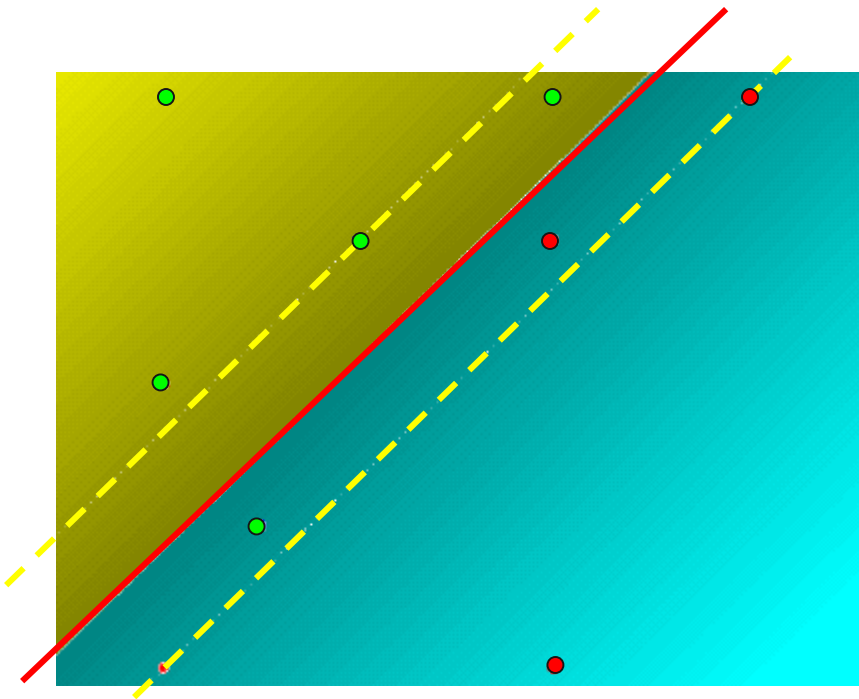
核=一次多项式

支持向量/总样本数  
=100%

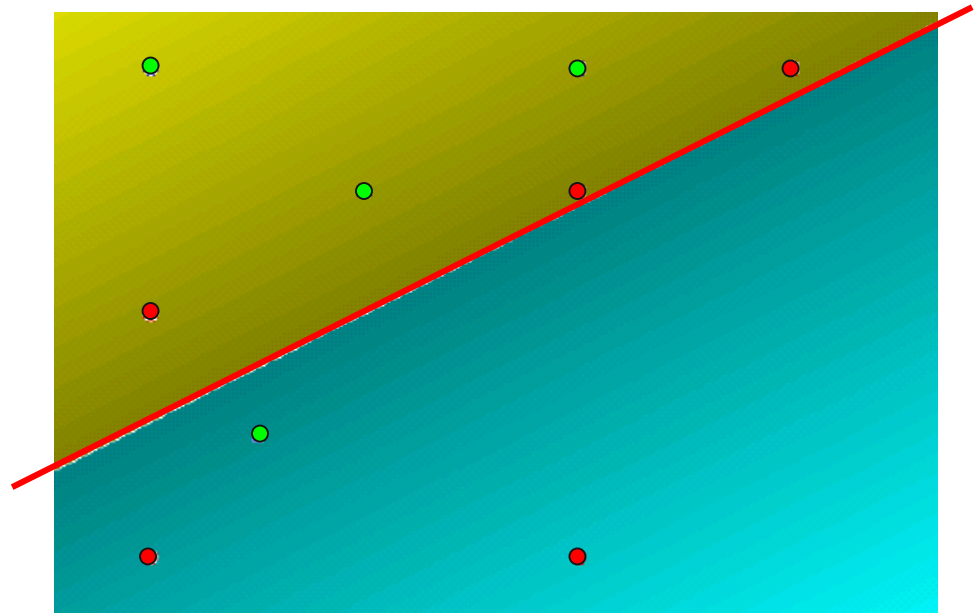
@并没有很好分开



# 松弛因子的影响

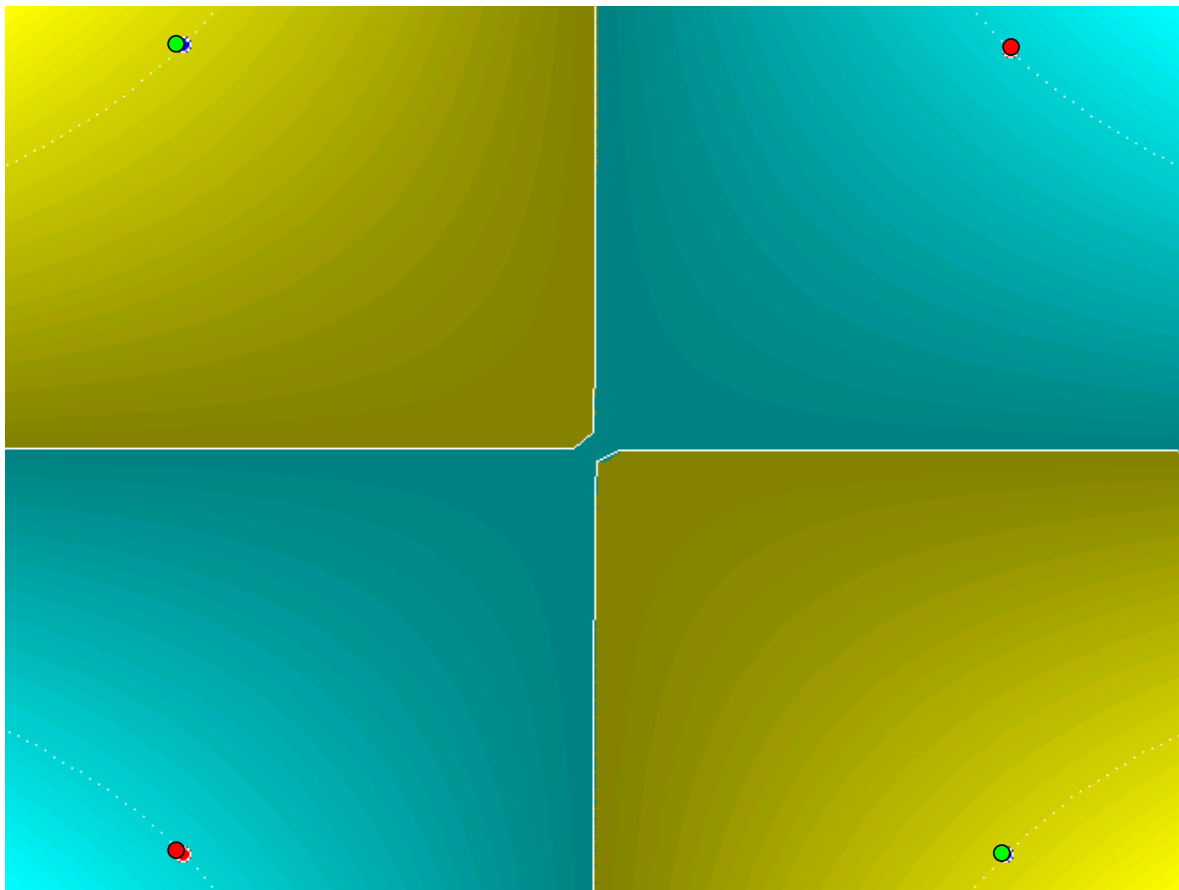


$C=10^5$



$C=10^{-8}$

# XOR问题分类面



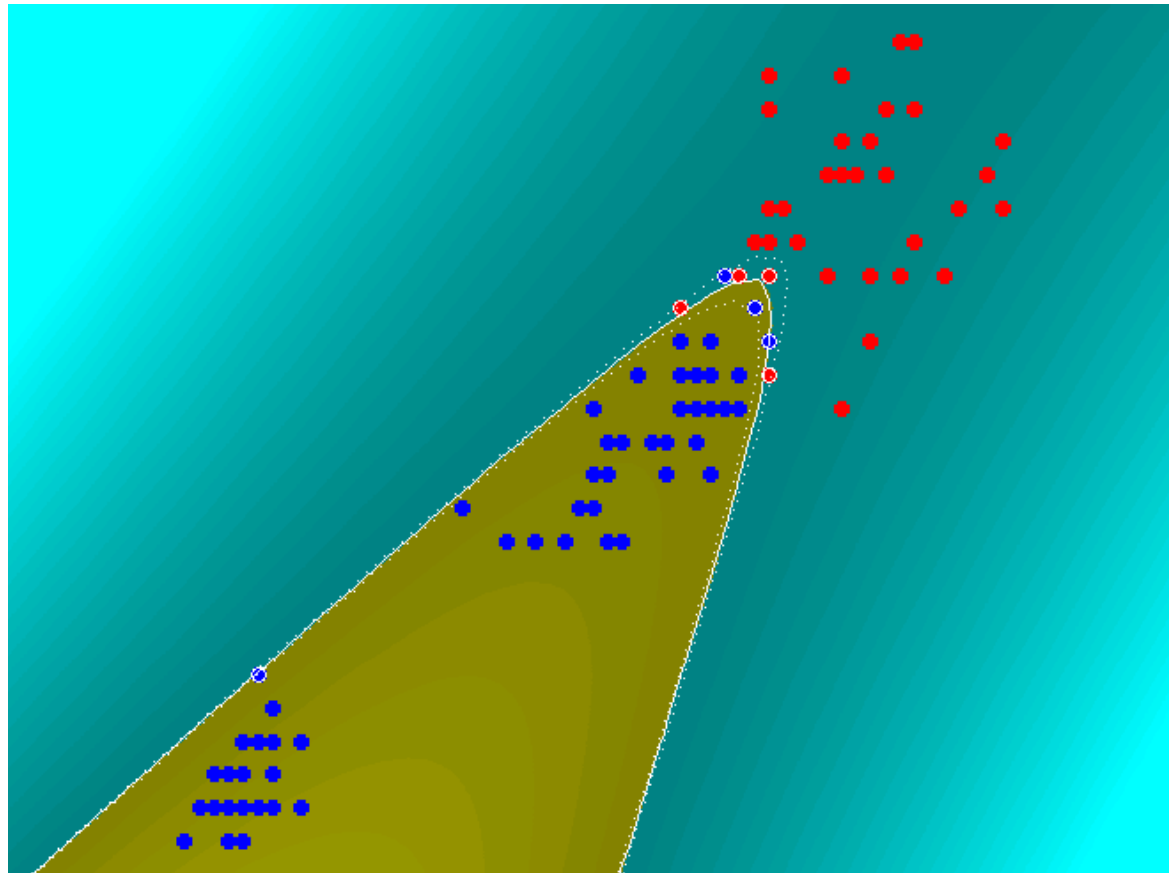
# 非线性问题的例子

样本数=120

核=二次多项式

支持向量/总样本数的  
=8.3%

@很好分开



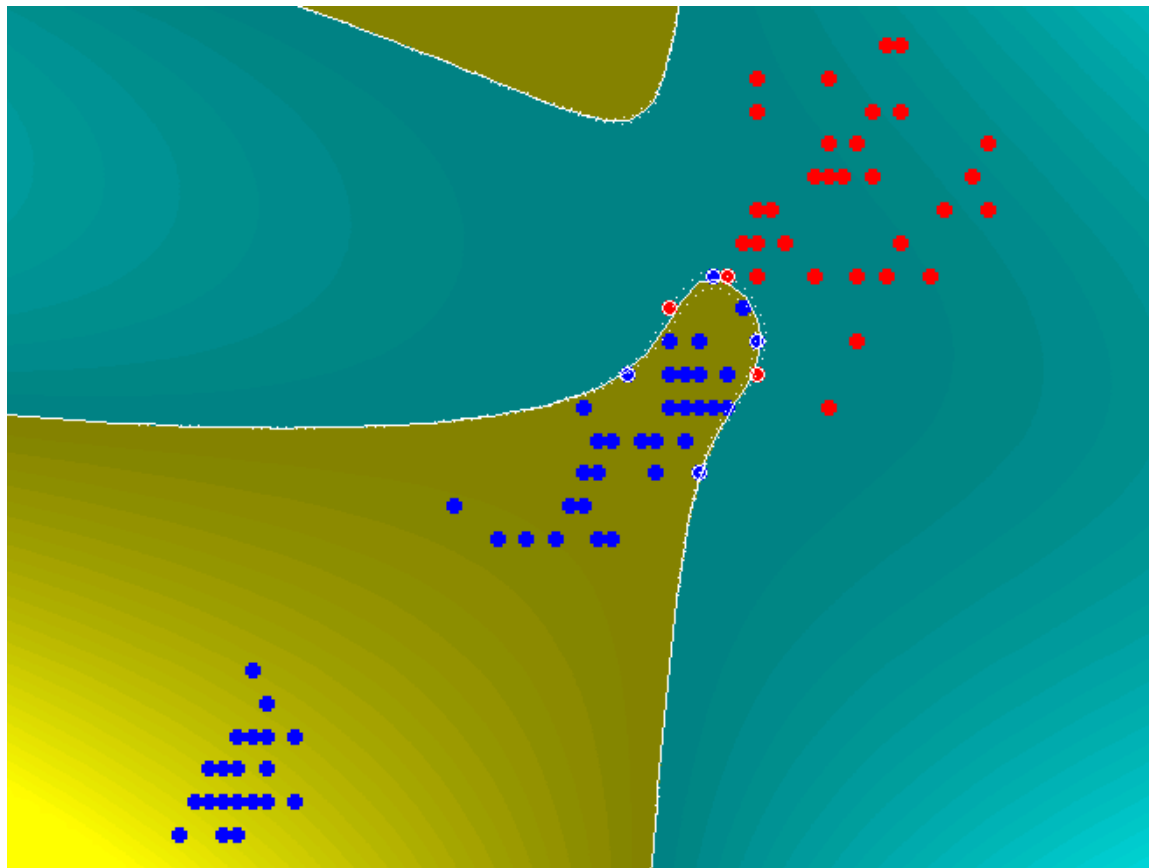
# 采用三次多项式

样本数=120

核=三次多项式

支持向量/总样本数的  
=7.5%

@很好分开但是分类  
面不同



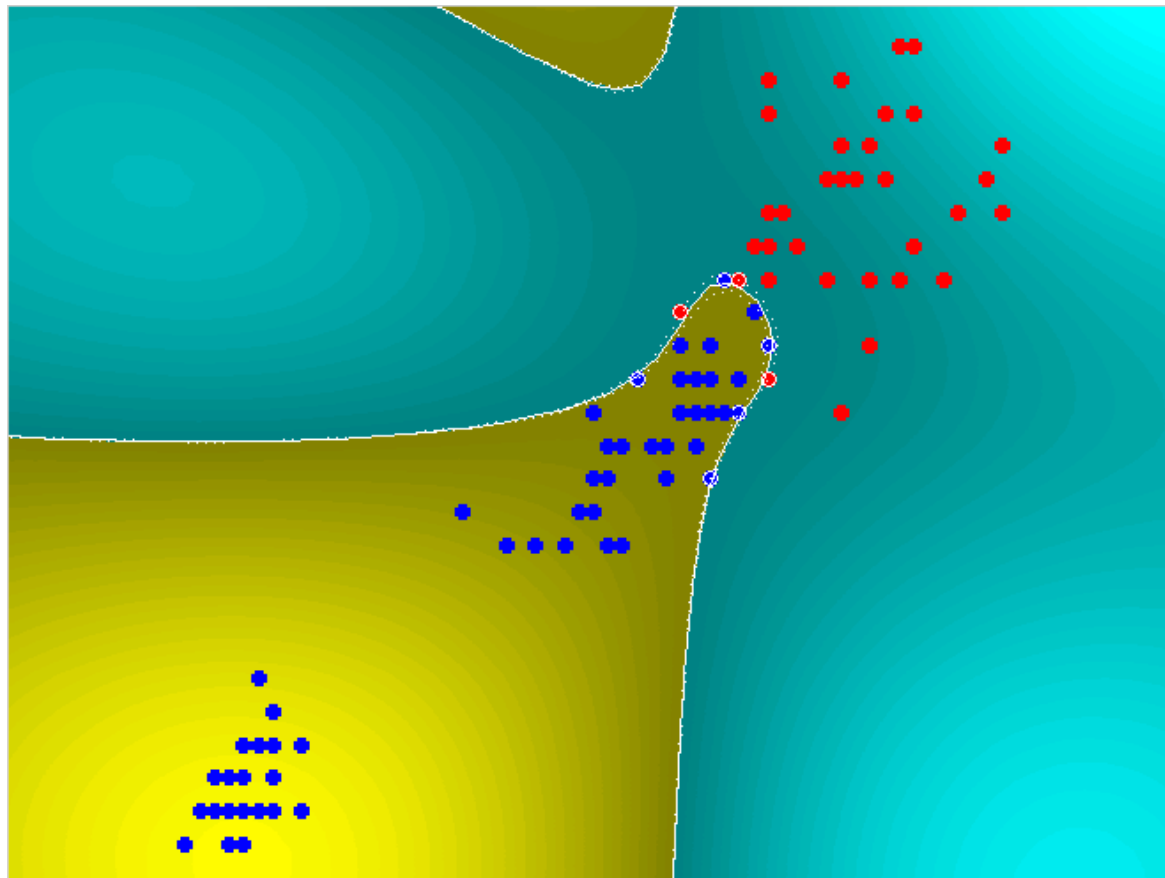
# 高斯核 $\text{sigm}=1$

样本数=120

核=高斯核

支持向量/总样本数的  
=8.3%

@高斯核相当于无限  
阶的多项式



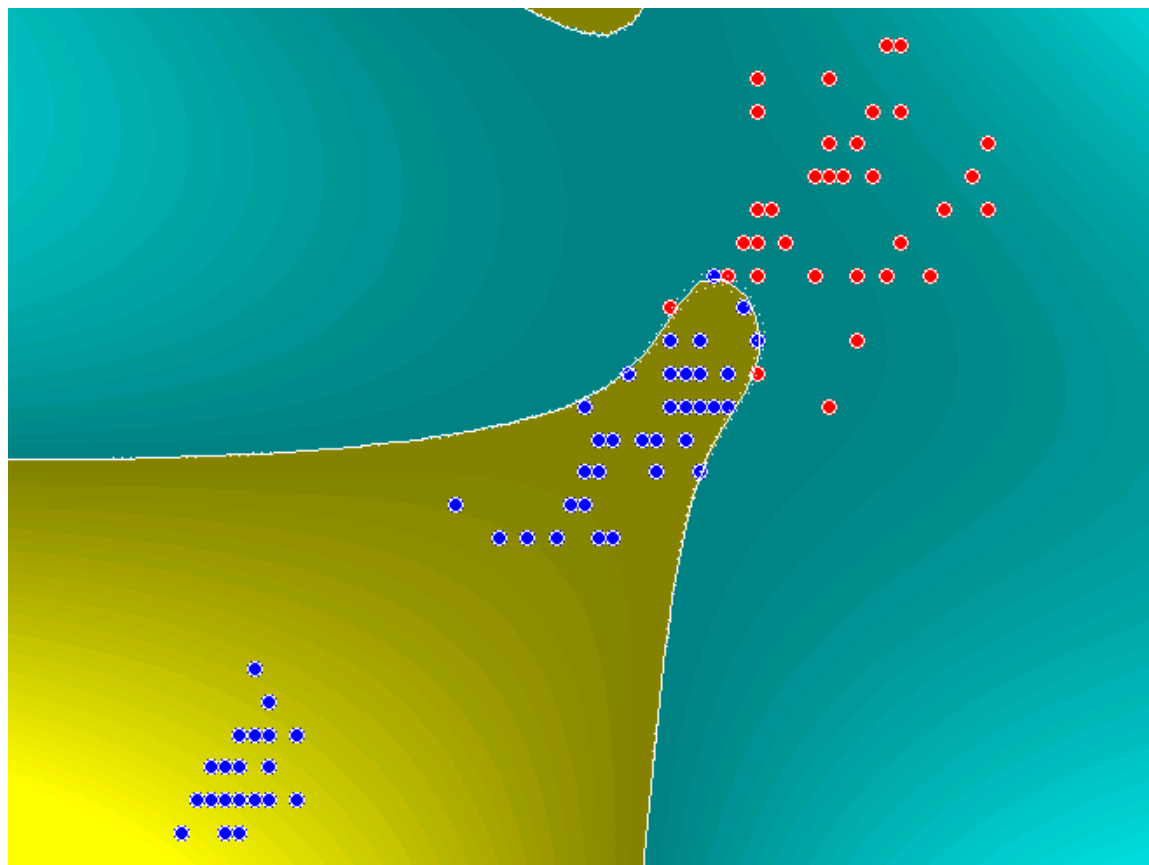


# 高斯核 $\sigma=2$

样本数=120

核=高斯核

@sigma的选取影响很大



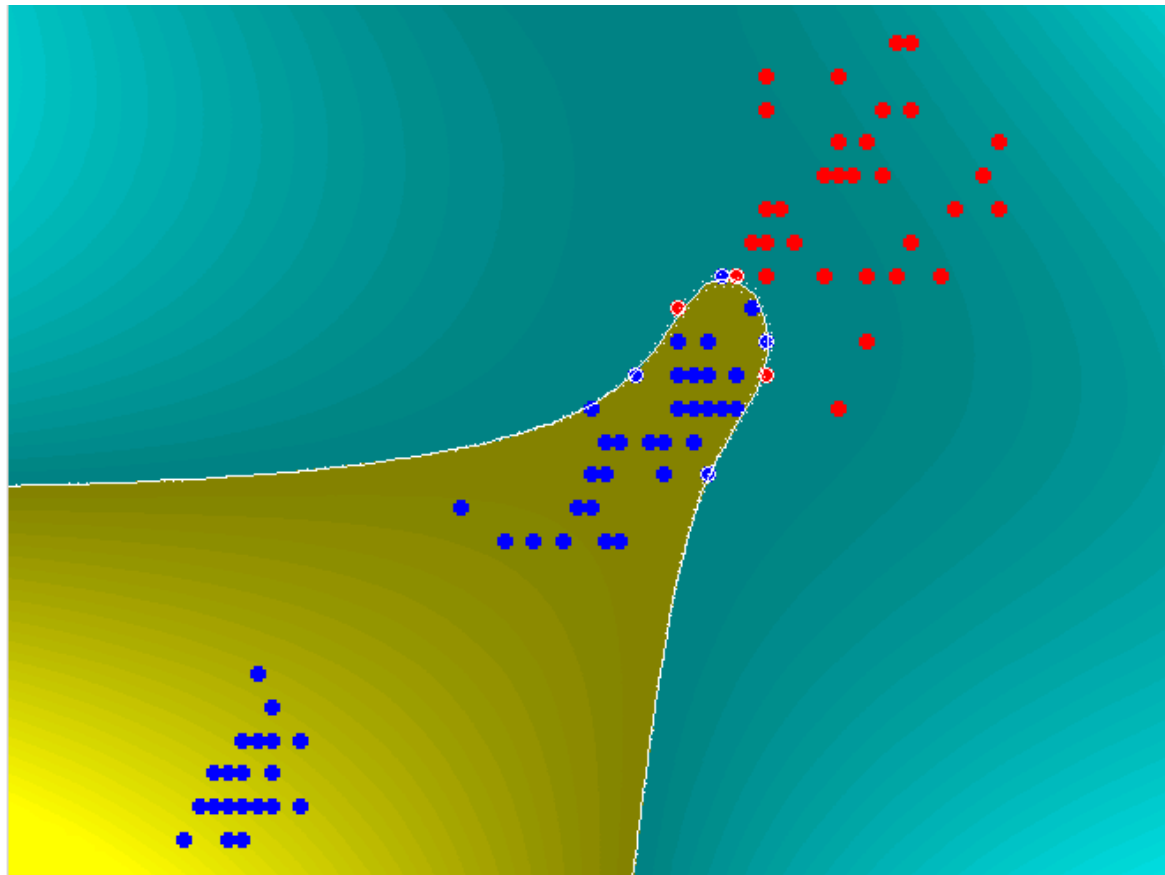
# 高斯核 $\sigma=3$

样本数=120

核=高斯内核

支持向量/总样本数的  
=7.5%

@高斯内核相当于无  
限阶的多项式

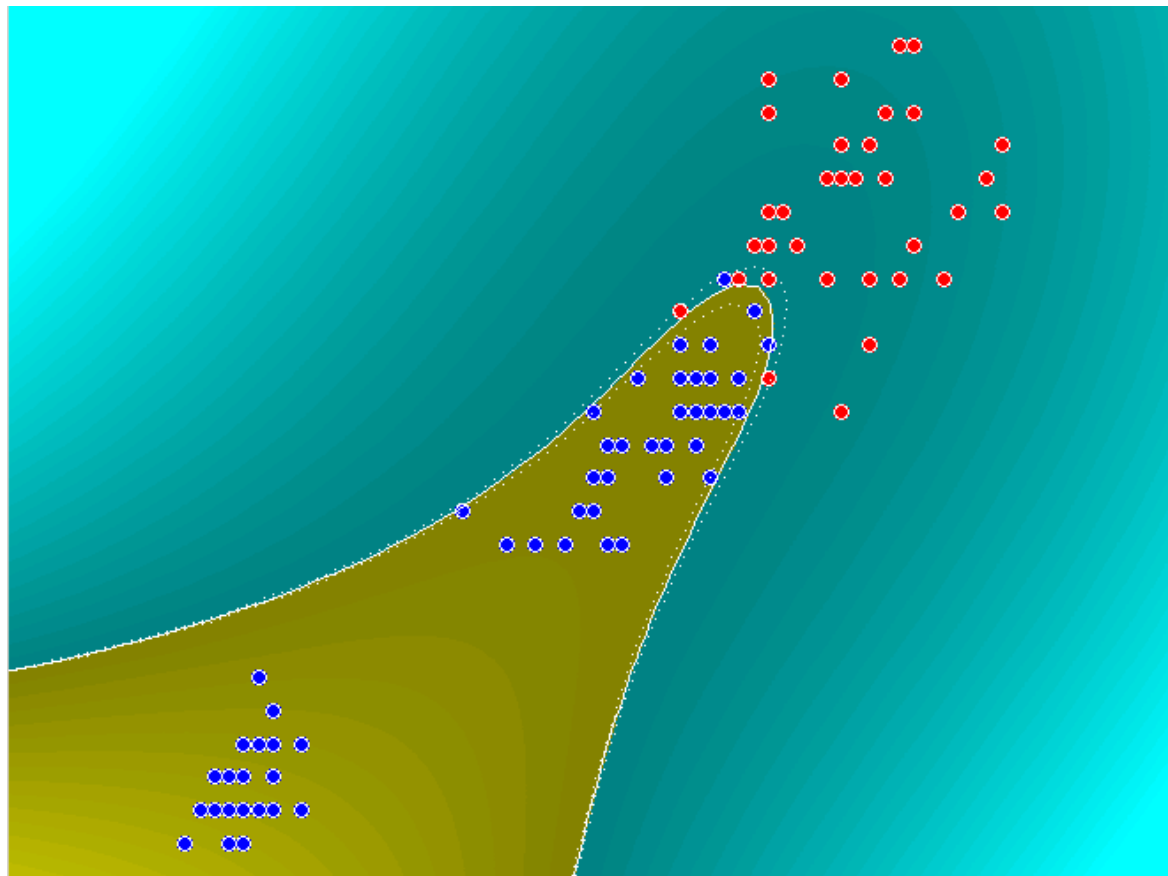


# 高斯核 $\sigma=9$

样本数=120

核=高斯核

@ $\sigma$ 增加，运算量  
增加





# 核的选择是很重要的

---

核的选择对结果有很大的影响

但是核的选择却没有统一的方法，大多要靠经验

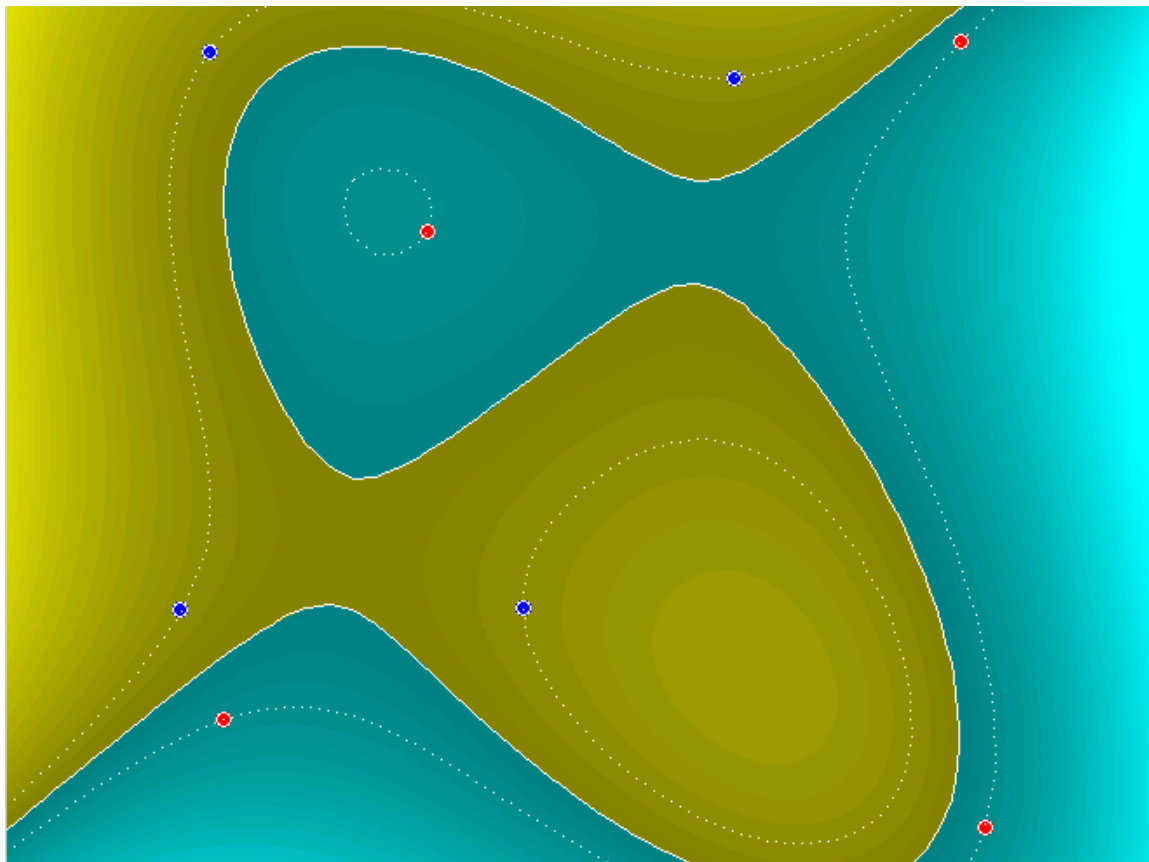
# 相互包含的例子

样本数=8

核=三次多项式

支持向量/总样本数的  
=100%

@小样本效果好





# SVM and Beyond

---

$$\min_{w, \xi} \Phi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i$$

subject to  $d_i(w^T x_i + b) \geq 1 - \xi_i$  for  $i = 1, 2, \dots, N$

$\xi_i \geq 0$ , for all  $i$

$\sum_{i=1}^N \xi_i$  : upper bound of misclassification error

$C$  : tradeoff between complexity of the machine  
and the number of nonseparable points



# Regularization

---

- SVM with general loss function

$$\min_w f(w) = \frac{1}{2} w^T w + C \sum_{i=1}^N L(w, x_i, y_i)$$

# Different Loss Functions (Bishop's book)

- SVM with general loss function

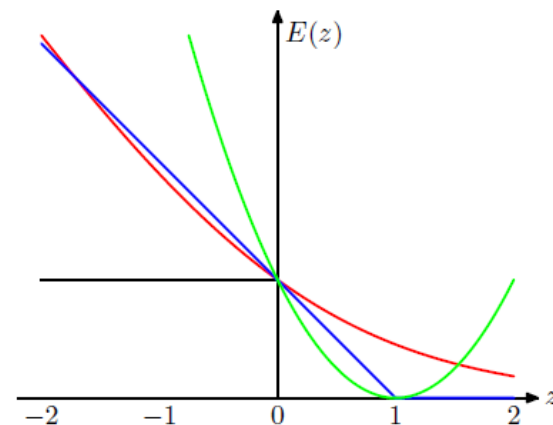
$$\min_w f(w) = \frac{1}{2} w^T w + C \sum_{i=1}^N L(w, x_i, y_i)$$

- SVM(Hinge Loss)

$$\min_w f(w) = \frac{1}{2} w^T w + C \sum_{i=1}^N \max(0, \xi_i)$$

- SVM(Differentiable Loss function)

$$\min_w f(w) = \frac{1}{2} w^T w + C \sum_{i=1}^N \max(0, \xi_i)^2$$







# L1-SVM

---

- L1-SVM  $\min_w f(w) = \|w\|_1 + C \sum_{i=1}^N L(w, x_i, y_i)$
- Sparsity

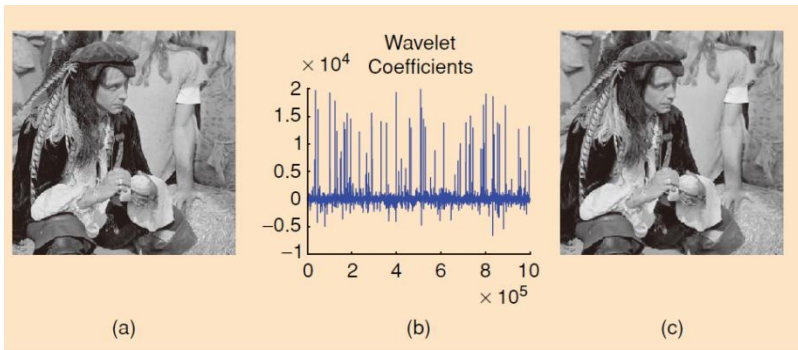
# 应用问题

## 文本分析

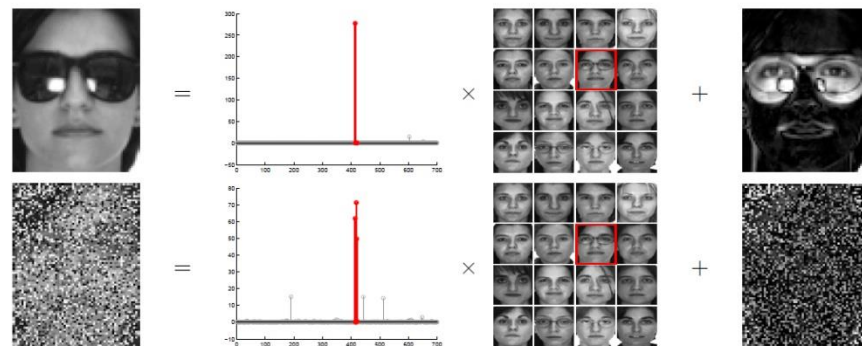
account	stock	politics	day	health	movie	video	market	farm	car
pay	trading	music	index	image	bank	imports	exports	dollars	surplus
club	game	bar	star	football	brain	sensor	study	effort	town
sell	people	oil	good	walk	city	billion	value	China	normal
top	money	card	job	start	run	beat	dance	wine	goal
arsenal	tax	large	salary	asset	China	federal	crisis	beer	small
price	house	Obama	wall	hubel	match	cell	state	card	bad
rate	school	manage	season	system	nose	league	play	union	climate
capital	people	results	credit	plan	time	USA	street	down	year
Bush	team	free	invest	make	win	set	news	NBA	wealth

account	stock	politics	day	health	movie	video	market	farm	car
pay	trading	music	index	image	bank	imports	exports	dollars	surplus
club	game	bar	star	football	brain	sensor	study	effort	town
sell	people	oil	good	walk	city	billion	value	China	normal
top	money	card	job	start	run	beat	dance	wine	goal
arsenal	tax	large	salary	asset	China	federal	crisis	beer	small
price	house	Obama	wall	hubel	match	cell	state	card	bad
rate	school	manage	season	system	nose	league	play	union	climate
capital	people	results	credit	plan	time	USA	street	down	year
Bush	team	free	invest	make	win	set	news	NBA	wealth

## 信号处理

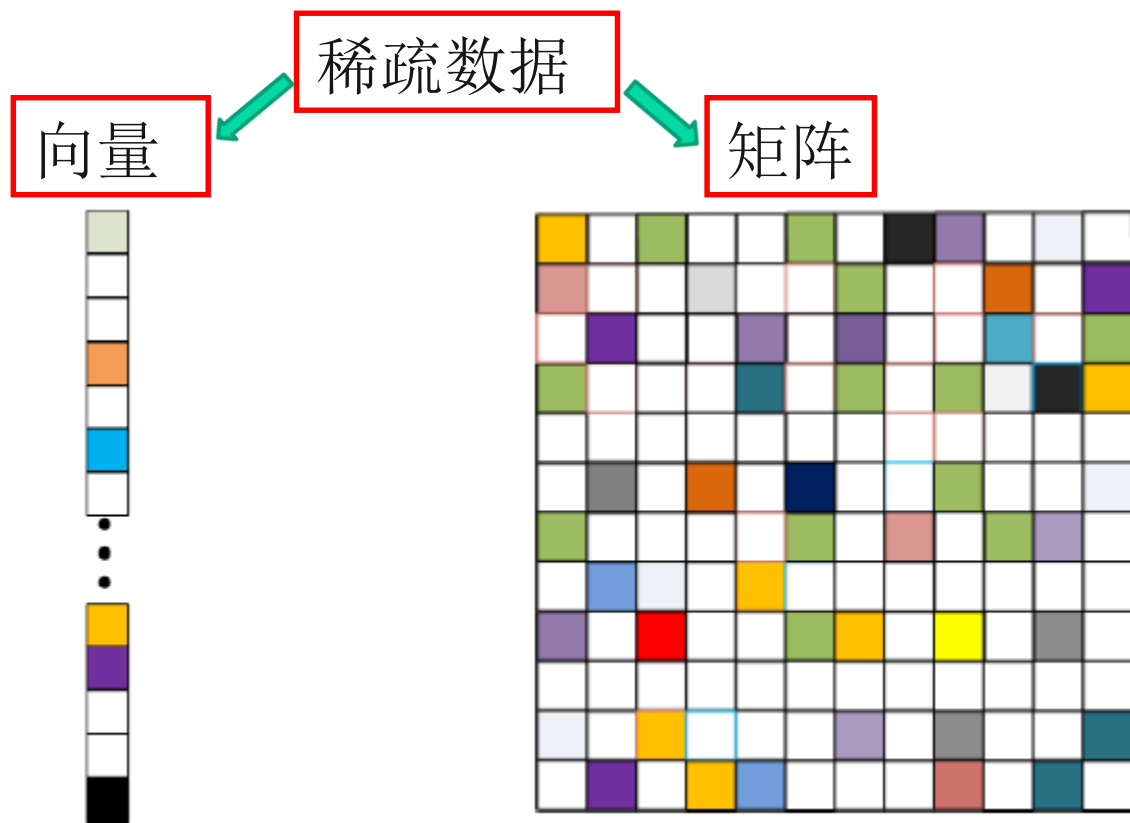


## 人脸识别



# 稀疏学习

□ 稀疏学习：带有稀疏结构的机器学习问题



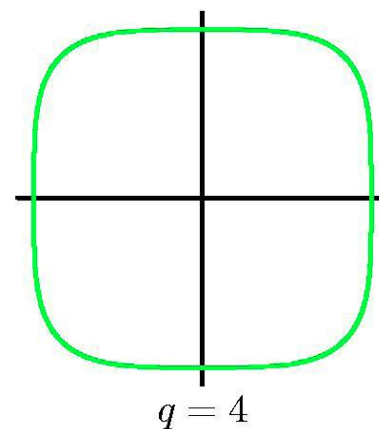
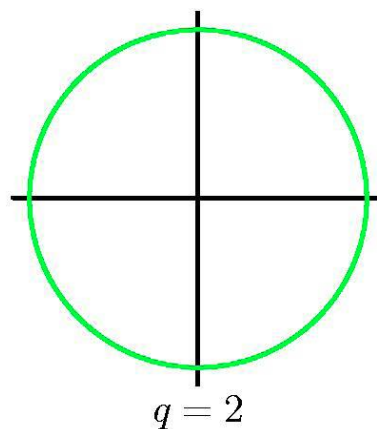
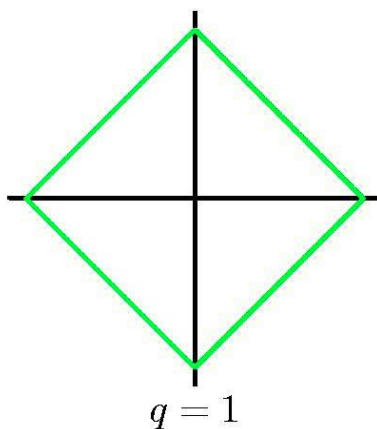
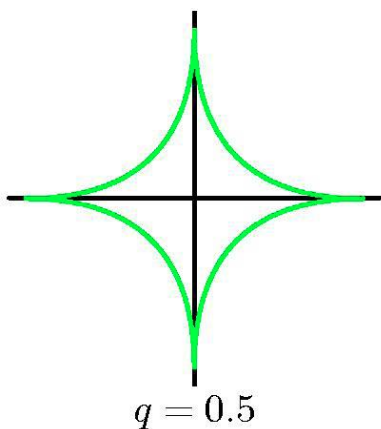
# 正则化方法

- 最小二乘正则化方法

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \sum_{j=1}^M |w_j|^q$$

- Ridge regression:  $q=2$

- Lasso:  $q=1$

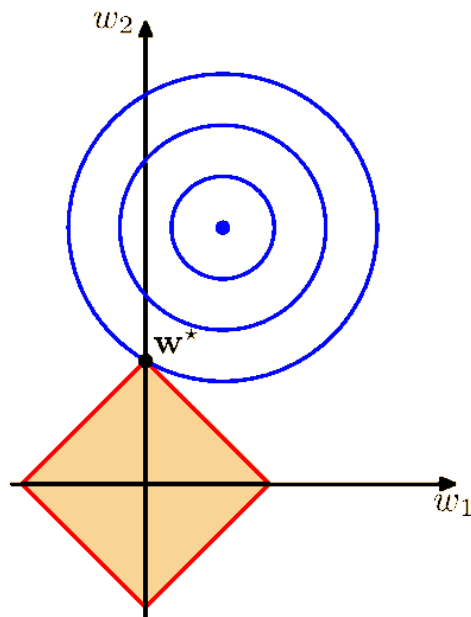


# L1范数可以导致稀疏解?

- 优化的角度 (Bishop, 2006, Hastie et al., 2009)

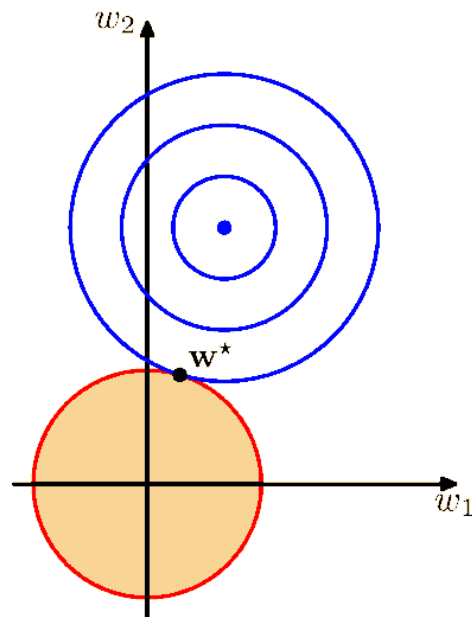
$$\min \text{loss}(x)$$

$$\text{s.t. } \|x\|_1 \leq 1$$

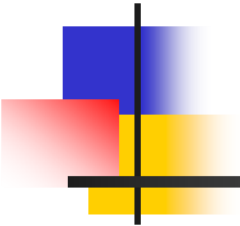


$$\min \text{loss}(x)$$

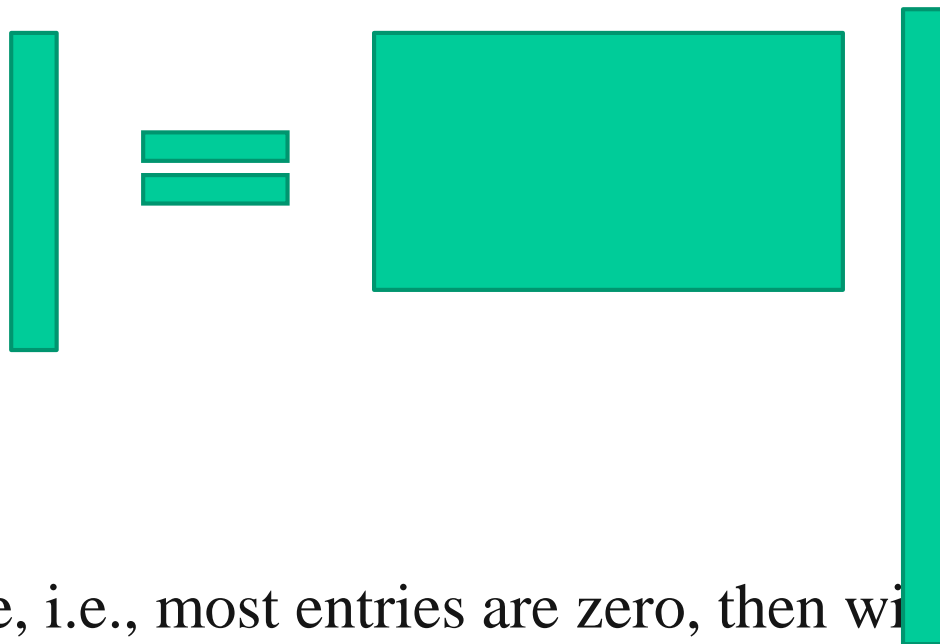
$$\text{s.t. } \|x\|_2 \leq 1$$



# 稀疏模型



- $y = Ax$



- 欠定方程
- If  $x$  is known to be sparse, i.e., most entries are zero, then with a large probability we can recover  $x$  exactly by solving a linear programming. (**Candes and Tao, 2005**)

# 稀疏学习一般模型

$$\min_{\mathbf{w}} l(\mathbf{w}) + \lambda r(\mathbf{w})$$

$$l(\mathbf{w}) = \frac{1}{2} \|\mathbf{y} - X\mathbf{w}\|^2$$

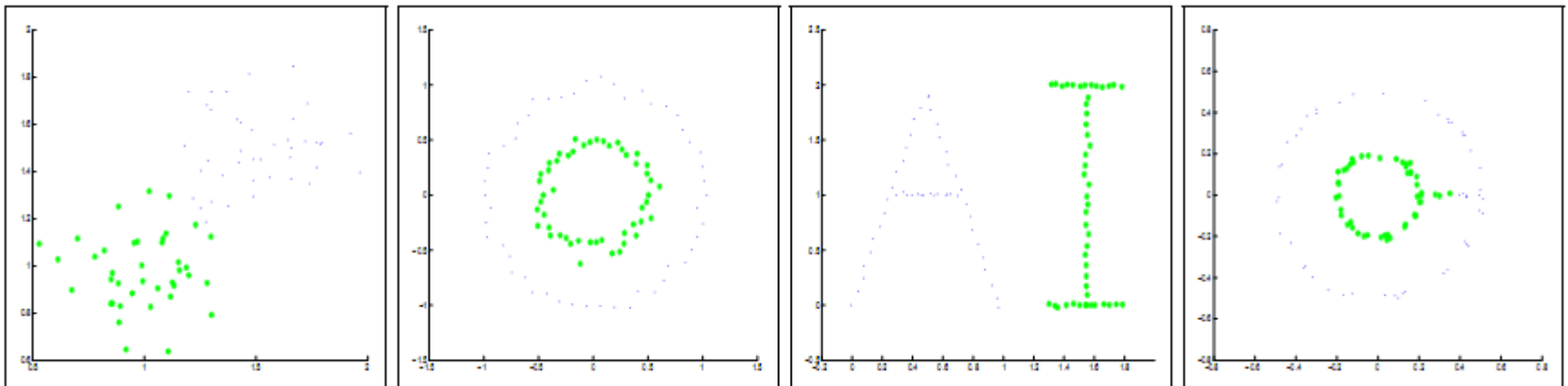
$$\min_{\mathbf{w}} l(\mathbf{w}) \quad s.t. \quad \mathbf{w} \in \mathcal{C}$$

$$l(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i))$$

名称	正则项	约束
$\ell_1$ 范数	$\ \mathbf{w}\ _1 = \sum_{i=1}^p  w_i $	$\ \mathbf{w}\ _1 \leq t$
$\ell_q$ 范数	$\ \mathbf{w}\ _q = \left( \sum_{i=1}^p  w_i ^q \right)^{1/q}$	$\ \mathbf{w}\ _q \leq t$
$\ell_{1,q}$ 范数	$\ \mathbf{w}\ _{1,q} = \sum_i \ \mathbf{w}_{G_i}\ _q$	$\ \mathbf{w}\ _{1,q} \leq t$
融合 $\ell_1$ 范数	$\ \mathbf{w}\ _{\text{FL}} = \lambda_1 \sum_{i=1}^p  w_i  + \lambda_2 \sum_{i=1}^{p-1}  w_i - w_{i+1} $	$\ \mathbf{w}\ _{\text{FL}} \leq t$
图上的 $\ell_1$ 范数	$\ \mathbf{w}\ _{\text{GL}} = \lambda_1 \sum_{i=1}^p  w_i  + \lambda_2 \sum_{(j,k) \in \mathcal{E}}  w_j - w_k $	$\ \mathbf{w}\ _{\text{GL}} \leq t$
迹范数	$\ W\ _{\text{tr}} = \sum_i \sigma_i(W)$	$\ \mathbf{w}\ _{\text{tr}} \leq t$

# Unsupervised Learning

- Maximum Margin Clustering[Xu et al., 2004]
- Find a labeling so that if one were to subsequently run an SVM, the margin obtained would be maximal over all possible labellings.





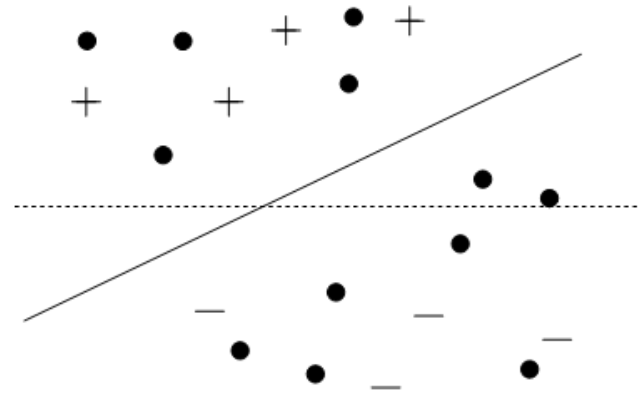
# Semi-supervised Learning

- Transductive SVM[Joachims, 1999]
- Training  $(\mathbf{x}_i, y_i), i = 1 \dots, N$
- Testing  $\mathbf{x}_j^*, j = 1 \dots, K$

$$\min_{y_1^*, \dots, y_K^*, \mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

$$y_j^* (\mathbf{w}^T \mathbf{x}_j^* + b) \geq 1, \forall j = 1, \dots, K$$

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





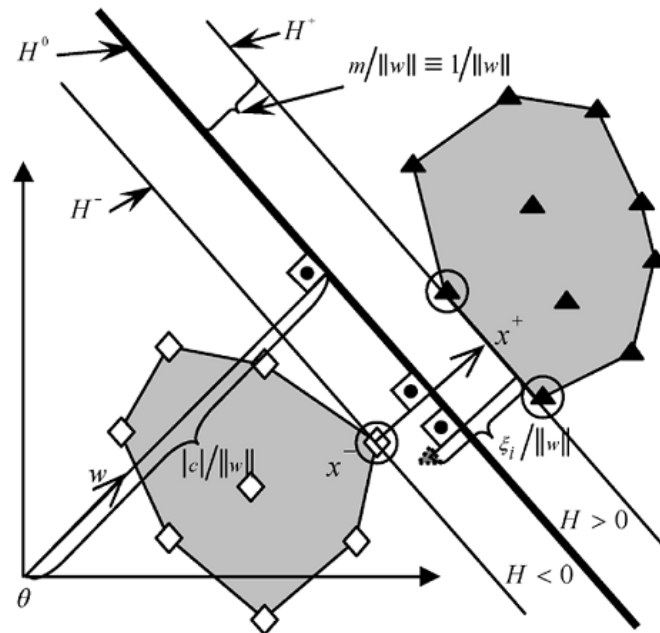
# Optimization

---

- Convex optimization
- Non-Convex optimization
- Large-Scale SVM[Yu et al,2010 (KDD best paper)]
  - Large scale: data can not fit in memory.
  - A block minimization method for linear SVM.
  - Experiment: Data size  $> 20 * \text{memory size}$ .

# Geometric Interpretation

- Geometrical Approach[Mavroforakis and Theodoridis, 2006]
- SVM  $\Leftrightarrow$  finding the points of the two convex hulls





# Multiple Kernels Learning

---

- The classification function of kernel SVM

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

- Multiple Kernel Learning

$$K(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^M d_m K_m(\mathbf{x}, \mathbf{x}')$$

where  $d_m \geq 0$ ,  $\sum_{m=1}^M d_m = 1$ ,  $K_m(\mathbf{x}, \mathbf{x}')$  is basis kernel,  
e.g. Gaussian kernels.



# Topics

---

- SVM and Statistical Learning Theory
- Fast algorithm
- Multi-class SVM



# 产生式模型

## Generative models

---

- 设计分类器的三种方法

1. 计算  $p(\mathbf{x}|C_k)$

计算

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$$

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|C_k)p(C_k).$$


表达了数据产生的方法



# 一些产生式模型

---

- 高斯混合模型GMM
- 隐马尔科夫模型HMM
- 独立成分分析ICA
- 因子分析FA
- ...

- 
- 缺点：需要大量样本计算  $p(\mathbf{x}|C_k)$
  - 优点：可以得到

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|C_k)p(C_k).$$

可以结合应用领域的知识



# 判别式模型 Discriminative model

2. 直接计算  $p(C_k|\mathbf{x})$ ,

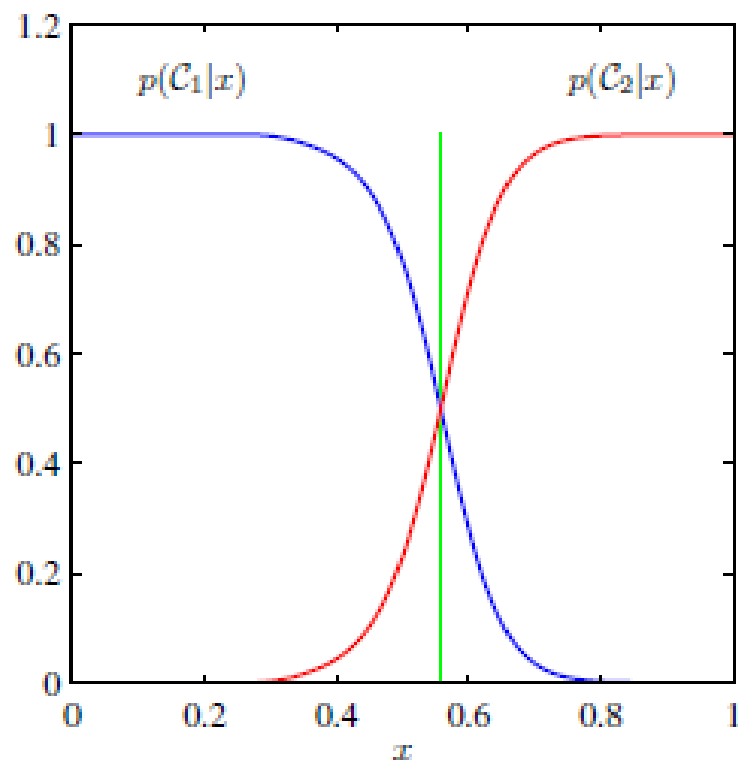
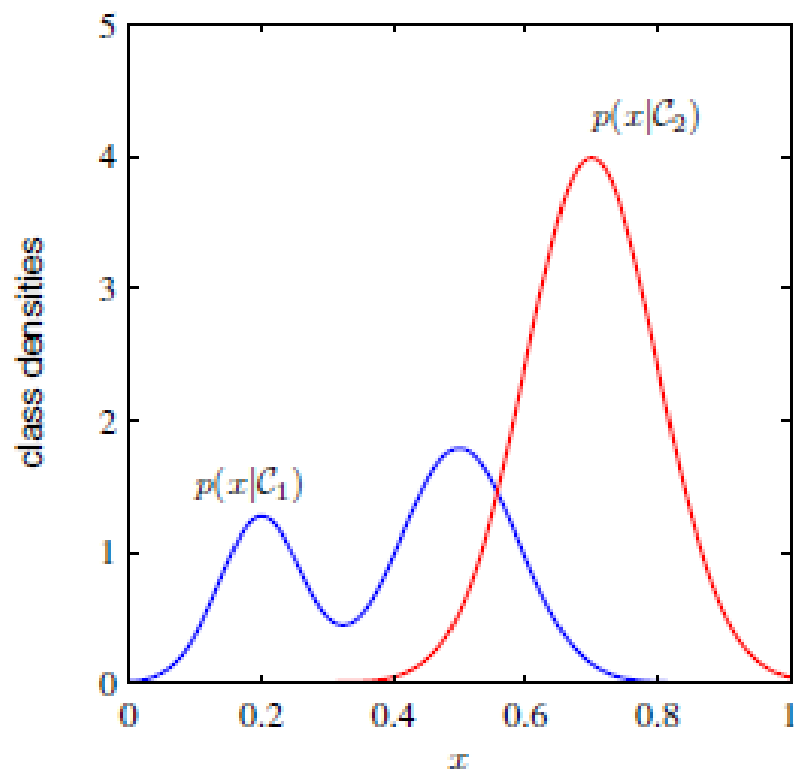
3. 直接计算判别函数  $f(\mathbf{x})$  ,如支持向量机

$f = 0$  represents class  $C_1$

$f = 1$  represents class  $C_2$

# 产生式模型与判别式模型

- 类条件概率密度函数的某些细节对分类没有用处





# 判别式模型

---

- 分类的准确率往往更高



# 计算后验的好处

---

- 很容易地修正最小化风险决策准则。如果仅仅有一个判别函数，那么损失矩阵的任何改变都要求用训练数据并重新解决分类问题
- 选择拒绝策略

- 
- 
- 如何把产生式模型和判别式模型结合？



# Reference

---

- Vladimir N. Vapnik著，张学工译，《统计学习理论的本质》，清华大学出版社，2002年9月第1版
- Burges, C.J.C. (1998) “A Tutorial on Support Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery, 2(2), pp. 121-167.
- Bernhard Schölkopf, “Statistical Learning and Kernel Methods”, <http://research.microsoft.com/~bsc>