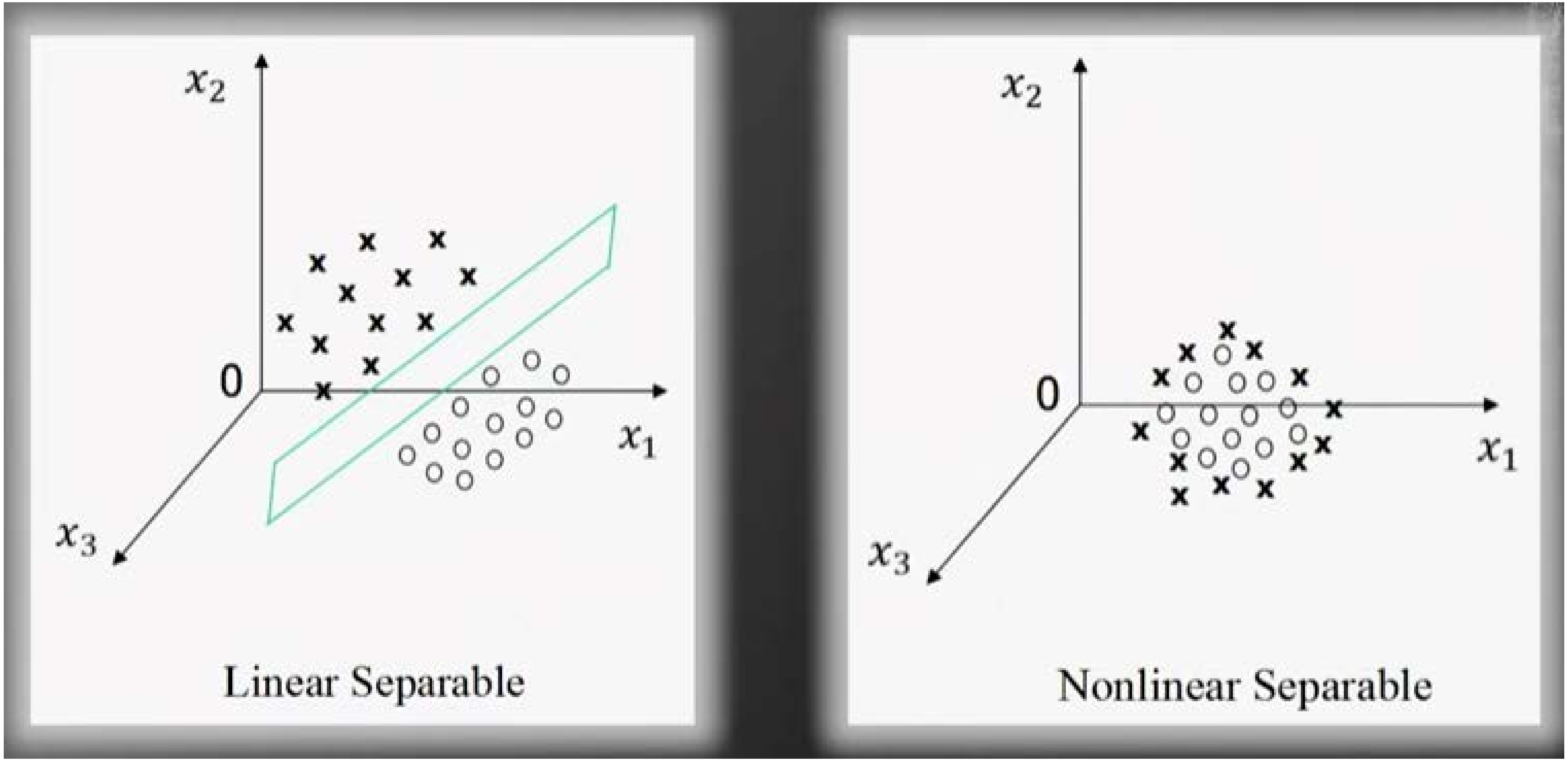
支持向量机



Linear Separable

Nonlinear Separable

用数学严格定义训练样本以及他们的标签

假设：

我们有N个训练样本和他们的标签

$$\{(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)\}$$

其中 $X_i = [x_{i1}, x_{i2}]^T$

$$y_i = \{+1, -1\}$$

用数学严格的定义线性可分

线性可分的严格定义：一个训练样本集 $\{(X_i, y_i), \ldots, (X_N, y_N)\}$，
在i=1~N线性可分，是指存在（ω_1，ω_2，b），
使得对 i=1~N，有：

(1) 若$y_i = +1$，则 $\omega_1 x_{i1} + \omega_2 x_{i2} + b > 0$

(2) 若$y_i = -1$，则 $\omega_1 x_{i1} + \omega_2 x_{i2} + b < 0$

用向量形式来定义线性可分

假设：

$$X_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}^{\mathrm{T}} \quad \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}^{\mathrm{T}}$$

(1) 若 $y_i = +1$，则 $\omega^T X_i + b > 0$

(2) 若 $y_i = -1$，则 $\omega^T X_i + b < 0$

线性可分定义的最简化形式

(1) 若$y_i = +1$，则 $\omega^T X_i + b > 0$

(2) 若$y_i = -1$，则 $\omega^T X_i + b < 0$

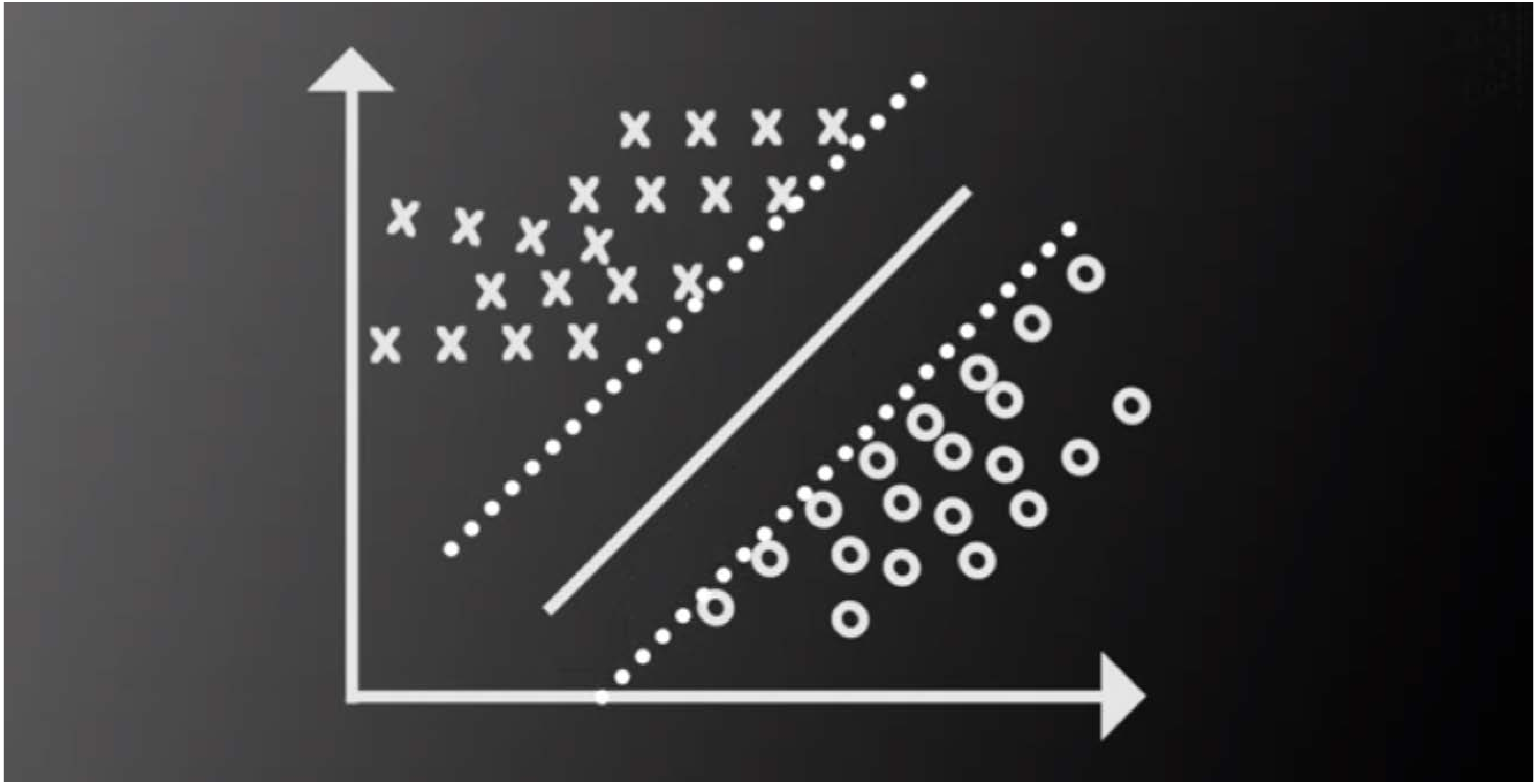
如果

y_i=+1 或 −1

一个训练样本集 $\{(X_i, y_i)\}$, 在i =1~N线性可分，

是指存在（ω,b), 使得对 i=1~N，有：

$$y_i(\omega^T X_i + b) > 0$$



支持向量机寻找的最优分类直线应满足：

（1）该直线分开了两类；

（2）该直线最大化间隔（margin）；

（3）该直线处于间隔的中间，到所有支持向量距离相等。

最优分类超平面应该满足：

1. 该超平面分开了两类；
2. 该超平面有最大化间隔；
3. 该超平面处于间隔的中间，到所有支持向量距离相等。

线性可分的定义

一个训练样本集$\{(x_i, y_i)\}$，$i=1\sim N$线性可分，是指存在（ω，b）使：

① 当 $y_i = +1$ 时，$\omega^T x_i + b > 0$

② 当 $y_i = -1$ 时，$\omega^T x_i + b < 0$

假定训练样本集是线性可分的

支持向量机需要寻找的是最大化间隔（MARGIN）的超平面

最小化（Minimize）：$\frac{1}{2}\|\omega\|^2$

限制条件：$y_i(\omega^T x_i + b) \geq 1, (i = 1 \sim N)$

限制条件：$y_i = (\omega^T x_i + b) \geq 1, (i = 1 \sim N)$

最小化（Minimize）：$\frac{1}{2}\|\omega\|^2$

限制条件：$y_i(\omega^T x_i + b) \geq 1, (i = 1 \sim N)$

已知：

训练样本集{(X_i, y_i)}，i=1到N；

待求：（ω，b）

事实 1

事实 2

一个点 X_0 到超平面 $\omega^T x + b = 0$ 的距离

$$d = \frac{\left|\omega^T x_0 + b\right|}{\|\omega\|}$$

事实2　点到超平面的距离公式

一个点(x_0, y_0)到超平面

$\omega_1 x + \omega_2 y_0 + b = 0$的距离

$$d = \frac{\left|\omega_1 x_0 + \omega_2 y_0 + b\right|}{\sqrt{\omega_1^2 + \omega_2^2}}$$

$(\omega\ ,\ b) \rightarrow (\alpha\omega\ ,\ \alpha b)$

最终使在支持向量 x_0 上有 $\left|\omega^T x_0 + b\right| = 1$，

而在非支持向量上 $\left|\omega^T x_0 + b\right| > 1$，

(ω，b)表示的超平面和(aω，ab)表示的超平面是同一个平面。

参数a去缩放（ω，b）

据事实2，支持向量x_0到超平面的距离将会变为：

$$d = \frac{\left|\omega^T x_0 + b\right|}{\|\omega\|} = \frac{1}{\|\omega\|}$$

优化问题定为：

$\boxed{最小化：\frac{1}{2}\|\omega\|^2}$ $\boxed{最小化：\|\omega\|}$

限制条件：

支持向量到超平面的距离为：$\frac{1}{2}\|\varepsilon\|$

在非支持向量上

$$|\omega^T x_0 + b| > 1$$

限制条件：

$$y_i\left(\omega^T x_i + b\right) \geq 1 \quad i = 1 \sim N$$

$y_i \Longrightarrow$ 协调超平面的左右

$\Longrightarrow (\omega^T x + b) > 1$

$\Longrightarrow (\omega^T x + b) < 1$

限制条件：

(ω, b)

相差一个a

$$y_i(\omega^T x_i + b) \geq 1 \quad i = 1 \sim N$$

限制条件：$y_i(\boldsymbol{\omega}^T \boldsymbol{x}_i + \boldsymbol{b}) \geq 2$

线性可分情况下，支持向量机寻找最佳超平面的优化问题可以表示为：

凸优化 (CONVEX OPTIMIZATION)

最小化（Minimize）：$\frac{1}{2}\|\boldsymbol{\omega}\|^2$

限制条件：$y_i = (\boldsymbol{\omega}^T \boldsymbol{x}_i + \boldsymbol{b}) \geq \mathbf{1}, (\boldsymbol{i} = \mathbf{1} \sim \boldsymbol{N})$

$(X_i, y_i), i = 1 \sim N$ 是已知的

（ω ,b）是待求的

二次规划的定义：

（1）目标函数 (Objective Function) 是二次项。

（2）限制条件是一次项。

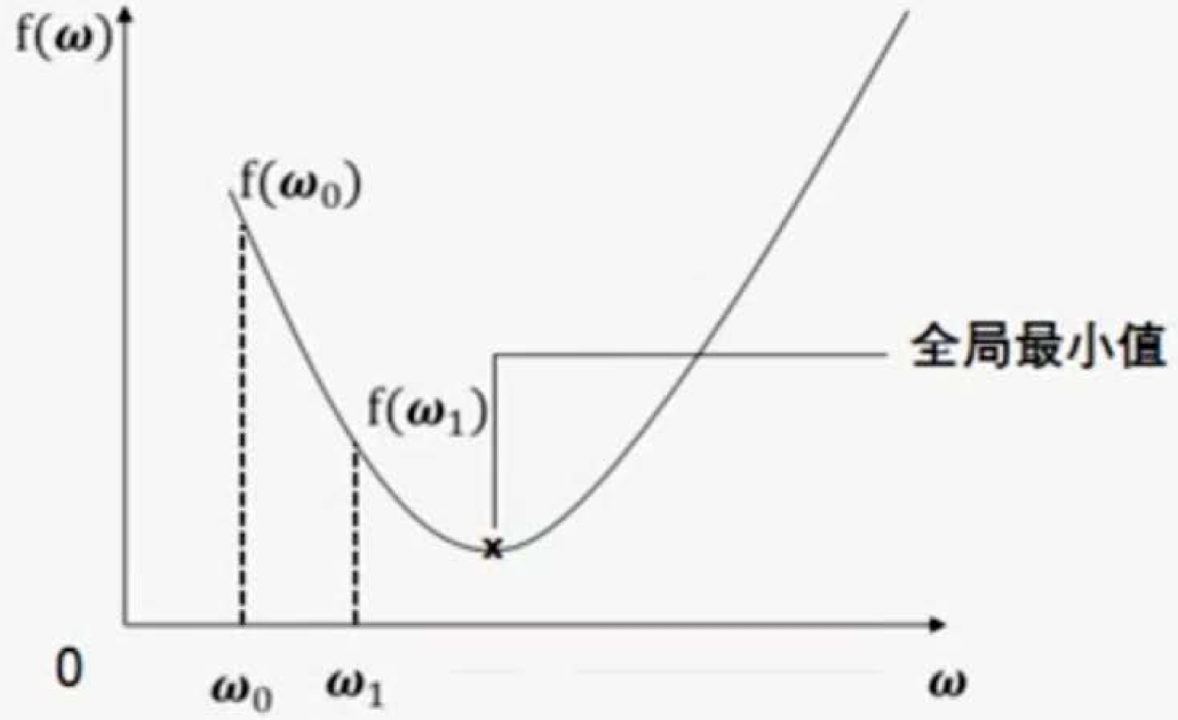
目标函数：$\frac{1}{2}\|\omega\|^2$ $\Longrightarrow$ $\frac{1}{2}\|\omega\|^2 = \frac{1}{2}\omega_1^2 + \frac{1}{2}\omega_2^2 + \dots + \frac{1}{2}\omega_M^2$

限制条件：$y_i=\left(\omega^T x_i + b\right) \geq 1, (i = 1 \sim N)$

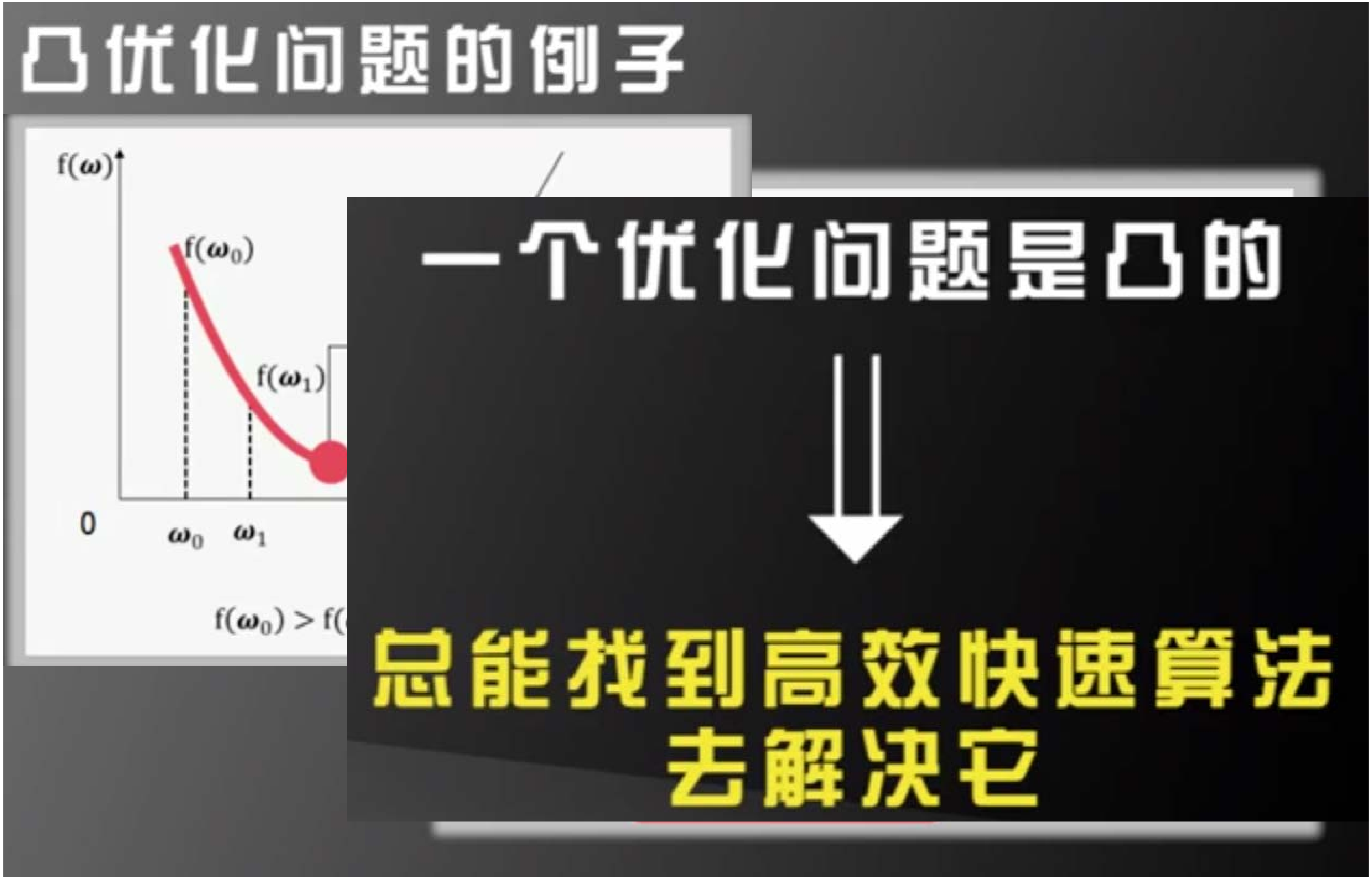
要么无解，要么只有唯一的最小值。

凸优化问题的例子

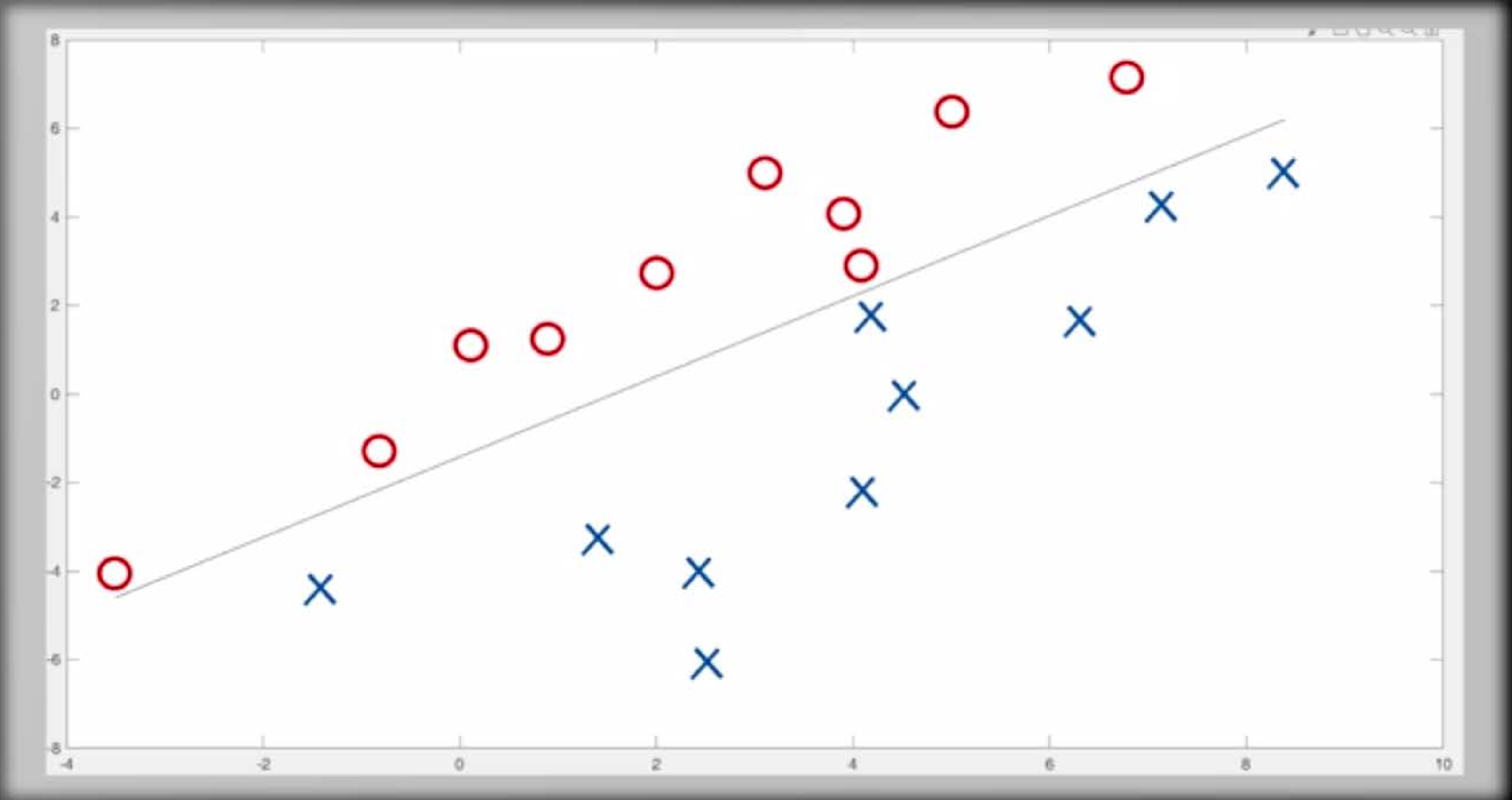
$f(\omega)$



$f(\omega_0) > f(\omega_1)$


凸优化问题的例子
f(ω)
f(ω0)
f(ω1)
0
ω0
ω1
一个优化问题是凸的
⇓
总能找到高效快速算法
去解决它

用凸优化解出的支持向量机最佳分类超平面的例子



放松限制条件的基本思路

$\Downarrow$

对每个训练样本及标签(X_i, Y_i)

$\Downarrow$

松弛变量 δ_i

(slack variable)

对于线性不可分情况，需适当放松限制条件。

限制条件改写：$\boxed{y_i(\omega^T X_i + b) \geq} \mathbf{1} - \delta_i, (i = \mathbf{1} \sim N)$

松弛变量 δ_i

(slack variable)

改造后的支持向量机优化版本

最小化：$\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\delta_i$ 或 $\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\delta_i^2$

限制条件：（1）$\delta_i \geq 0, (i = 1 \sim N)$

（2）$y_i(\omega^T X_i + b) \geq 1 - \delta_i, (i = 1 \sim N)$

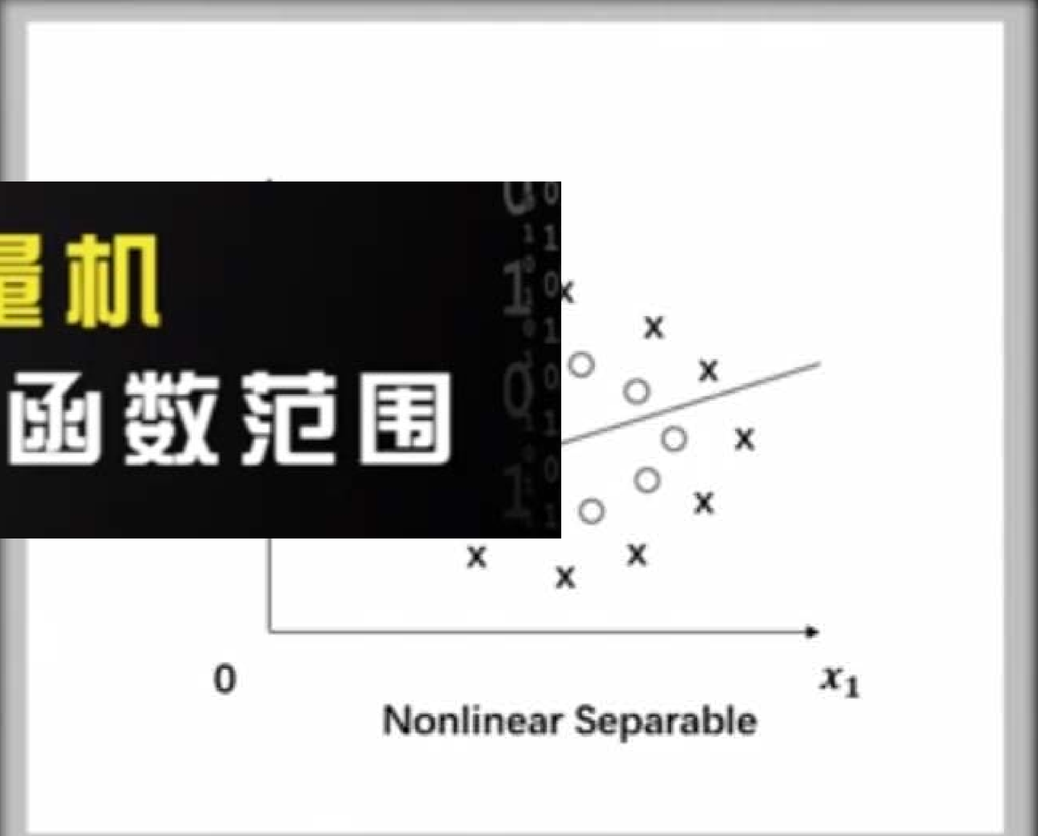
以前的目标函数只需要最小化 $\frac{1}{2}\|\omega\|^2$

现在的目标函数增加了一项 所有 δ_i 的和


不断变化C的值
同时测试算法的识别率
选取超参数C

在线性不可分情况下应用支持向量机

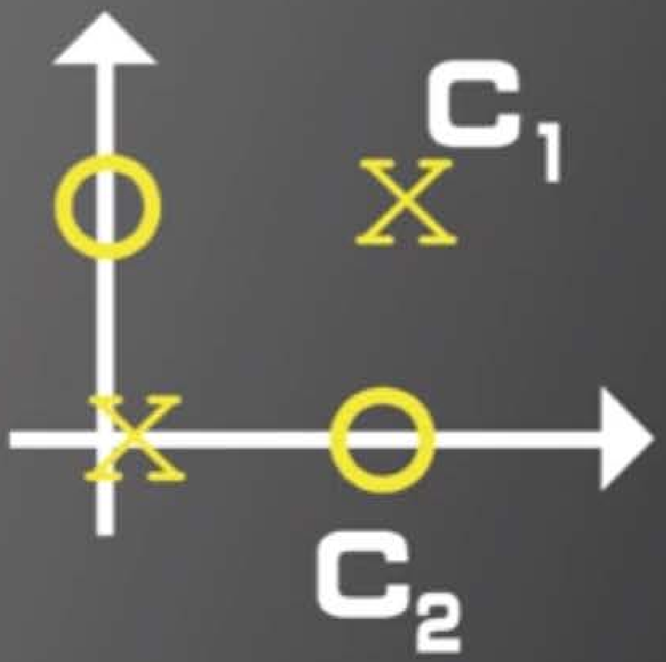
取目标函数：$\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\delta_i$ ，$C = 10000$



支持向量机

如何扩大可选函数范围

考察如图的异或问题



即：

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in C_1 \qquad x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \in C_1$$

$$x_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in C_2 \qquad x_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in C_2$$

构造一个二维到五维的映射 $\varphi(x)$

$$\varphi(x): \quad x = \begin{bmatrix} a \\ b \end{bmatrix} \longrightarrow \varphi(x) = \begin{bmatrix} a^2 \\ b^2 \\ a \\ b \\ ab \end{bmatrix}$$

$$\varphi(x_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \varphi(x_2) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\varphi(x_3) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \varphi(x_4) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

设：

$$\omega = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 6 \end{bmatrix} \qquad b=1$$

$$\omega^T\varphi(x_1)+b=1\geqslant 0 \qquad \omega^T\varphi(x_2)+b=3\geqslant 0$$

$$\omega^T\varphi(x_3)+b=-1<0 \qquad \omega^T\varphi(x_4)+b=-1<0$$

假设：

在一个M维空间上随机取N个训练样本

随机的对每个训练样本赋予标签+1或-1

假设：

这些训练样本线性可分的概率为P(M)

当M趋于无穷大时　P(M)=1

将训练样本
由低维映射到高维

$\Downarrow$

增大线性可分的概率

由低维到高维的映射 $\varphi(x)$

最小化：$\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\delta_i$ 或 $\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\delta_i^2$

限制条件：① $\delta_i \geq 0, (i = 1 \sim N)$

② $y_i[\omega^T\varphi(X_i) + b] \geq 1 - \delta_i, (i = 1 \sim N)$

$X_i \Longrightarrow$ 被 $\varphi(X_i)$ 替换

$$K(X_1, X_2) = \varphi(X_1)^T \varphi(X_2)$$

举两个例子

核函数以及低维到高维的映射φ(X)之间的相互关系。

假设：

φ(X)是一个将二维向量映射为三维向量的映射

$$\begin{aligned} K(X_1, X_2) &= \varphi(X_1)^T \varphi(X_2) \\ &= [x_{11}^2, x_{11}x_{12}, x_{12}^2][x_{21}^2, x_{21}x_{22}, x_{22}^2]^T \\ &= x_{11}^2 x_{21}^2 + x_{11}x_{12}x_{21}x_{22} + x_{12}^2 x_{22}^2 \end{aligned}$$

已知核函数K求映射φ的例子

假设：

$$K(X_1,X_2) = (x_{11}x_{21} + x_{12}x_{22} + 1)^2$$

$$= x_{11}^2\,x_{21}^2 + x_{12}^2\,x_{22}^2 + 1 + 2x_{11}x_{21}x_{12}x_{22} + 2x_{11}x_{21} + 2x_{12}x_{22}$$

$$\boxed{K(X_1,X_2) = \varphi(X_1)^T\varphi(X_2)}$$

$$\varphi(X) = \varphi([x_1,x_2]^T)$$

$$= [x_1^2, x_2^2, 1, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2]^T$$

核函数K和映射φ是一一对应的关系

核函数的形式不能随意的取

$\Big\Downarrow$ 满足一定的条件

两个φ内积的形式

$K(X_1,X_2)$能写成$\varphi(X_1)^T\varphi(X_2)$的充要条件

① $K(X_1,X_2) = K(X_2,X_1)$ (交换性)

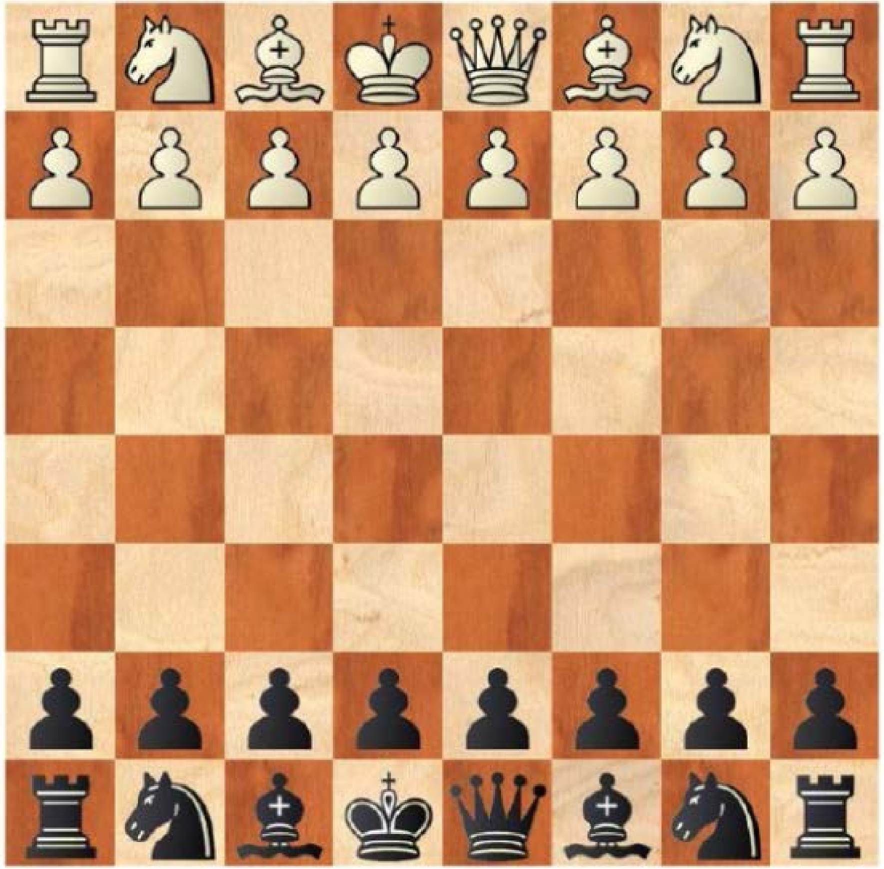
② $\forall C_i(i = 1 \sim N)$ ，$\forall N$ 有$\sum_{i=1}^{N}\sum_{j=1}^{N} C_i C_j K(X_i X_j) \geq 0$ (半正定性)

核函数

- **Linear**（线性内核）： $K(x,y)=x^Ty$
- **Ploy**（多项式核）： $K(x,y)=(x^Ty+1)^d$
- **Rbf**（高斯径向基函数核）： $K(x,y)=e^{-\frac{\|x-y\|^2}{\sigma^2}}$
- **Tanh**（**sigmoid** 核）：

$$K(x,y)=\tanh(\beta x^Ty+b) \qquad \tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$$

国际象棋规则介绍



- 兵 （黑白各**8**个）：第一步向前可走一格或两格，以后每次只能向前走一格，不能后退。但在吃对方子时，则是向位于斜前方的那格去吃，并落在那个格。

- 王 （黑白各1个）：是国际象棋中最为重要的棋子，王被将死即告负。走法是每次横直斜走均可，但每次只能走一格。吃子与走法相同。

- 兵王问题：黑方只剩一个王，白方剩一个兵一个王。
- 两种可能

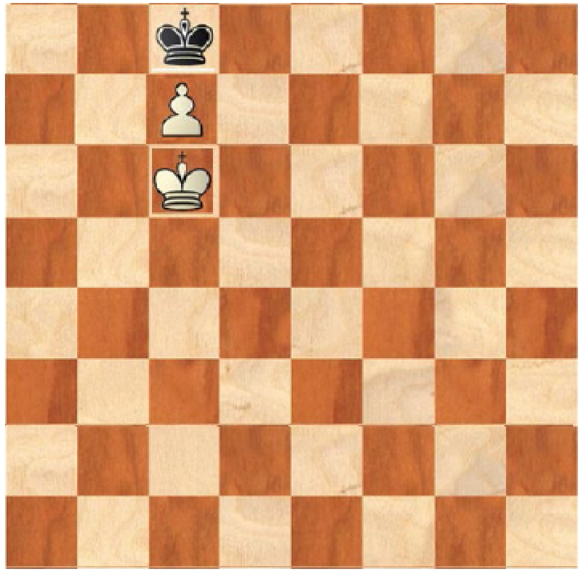
（1）白方将死黑方，获胜。

（2）和棋。

这两种可能视三个棋子在棋盘的位置而确定。

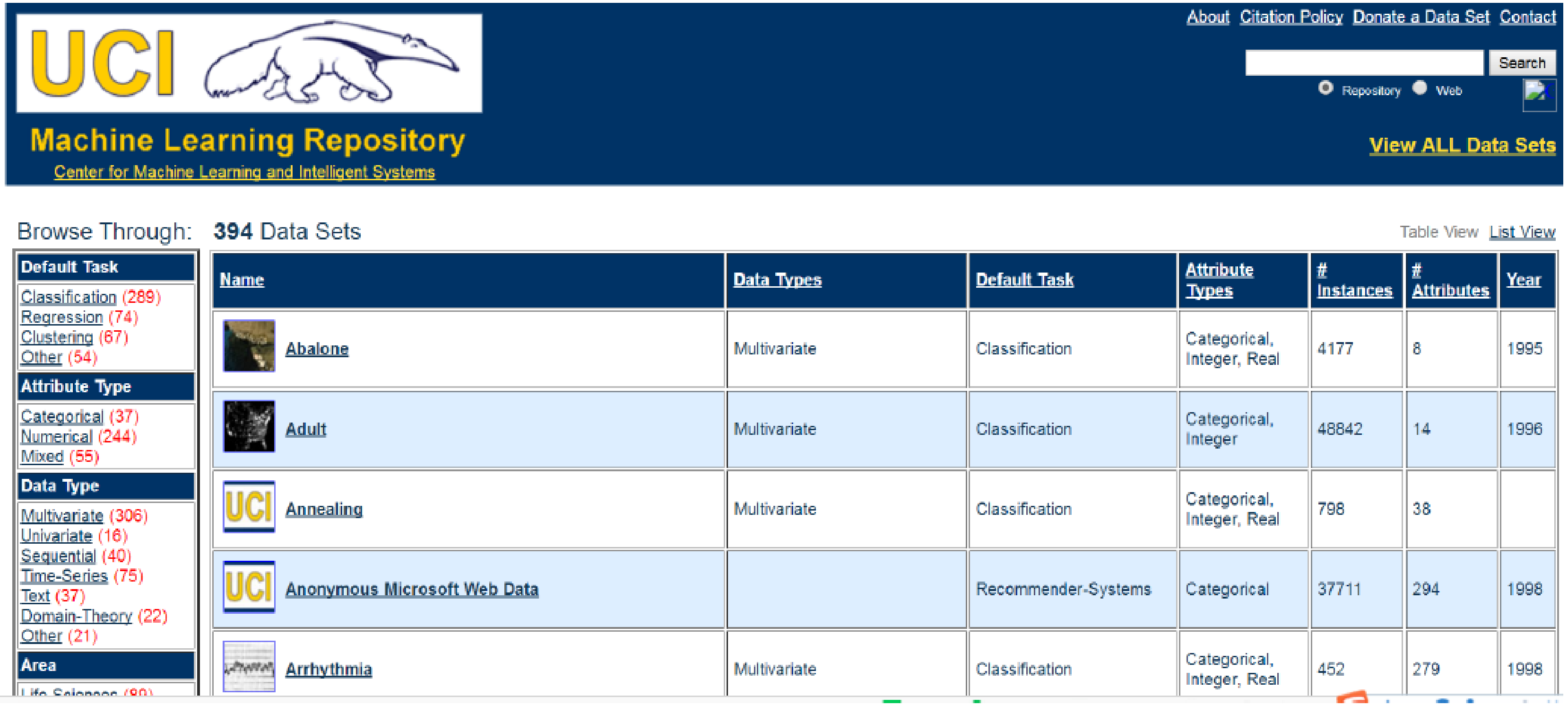
- 兵的升变：兵走至对方底线，可以升变为除王以外的任意一子。

- 逼和：一方的王未被将军，但移动到任意地方都会被对方将死，则此时是和棋。

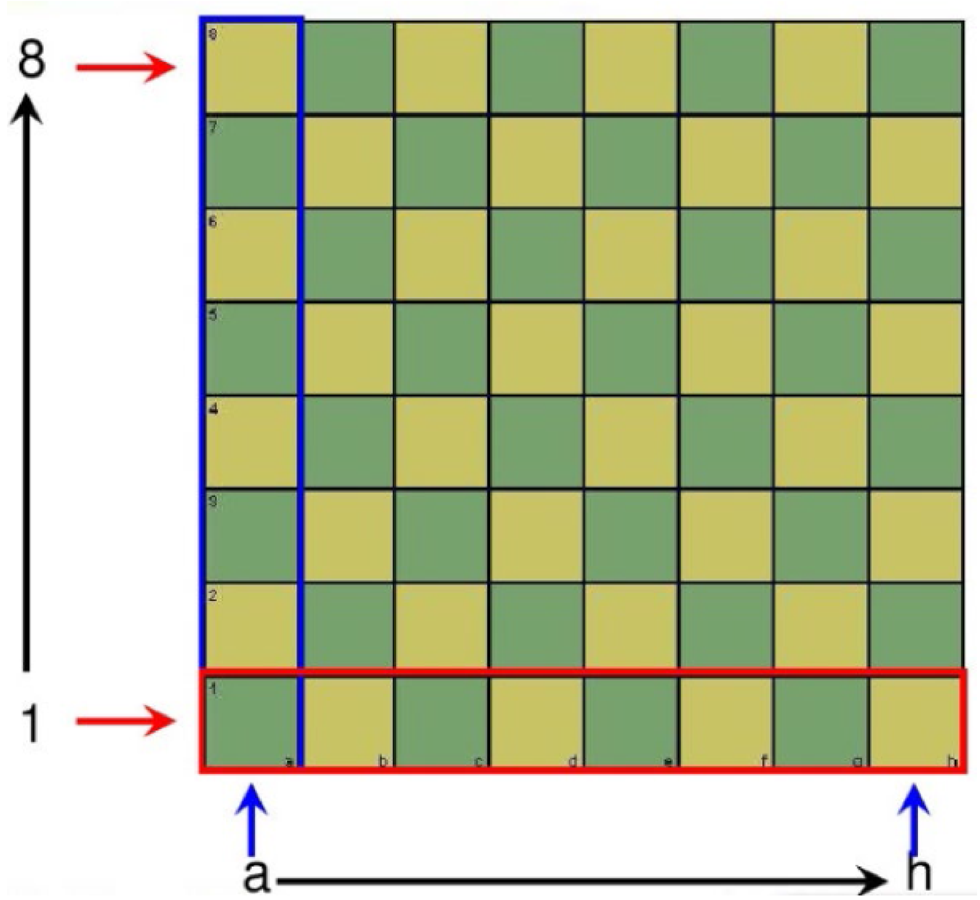


用SVM解兵王问题

- **UCI Machine Learning Repository**



- **krkopt.data文件**




```
a,1,b,3,c,2,draw
a,1,c,1,c,2,draw
a,1,c,1,d,1,draw
a,1,c,1,d,2,draw
a,1,c,2,c,1,draw
a,1,c,2,c,3,draw
a,1,c,2,d,1,draw
a,1,c,2,d,2,draw
a,1,c,2,d,3,draw
a,1,c,3,c,2,draw
a,1,c,3,d,2,draw
a,1,c,3,d,3,draw
a,1,c,3,d,4,draw
。。。
d,1,e,3,f,1,six
d,1,e,3,g,1,six
d,1,e,3,g,2,six
d,1,e,4,h,1,six
d,1,e,4,h,2,six
d,1,e,4,h,3,six
d,1,e,5,h,1,six
d,1,e,6,h,1,six
。。。
c,1,e,7,c,5,fifteen
c,1,e,7,c,6,fifteen
c,1,e,7,c,7,fifteen
c,1,e,7,d,5,fifteen
c,1,e,7,e,5,fifteen
c,1,f,1,c,3,fifteen
```

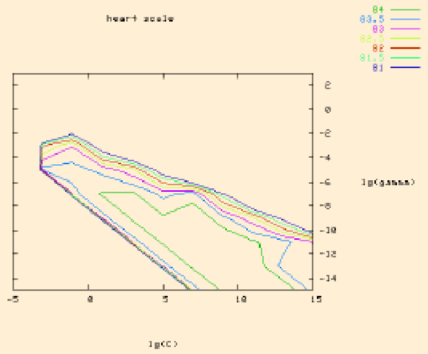
- **LIBSVM -- A Library for Support Vector Machines**
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/

LIBSVM -- A Library for Support Vector Machines

Chih-Chung Chang and Chih-Jen Lin

- NEW Version 3.22 released on December 22, 2016. It conducts some minor fixes.
- NEW LIBSVM tools provides **many extensions** of LIBSVM. Please check it if you need some functions not supported in LIBSVM.
- NEW We now have a nice page LIBSVM data sets providing problems in LIBSVM format.
- NEW A practical guide to SVM classification is available now! (mainly written for beginners)

We now have an easy script (easy.py) for users who know NOTHING about SVM. It makes everything automatic--from data scaling to parameter selection.
The parameter selection tool grid.py generates the following contour of cross-validation accuracy. To use this tool, you also need to install python and gnuplot.



To see the importance of parameter selection, please see our guide for beginners.
NEW Using libsvm, our group is the winner of IJCNN 2001 Challenge (two of the three competitions), EUNITE world wide competition on electricity load prediction, NIPS 2003 feature selection challenge (third place), WCCI 2008 Causation and Prediction challenge (one of the two winners), and Active Learning Challenge 2010 (2nd place).

- 总样本数**28056**， 其中正样本**2796**, 负样本**25260**。
- 随机取**5000**个样本训练，其余测试。
- 样本归一化，在训练样本上，求出每个维度的均值和方差，在训练和测试样本上同时归一化。

$$newX = \frac{X - mean(X)}{std(X)}$$

- 高斯核
- **5-fold cross validation**，在

CScale = [2^(-5), 2^15]; gamma = [2^(-15),2^3];

上遍历求识别率的最大值。

上述**C**和**gamma**的区间设置参见**LIBSVM**自带的介绍：

a practical guide to support vector classification

- 训练参数设置 **svmtrain(yTraining, xTraining, cmd)**

cmd参数如下:

(1) –s 0 "-s svm_type : set type of SVM (default 0)\n"

" 0 -- C-SVC   (multi-class classification)\n"

" 1 -- nu-SVC   (multi-class classification)\n"

" 2 -- one-class SVM\n"

" 3 -- epsilon-SVR   (regression)\n"

" 4 -- nu-SVR   (regression)\n"

(2) –t 2

“-t kernel_type : set type of kernel function (default 2)\n"

"   0 -- linear: u'*v\n"

"   1 -- polynomial: (gamma*u'*v + coef0)^degree\n"

"   2 -- radial basis function: exp(-gamma*|u-v|^2)\n"

"   3 -- sigmoid: tanh(gamma*u'*v + coef0)\n"

"   4 -- precomputed kernel (kernel values in training_instance_matrix)\n"

(3) –c CVALUE

"-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)\n"

(4) –g gammaValue

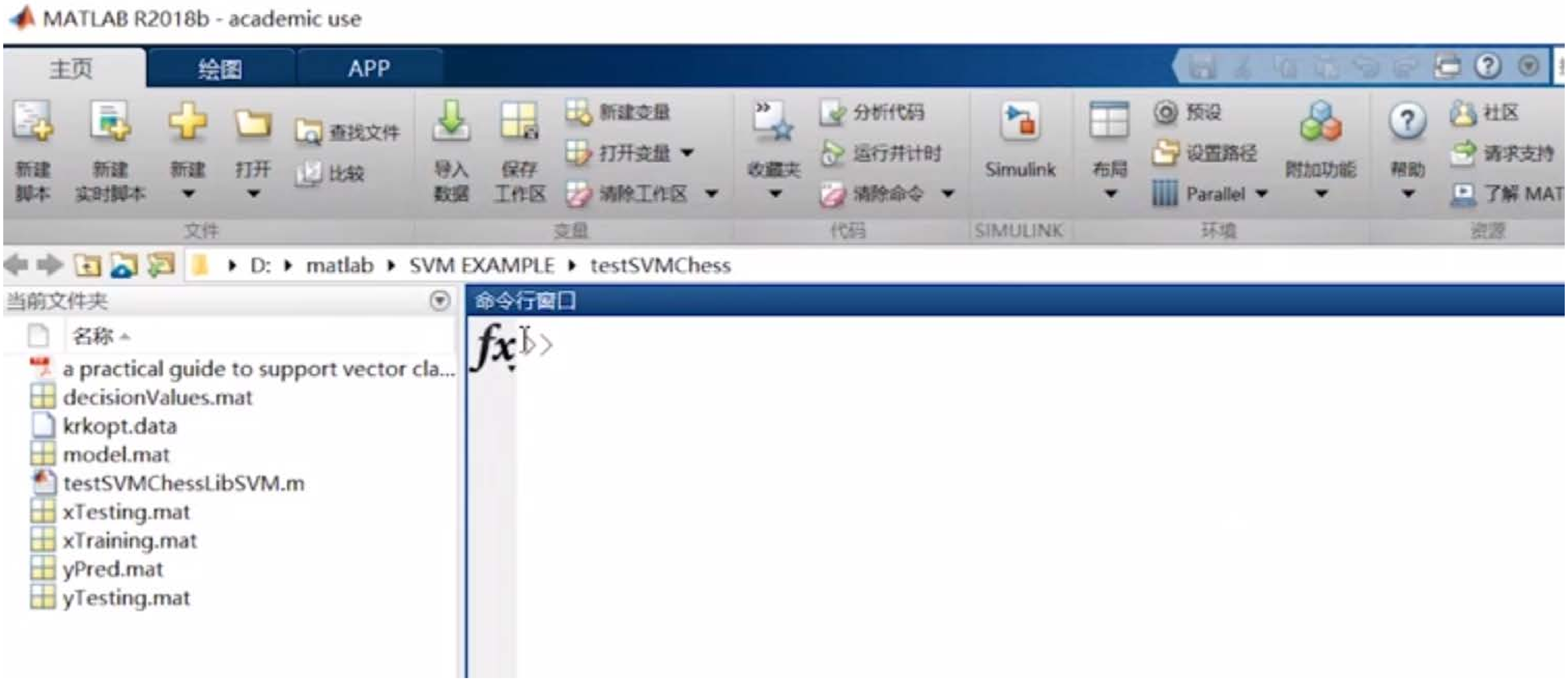
"-g gamma : set gamma in kernel function (default 1/num_features)\n“

和棋（draw）

$$y_i = +1$$

$$y_i = -1$$

其他 (ONE-FIFTEEN)


MATLAB R2018b - academic use
主页
绘图
APP
新建脚本
新建实时脚本
新建
打开
查找文件
比较
导入数据
保存工作区
新建变量
打开变量
清除工作区
收藏夹
分析代码
运行并计时
清除命令
Simulink
布局
预设
设置路径
Parallel
附加功能
帮助
社区
请求支持
文件
变量
代码
SIMULINK
环境
资源
D: matlab SVM EXAMPLE testSVMChess
当前文件夹
命令行窗口
名称
a practical guide to support vector cla...
decisionValues.mat
krkopt.data
model.mat
testSVMChessLibSVM.m
xTesting.mat
xTraining.mat
yPred.mat
yTesting.mat
fx >>

```
vec = zeros(6,1);
xapp = [];
yapp = [];
while ~feof(fid)
    string = [];
    c = fread(fid,1);
    flag = flag+1;
    while c~=13
        string = [string, c];
        c=fread(fid,1);
    end;
    fread(fid,1);
    if length(string)>10
        vec(1) = string(1) - 96;
        vec(2) = string(3) - 48;
        vec(3) = string(5) - 96;
        vec(4) = string(7) - 48;
        vec(5) = string(9) - 96;
        vec(6) = string(11) - 48;
        xapp = [xapp,vec];
        if string(13) == 100
            yapp = [yapp,1];
        else
            yapp = [yapp,-1];
        end;
    end;
end;
fclose(fid);
```

```
>> testSVMChessLibSVM
K>> size(xapp)

ans =

        6       28056
```

```
K>> xapp(:, 1)

ans =

     1
     1
     2
     3
     3
     2
```

- 第一步：用ABCD训练，用E测试，获得识别率。
- 第二步：用ABCE训练，用D测试，获得识别率。
- 第三步：用ABDE训练，用C测试，获得识别率。
- 第四步：用ACDE训练，用B测试，获得识别率。
- 第五步：用BCDE训练，用A测试，获得识别率。

- 训练后获得的参数

（1）**C = 16， gamma = 0.0825**

（2）支持向量（即**alpha**不为**0**的向量）：**358**个（**162**个正样本，**196**个负样本)

（3）**b = 6.2863**

- 识别结果

		预测	
		正样本	负样本
实际	正样本	TP (2249)	FN (39)
	负样本	FP (51)	TN (20717)

		预测	
		正样本	负样本
实际	正样本	TP (98.295%)	FN (1.705%)
	负样本	FP (0.246%)	TN (99.754%)

表. 在测试集上的混淆矩阵

(Accuracy = 99.60%)