支持向量机

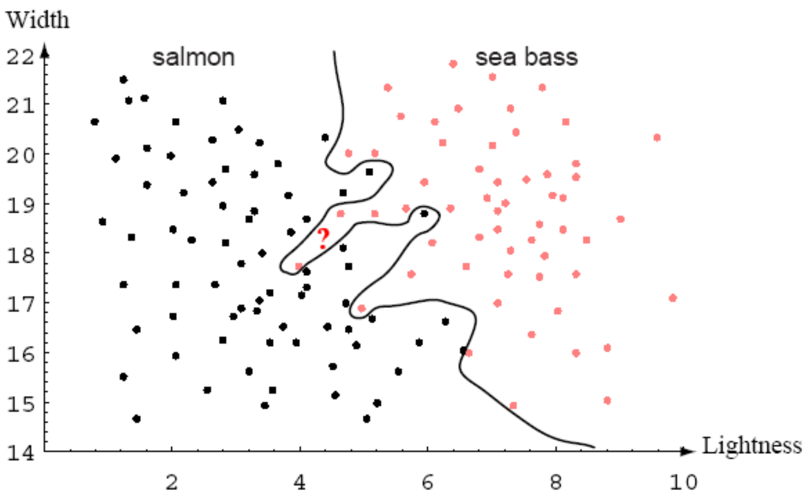
Support Vector Machine

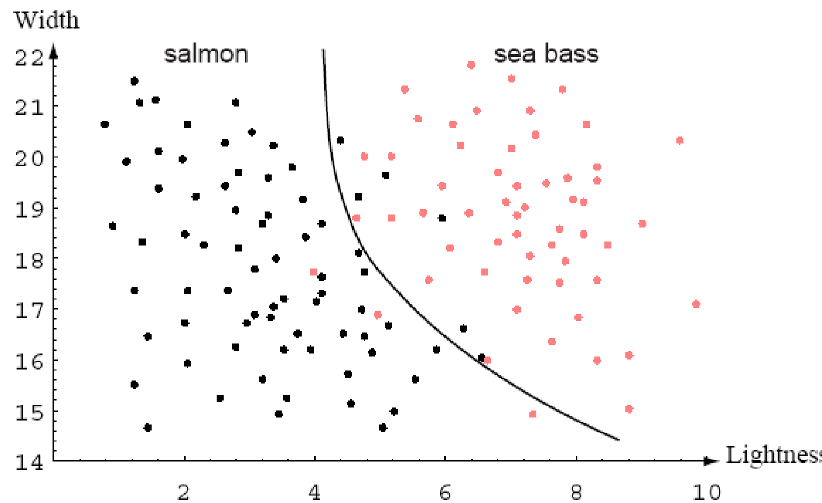
洪 波
Dept. of Biomedical Engineering
Tsinghua University


SUPPORT
VECTOR
MACHINES

机器学习“永远的痛”：泛化能力

Generalization ability





通常情况下，我们总是努力在有限样本的训练集上能找到使得经验风险最小（**ERM**）的分类器，而且希望当样本数增多时，经验风险趋近于期望风险.

Issue of generalization

A small emprical risk does not imply small true expected risk.

$$R_{emp} \xrightarrow[?]{n \to \infty} R$$

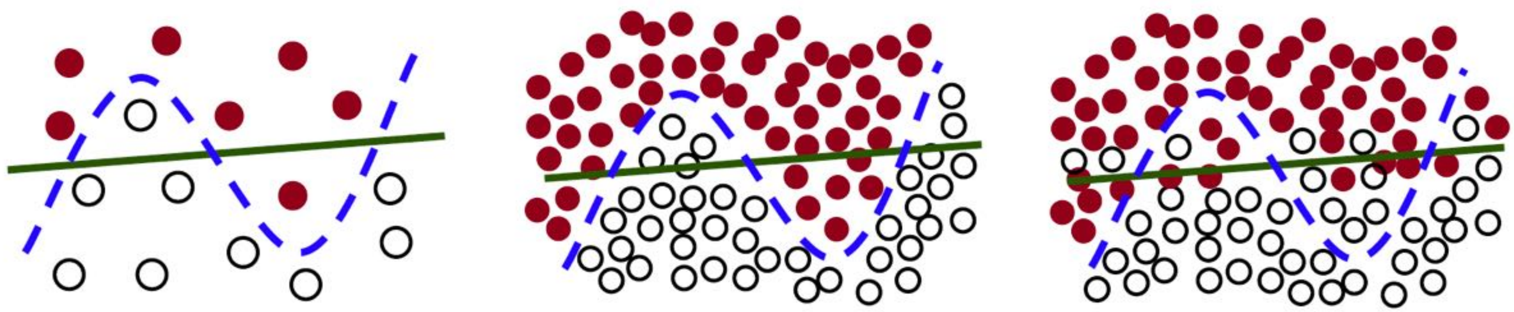
经验风险最小化与奥坎剃刀

Empirical Risk Minimization and Occam’s razor

Tradeoff:

- **Performance of classification (ERM)**
- **Simplicity of classifier (Occam’s razor)**

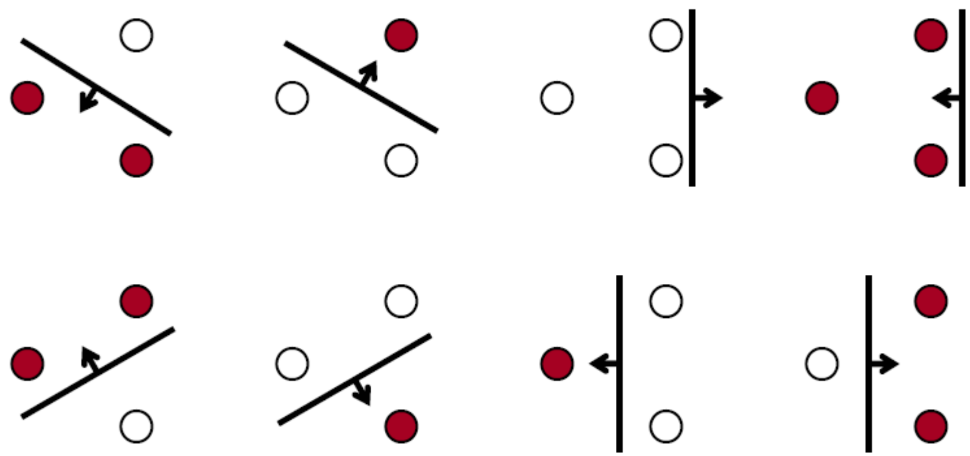
经验证明：简单的分类器 **(Low Complexity)** 推广能力强



Statistical learning theory (STL) - Vapnik & Chervonenkis (1960s-1990s)

VC维——函数复杂性的度量
Vapnik-Chervonenkis dimension

- The VC dimension VC(f) is the size of the largest dataset that can be shattered by the set of functions f(α)



- The VC dimension of the set of oriented lines in R2 is three
- VC dimension of the family of oriented separating hyperplanes in R^D is at least D+1

VC维的作用：给出期望风险的界

VC dimension provides bounds on the expected risk

VC dimension provides bounds on the expected risk as a function of the empirical risk and the number of available examples. It can be shown that, with probability 1-η, the following bound holds

$$R(f) \le R_{emp}(f) + \underbrace{\sqrt{\frac{h(\ln(2N/h)+1)-\ln(\eta/4)}{N}}}_{\text{VC confidence}}$$

$$\Downarrow$$

$$\boxed{R(f) \le R_{emp}(f) + \Phi(\frac{N}{h})}$$

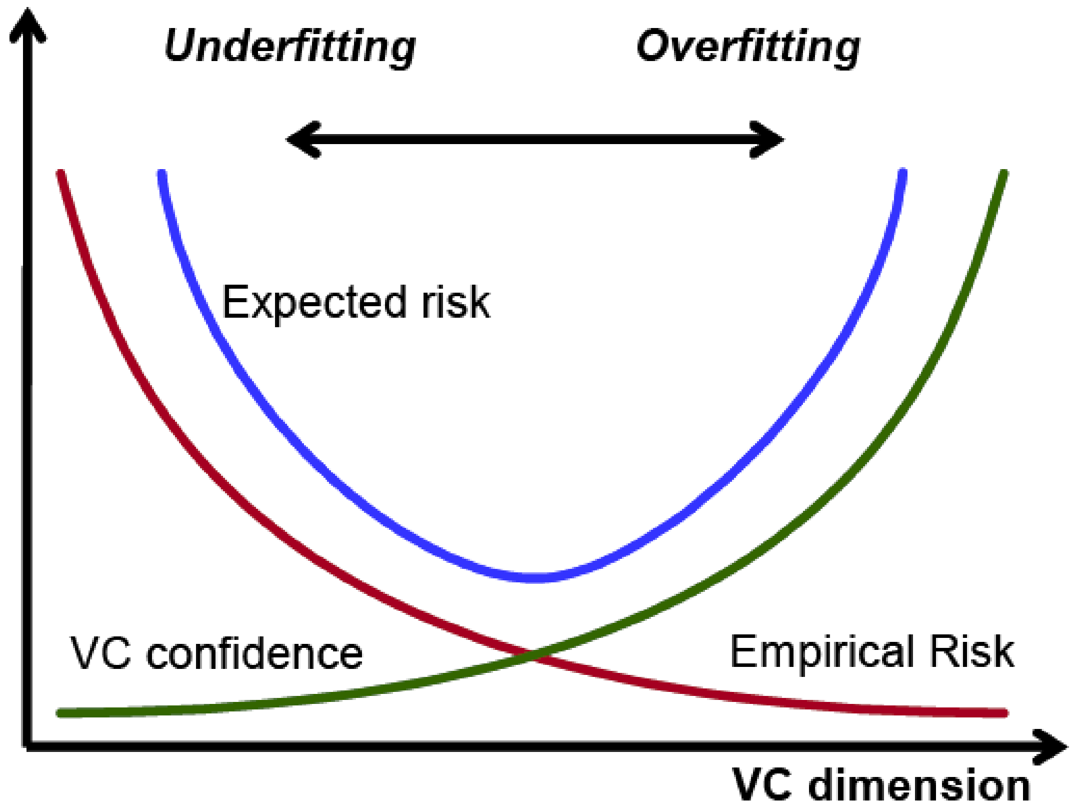
N为样本数

h为VC维

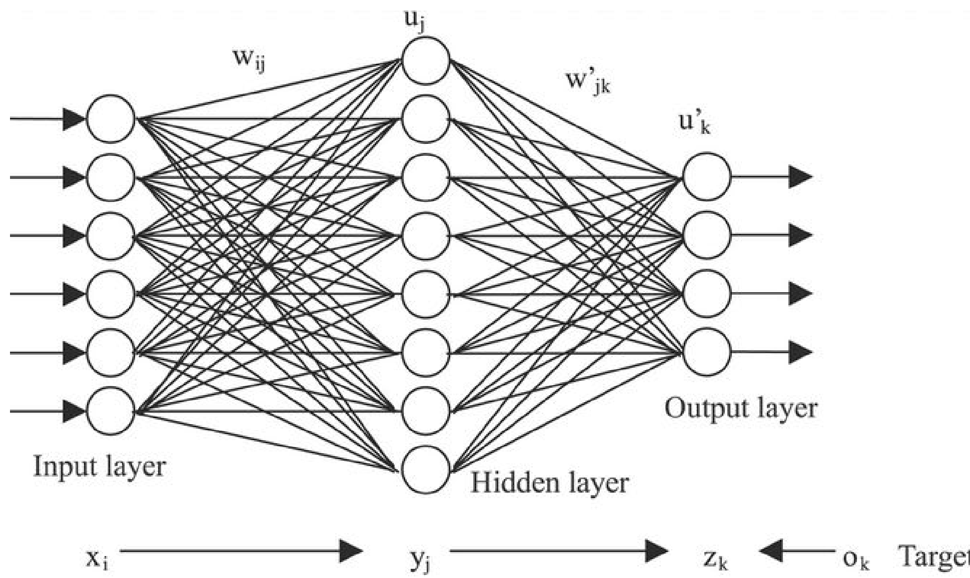
Φ是单调递减函数

- 为了提高推广能力，样本数**N**应该尽量大，而分类器应该尽量简单（**VC**维小）
- 经验风险最小化和**VC**维之间存在矛盾，$\mathbf{R_{emp}}$减小了，**VC**维可能会增大

基本矛盾和平衡策略



Underfitting
Overfitting
Expected risk
VC confidence
Empirical Risk
VC dimension

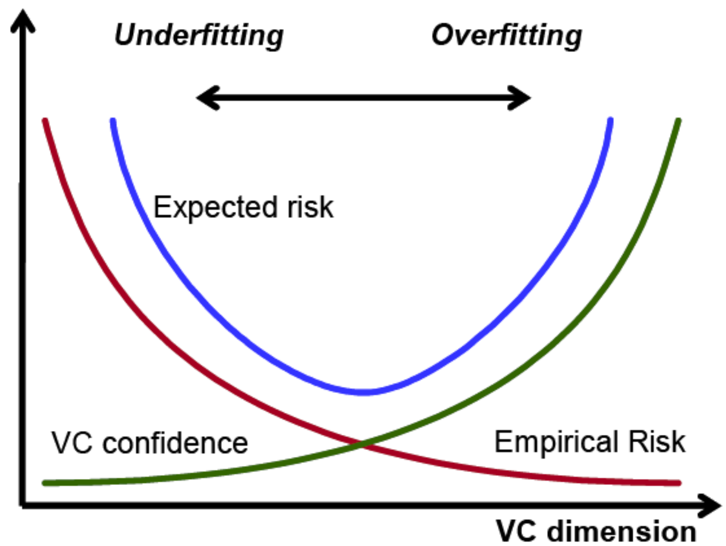


为什么人工神经网络的推广能力（泛化能力）差？

$$R(f) \le R_{emp}(f) + \Phi(\frac{N}{h})$$

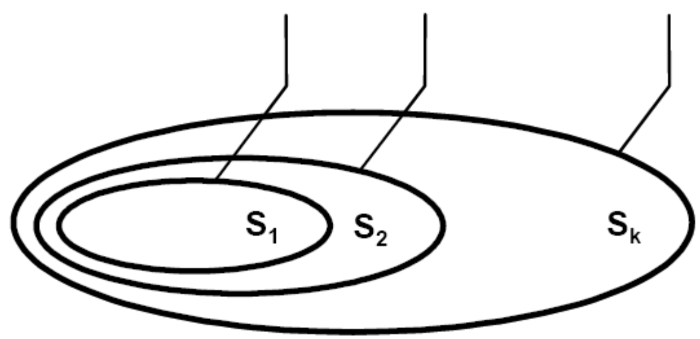
新思路：结构风险最小化
Structural Risk Minimization

- **Construct a nested *structure* for family of function classes $F_1 \subset F_2 \subset \ldots \subset F_k$ with non-decreasing VC dimensions $(h_1 \le h_2 \le \ldots \le h_k)$**
- **For each class F_i, compute the solution f_i that minimizes the empirical risk**
- **Choose the function class F_i, and the corresponding solution f_i, that minimizes the risk bound**



实际使用SRM原则设计分类器的两个步骤

- 首先选择一个具有最优分类能力的函数子集（VC维较小）
- 然后从这个子集中选择一个使经验风险最小的判别函数

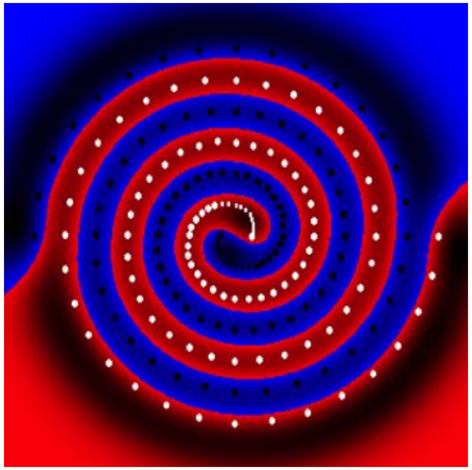


统计学习理论是支持向量机的理论基础

- 基于经验风险最小化的机器学习存在学习函数复杂性和推广能力的矛盾
- 基于结构风险最小化的统计学习理论给出了学习函数推广能力的上界
- VC维是函数推广能力的指标，较小的VC维具有较好的推广能力

传说中的支持向量机

Support Vector Machine

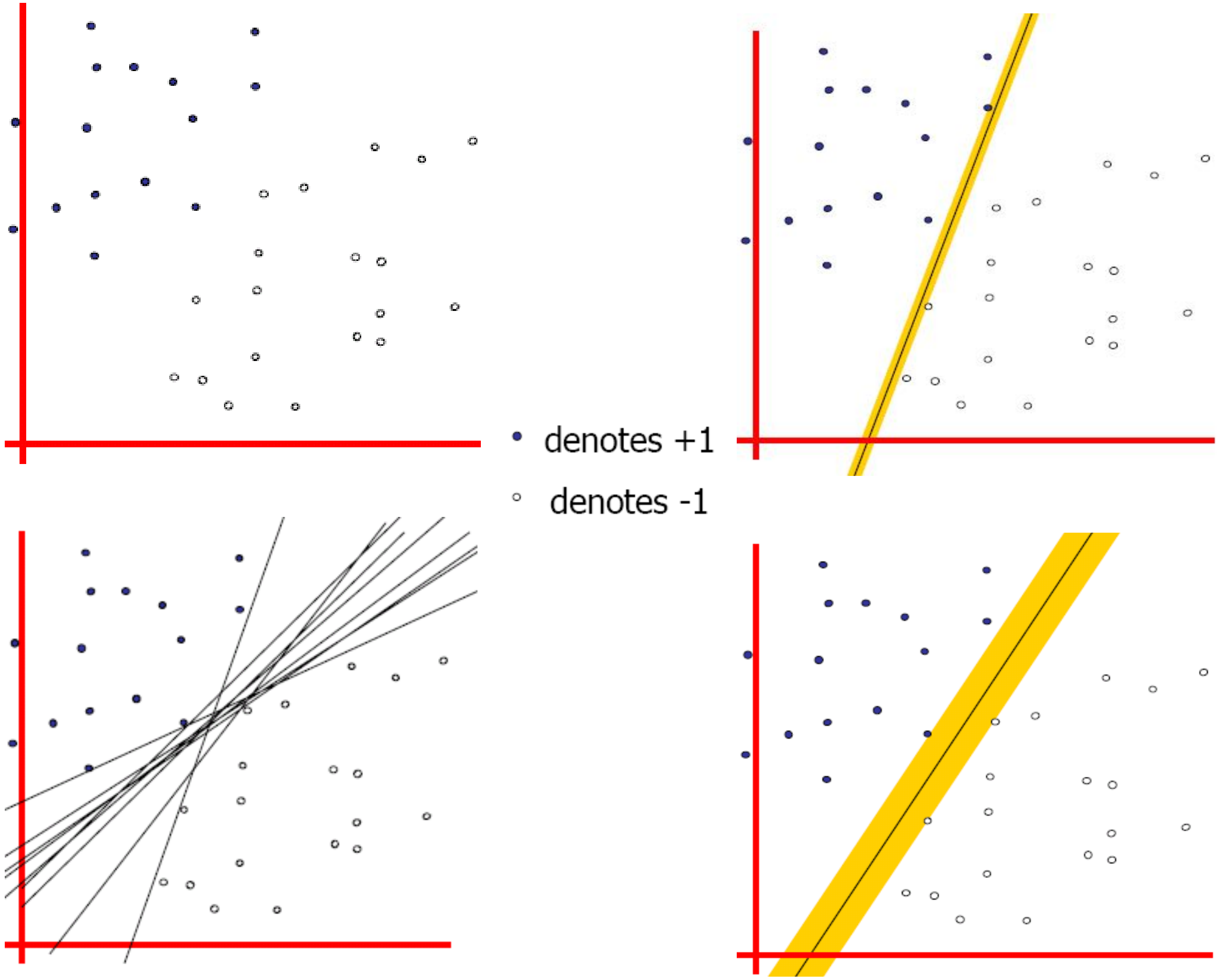


- 秘笈1：最大间隔线性分类器 （SLT核心理念）
- 秘笈2：转化为对偶优化问题
- 秘笈3：利用核函数升维

秘笈1　最大间隔线性分类器

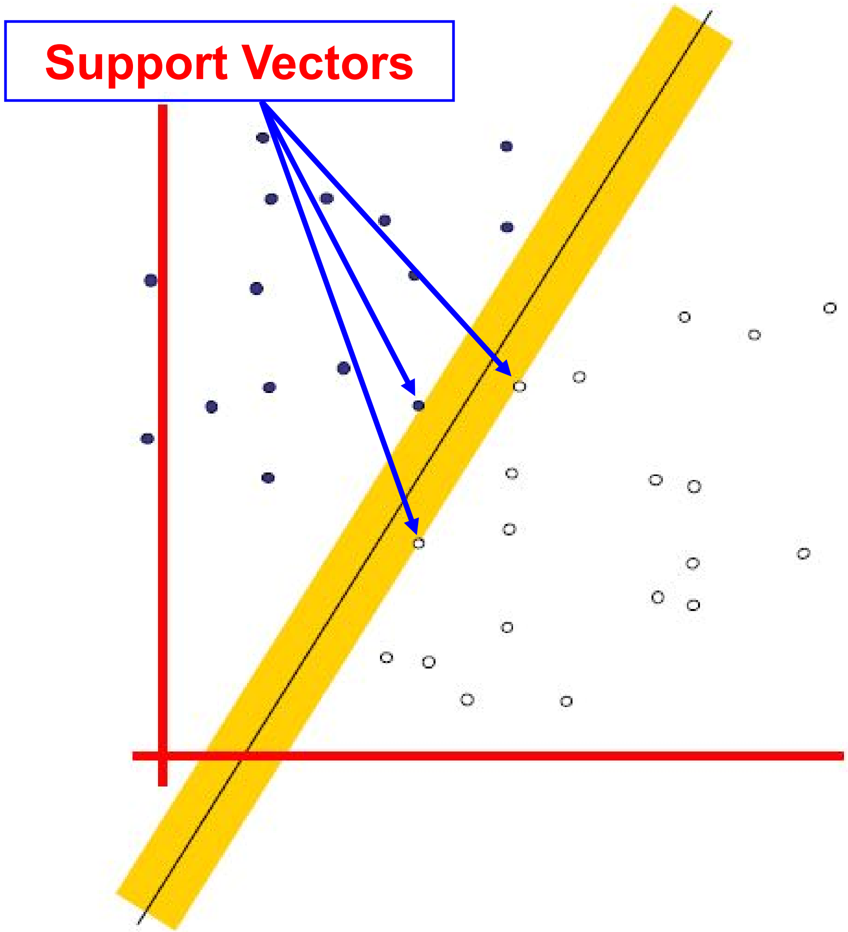
Maximum margin linear classifier

Linear Classifiers —— Which is the best?



最大间隔线性分类器

Maximum margin linear classifier



Why is the **maximum margin linear classifier** the best?

利用统计学习理论可以证明：

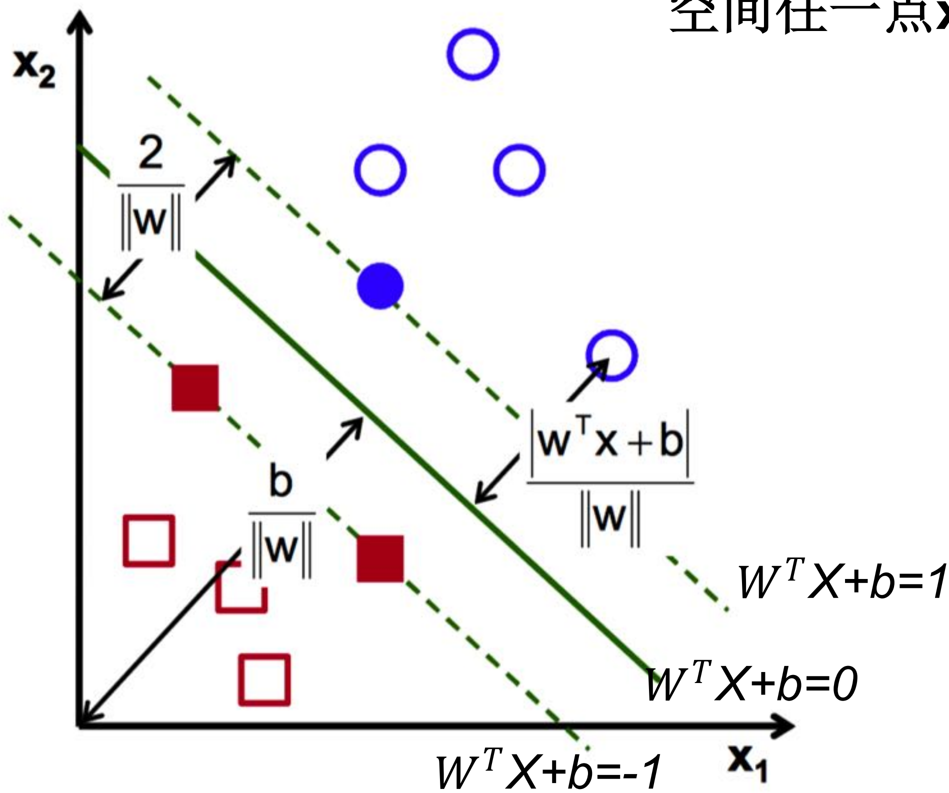
间隔为**m**的超平面的**VC**维的上界是

$$h \le \min\left(\left\lceil \frac{R^2}{m^2} \right\rceil, D\right) + 1$$

其中，**D**是样本空间的维数，**R**是包含所有样本的超球的半径

所以，间隔越大，**VC**维越小，相应的推广能力越强

如何计算间隔Margin？



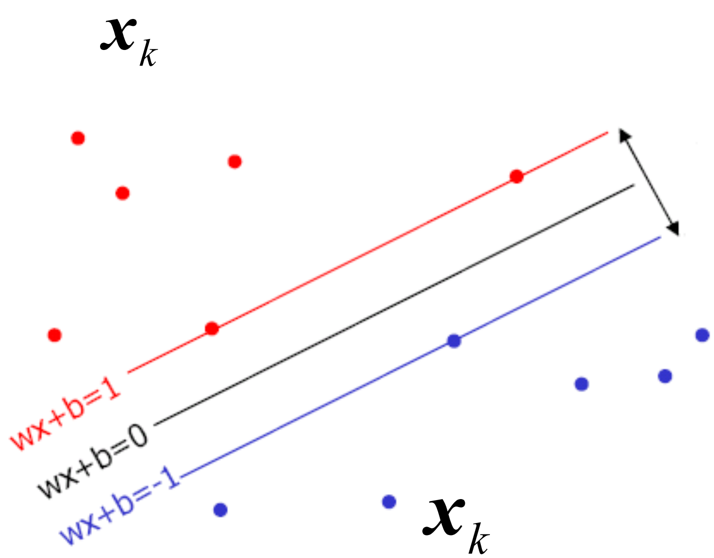
空间任一点**x**到超平面**w^{T}x+b=0**的距离

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

间隔Margin

$$d = \frac{2}{\|W\|} = \frac{2}{\sqrt{W \cdot W}}$$

优化问题的表述



每个分类样本表示为

(x_k, y_k) where $y_k = +/- 1$

最大化 $d = \frac{2}{\sqrt{W \cdot W}}$

约束条件

$w \,.\, x_k + b >= 1$ if $y_k = 1$

$w \,.\, x_k + b <= -1$ if $y_k = -1$

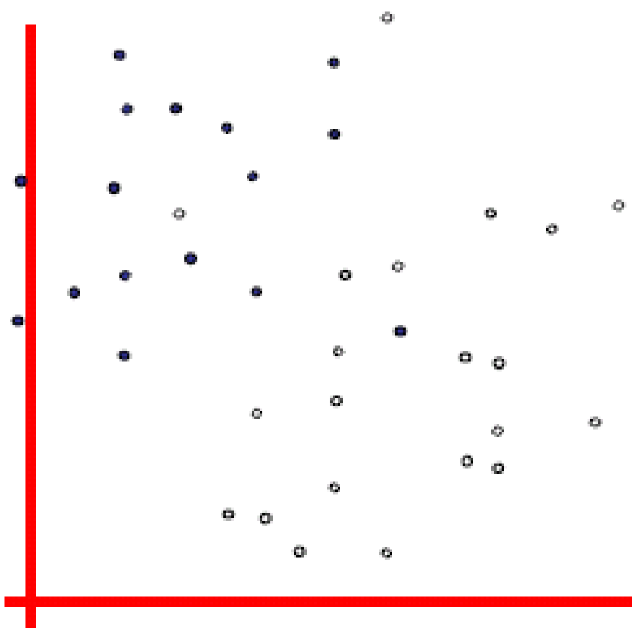
优化问题A（线性可分情况）

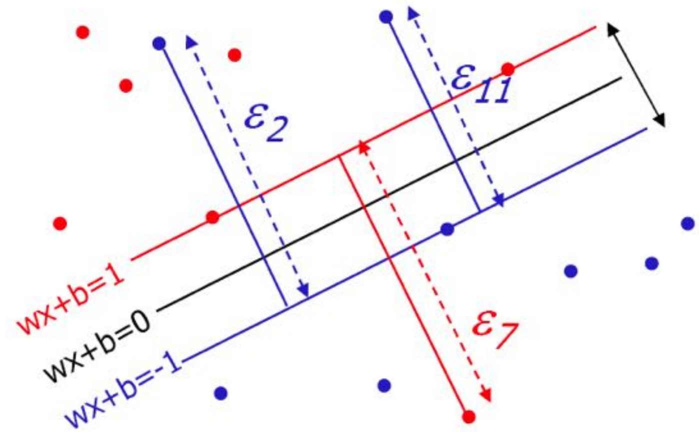
最小化 $\frac{1}{2} W \cdot W$

约束条件

$$y_i \cdot [(W \cdot x_i) + b] - 1 \geq 0$$

线性不可分的情况





$\mathbf{w} \cdot \mathbf{x}_k + b >= 1-\varepsilon_k$ if $y_k = 1$

$\mathbf{w} \cdot \mathbf{x}_k + b <= -1+\varepsilon_k$ if $y_k = -1$

$\varepsilon_k >= 0$ for all k

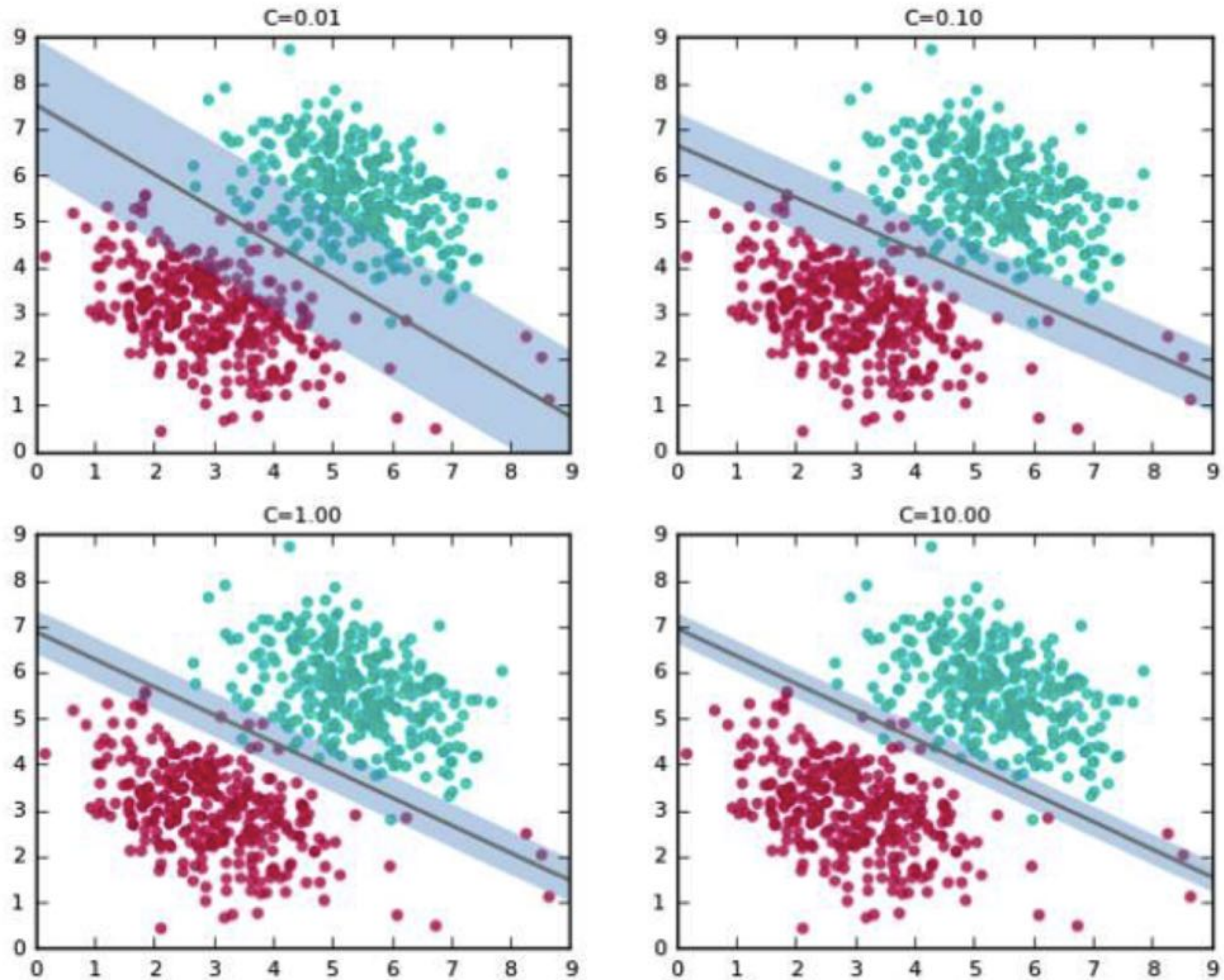
优化问题***B***

最小化 $\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$ 约束条件 $y_i \cdot [(\boldsymbol{W} \cdot \boldsymbol{x}_i) + b] - 1 + \varepsilon_k \geq 0$

训练集错误率和推广能力的平衡参数

Tradeoff between C and Margin

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$



秘笈2 对偶优化问题

Dual problem of optimization

二次型寻优的对偶问题

优化问题*A*

最小化 $\frac{1}{2}\boldsymbol{W}\cdot\boldsymbol{W}$

约束条件 $y_i\cdot[(\boldsymbol{W}\cdot\boldsymbol{x}_i)+b]-1\geq 0$

$$\max_{\alpha_i\geq 0}\mathcal{L}(w,b,\alpha)=\max_{\alpha_i\geq 0}\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i\left(y_i(w^Tx_i+b)-1\right)$$

拉格朗日乘子 Lagrange multiplier

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

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i \qquad \frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$$

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

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

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

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

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

http://blog.pluskid.org/?p=682

对偶优化问题

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

优化问题A $\quad s.t. \quad \sum_{i=1}^{N} y_i \quad \alpha_i = 0, \qquad \alpha_i \geq 0 \qquad i = 1, \cdots, N$

优化问题B $\quad s.t. \quad \sum_{i=1}^{N} y_i \quad \alpha_i = 0, \qquad 0 \leq \alpha_i \leq C \qquad i = 1, \cdots, N$

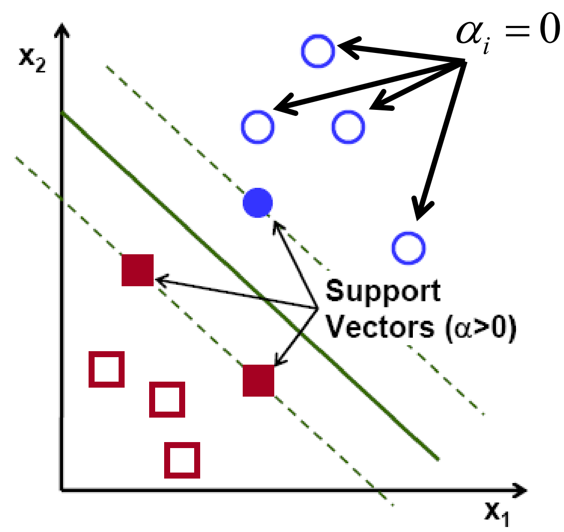
由最优的 α_i^* 得到最优的 和 $\boldsymbol{w}^*$ $\boldsymbol{b}^*$

$$\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i$$

$$y = \operatorname{sgn}(\mathbf{w}^* \mathbf{x} + \mathbf{b}^*) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \boxed{(\mathbf{x}_i \cdot \mathbf{x})} + b^* \right)$$

"支持向量"如何脱颖而出

$$\max_{\alpha_i \geq 0} \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(w^T x_i + b) - 1\right)$$



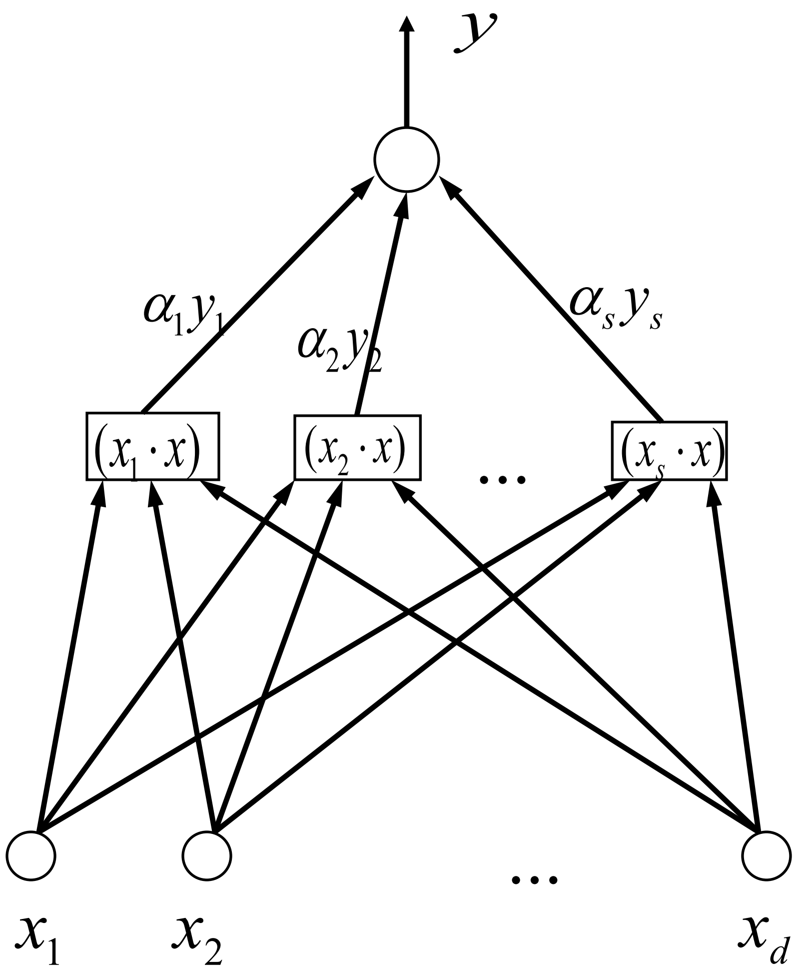
$\alpha_i = 0$ 上式括号中的值大于零，所以对应的是大多数在Margin以外的样本点

$\alpha_i > 0$ 上式中括号中的值必然等于零，所以对应的是Margin边界上的点

$$y = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^*\right) = \operatorname{sgn}\left(\sum_{i=1}^{S} \alpha_i y_i (\underline{\mathbf{x}}_i \cdot \mathbf{x}) + b^*\right)$$

在最终的判别函数式中只有S个支持向量起作用，而不是所有的N个样本

SVM的实现形式（线性SVM）

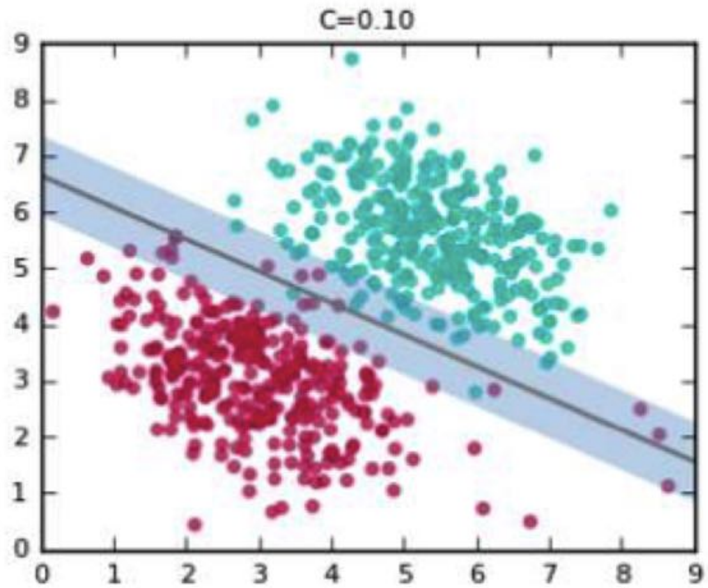


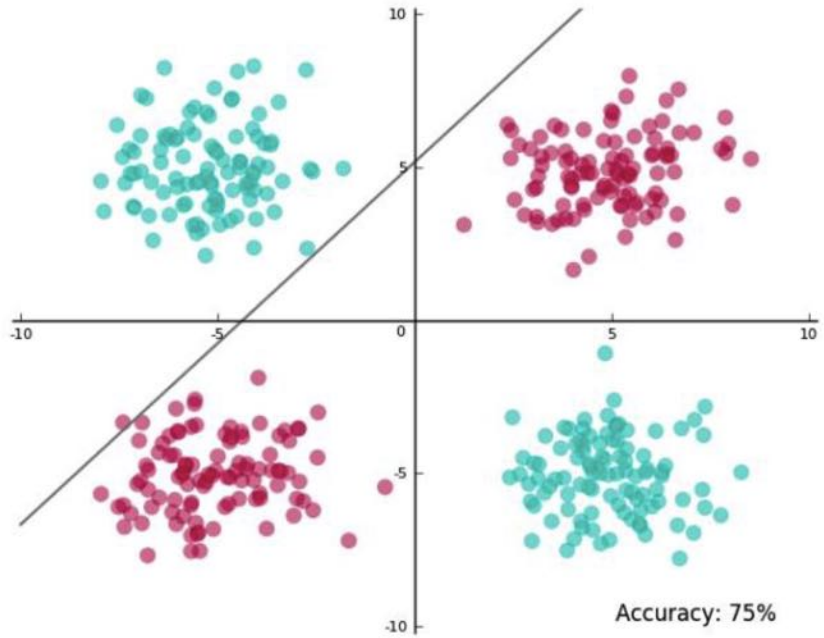
$$y = \mathrm{sgn}\left(\sum_{i=1}^{s} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \right)$$

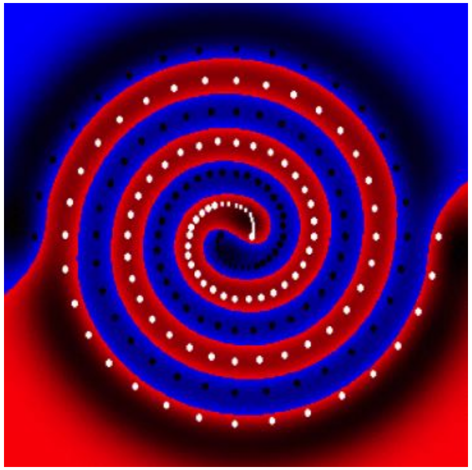
权值 $w_i = \alpha_i y_i$

基于s个支持向量的**线性变换**（内积）

输入向量 $x = (x_1, x_2, \ldots, x_d)$


C=0.10
0
1
2
3
4
5
6
7
8
9

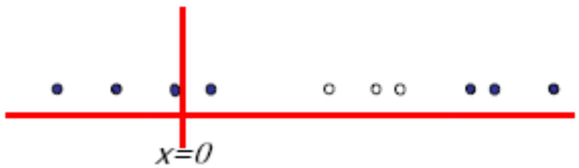

10
5
-10
-5
0
5
10
-5
-10
Accuracy: 75%

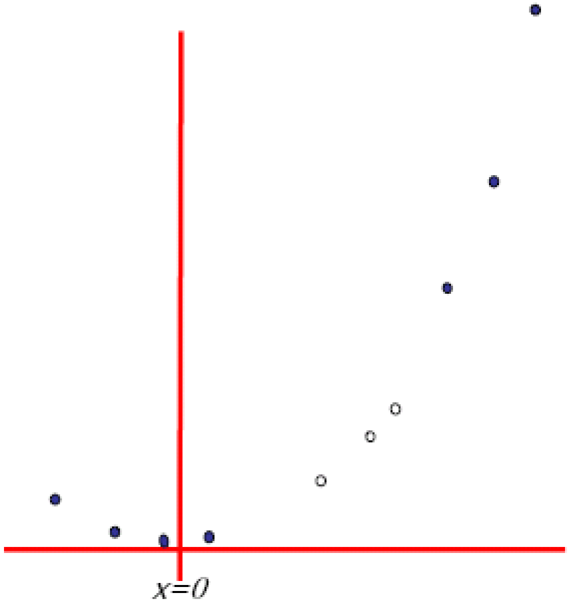


秘笈3：通过核函数升维

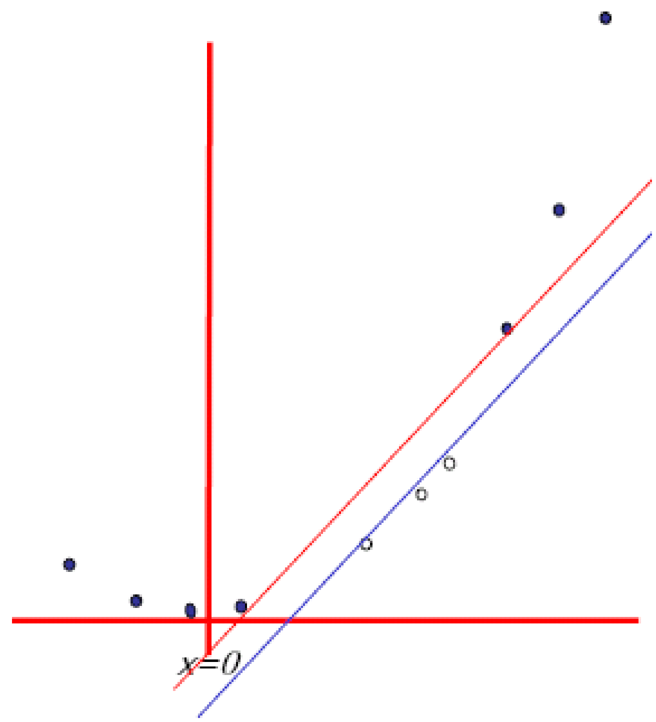
Kernel trick

通过恰当升维解决线性不可分的问题



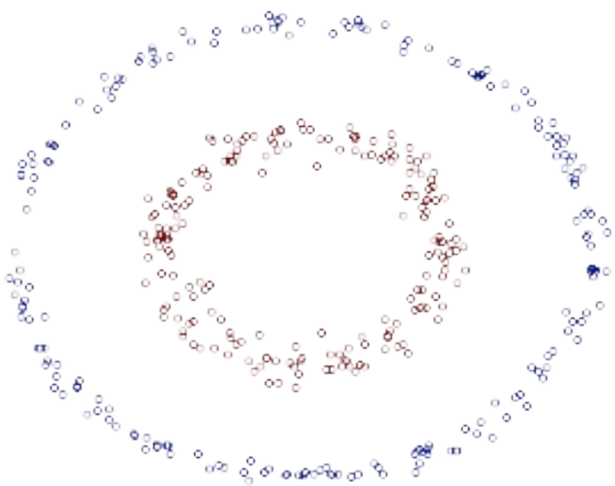


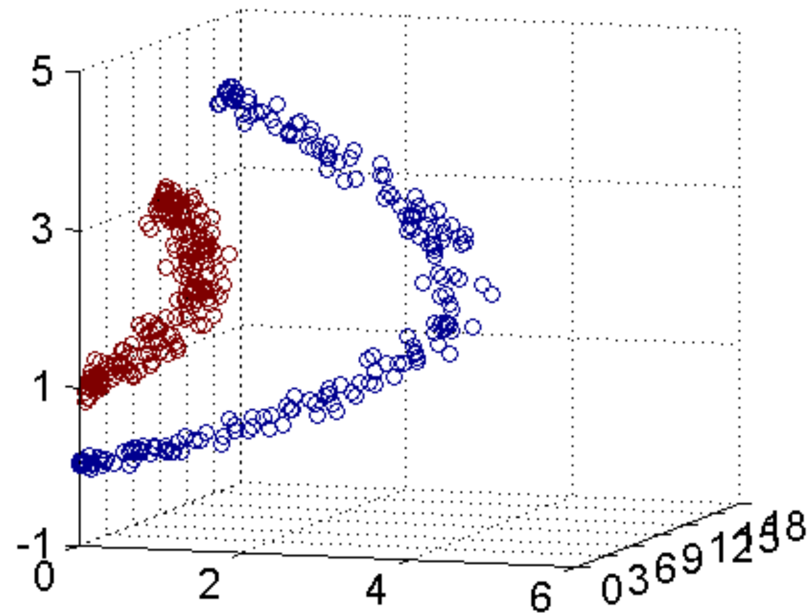
$$\mathbf{z}_k = (x_k, x_k^2)$$



$$\varphi : \mathrm{R}^2 \rightarrow \mathrm{R}^3$$

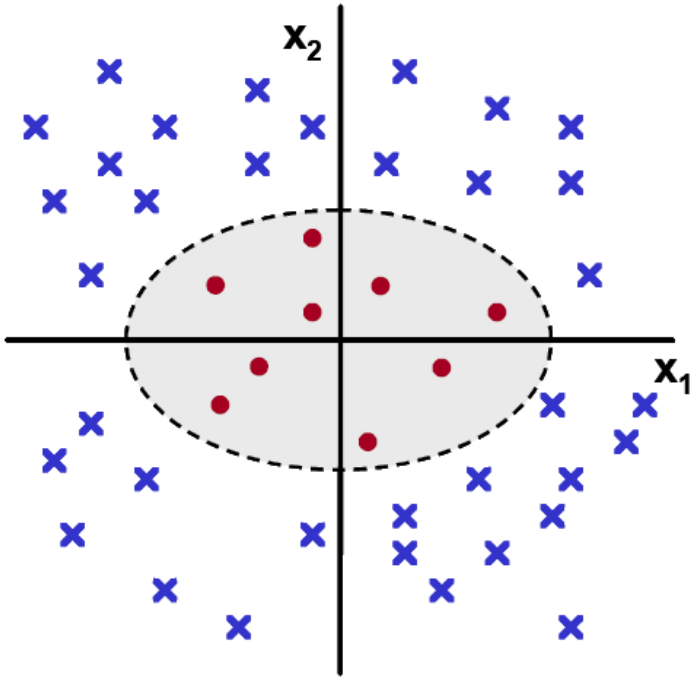
$$Z_1 = X_1^2, Z_2 = X_2^2, Z_3 = X_2$$

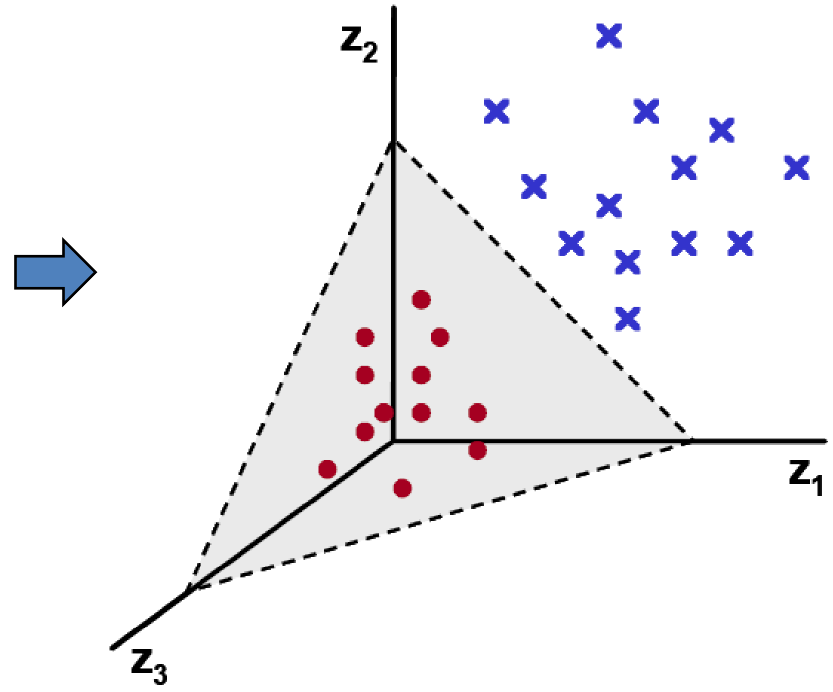


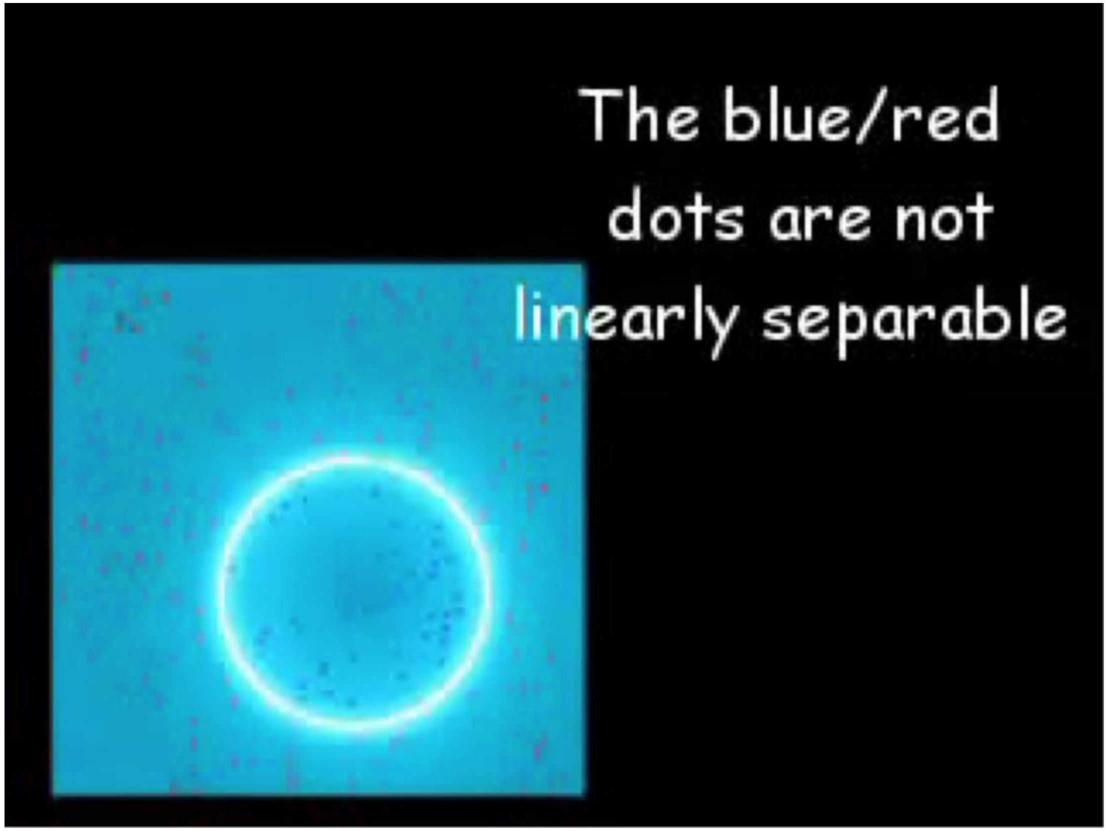


$$\varphi : \mathrm{R}^2 \to \mathrm{R}^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$







Cover's theorem *"A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space"*

盗梦空间：高维非欧空间？

http://www.360doc.com/content/10/1126/19/2119143_72692056.shtml

升维带来的问题及SVM的解决方案

- **Statistical problem 维数灾难**: operation on high-dimensional spaces is ill-conditioned due to the “curse of dimensionality” and the subsequent risk of overfitting
  - Generalization capabilities in the high-dimensional manifold are ensured by enforcing a **largest margin** classifier （在高维空间采用最大间隔分类器）

- **Computational problem 计算复杂**: working in high-dimensions requires higher computational power, which poses limits on the size of the problems that can be tackled
  - projection onto a high-dimensional manifold is only **implicit** （采用特殊的核函数，使得升维变换不增加内积计算复杂度）

通过核函数升维

原优化问题B

$$\max_{\alpha} \quad \tfrac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{R} y_i y_j \alpha_i \alpha_j \left(\mathbf{x}_i \mathbf{x}_j\right) - \sum_{j=1}^{R} \alpha_j$$

一般变换后

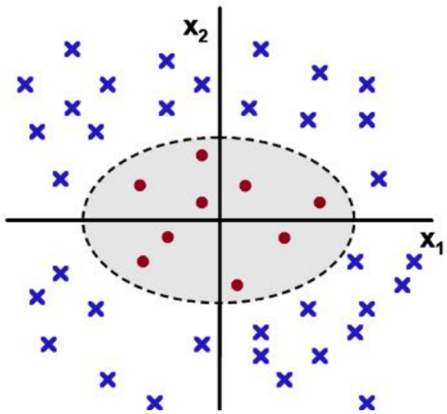
$$\min_{\alpha} \quad \tfrac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{R} y_i y_j \alpha_i \alpha_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) - \sum_{j=1}^{R} \alpha_j$$

核函数变换后

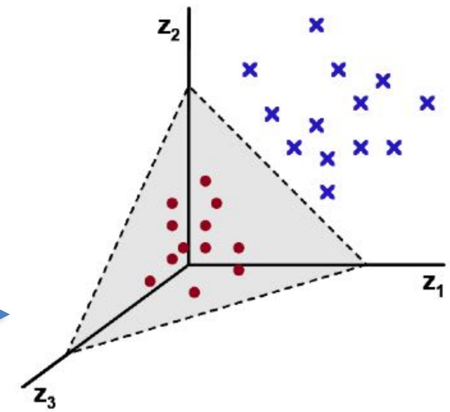
核函数
Kernel function

$$\min_{\alpha} \quad \tfrac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{R} y_i y_j \alpha_i \alpha_j K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) - \sum_{j=1}^{R} \alpha_j$$

核函数技巧 Kernel Trick



$$\phi(\mathbf{x}) = \phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} x_1{}^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2{}^2 \end{pmatrix}$$



$$\phi(\mathbf{a})^T \cdot \phi(\mathbf{b}) = \begin{pmatrix} a_1{}^2 \\ \sqrt{2}\, a_1 a_2 \\ a_2{}^2 \end{pmatrix}^T \cdot \begin{pmatrix} b_1{}^2 \\ \sqrt{2}\, b_1 b_2 \\ b_2{}^2 \end{pmatrix} = a_1{}^2 b_1{}^2 + 2 a_1 b_1 a_2 b_2 + a_2{}^2 b_2{}^2$$

$$= (a_1 b_1 + a_2 b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}\right)^2 = \left(\mathbf{a}^T \cdot \mathbf{b}\right)^2$$

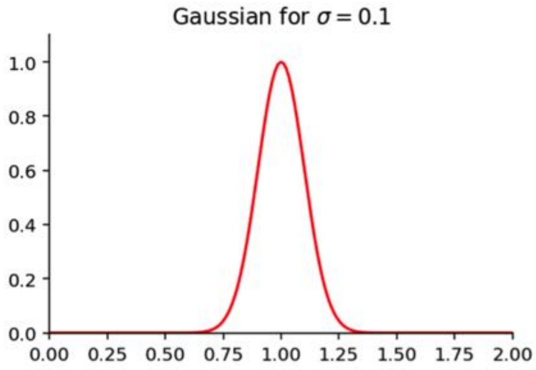
$$\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) \longrightarrow \boxed{K(x_i, x_j) = (x_i{}^T \cdot x_j)^2}$$

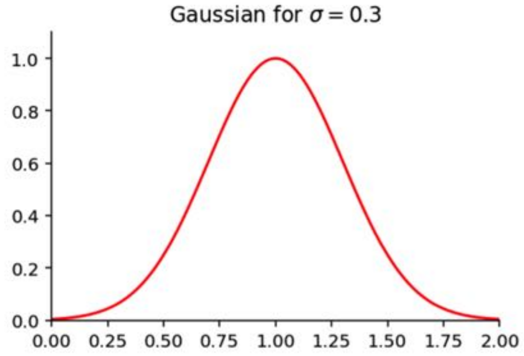
SVM常用核函数

线性(Linear) $K(\boldsymbol{x},\boldsymbol{x}_i)=\boldsymbol{x}\cdot\boldsymbol{x}_i$

多项式(Polynomial) $K(\boldsymbol{x},\boldsymbol{x}_i)=[(\boldsymbol{x}\cdot\boldsymbol{x}_i)+1]^q$

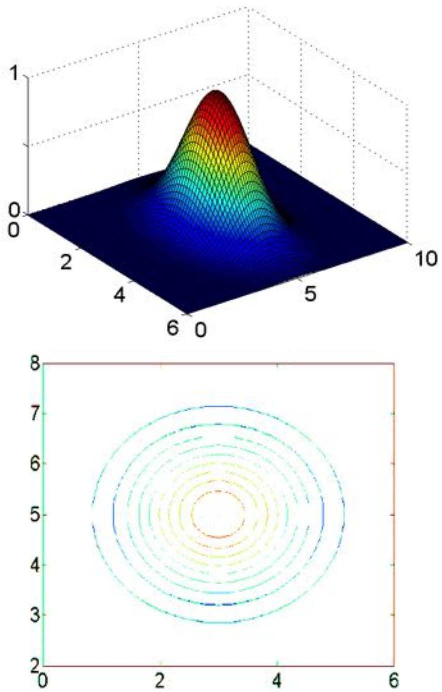
高斯核函数
径向基函数(RBF) $K(\boldsymbol{x},\boldsymbol{x}_i)=\exp\left\{-\frac{|\boldsymbol{x}-\boldsymbol{x}_i|^2}{\sigma^2}\right\}$



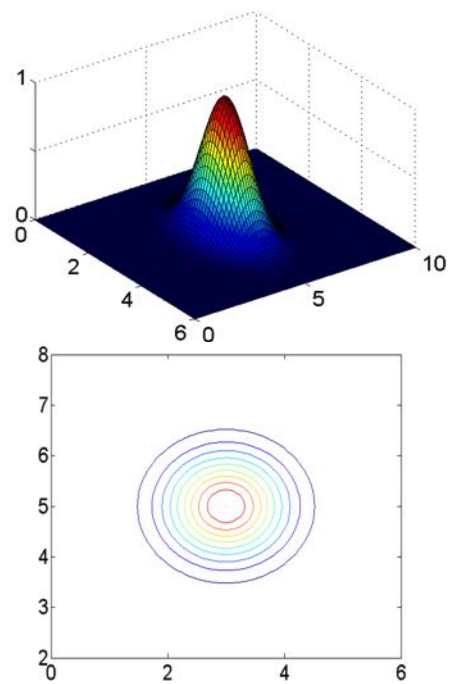


$$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

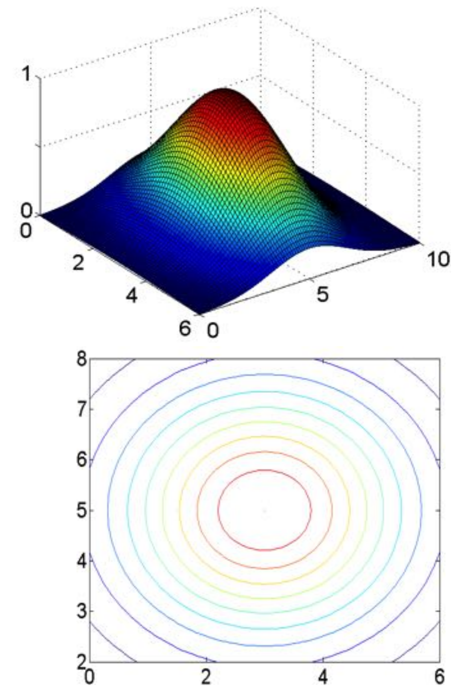
$\sigma^2 = 1$



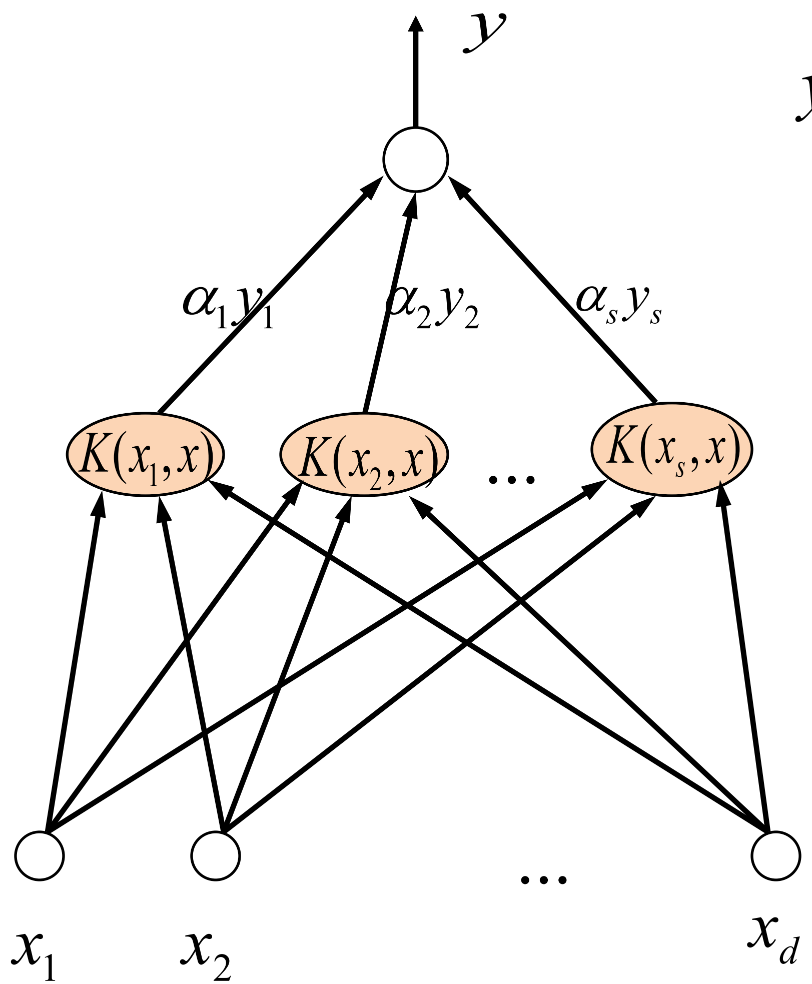
$\sigma^2 = 0.5$



$\sigma^2 = 3$



SVM的实现形式（非线性核函数K）



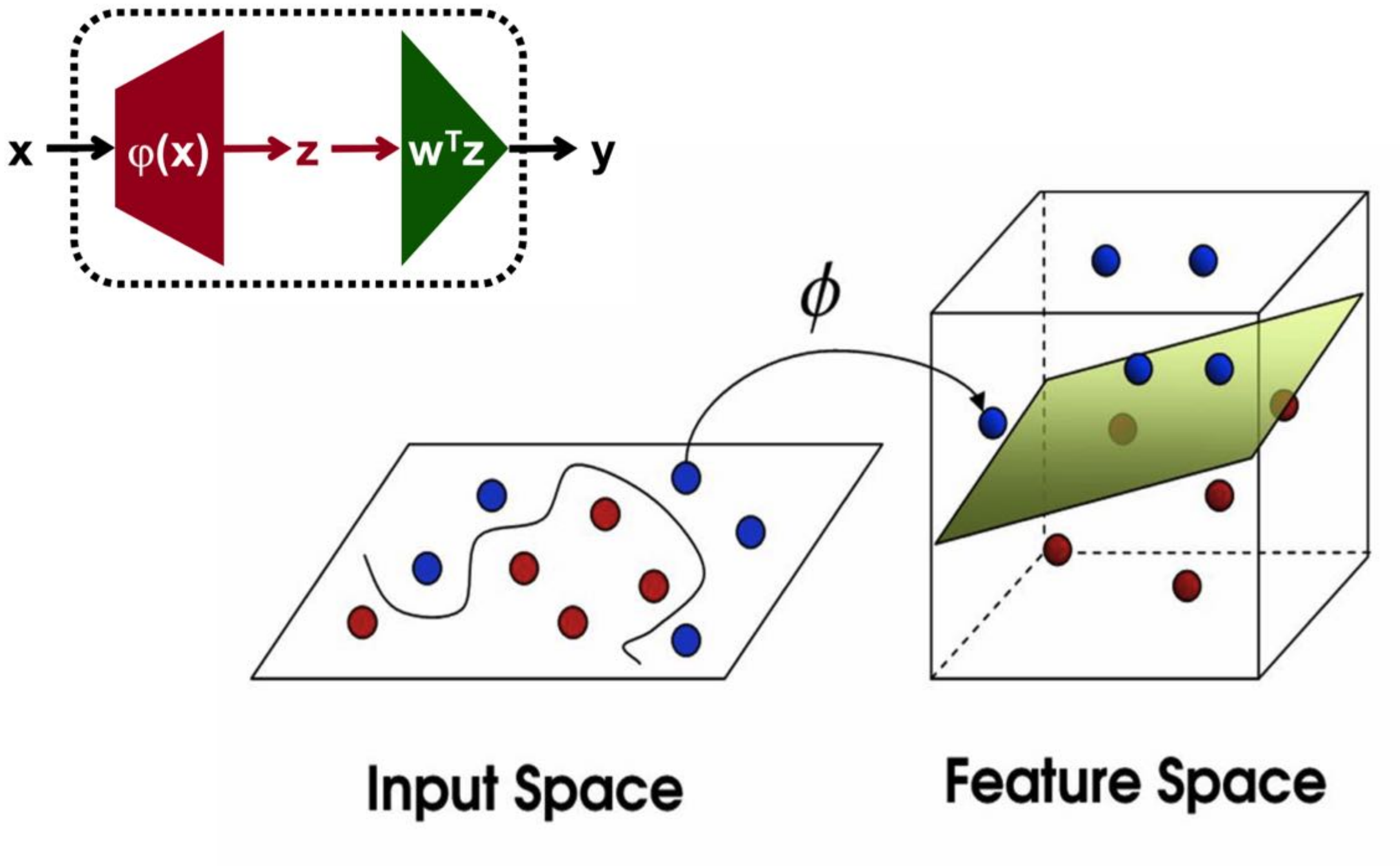
$$y = \text{sgn}\left(\sum_{i=1}^{s} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right)$$

权值 $w_i = \alpha_i y_i$

基于s个支持向量的**非线性变换**（内积）

输入向量 $x = (x_1, x_2, \ldots, x_d)$

SVM is a hyperplane in hyperspace



高维变换空间最大间隔分类器的推广能力

VC维

$$h \le \min\left(\left\lceil \frac{R^2}{m^2} \right\rceil, D\right) + 1$$

测试样本错误率的期望

$$E[P(error)] \le \frac{E[\text{支持向量个数}]}{\text{训练样本总数-1}}$$

- 支持向量机的泛化能力与变换空间的维数无关，只要能够适当选择一种核函数，构造一个**支持向量数较少**的最大间隔分类面，就可以得到较好的泛化能力

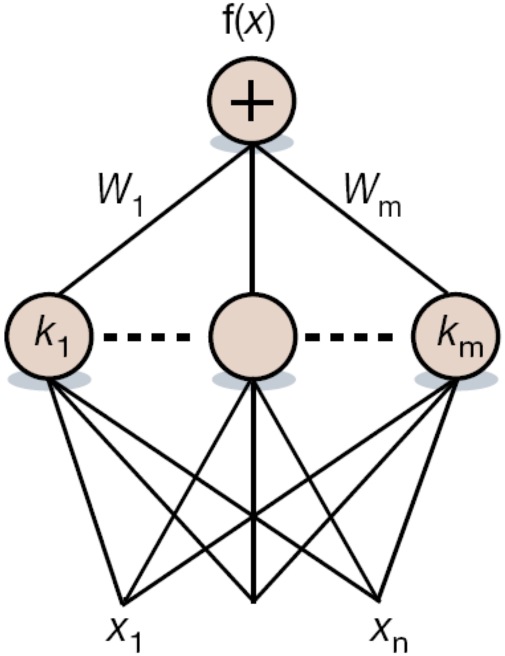
支持向量机的特点

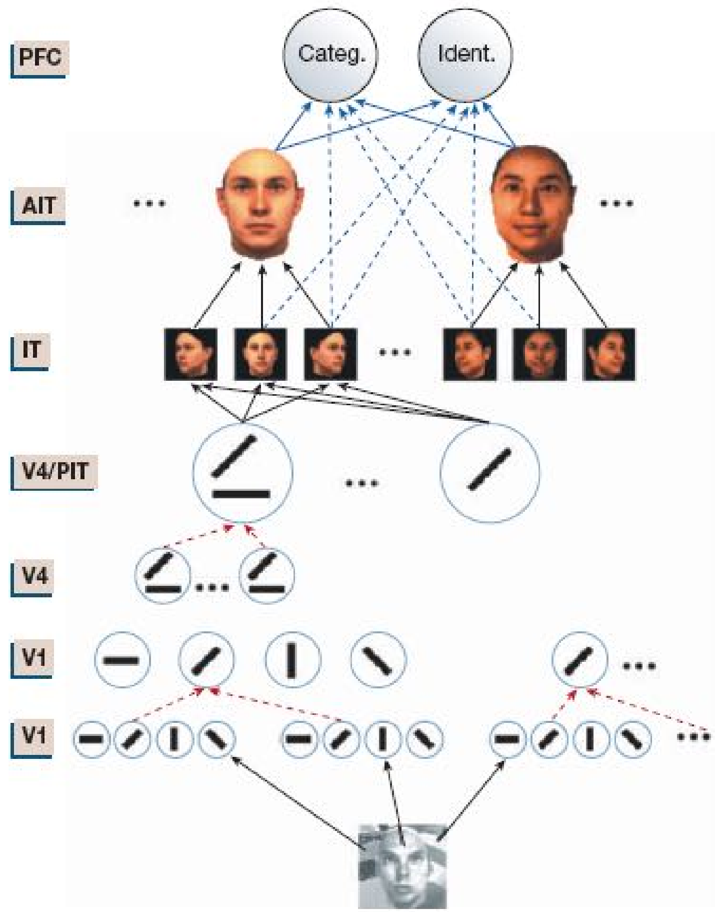
- **最大间隔线性分类器**是专门针对有限样本情况的，其目标是找到推广能力最强的分类器参数
- 算法最终将转化成为一个**二次型寻优问题**，从理论上说，得到的将是全局最优点，解决了在神经网络方法中无法避免的局部极值问题
- 算法将实际问题通过**非线性映射转换到高维特征空间**，在高维空间中构造线性判别函数来实现原空间中的非线性判别函数，同时它通过**内积不变核函数**巧妙地解决了维数灾难问题，其算法复杂度与样本维数无关

思考与讨论

SVM is probably a brain-like learning machine

$$f(\bar{x}) = \sum_{i}^{n} w_i k(\bar{x}, \bar{x}_i)$$



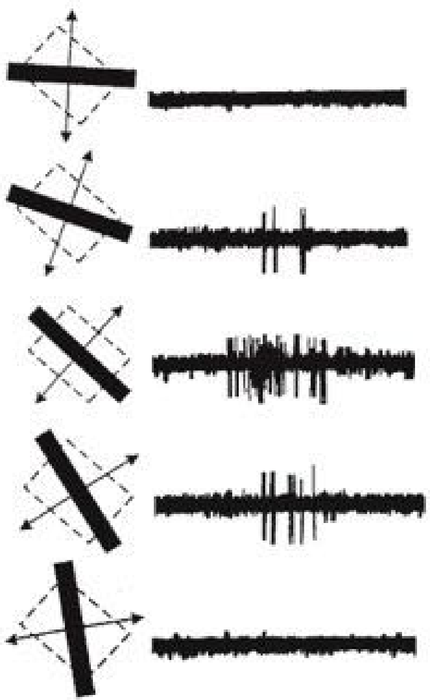


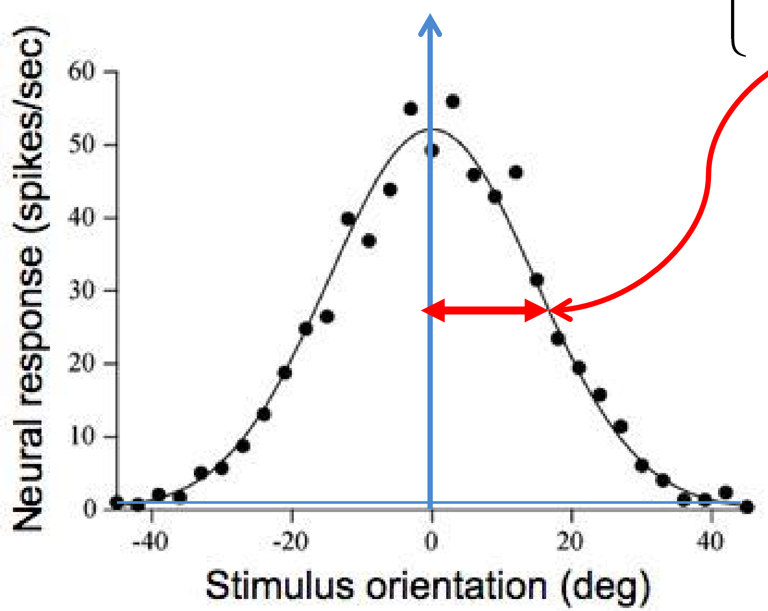
[Poggio 2004, Nature](#)

Generalization in vision and motor control

神经细胞的调谐曲线与高斯核函数
Tuning curve of a visual neuron and Gaussian Kernel

$$K(\boldsymbol{x}, \boldsymbol{x}_i) = \exp\left\{ -\frac{\left|\boldsymbol{x} - \boldsymbol{x}_i\right|^2}{\sigma^2} \right\}$$



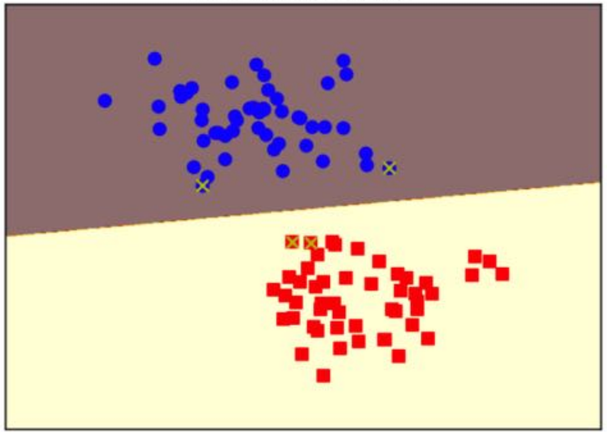


Hubel & Wiesel, 1968

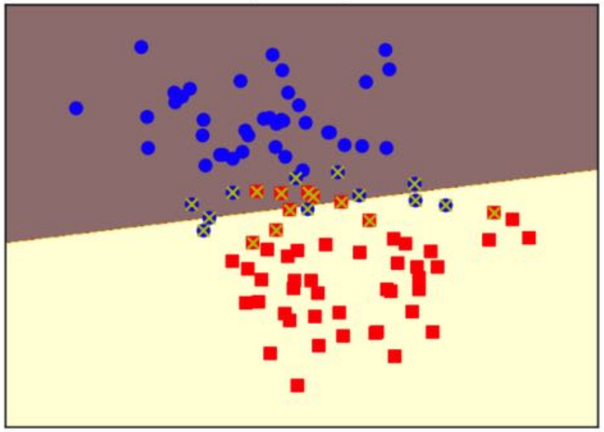
SVM in Scikit-learn

```
from sklearn import svm
from sklearn.datasets import make_blobs

X1, y1 = make_blobs(n_samples=100, centers=2,
                    random_state=0, cluster_std=0.6)
clf = svm.SVC(C=1.0, kernel='linear')
clf.fit(X1, y1)
plt.figure(figsize=(12,4),dpi=144)
plt.subplot(1, 2, 1)
plot_hyperplane(clf, X1, y1, h=0.01,
                title='Linearly separable')

X2, y2 = make_blobs(n_samples=100, centers=2,
                    random_state=0, cluster_std=1.1)
clf = svm.SVC(C=1.0, kernel='linear')
clf.fit(X2, y2)
plt.subplot(1, 2, 2)
plot_hyperplane(clf, X2, y2, h=0.01,
                title='Linearly nonseparable')
```

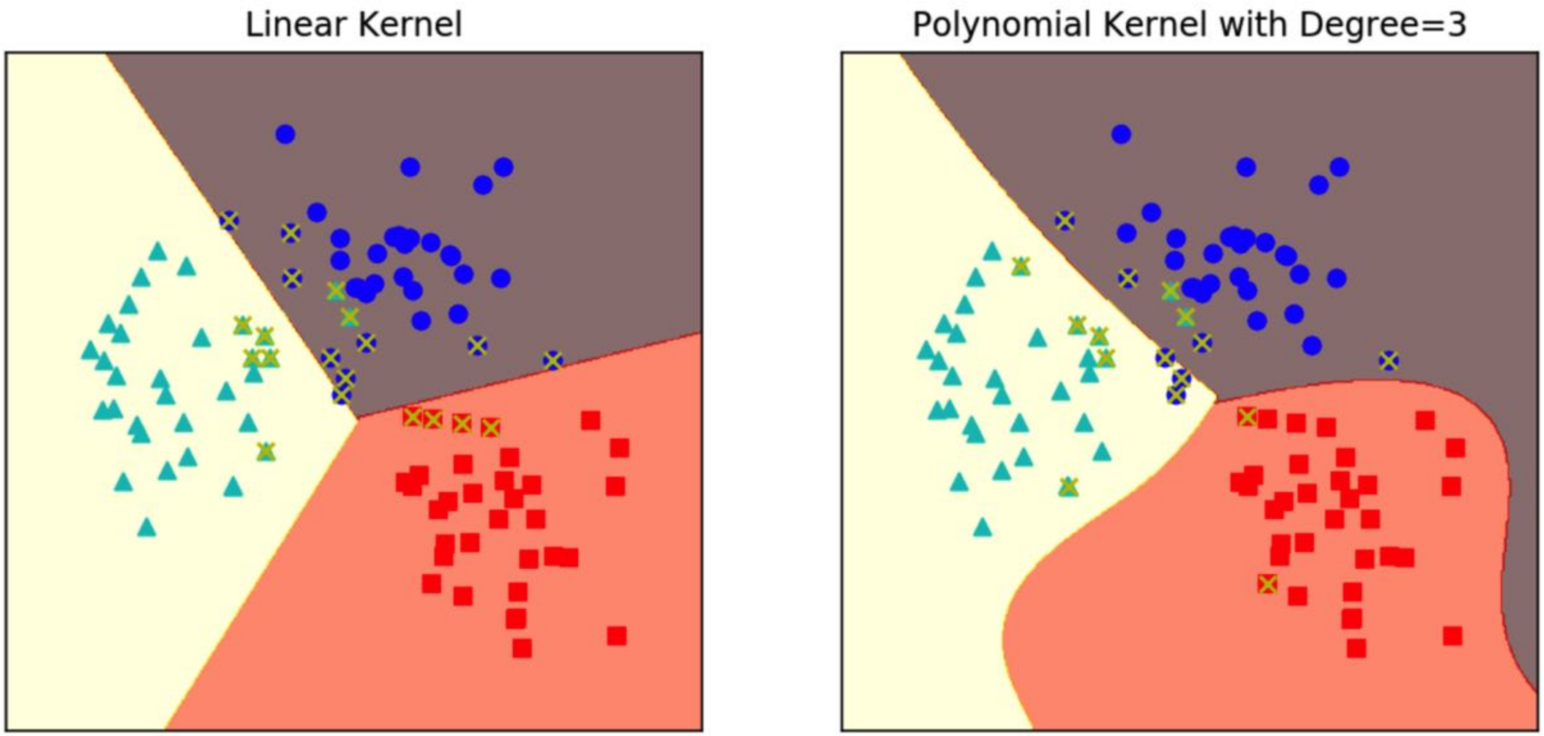
Linearly separable



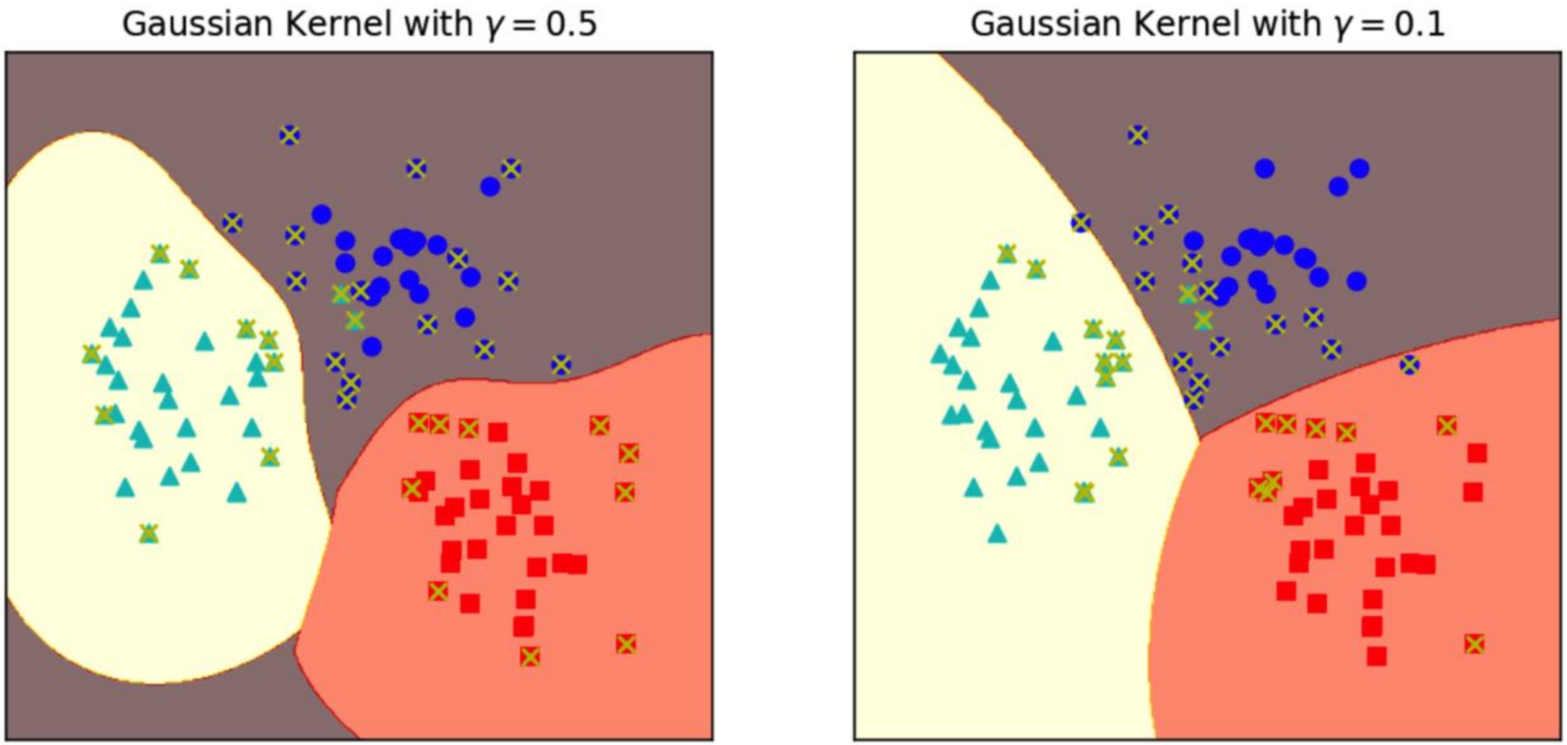
Linearly nonseparable



SVM in Scikit-learn



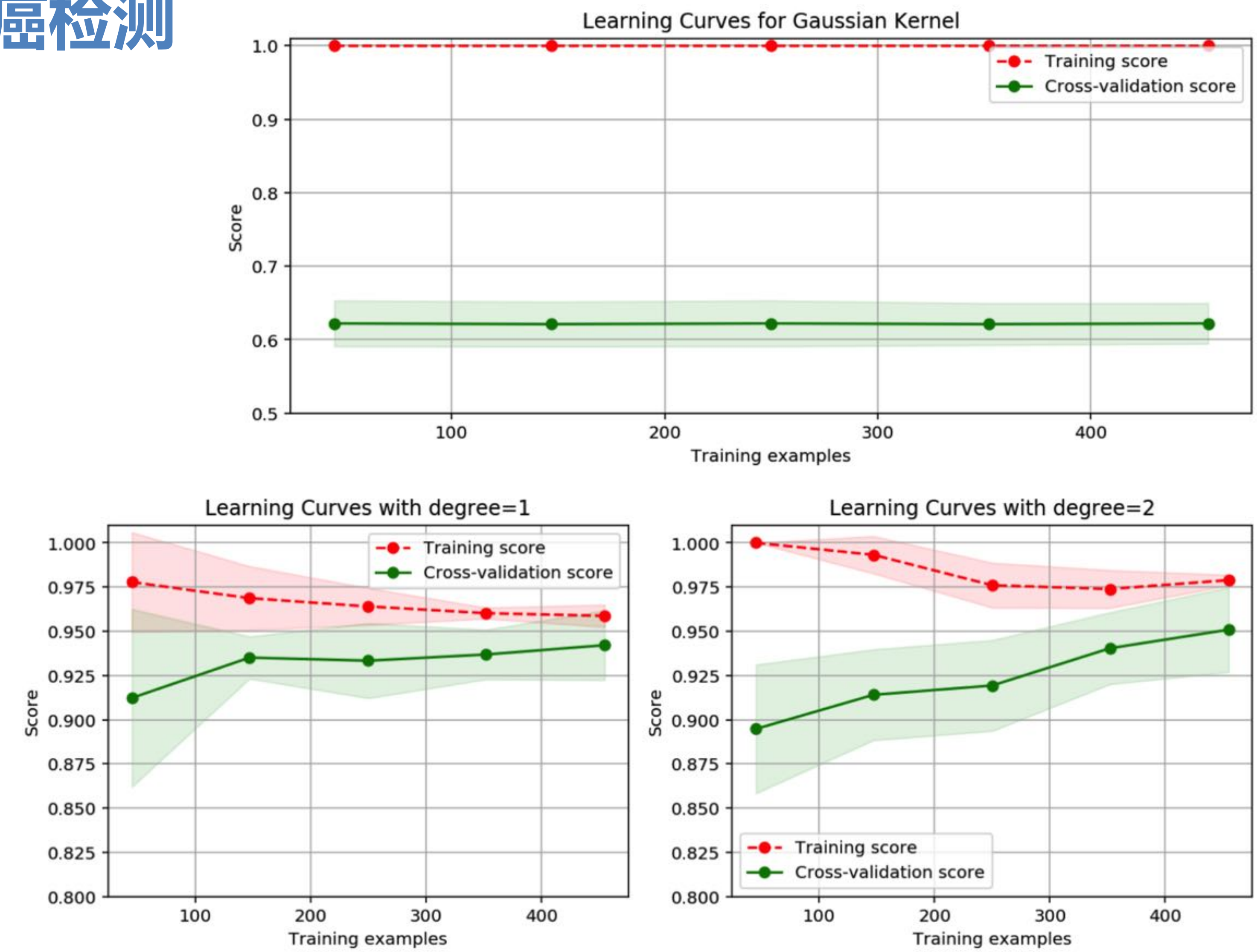
SVM in Scikit-learn



SVM in Scikit-learn

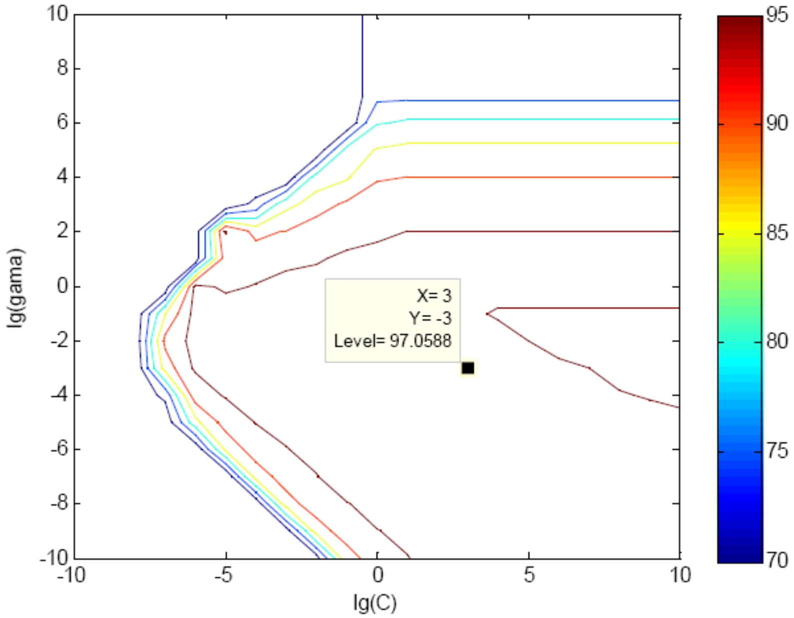
```
from sklearn import svm
from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples=100, centers=3,
                  random_state=0, cluster_std=0.8)
clf_linear = svm.SVC(C=1.0, kernel='linear')
clf_poly = svm.SVC(C=1.0, kernel='poly', degree=3)
clf_rbf = svm.SVC(C=1.0, kernel='rbf', gamma=0.5)
clf_rbf2 = svm.SVC(C=1.0, kernel='rbf', gamma=0.1)

plt.figure(figsize=(10, 10), dpi=144)

clfs = [clf_linear, clf_poly, clf_rbf, clf_rbf2]
titles = ['Linear Kernel',
          'Polynomial Kernel with Degree=3',
          'Gaussian Kernel with $\gamma=0.5$',
          'Gaussian Kernel with $\gamma=0.1$']
for clf, i in zip(clfs, list(range(len(clfs)))):
    clf.fit(X, y)
    plt.subplot(2, 2, i+1)
    plot_hyperplane(clf, X, y, title=titles[i])
```

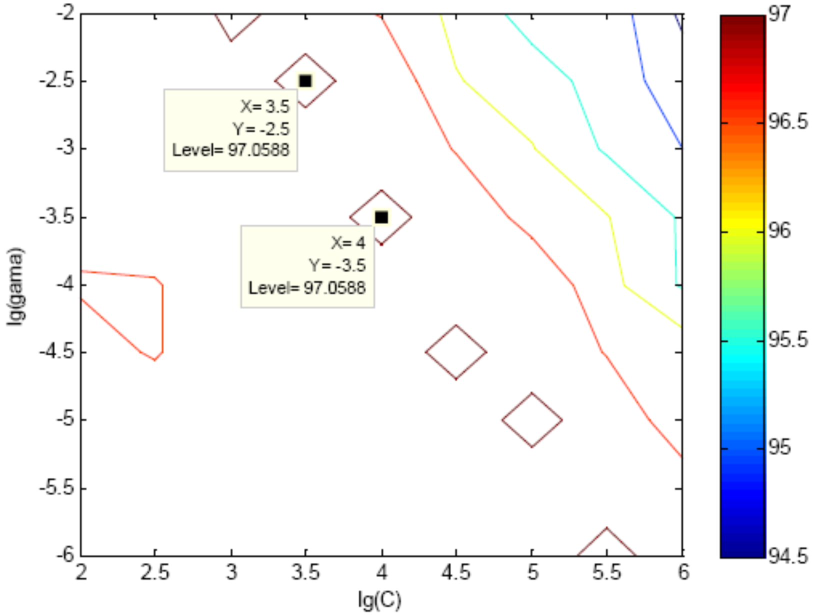
乳腺癌检测



Learning Curves for Gaussian Kernel
Training score
Cross-validation score
Score
Training examples
Learning Curves with degree=1
Training score
Cross-validation score
Score
Training examples
Learning Curves with degree=2
Training score
Cross-validation score
Score
Training examples

RBF参数的网格搜索 (Grid Search)





Logistic regression vs. SVMs

Andrew Ng, ML

$n =$ number of features , $m =$ number of training examples

If n is large (relative to m):
Use logistic regression, or SVM without a kernel (“linear kernel”)

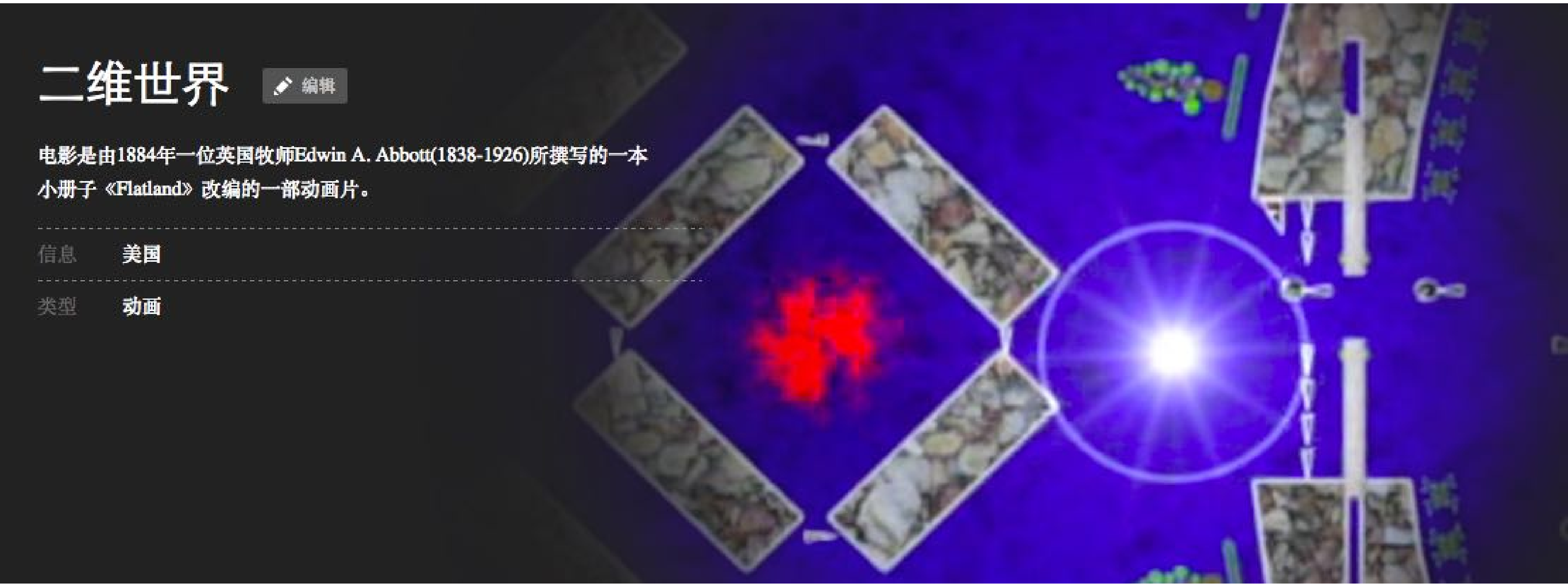
If n is small, m is intermediate:
Use SVM with Gaussian kernel

If n is small, m is large:
Create/add more features, then use logistic regression or SVM without a kernel

Neural network likely to work well for most of these settings, but may be slower to train.

机器学习的核心方法

- ☑ 模型选择 Model selection
- ☑ 代价函数 Cost function
- ☑ 优化算法 Optimization algorithm
- ☑ **正则化 Regularization**
- ☑ **核函数升维 Kernel trick**


二维世界
编辑
电影是由1884年一位英国牧师Edwin A. Abbott(1838-1926)所撰写的一本小册子《Flatland》改编的一部动画片。
信息 美国
类型 动画