机器学习
Machine Learning

北京航空航天大学计算机学院智能识别与图像处理实验室
IRIP Lab, School of Computer Science and Engineering, Beihang University
黄 迪　刘庆杰

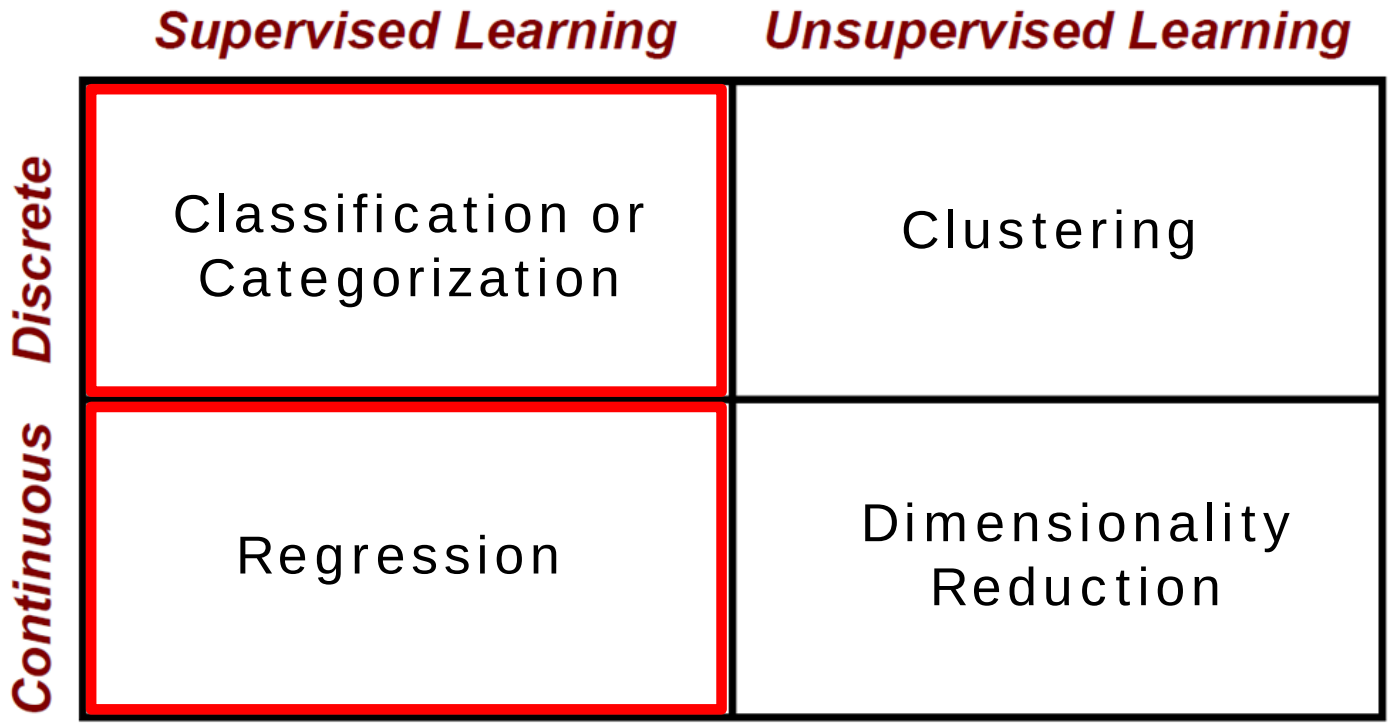
2018年秋季学期
Fall 2018

部分内容来源于C. Bishop和A. NG等人的课程以及互联网资源

课前回顾

机器学习算法

机器学习主要问题



一些问题

- 已知类条件概率密度 $p(\mathbf{x}|w_i)$ 和先验概率 $P(w_i)$，计算后验概率 $P(w_i|\mathbf{x})$ 进行决策

若类条件概率密度参数未知?

- 已知类条件概率密度 $p(\mathbf{x}|w_i)$ 的参数表达式，利用样本估计 $p(\mathbf{x}|w_i)$ 的未知参数，再利用贝叶斯定理将其转化成后验概率 $P(w_i|\mathbf{x})$ 进行决策

若类条件概率密度形式难以确定?

- 非参数方法估计

需要大量样本…

线性分类器设计

- 利用训练样本建立线性判别函数

$$g(x) = w^T x + w_0$$

$$g(x) = w_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=1}^{d} a_i y_i = a^T y$$

最好的结果一般出现在准则函数的极值点上，所以**将分类器设计问题转化为求准则函数极值** w^*, w_0^* **或** a^* **的问题。**

步骤1： 具有类别标志的样本集$\boldsymbol{\mathcal{X}} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots \boldsymbol{x_N}\}$或其增广样本集$\boldsymbol{\mathcal{Y}}$。

步骤2： 确定准则函数$\boldsymbol{\mathcal{J}}$，满足①$\boldsymbol{\mathcal{J}}$是样本集和$\boldsymbol{w}$, $\boldsymbol{w_0}$或$\boldsymbol{a}$的函数；②$\boldsymbol{\mathcal{J}}$的值反应分类器的性能，其极值对应“最好”的决策。

步骤3： 优化求解准则函数极值 w^*, w_0^* 或 a^* 。

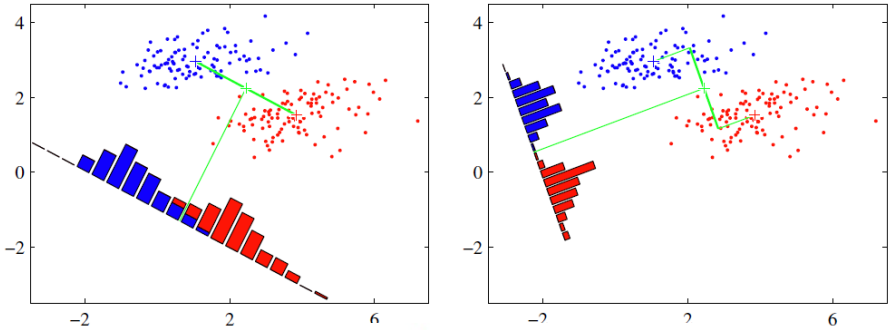
最终得到线性判别函数：$g(x) = w^{*T} x + w_0^*$ 或 $g(x) = a^{*T} y$，对于位置类别样本x_k，计算$g(x_k)$并通过决策规则判断其类别。

准则函数

- **Fisher准则**

$$J_F(w) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2}$$

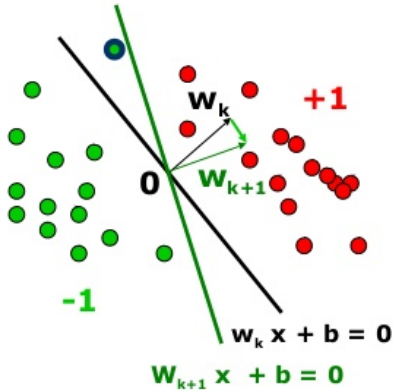
$$w^* = S_w^{-1}(m_1 - m_2)$$



- **感知机准则**

$$J_P(a) = \sum_{y \in \eta^k} (-a^T y)$$

$$a(k+1) = a(k) + \rho_k \nabla \sum_{y \in \eta^k} y$$

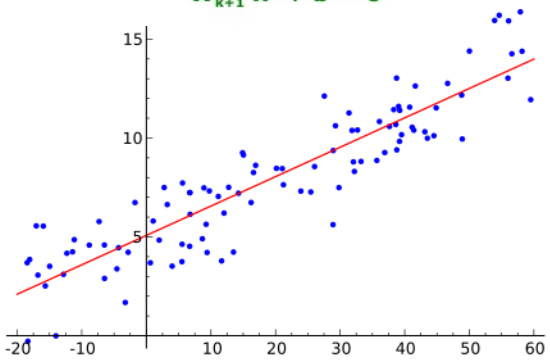


- **最小平方误差准则**

$$J_S(a) = \|e\|^2 = \|Ya - b\|^2 = \sum_{n-1}^{N} (a^T y_n - b_n)^2$$

$$a^* = (Y^T Y)^{-}1 Y^T b = Y^{+} b$$

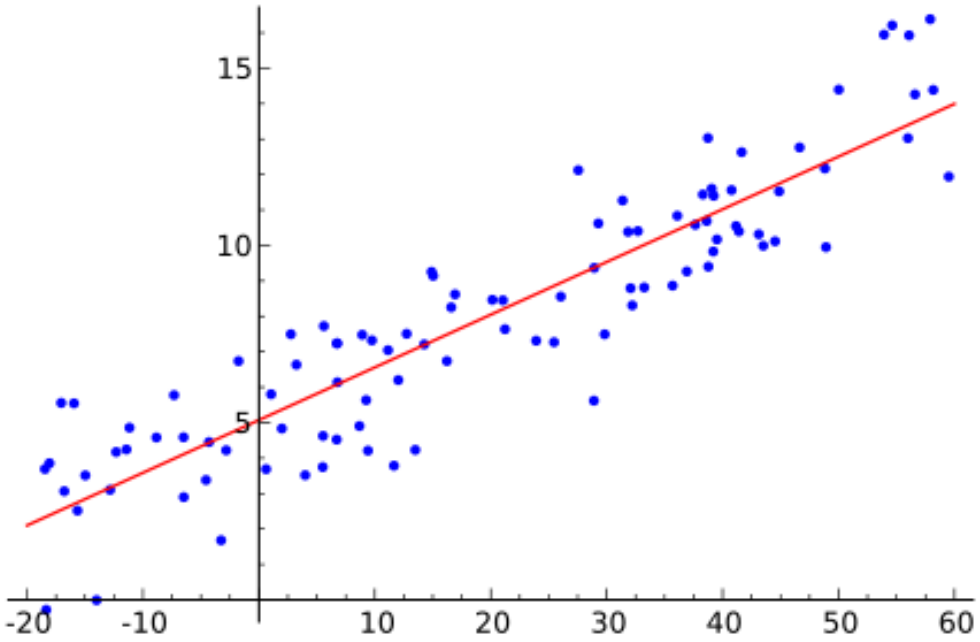
$$a(k+1) = a(k) - \rho_k Y^T (Ya - b)$$



最小二乘准则

- A.-M. Legendre(1806提出)，C. Gauss(1809提出/1829证明)

最小二乘法(最小平方误差法)通过最小化误差的平方和寻找数据的最佳函数匹配，即可以使求得的数据与实际数据之间误差的平方和最小。



最小二乘准则

- 寻找最好投影方向 a^*

$$a^T y_n > 0$$

$$a^T y_n = b_n > 0$$

b_n是任意给定的正常数

方程组形式：

$$Ya = b$$

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1\hat{d}} \\ y_{21} & y_{22} & \cdots & y_{2\hat{d}} \\ \cdots & \cdots & \cdots & \cdots \\ y_{N1} & y_{N2} & \cdots & y_{N\hat{d}} \end{bmatrix}$$

y_n是规范化增广向量样本

Y是$N \times \hat{d}$维矩阵，通常$N > \hat{d}$，一般为列满秩阵

$b = [b_1 \quad b_2 \quad \cdots \quad b_N]$

b是N维向量，b_n>0， n=1, 2, …, N

方程数多于未知数的矛盾方程组通常没有精确解

定义误差向量：$e = Ya - b$ 及平方误差准则函数

$$J_S(a) = \|e\|^2 = \|Ya - b\|^2 = \sum_{n-1}^{N} (a^T y_n - b_n)^2$$

最小二乘准则

- 求使 $J_S(a)$ 最小的**a***（最小二乘近似解/伪逆解/MSE解）

采用解析法求伪逆解 $J_S(a) = \|e\|^2 = \|Ya - b\|^2 = \sum_{1}^{N}(a^T y_n - b_n)^2$

$$\nabla J_S(a) = \sum_{n-1}^{N} 2(a^T y_n - b_n) y_n = 2Y^T(Ya - b)$$

令 $\nabla J_S(a) = 0$

得 $Y^T Y a^* = Y^T b$　　矩阵Y^TY是$\hat{d} \times \hat{d}$方阵一般非奇异

唯一解 $a^* = (Y^T Y)^{-}1 Y^T b = Y^+ b$

其中$\hat{d} \times N$矩阵 $Y^+ = (Y^T Y)^{-}1 Y^T$ 是Y的左逆矩阵

如何选b?

$$\boldsymbol{b} = \begin{bmatrix} N/N_1 \\ \cdots \\ N/N_1 \\ N/N_2 \\ \cdots \\ N/N_2 \end{bmatrix} \begin{matrix} \\ \\ N_1\text{个} \\ \\ \\ N_2\text{个} \end{matrix}$$

➡ a*等价于Fisher解

$$g_0(x) = P(w_1|x) - P(w_2|x)$$

$N \to \infty, \boldsymbol{b} = [\underbrace{1, 1, \cdots, 1}_{N\text{个}}]^T$ ➡ 以最小均方误差逼近贝叶斯判别函数

最小二乘准则

- 求使 $J_S(a)$ 最小的a^* (最小二乘近似解/伪逆解/MSE解)

$$a^* = Y^+ b \qquad Y^+ = (Y^T Y)^{-}1 Y^T$$

问题：①要求Y^TY非奇异；②求Y^+计算量大同时可能引入较大误差。

采用梯度下降法求解：$\nabla J_S(a) = 2Y^T(Ya - b)$

$$\begin{cases} a(1), Random \\ a(k+1) = a(k) - \rho_k Y^T(Ya - b) \end{cases}$$

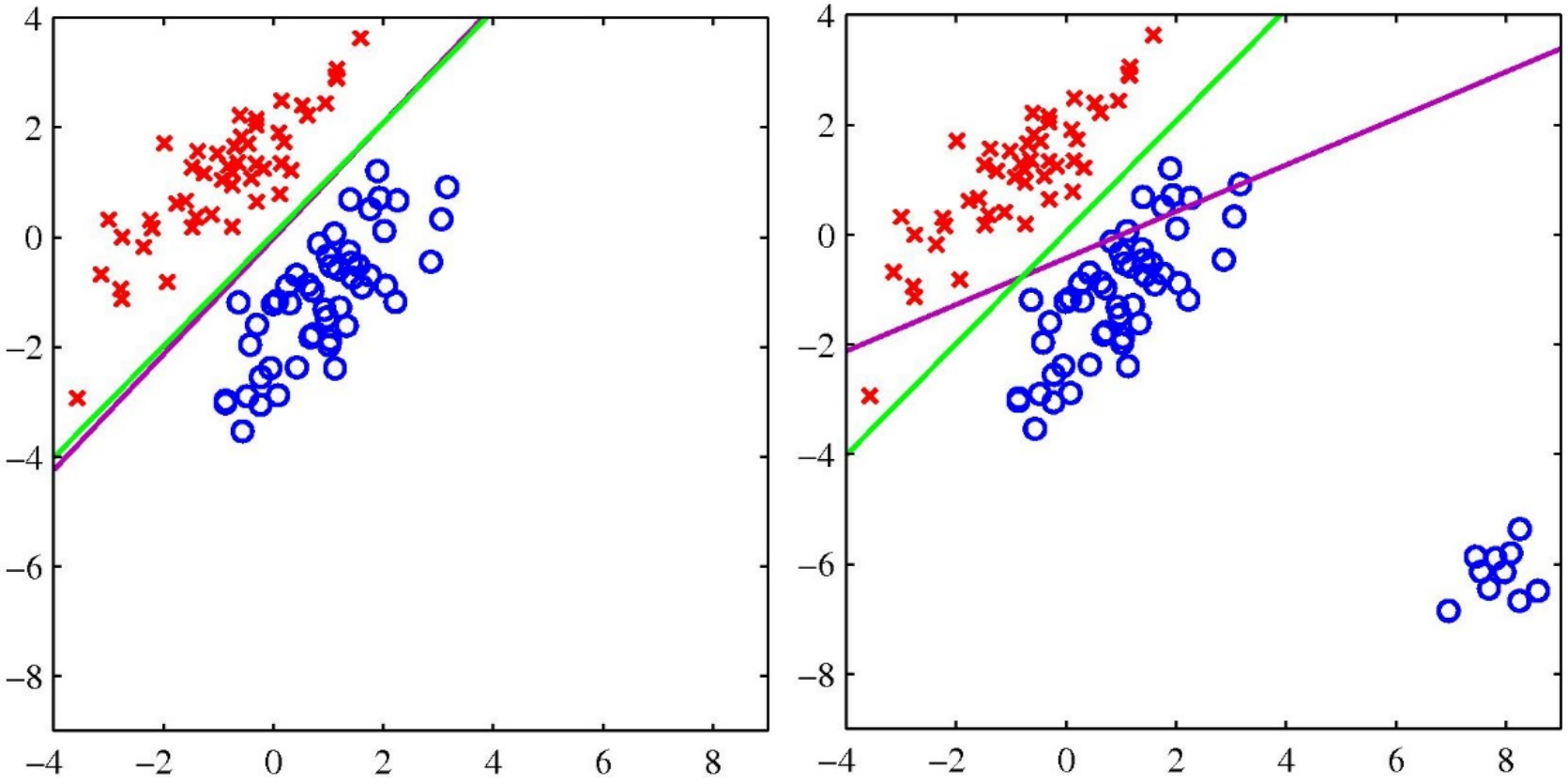
可以证明，选择 $\rho_k = \frac{\rho_1}{k}$ ，ρ_1是任意常数

该算法权向量收敛于使 $\nabla J_S(a) = 2Y^T(Ya - b) = 0$ 的权向量a^*

不要求Y^TY奇异与否，只计算$\hat{d} \times \hat{d}$方阵Y^TY，比$\hat{d} \times N$阵Y^+计算量小

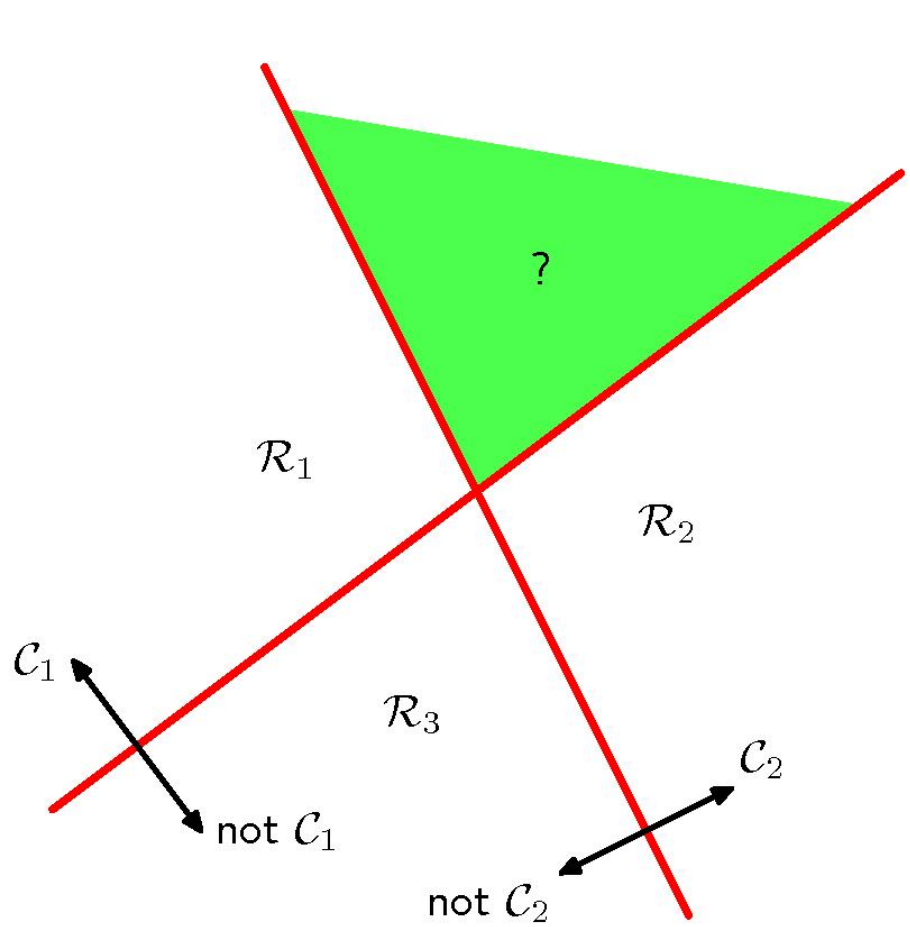
最小二乘准则

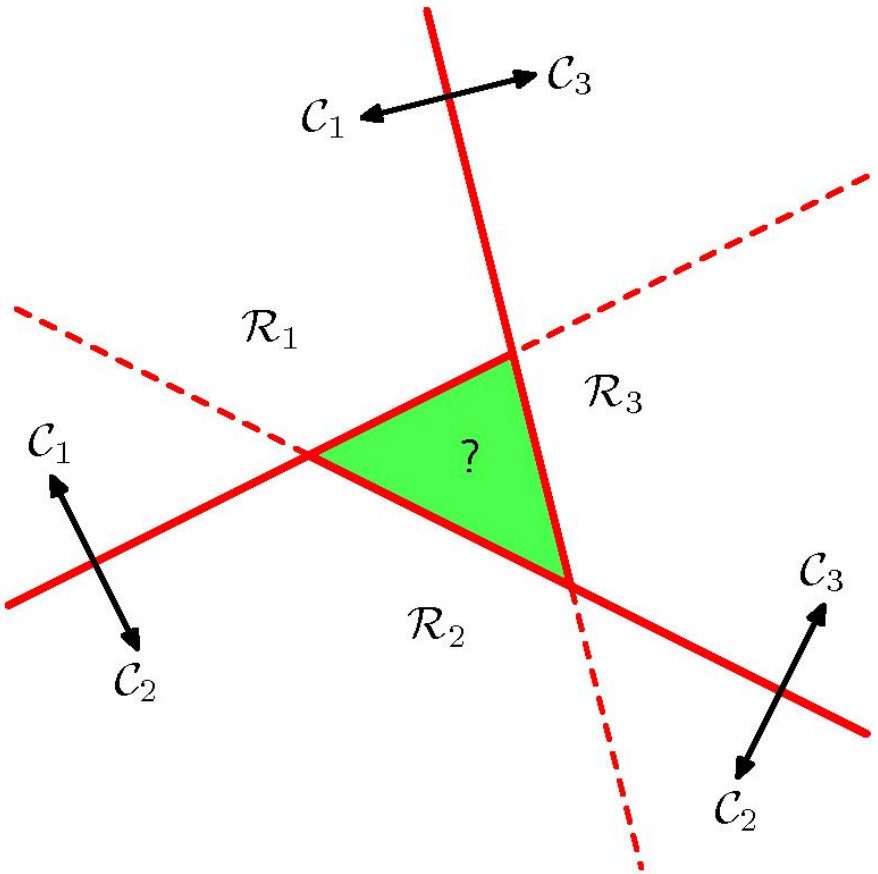
- 对于异常值(Outlier)非常敏感



多分类问题

- **1 *vs.* (N-1) or 1 *vs.* 1**





(补)：生成式模型和判别式模型

Generative Model and Discriminative Model

生成式模型和判别式模型

- **生成式模型(Generative Model)**

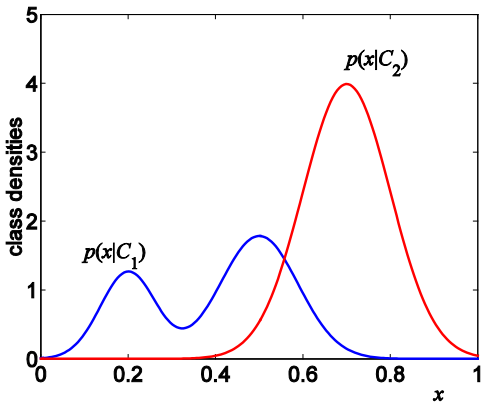
分别对各类的**类条件密度** $p(\mathbf{x}|C_k)$ 和**先验概率** $p(C_k)$ 进行建模，之后利用贝叶斯定理计算**后验概率**

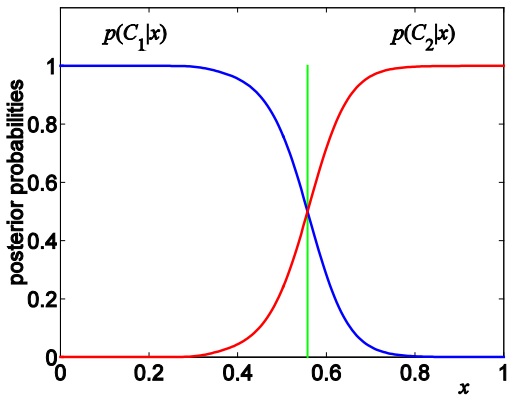
$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)}$$

或者直接对**联合分布** $p(\mathbf{x}, C_k)$ 建模得到后验概率

- **判别式模型(Discriminative Model)**

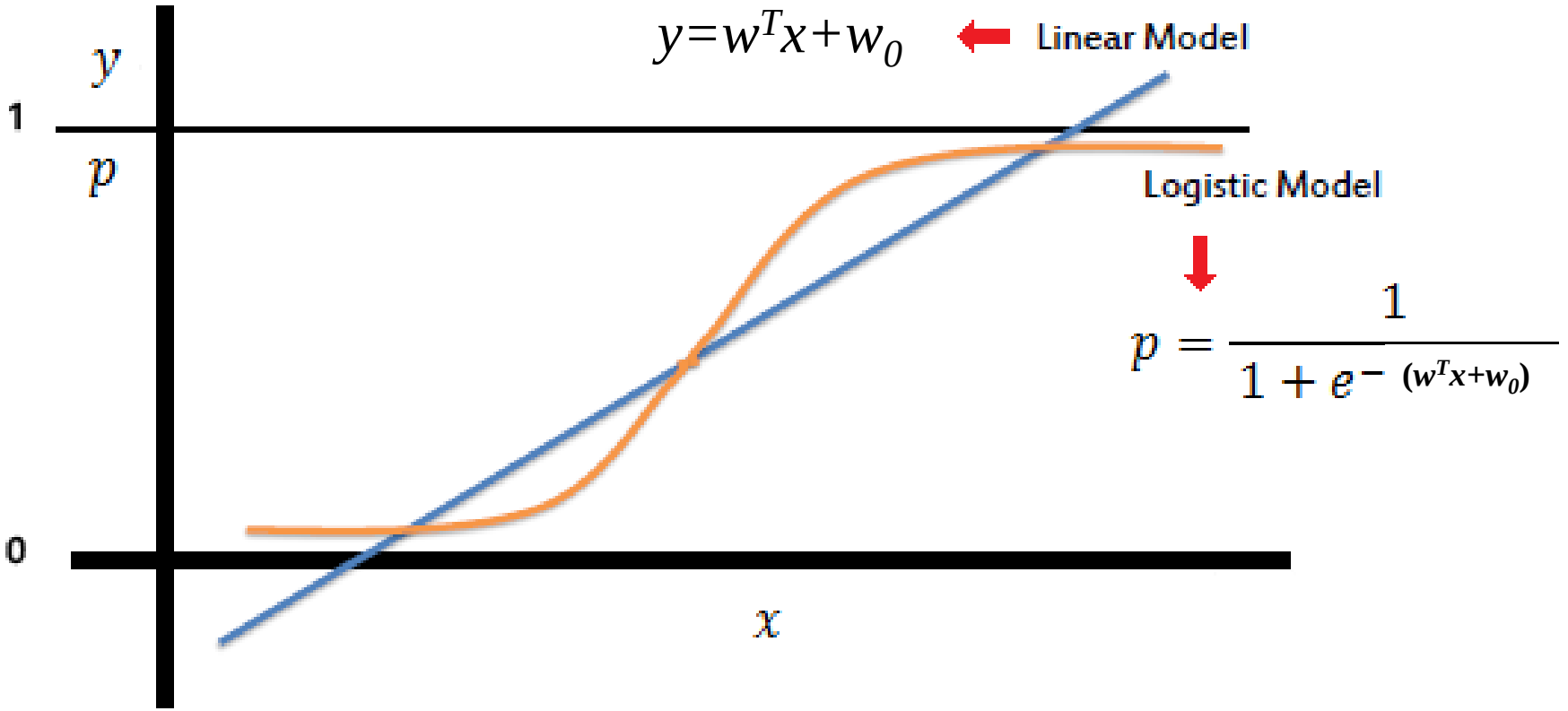
直接对**后验概率** $p(C_k|\mathbf{x})$ 建模





回归和分类

- 线性回归和逻辑回归



逻辑回归(Logistic Regression)

- 将回归问题转为分类问题

逻辑回归是概率型非线性回归，但其本质是线性回归，只是在特征到结果的映射中加入了一层函数映射，即先把特征线性求和，然后使用函数σ(z) 来预测。

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(c_1)}{p(\mathbf{x}|C_1)p(c_1)+p(\mathbf{x}|C_2)p(c_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) \quad a = ln\frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)}$$

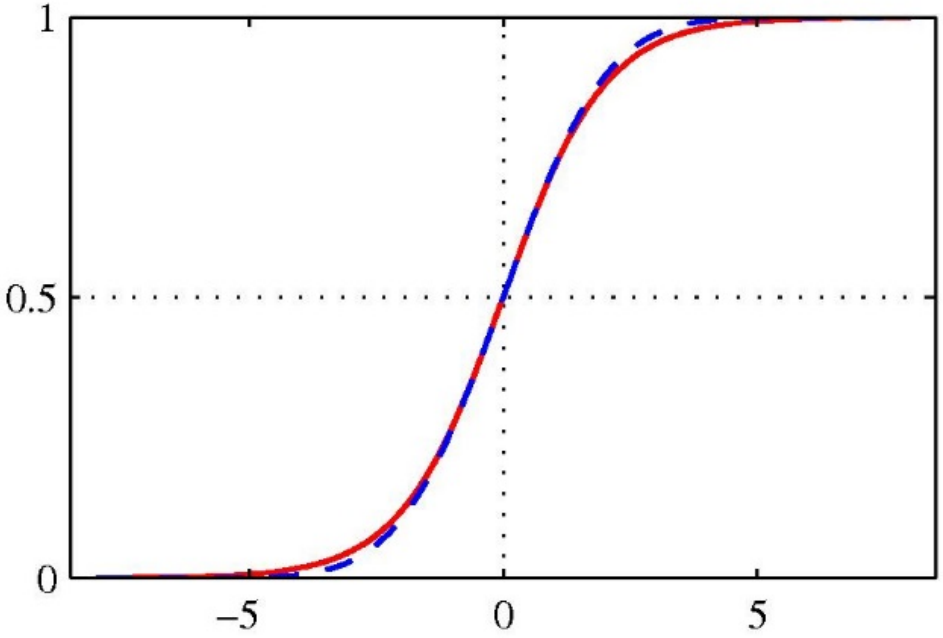
Sigmoid函数性质：

①可以将连续值映射到0和1区间上

②对称性 $\sigma(-a) = 1 - \sigma(a)$

③反转性 $a = ln(\frac{\sigma}{1-\sigma})$

④可导性 $\frac{d\sigma}{da} = \sigma(1-\sigma)$



逻辑回归(Logistic Regression)

- 将回归问题转为分类问题

逻辑回归是概率型非线性回归，但其本质是线性回归，只是在特征到结果的映射中加入了一层函数映射，即先把特征线性求和，然后使用函数 σ(z) 来预测。

多类问题:

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(c_k)}{\sum_j p(\mathbf{x}|C_j)p(c_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

$$a_k = \ln p(\mathbf{x}|C_k)p(c_k)$$

归一化指数
Normalized Exponential

Softmax函数

是一个平滑的max函数，如果 $a_k \gg a_j$ **，对于所有的** $k \neq j$**，有** $p(C_k|\mathbf{x}) \simeq 1$ $p(C_j|\mathbf{x}) \simeq 0$

生成式模型和判别式模型

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(c_1)}{p(\mathbf{x}|C_1)p(c_1)+p(\mathbf{x}|C_2)p(c_2)} = \frac{1}{1+exp(-a)} = \sigma(a)$$

$$a = ln\frac{p(C_1|x)}{p(C_2|x)} = ln\frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)}$$

- **生成式模型(Generative Model)**

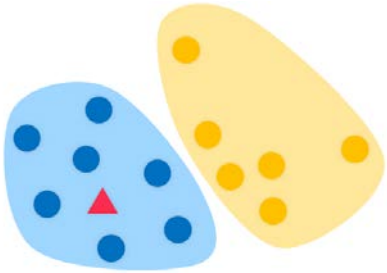
估计类条件密度并计算*a*

- **判别式模型(Discriminative Model)**

把*a*视为线性函数 $a = w^T x + w_0$ 直接估计

生成式模型和判别式模型

● 生成式模型



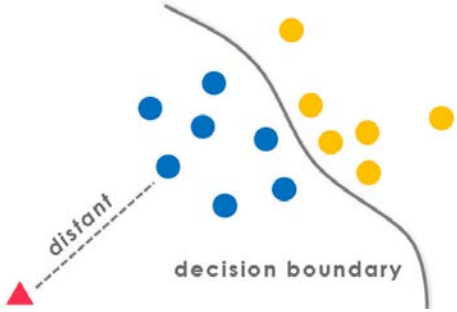
优点：

- 信息丰富
- 单类问题灵活性强
- 增量学习
- 合成缺失数据

缺点：

- 学习过程复杂
- 为分布牺牲分类性能

● 判别式模型



优点：

- 类间差异清晰
- 分类边界灵活
- 学习简单
- 性能较好

缺点：

- 不能反应数据特性
- 需要全部数据进行学习

由生成模型可以得到判别模型，
但由判别模型得不到生成模型。

生成式模型和判别式模型

● 生成式模型

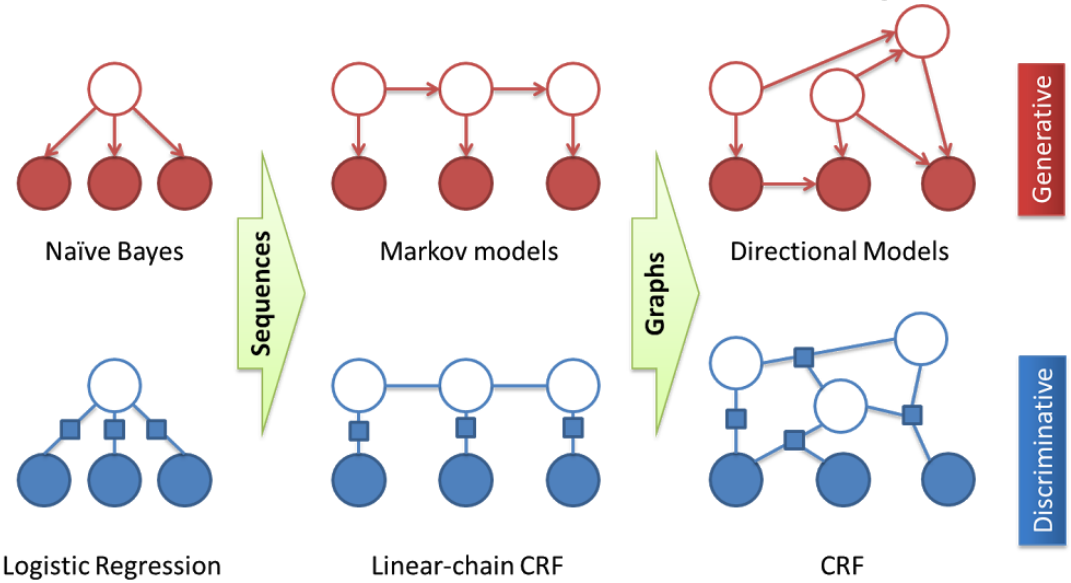
代表算法：

- Naive Bayes
- Mixtures of Gaussians
- Hidden Markov Models
- Bayesian Networks
- Deep Belief Network

● 判别式模型

代表算法：

- Linear & Logistic Regression
- Support Vector Machine
- Nearest Neighbor
- Conditional Random Fields
- Boosting



第四章：支持向量机

Support Vector Machine (SVM)

概述

- C. Cortes和V. Vapnik (1995年提出)

支持向量机是基于统计学习理论(Statistical Learning Theory, SLT)发展起来的一种机器学习的方法。

统计学习理论主要创立者是Vladimir N. Vapnik。

AT&T

概述

- **Vladimir N. Vapnik**

1936年 出生于苏联

1958年 乌兹别克国立大学 硕士
1964年 莫斯科控制科学学院 博士

1964-1990年 莫斯科控制科学学院
曾担任计算机科学与研究系主任
1991-2001年 美国AT&T贝尔实验室
发明支持向量机理论
2002-2014年 NEC实验室(美国)
从事机器学习研究
2014-2016年 美国Facebook公司
从事人工智能研究
2016年至今 美国Vencore实验室
继续研究工作

1995年和2003年，他分别被伦敦大学皇家霍洛威学院和美国哥伦比亚大学聘为计算机专业的教授。
2006年，他成为美国国家工程院院士。

概述

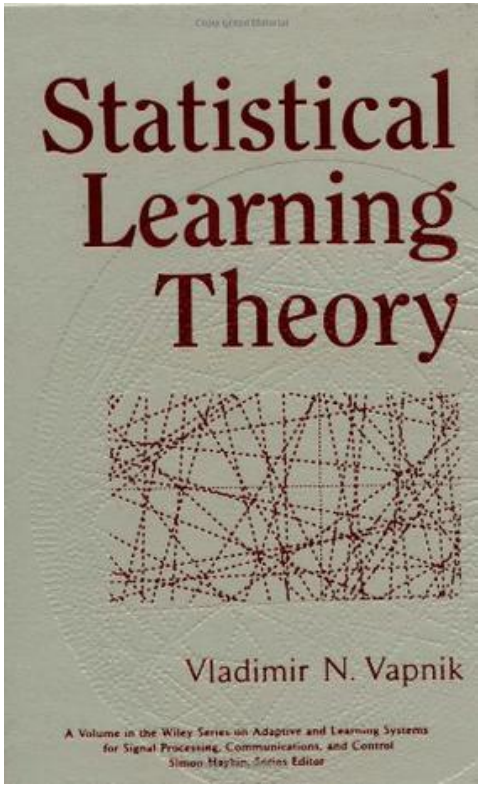
- V. Vapnik对于统计机器学习的贡献

1968年，Vapnik和Chervonenkis提出了VC熵和VC维的概念，这些是统计学习理论的核心概念。同时，他们发现了泛函空间的大数定理，得到了关于收敛速度的非渐进界的主要结论。

1974年，Vapnik和Chervonenkis提出了结构风险最小化归纳原则。

1989年，Vapnik和Chervonenkis发现了经验风险最小化归纳原则和最大似然方法一致性的充分必要条件，完成了对经验风险最小化归纳推理的分析。

90年代中期，有限样本情况下的机器学习理论研究逐渐成熟起来，形成了较完善的理论体系—统计学习理论。



概述

- 支持向量机的发展

1963年，Vapnik在解决模式识别问题时提出了支持向量方法，这种方法从训练集中选择一组特征子集，使得对特征子集的划分等价于对整个数据集的划分，这组特征子集就被称为支持向量(SV)。

1971年，Kimeldorf提出使用线性不等约束重新构造SV的核空间，解决了一部分线性不可分问题。

1990年，Grace、Boser和Vapnik等人开始对SVM进行研究。

1995年，Vapnik的书《The Nature of Statistical Learning Theory》出版，详细叙述了SVM理论，同时也标志着统计学习理论体系已经走向成熟。

1999年， IEEE Trans. on Neural Network (IEEE T-NN) 为统计学习理论出版了专刊，MIT出版了《Advances in Kernel Method》，使SVM理论的研究与应用推向了一个高潮。

近年来，SVM的研究主要集中在对SVM本身性质的研究和完善以及加大SVM应用研究的深度和广度两方面。

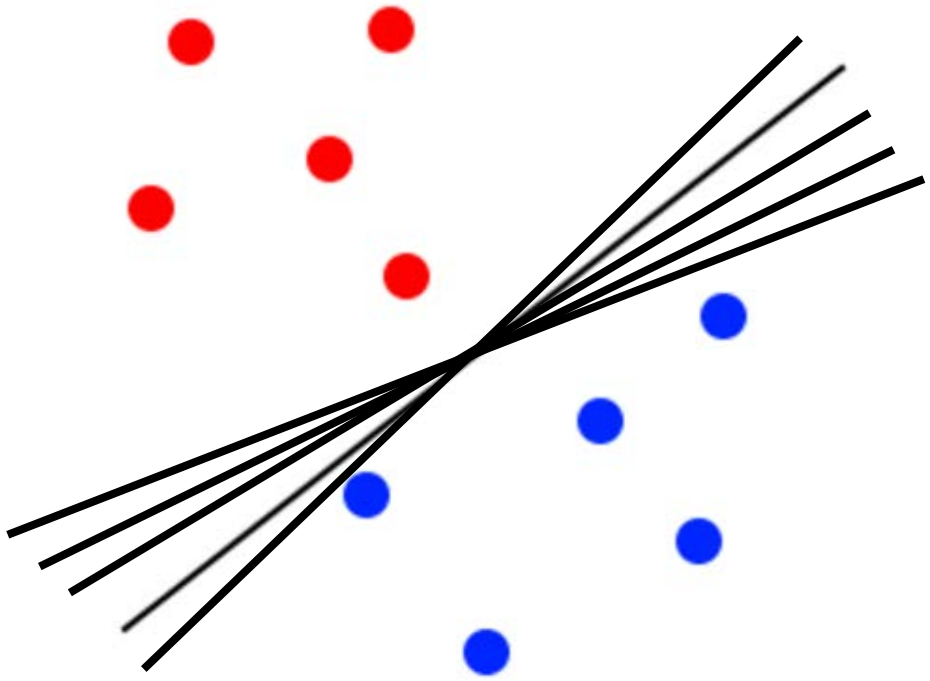
The Nature of Statistical Learning Theory 77736 * 1995
V Vapnik
Data mining and knowledge discovery

线性分类模型

- **两类样本的线性分类问题**

$$y(x) = w^T x + b$$

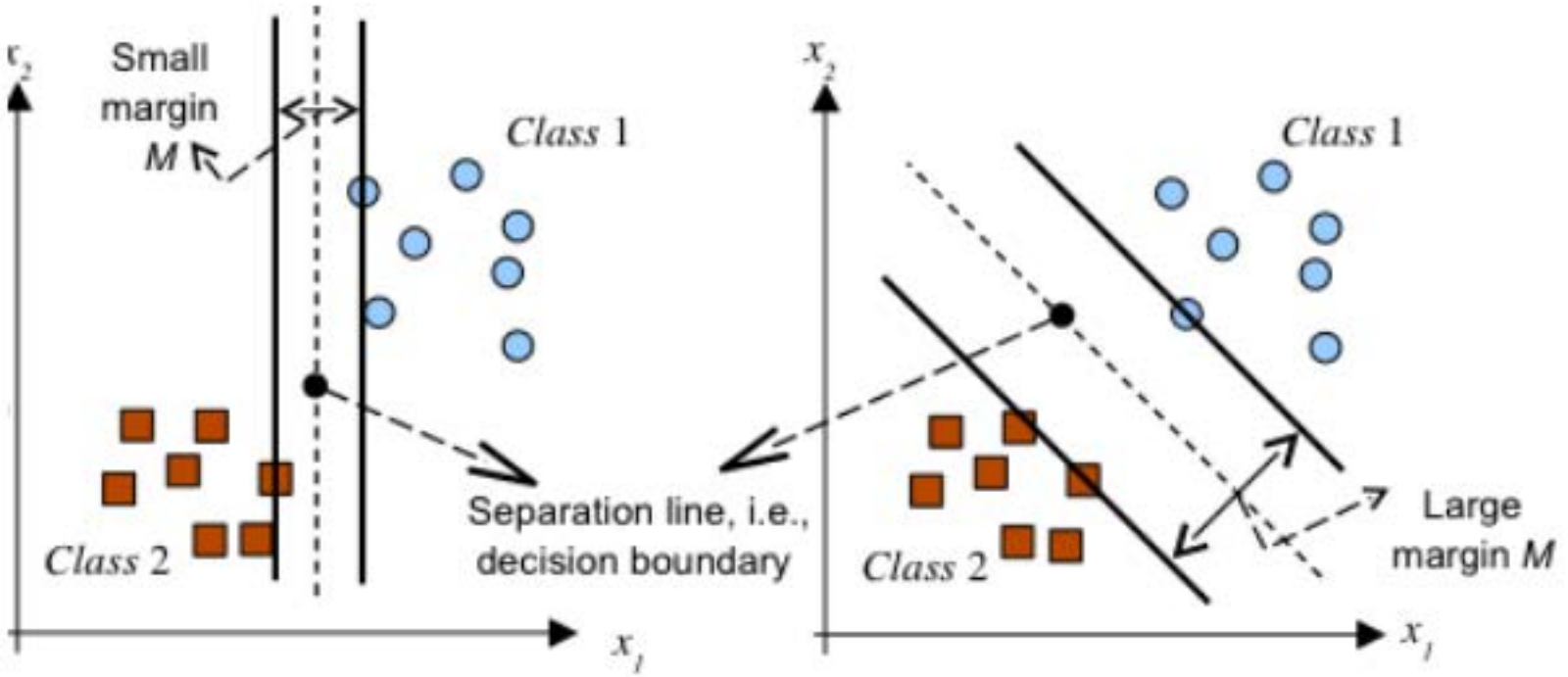
$$y(x, w) = f(\sum_{j=1}^{M} w_j x_j)$$



支持向量机

- SVM从线性可分情况下的**最优分类面**发展而来。

最优分类面就是要求分类线**不但能将两类正确分开**(训练错误率为0)，且使**分类间隔**最大。SVM考虑寻找一个满足分类要求的超平面，并且**使训练集中的点距离分类面尽可能的远**，也就是寻找一个分类面**使它两侧的空白区域(Margin)最大**。



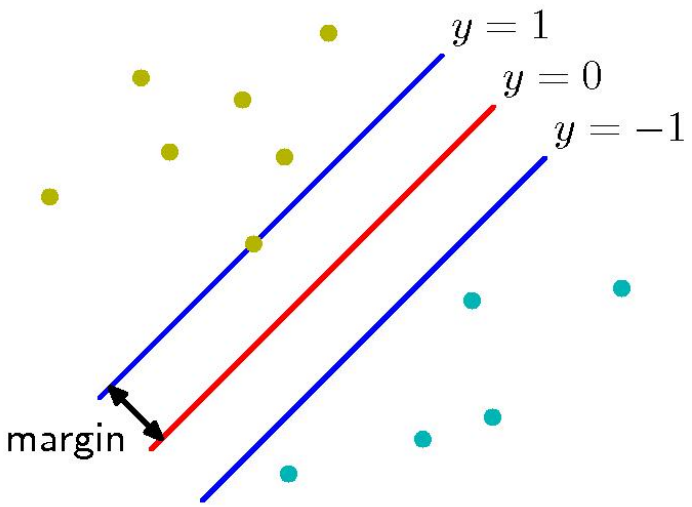
支持向量机

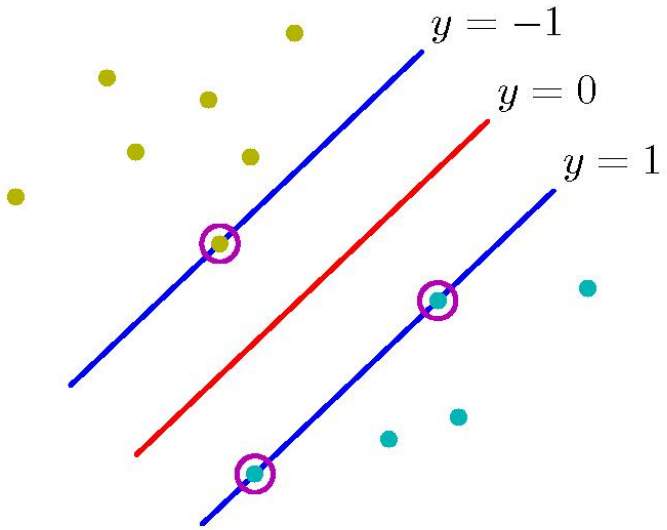
- **线性支持向量机**

样本集 $\{x_n, t_n\}, n = 1, 2, ..., N,\ x_n \in \mathcal{R}^d; t_n \in \{-1, 1\}$

分类器 $y(x) = w^T x + b$

$$t_n = \begin{cases} 1, y(x_n) > 0 \text{ if } x_i \in w_1 \\ -1, y(x_n) < 0 \text{ if } x_i \in w_2 \end{cases} \quad \Rightarrow \quad t_n y(x_n) > 0$$





支持向量机

- **线性支持向量机**

样本集任意一点x_n到分类面(满足$t_n y(x_n) > 0$)的距离

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n(w^T x_n + b)}{\|w\|}$$

优化*w*和*b*使Margin最大

$$\arg\max_{w,b}\{\frac{1}{\|w\|}\min_n[t_n(w^T x_n + b)]\}$$

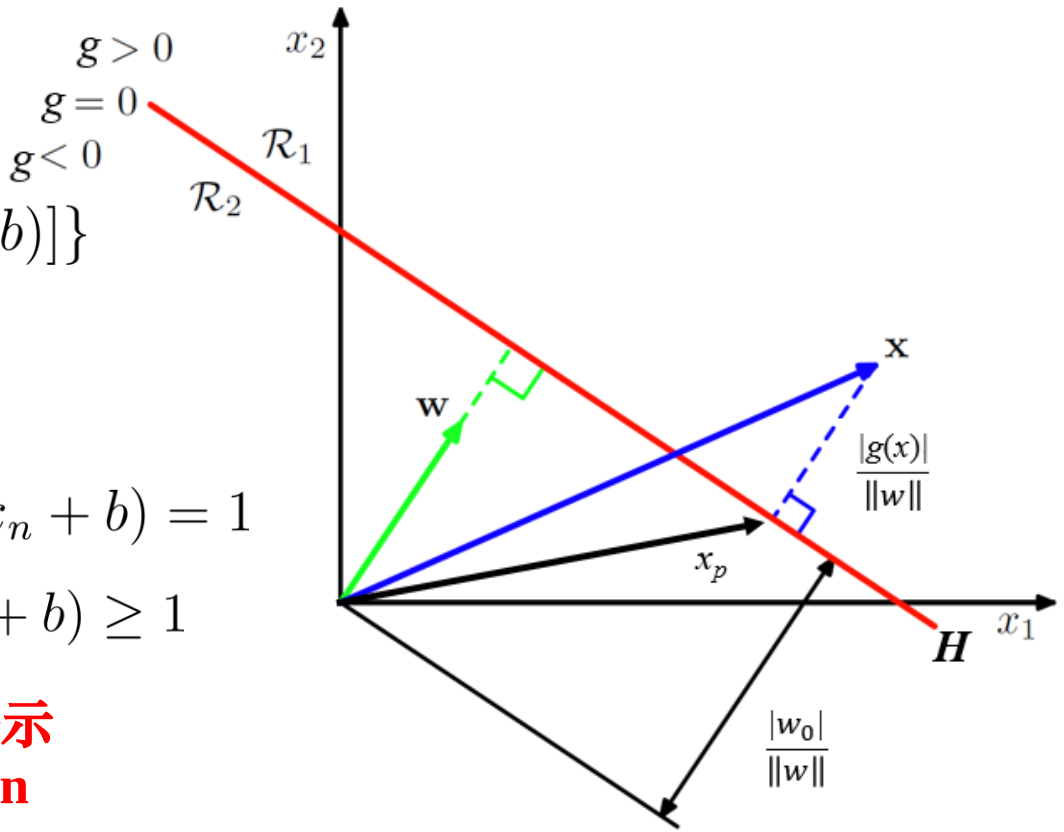
求解复杂

$$w \to kw, b \to kb$$

对于离超平面最近的点 $t_n(w^T x_n + b) = 1$

那么对于所有点满足 $t_n(w^T x_n + b) \geq 1$

对于决策超平面的标准表示
Canonical Representation



支持向量机

- **线性支持向量机**

问题转化为最大化 $\|w\|^{-1}$**, 等价于** $\arg\min_{w,b} \frac{1}{2}\|w\|^2$ **二次规划问题**

$$s.t.\ t_n(w^T x_n + b) \geq 1$$

拉格朗日乘子法 $L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n\{t_n(w^T x_n + b) - 1\}, a_n \geq 0$

分别对变量求导

$$\frac{\partial L(w,b,a)}{\partial w} = w - \sum_{n=1}^{N} a_n t_n x_n = 0$$

$$\frac{\partial L(w,b,a)}{\partial b} = \sum_{n=1}^{N} a_n t_n = 0$$

代入*L*得到对偶形式:

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m x_n^T x_m$$

二次规划问题

$$w.r.t.\ a_n \geq 0, n = 1, ..., N, \sum_{n=1}^{N} a_n t_n = 0$$

对偶问题

- **拉格朗日对偶性**

$$\min_w \ f(w)$$

存在等式约束的函数极值问题 $s.t.\ h_i(w) = 0, i = 1, ..., l.$

$$\Rightarrow \quad L(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

存在不等式约束的函数极值问题 $\min_w \ f(w)$

$$s.t.\ g_i(w) \le 0, i = 1, ..., k.$$
$$h_i(w) = 0, i = 1, ..., l.$$

$$\Rightarrow \quad L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

求*L*极小值，但如果 $g_i(w) \neq 0$ **，当** $\alpha_i \to +\infty$ **，** $L \to -\infty$

定义 $\theta_P(w) = \max_{\alpha, \beta; \alpha_i \ge 0} L(w, \alpha, \beta)$**，只有当满足约束条件，才有最大值。**

$$\Rightarrow \quad \theta_P(w) = \begin{cases} f(w), \ if \ w \ satisfies \ primal \ constraints. \\ +\infty, \ otherwise \end{cases}$$

对偶问题

- **拉格朗日对偶性**

原问题 $\min_w \ f(w)$ **转化为** $\min_w \theta_P(w) = \min\limits_{w} \max\limits_{\alpha,\beta;\alpha_i \geq 0} L(w,\alpha,\beta)$**，记为** p^*

直接求解不容易进而转向另一个问题 $\theta_D(\alpha,\beta) = \min\limits_{w} L(w,\alpha,\beta)$**，先固定** α, β**，求拉格朗日函数关于w的最小值，之后再求** $\theta_D(\alpha,\beta)$ **的最大值，即**

$\max\limits_{\alpha,\beta;\alpha_i \geq 0} \theta_D(\alpha,\beta) = \max\limits_{\alpha,\beta;\alpha_i \geq 0} \min\limits_{w} L(w,\alpha,\beta)$ **原问题的对偶问题，记为** d^*

$$\Rightarrow \quad d^* = \max_{\alpha,\beta;\alpha_i \geq 0} \min_{w} L(w,\alpha,\beta) \leq \min_{w} \max_{\alpha,\beta;\alpha_i \geq 0} L(w,\alpha,\beta) = p^*$$

假设*f*和*g*都是凸函数，*h*是仿射的(Affine，线性函数的一般形式)，且对于所有的*i*，$g_i(w) < 0$**，那么一定存在*w**, *α**, *β**，使*w**是原问题的解，*α**, *β**是对偶问题的解，即** $d^* = p^* = L(w^*,\alpha^*,\beta^*)$**，且*w**, *α**, *β**满足KKT条件。**

KKT条件

- **KKT(Karush-Kuhn-Tucker)条件**

$$\frac{\partial}{\partial w_i} L(w^*, \alpha^*, \beta^*) = 0, \ i = 1, ...n$$

$$\frac{\partial}{\partial \beta_i} L(w^*, \alpha^*, \beta^*) = 0, \ i = 1, ...l$$

$$\alpha_i^* g_i(w^*) = 0, \ i = 1, ...k$$

$$g_i(w^*) \leq 0, \ i = 1, ...k$$

$$\alpha^* \geq 0, \ i = 1, ...k$$

KKT对偶互补条件

如果*w, *α**, *β**满足KKT条件，那么它们就是原问题和对偶问题的解。**

补充条件隐含如果 $\alpha^* > 0$，那么 $g_i(w^*) = 0$，即*w*处于可行域的边界上，是起作用的(Active)约束，而位于可行域内部的点都是不起作用的约束，其 $\alpha^* = 0$。

支持向量机

- **线性支持向量机**

KKT条件：

$$a_n \geq 0$$

$$t_n y(x_n) - 1 \geq 0$$

$$a_n\{t_n y(x_n) - 1\} = 0$$

支持向量：

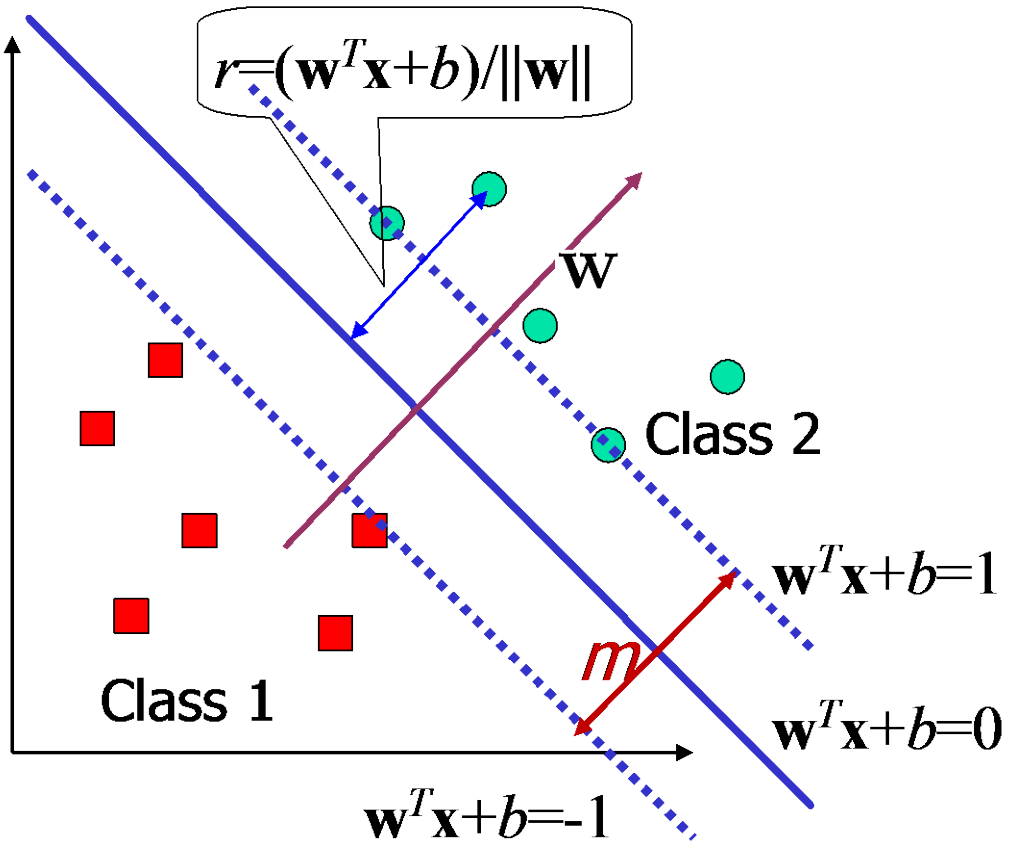
$$t_n(w^T x + b) = 1, a_n > 0$$

非支持向量：

$$t_n(w^T x + b) > 1, a_n = 0$$

$$y(x) = \sum_{n=1}^{N} a_n t_n x^T x_n + b$$

$$b = \frac{1}{N_S} \sum_{n \in S} (t_n - \sum_{m \in S} a_m t_m x_n^T x_m)$$



超平面法向量是支持向量的线性组合

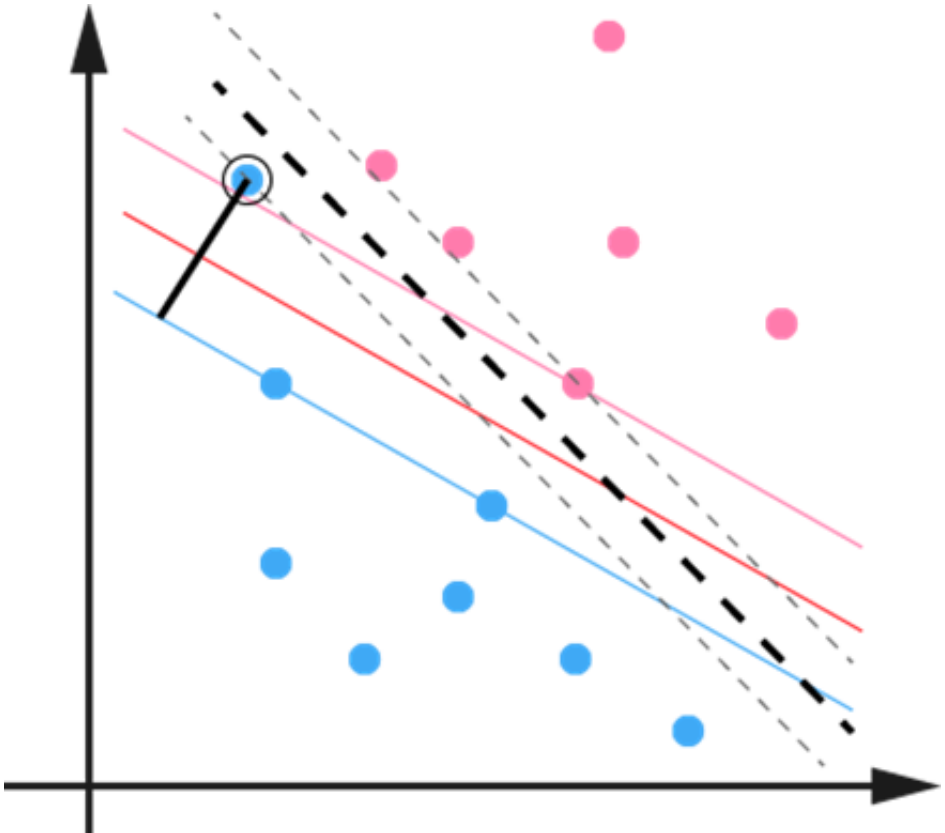
支持向量机

- 处理噪声和离群点

求解最优分类面的时间代价大还可能导致泛化性能差。因此，对于分布有交集的数据需要有一定范围内的“错分”，又有较大分界区域的广义最优分类面。

准确性

泛化性



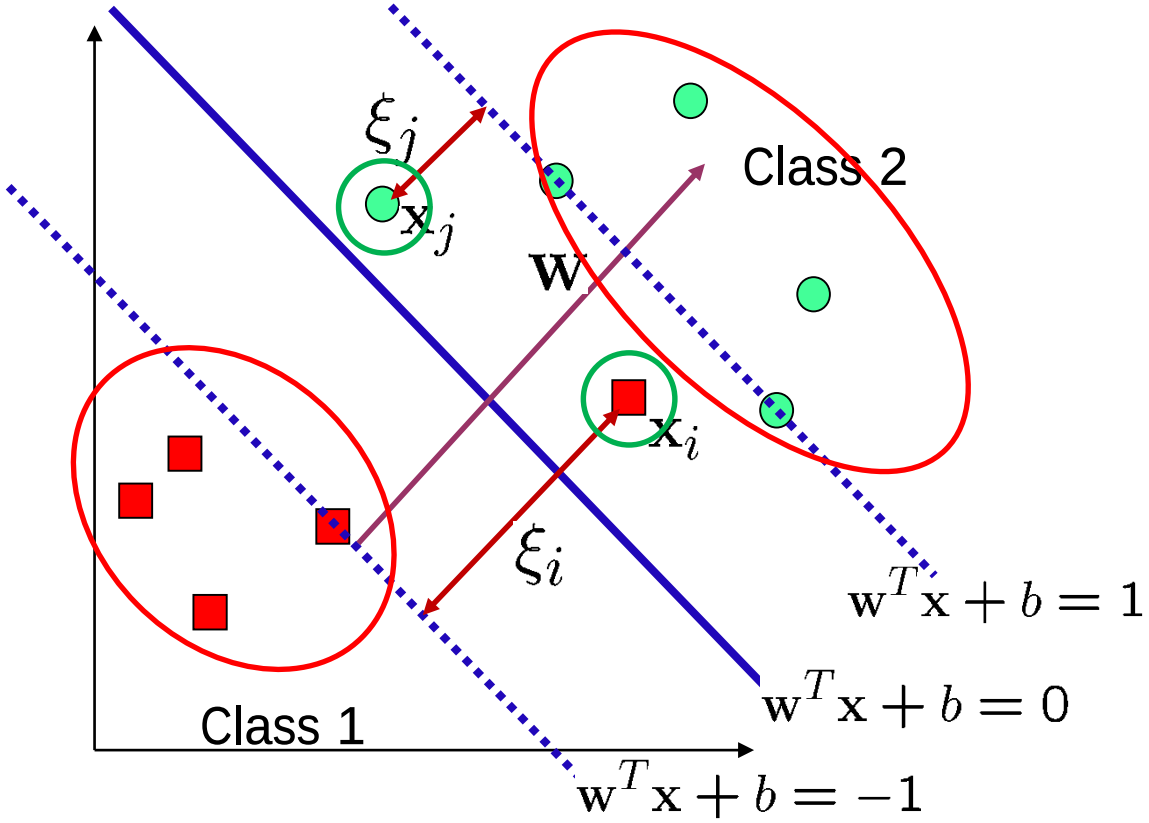
支持向量机

- **处理噪声和离群点**

引入松弛变量 $\xi_n \geq 0$

$$\xi_n = 0$$
$$\xi_n = |t_n - y(x_n)|$$

$$\xi_n = 0$$
$$0 < \xi_n \leq 1$$
$$\xi_n > 1$$



原有约束 $t_n y(x_n) \geq 1$ ➡ $t_n y(x_n) \geq 1 - \xi_n$

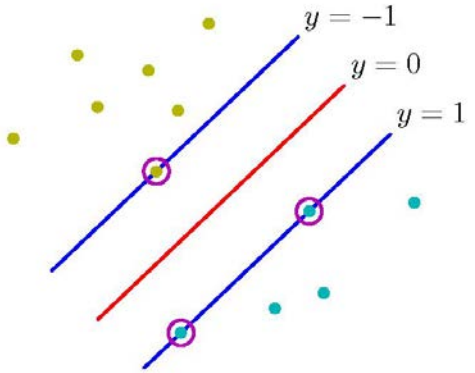
支持向量机

- **处理噪声和离群点**

这种处理方式也被视为是从硬间隔(Hard Margin)向软间隔(Soft Margin)的转变。

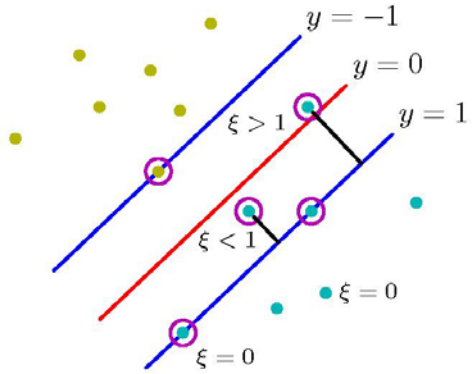
硬间隔

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t.\ t_n(w^T x_n + b) \geq 1,\ n = 1, ..., N$$



软间隔

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N} \xi_n$$
$$s.t.\ t_n(w^T x_n + b) \geq 1 - \xi_n,\ n = 1, ..., N$$
$$\xi_n \geq 0$$



支持向量机

- **处理噪声和离群点**

利用拉格朗日乘子法求解：

$$L(w,b,\xi,a,\mu)=\tfrac{1}{2}\|w\|^2+C\sum_{n=1}^{N}\xi_n-\sum_{n=1}^{N}a_n\{t_n y(x_n)-1+\xi_n\}-\sum_{n=1}^{N}\mu_n\xi_n$$

$$a_n\geq 0;\mu_n\geq 0$$

KKT条件：

$$a_n\geq 0$$

$$t_n y(x_n)-1+\xi_n\geq 0$$

$$a_n(t_n y(x_n)-1+\xi_n)=0$$

$$\mu_n\geq 0$$

$$\xi_n\geq 0$$

$$\mu_n\xi_n=0$$

优化*w*, *b*, {ξ_n}

$$\frac{\partial L}{\partial w}=0\Rightarrow w=\sum_{n=1}^{N}a_n t_n x_n$$

$$\frac{\partial L}{\partial b}=0\Rightarrow \sum_{n=1}^{N}a_n t_n=0$$

$$\frac{\partial L}{\partial \xi_n}=0\Rightarrow a_n=C-\mu_n$$

代入*L*化简：

$$\tilde{L}(a)=\sum_{n=1}^{N}a_n-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_n a_m t_n t_m x_n^T x_m$$

支持向量机

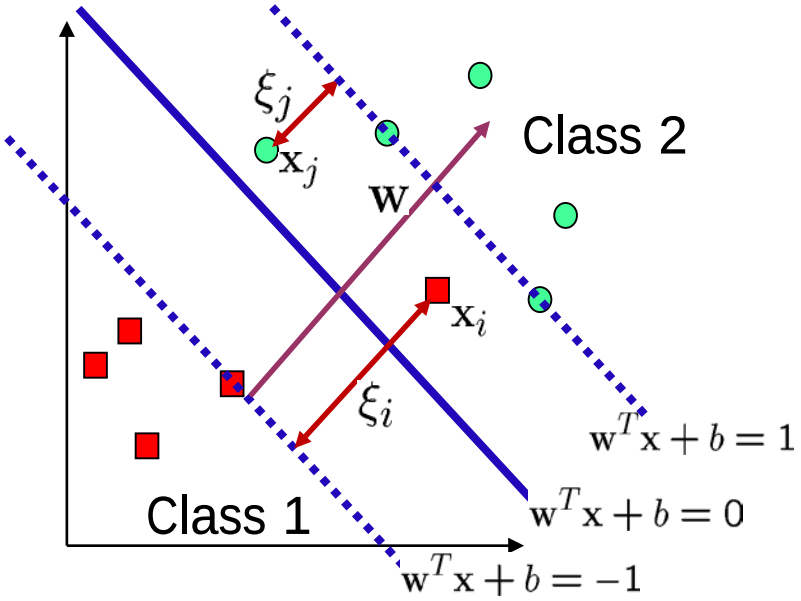
- **处理噪声和离群点**

得到其对偶形式：

$$\max_a \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m x_n^T x_m$$

$$s.t.\ 0 \le a_n \le C, n = 1, 2, ..., N$$

$$\sum_{n=1}^{N} a_n t_n = 0$$ **二次规划问题**



对于新样本预测的分类器：

$$y(x) = \sum_{n=1}^{N} a_n t_n x^T x_n + b$$

非支持向量： $a_n = 0$

支持向量： $t_n y(x_n) = 1 - \xi_n$

$$a_n < C \Rightarrow \mu > 0 \Rightarrow \xi_n = 0$$

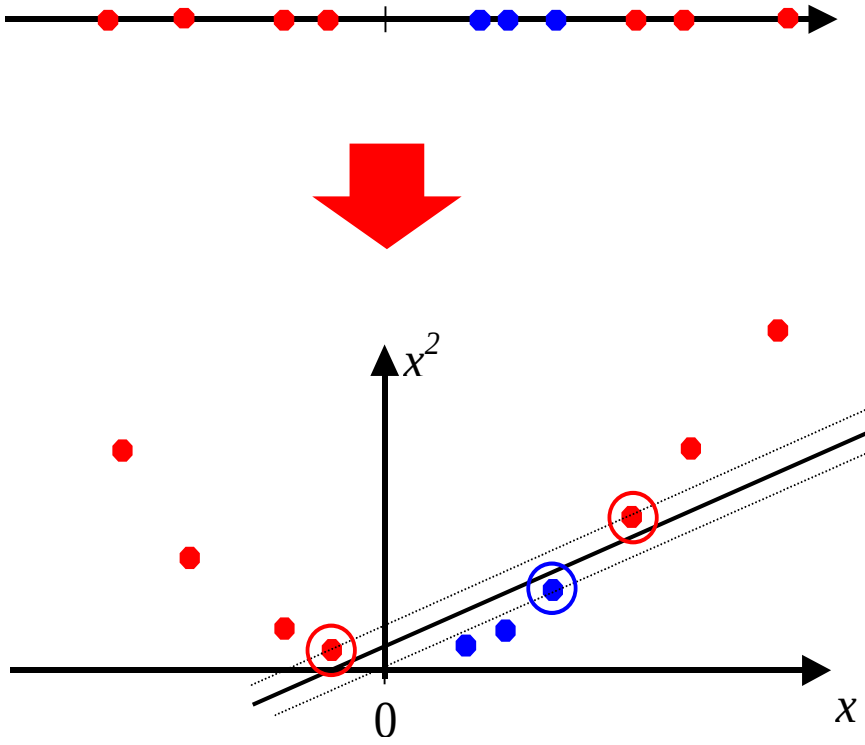
$$a_n = C \Rightarrow \xi_n \le 1 \ or \ \xi_n > 1$$

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} (t_n - \sum_{m \in S} a_m t_m x_n^T x_m)$$

支持向量机

- 非线性支持向量机

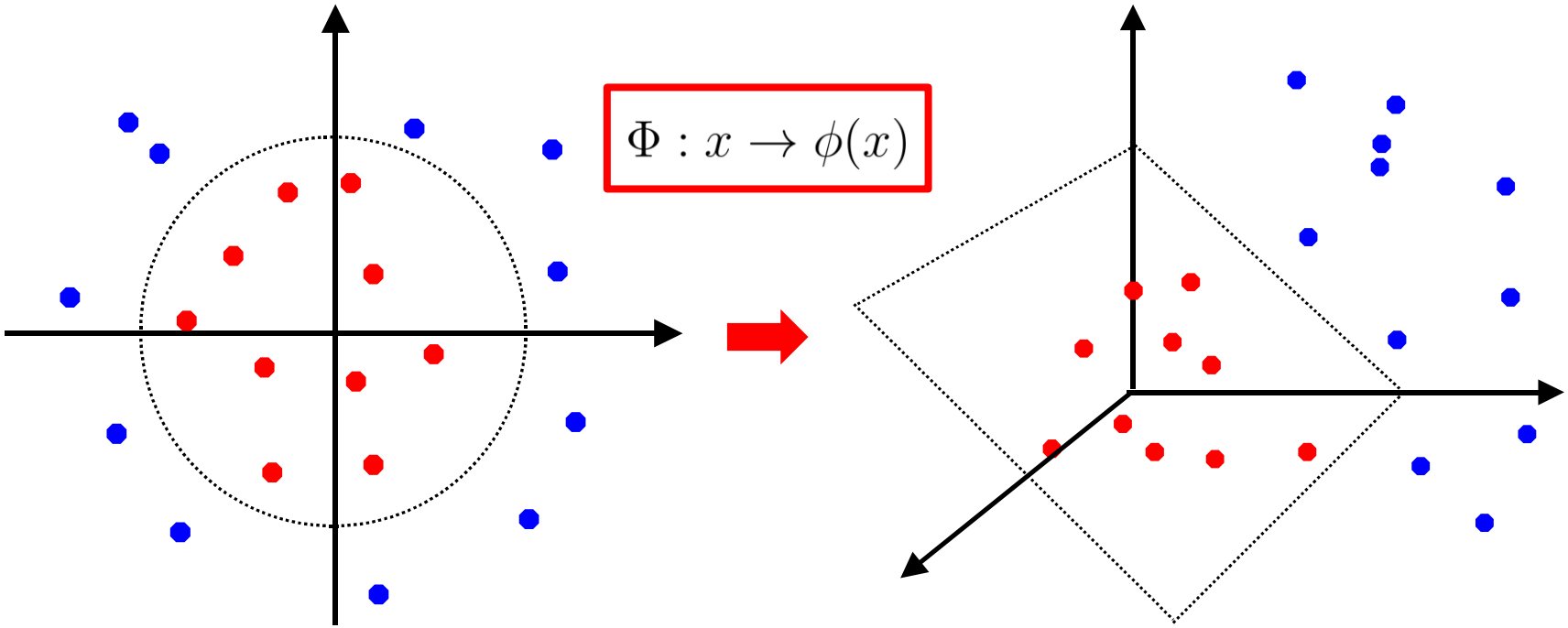
线性模型在解决复杂分类问题时适应性较差。而对于非线性可分的数据样本，可能通过适当的函数变换，将其在高维空间中转化为线性可分。



支持向量机

- **非线性支持向量机**

可以把样本*x*映射到某个高维特征空间$\phi(x)$，并在其中使用线性分类器。

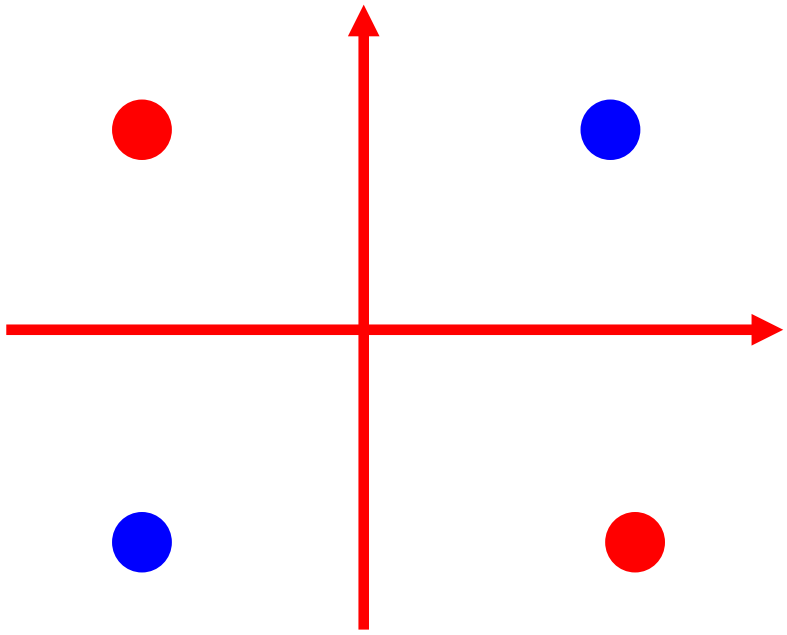


支持向量机

- **XOR问题**

二维样本集 $x = (x_1, x_2)$

第一类(0, 0)和 (1, 1)，第二类(1, 0)和 (0, 1)



将二维数据映射到三维

映射函数 $\phi(x) = (x_1, x_2, x_1 x_2)$

$$(0,0) \to (0,0,0)$$
$$(1,1) \mapsto (1,1,1)$$
$$(0,1) \mapsto (0,1,0)$$
$$(1,0) \to (1,0,0)$$

线性不可分 线性可分

支持向量机

- 非线性支持向量机

利用一个固定的非线性映射将数据映射到特征空间学习的线性分类器等价于基于原始数据学习的非线性分类器。

$$y(x) = w^T x + b \quad \Rightarrow \quad y(x) = w^T \phi(x) + b$$

决策时

$$y(x) = \sum_{n=1}^{N} a_n t_n x^T x_n + b \quad \Rightarrow \quad y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b$$

$$k(x, x_n) = \phi(x)^T \phi(x_n)$$

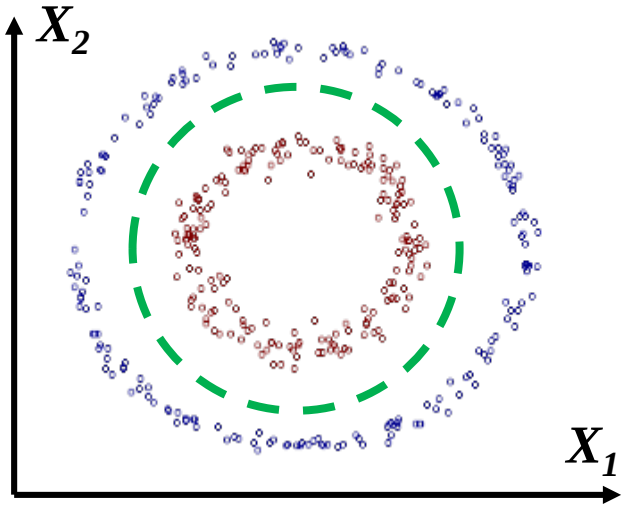
核函数

核函数在特征空间中直接计算数据映射后的内积就像在原始输入数据的函数中计算一样，大大简化了计算过程。

支持向量机

- 非线性支持向量机

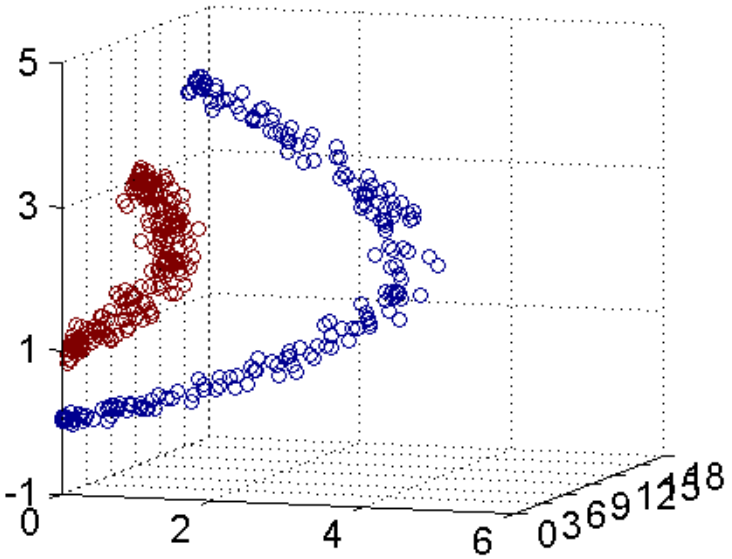
利用一个固定的非线性映射将数据映射到特征空间学习的线性分类器等价于基于原始数据学习的非线性分类器。



$$a_1 X_1 + a_2 X_1^2 + a_3 X_2 + a_4 X_2^2 + a_5 X_1 X_2 + a_6 = 0$$

$$Z_1 = X_1; Z_2 = X_1^2; Z_3 = X_2; Z_4 = X_2^2; Z_5 = X_1 X_2$$

$$\sum_{i=1}^{5} a_i Z_i + a_6 = 0 \qquad \phi : \mathbb{R}^2 \to \mathbb{R}^5$$



$$a_1 X_1^2 + a_2 (X_2 - c)^2 + a_3 = 0$$

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

$$\phi : \mathbb{R}^2 \to \mathbb{R}^3$$

支持向量机

- **非线性支持向量机**

利用一个固定的非线性映射将数据映射到特征空间学习的线性分类器等价于基于原始数据学习的非线性分类器。

$$a_1X_1 + a_2X_1^2 + a_3X_2 + a_4X_2^2 + a_5X_1X_2 + a_6 = 0 \quad \phi : \mathbb{R}^2 \to \mathbb{R}^5$$

原始样本增加到三维

$$a_1X_1^3 + a_2X_2^3 + a_3X_3^3 + a_4X_1^2X_2 + a_5X_1^2X_3 + a_6X_2^2X_1 + a_7X_2^2X_3 + a_8X_3^2X_1 + a_9X_3^2X_2 + a_{10}X_1X_2X_3 + a_{11}X_1^2 + a_{12}X_2^2 + a_{13}X_3^2 + a_{14}X_1X_2 + a_{15}X_2X_3 + a_{16}X_1X_3 + a_{17}X_1 + a_{18}X_2 + a_{19}X_3 + a_{20} = 0$$

$$\phi : \mathbb{R}^3 \to \mathbb{R}^{19}$$

维数大大增加
计算变得非常困难

$$k(x_1, x_2) = (< x_1, x_2 > +1)^2$$

利用核函数直接在原来的低维空间中进行计算不需要显式地写出映射后的结果，避免了先映射到高维空间中然后再根据内积的公式进行计算

支持向量机

- **非线性支持向量机**

根据问题和数据的不同，选择带有不同的核函数。

一些常用的核函数:

线性核: $k(x_1, x_2) = x_1^T x_2$

多项式核: $k(x_1, x_2) = (< x_1, x_2 > + R)^d$

高斯核: $k(x_1, x_2) = \exp\{-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\}$

Sigmoid核: $k(x_1, x_2) = \tanh(\beta_0 x_1^T x_2 + \beta_1)$

如何判断一个函数是否可以作为核函数?

Mercer定理:
$\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$**上的映射*k*是一个有效核函数(也称Mercer核函数)当且仅当对于训练样本其相应的核函数矩阵是对称半正定的,即对于任何平方可积函数*g*(*x*)有** $\int \int k(x, y) g(x) g(y) dx dy \geq 0$ **。**

序列最小优化算法

- **J. C. Platt(1999年提出)**

支持向量机的学习问题可以形式化为求解具有全局最优解的凸二次规划问题。许多方法可以用于求解这一问题，但当训练样本容量很大时，这些算法往往效率较低，以致无法使用。

序列最小优化算法(Sequential Minimal Optimization, SMO)是一种启发式算法。基本思想是：如果所有变量都满足此优化问题的KKT条件，那么这个问题的解就得到了。

SMO算法的特点是不断地将原二次规划问题分解为只有两个变量的二次规划问题，并对子问题进行解析求解，直到所有变量都满足KKT条件为止。因为子问题解析解存在，所以每次计算子问题都很快，虽然子问题次数很多，但是总体上还是高效的。

序列最小优化算法

- **SMO算法解凸二次规划问题**

$$\min_{a} \ \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m) - \sum_{n=1}^{N} a_n$$

$$s.t.\ 0 \le a_n \le C, n = 1, 2, ..., N$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

子问题有两个变量，一个是违反KKT条件最严重的，另一个有约束条件自动确定。两个变量中只有一个是自由变量。假设a_1, a_2为两个变量，$a_3, a_4, \ldots, a_N$固定，那么：

$$a_1 = -t_1 \sum_{n=2}^{N} a_n t_n = 0$$

即a_2确定，a_1也随之确定。

SMO算法包括：求解两个变量二次规划的解析方法和选择变量的启发式方法。

序列最小优化算法

- **两个变量二次规划的解析方法**

不失一般性，假设选择的两个变量是a_1和a_2，其他变量a_i (i=3, 4,…, N)是固定的，原问题的子问题可以写成：

$$\min_{a_1,a_2} W(a_1,a_2) = \frac{1}{2}K_{11}a_1^2 + \frac{1}{2}K_{22}a_2^2 + t_1t_2K_{12}a_1a_2$$
$$-(a_1+a_2) + t_1a_1\sum_{n=3}^{N} t_na_nK_{n1} + t_2a_2\sum_{n=3}^{N} t_na_nK_{n2} + const$$
$$s.t.\ a_1t_1 + a_2t_2 = -\sum_{n=3}^{N} t_na_n = \zeta$$
$$0 \le a_n \le C, i = 1,2$$

其中，$K_{mn} = K(x_m,x_n), m,n = 1,2,...N$**，**$\zeta$**是常数。**

序列最小优化算法

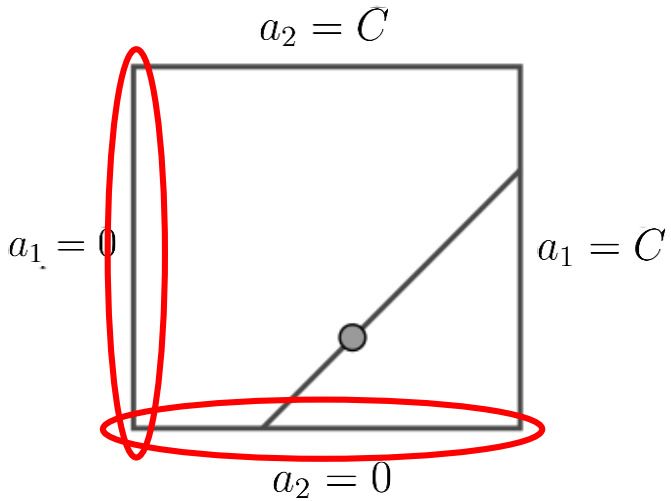
- **两个变量二次规划的解析方法**

两个变量(a_1, a_2)的约束可用二维空间中的图形表示

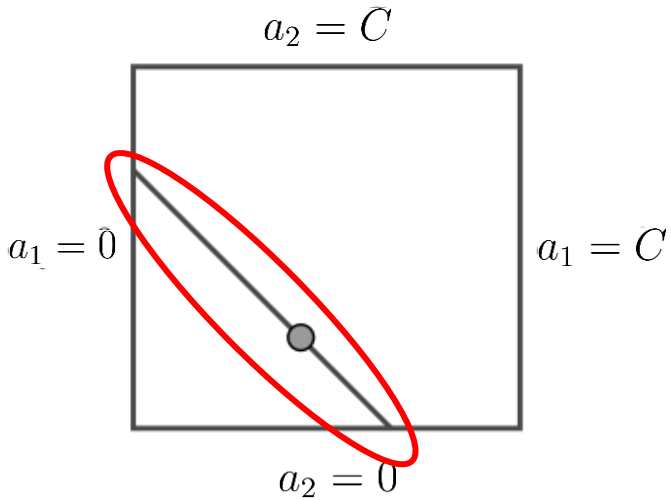
$$a_1 t_1 + a_2 t_2 = -\sum_{n=3}^{N} t_n a_n = \zeta$$

$$0 \le a_n \le C, i = 1, 2$$

目标函数在一条平行于对角线的线段上的最优值
两个变量的最优化问题转化为单变量最优化问题



$$t_1 \ne t_2 \Rightarrow a_1 - a_2 = k$$



$$t_1 = t_2 \Rightarrow a_1 + a_2 = k$$

序列最小优化算法

- **两个变量二次规划的解析方法**

考虑变量为a_2的最优化问题，假设问题的初始可行解为 a_1^{old}, a_2^{old}，最优解为 a_1^{new}, a_2^{new}，并且在沿着约束方向未经剪辑(未考虑不等式约束)时a_2的最优解为 $a_2^{new,\ unc}$。

$$0 \le a_n \le C, n = 1, 2$$

由于 a_2^{new} 需要满足不等式约束，所以其取值范围须满足 $L \le a_n^{new} \le H$，其中，*L*与*H*是 a_2^{new} 所在的对角线段端点的界。

如果 $t_1 \ne t_2$，**则** $L = \max(0, a_2^{old} - a_1^{old}),\ H = \min(C, C + a_2^{old} - a_1^{old})$

如果 $t_1 = t_2$，**则** $L = \max(0, a_2^{old} + a_1^{old} - C),\ H = \min(C, a_2^{old} + a_1^{old})$

首先求沿着约束方向未经剪辑时的a_2最优解 $a_2^{new,\ unc}$，然后再求剪辑后a_2的解 a_2^{new}。

序列最小优化算法

- **两个变量二次规划的解析方法**

记 $g(x)=\sum_{n=1}^{N}a_n y_n K(x_n,x)+b$

令 $E_n=g(x_n)-y_n=(\sum_{m=1}^{N}a_m y_m K(x_m,x_n)+b)-y_n, n=1,2$

即*n*=1, 2时，E_n为函数*g*(*x*)对输入x_n的预测值与真实输出y_n之差。

那么， $a_2^{new,unc}=a_2^{old}+\frac{y_2(E_1-E_2)}{\eta}$

其中， $\eta=K_{11}+K_{22}-2K_{12}=\|\Phi(x_1)-\Phi(x_2)\|^2$

经剪辑后a_2的解是 $a_2^{new}=\begin{cases}H, & a_2^{new,unc}>H\\ a_2^{new,unc}, & L\le a_2^{new,unc}\le H\\ L, & a_2^{new,unc}<L\end{cases}$

由 a_2^{new} **求得** $a_1^{new}=a_1^{old}+y_1y_2(a_2^{old}-a_2^{new})$

序列最小优化算法

- **变量的选择方法**

SMO算法在每个子问题中选择两个变量优化，其中至少一个变量是违反KKT条件的。

(1)第一个变量的选择(外层循环)

外层循环在训练样本中选取违反KKT条件最严重的样本点，并将其对应的变量作为第一个变量。检验是否满足KKT条件：

$$\begin{aligned} a_i = 0 &\iff y_i g(x_i) \geq 1 \\ 0 < a_i < C &\iff y_i g(x_i) = 1 \\ a_i = C &\iff y_i g(x_i) \leq 1 \end{aligned}$$

其中，$g(x_i) = \sum_{j=1}^{N} a_j y_j K(x_j, x_i) + b$

外循环首先遍历满足条件$0 < a_i < C$**的样本点，就是在边界上的支持向量点，如果都满足KKT条件，再遍历整个训练集。**

序列最小优化算法

- **变量的选择方法**

SMO算法在每个子问题中选择两个变量优化，其中至少一个变量是违反KKT条件的。

(2)第二个变量的选择(内层循环)

假设在外层循环已经找到第一个变量a_1，那么内层循环要找到第二个变量a_2，希望使a_2有足够大的变化。

$$a_2^{new,unc} = a_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$$

a_2^{new}是依赖于$|E_1 - E_2|$的

简单做法是选择a_2使$|E_1 - E_2|$最大

$$a_2^{new} = \begin{cases} H, & a_2^{new,unc} > H \\ a_2^{new,unc}, & L \le a_2^{new,unc} \le H \\ L, & a_2^{new,unc} < L \end{cases}$$

a_1已定，E_1确定，如果E_1为正，选择最小的E_i作为E_2；如果E_1为负，选择最大的E_i作为E_2。

序列最小优化算法

- **变量的选择方法**

SMO算法在每个子问题中选择两个变量优化，其中至少一个变量是违反KKT条件的。

(3)计算阈值*b*和差值E_i

在每次完成两个变量的优化后，都要重新计算阈值*b*。当$0 < a_1^{new} < C$**时，由KKT条件**$0 < a_i < C \iff y_i g(x_i) = 1$**可计算：**

$$b_1^{new} = -E_1 - y_1 K_{11}(a_1^{new} - a_1^{old}) - y_2 K_{21}(a_2^{new} - a_2^{old}) + b^{old}$$

同理，当$0 < a_2^{new} < C$**时，可计算：**

$$b_2^{new} = -E_2 - y_1 K_{12}(a_1^{new} - a_1^{old}) - y_2 K_{22}(a_2^{new} - a_2^{old}) + b^{old}$$

如果a_1^{new}, a_2^{new}**同时满足条件**$0 < a_i^{new} < C, i = 1, 2$**，那么** $b_1^{new} = b_2^{new}$**。**

如果a_1^{new}, a_2^{new}**是*0*或者*C*，那么**b_1^{new}, b_2^{new}**以及他们之间的数都是符合KKT条件的阈值，这是取它们的中点作为**b^{new}**。之后更新E_i**

$$E_i^{new} = \sum_{S} y_j a_j K(x_i, x_j) + b^{new} - y_i$$

序列最小优化算法

- **SMO算法**

输入：训练数据集 $T=(x_1,y_1),(x_2,y_2),...,(x_N,y_N)$**，其中** $x_i \in \mathcal{X}=R^n$，$y_i \in \mathcal{Y}=\{-1,\ +1\}, i=1,2,...,N$**，精度**$\epsilon$**；**

输出：近似解 $\hat{a}$ **。**

（1）取初值 $a^{(0)}=0$**，令** $k=0$**；**

（2）选取优化变量 a_1^k, a_2^k**，解析求解两个变量的最优化问题，求得最优解** a_1^{k+1}, a_2^{k+1}**，更新*a*为** a^{k+1}**；**

$$\sum_{i=1}^{N} a_i y_i = 0$$

$$0 \le a_i \le C, i=1,2,...,N$$

（3）若在精度ϵ **范围内满足停止条件**

其中，

$$g(x_i)=\sum_{j=1}^{N} a_j y_j K(x_j,x_i)+b$$

$$y_i \cdot g(x_i)=\begin{cases} \ge 1, & \{x_i | a_i=0\} \\ =1, & \{x_i | 0<a_i<C\} \\ \le 1, & \{x_i | a_i=C\} \end{cases}$$

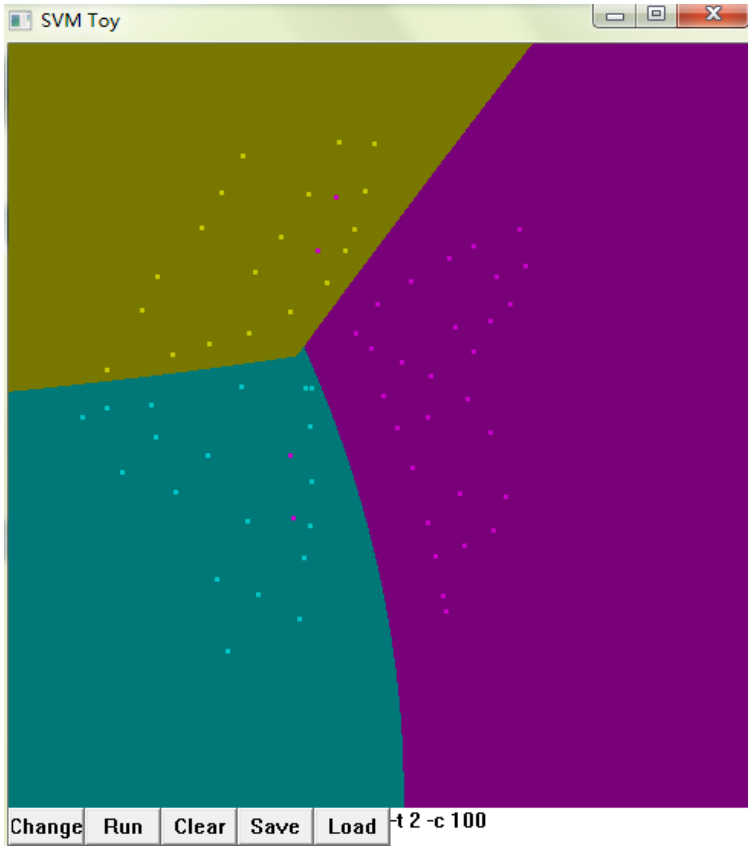
则转(4)；否则令 $k=k+1$**，转(2)；**

（4）取 $\hat{a}=a^{(k+1)}$**。**

支持向量机

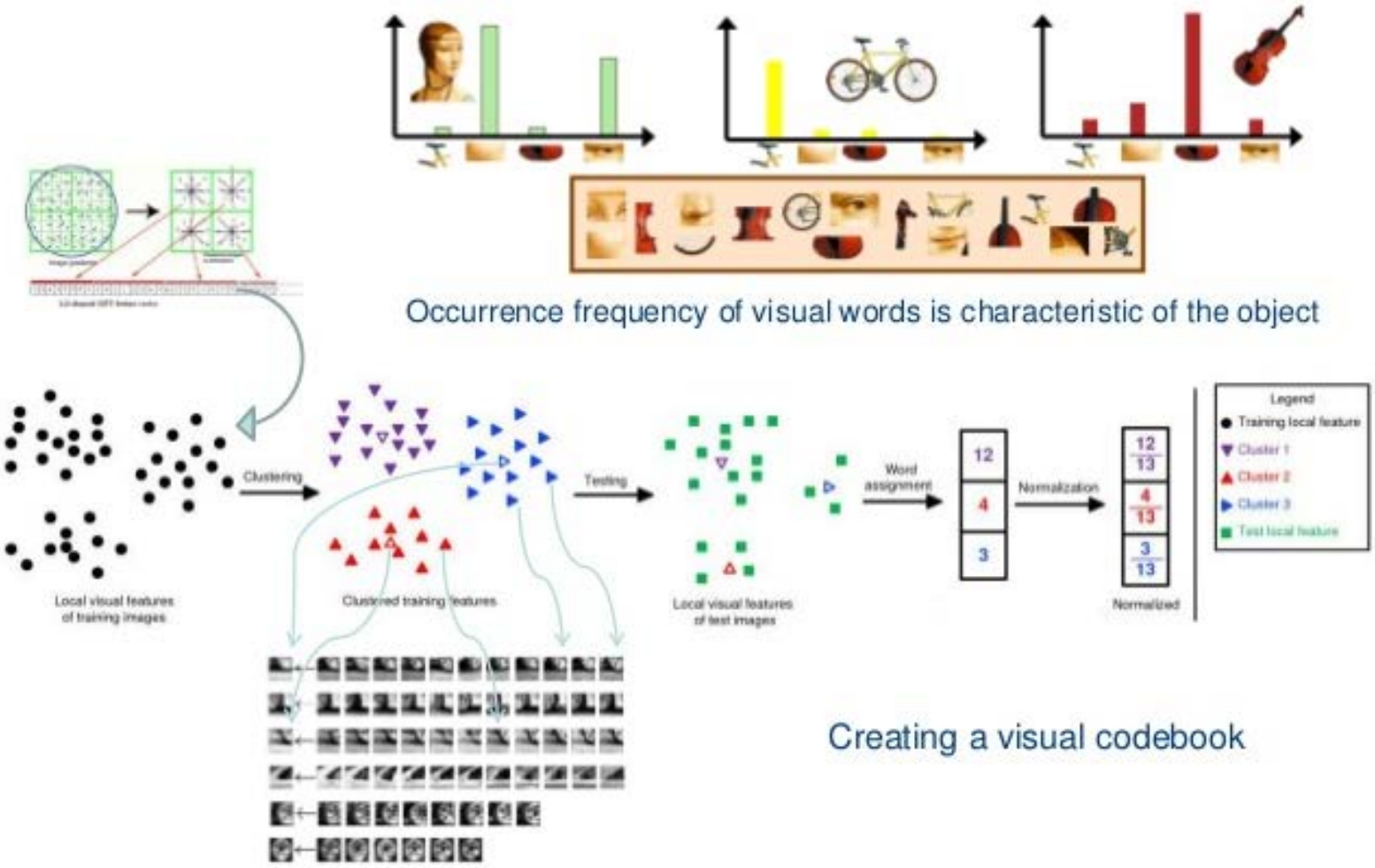
- 支持向量机工具

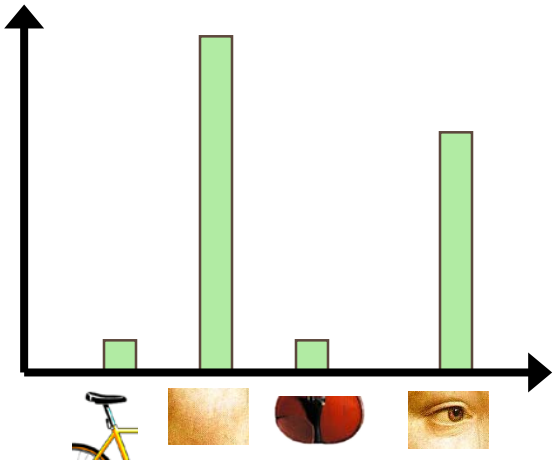
LibSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

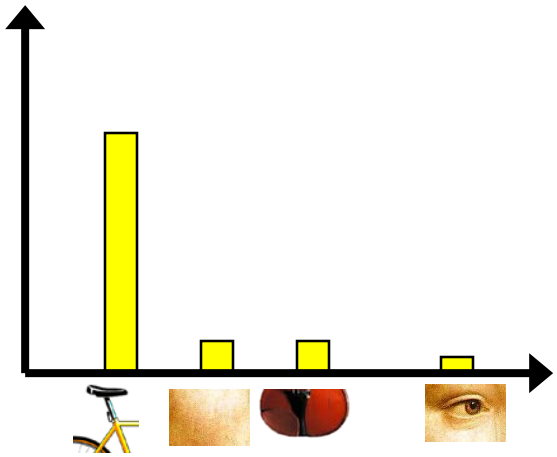


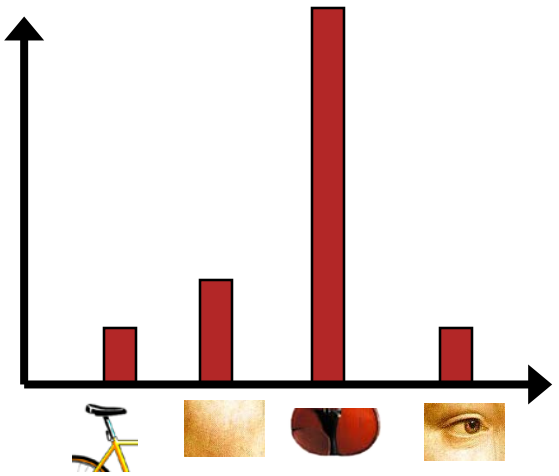
支持向量机应用

- 物体识别









VC维

- **SLT中衡量函数集性能的指标**

为了研究经验风险最小化函数集的学习一致收敛速度和推广性，统计学力理论定义了一些指标来衡量函数集的性能，其中最重要的就是VC维(Vapnik-Chervonenkis Dimension)

VC维定义：

对于一个指示函数(即只有0和1两种取值的函数)集，如果存在h个样本能够被函数集里的函数按照所有可能的2^h种形式分开，则称函数集能够把h个样本打散，函数集的VC维就是能够打散的最大样本数目。

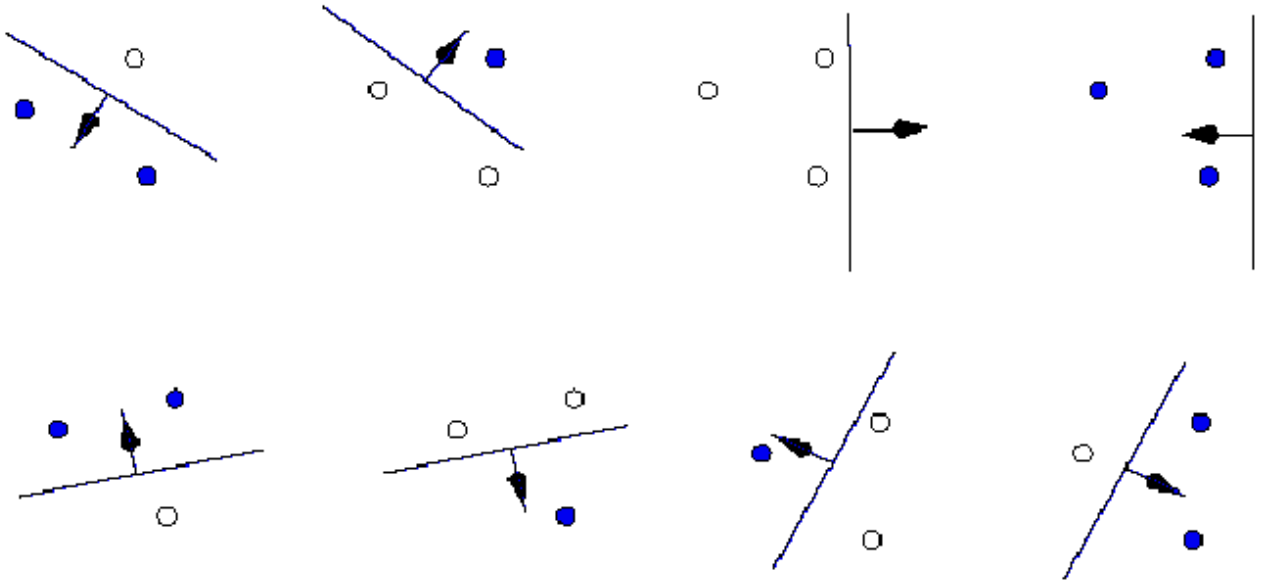
VC维

- **SLT中衡量函数集性能的指标**

对于2维空间的线性分类面函数

$$y = w_0 + w_1 x_1 + w_2 x_2$$

其能够以任意方式打散(分类)的最大样本数为3。



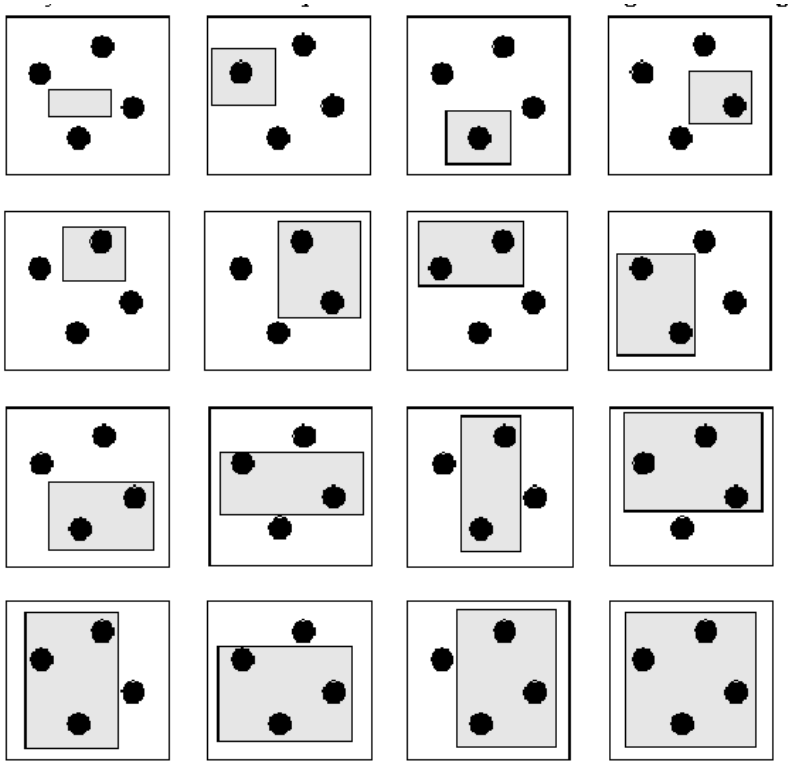
VC维

- **SLT中衡量函数集性能的指标**

一般来说,对于*n*维空间的线性分类面函数

$$y = w_0 + w_1 x_1 + \ldots + w_n x_n$$

的VC维为*n*+1。



VC维

- **SLT中衡量函数集性能的指标**

一般而言，VC维反映了函数集的学习能力，VC维越大则学习机器越复杂，学习内容量就越大。

目前没有通用的关于任意函数集VC维计算的理论，只对一些特殊的函数集知道其函数维。

- **在n维实数空间中线性分类器和线性实函数的VC维是$n+1$**
- **而$f(x, a)=\sin(ax)$的VC维则为无穷大**

如何用理论和试验的方法计算其VC维是当前SLT中一个待研究的问题。