软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

机器学习

刘祥龙
北京航空航天大学计算机学院
软件开发环境国家重点实验室

2018年11月6日

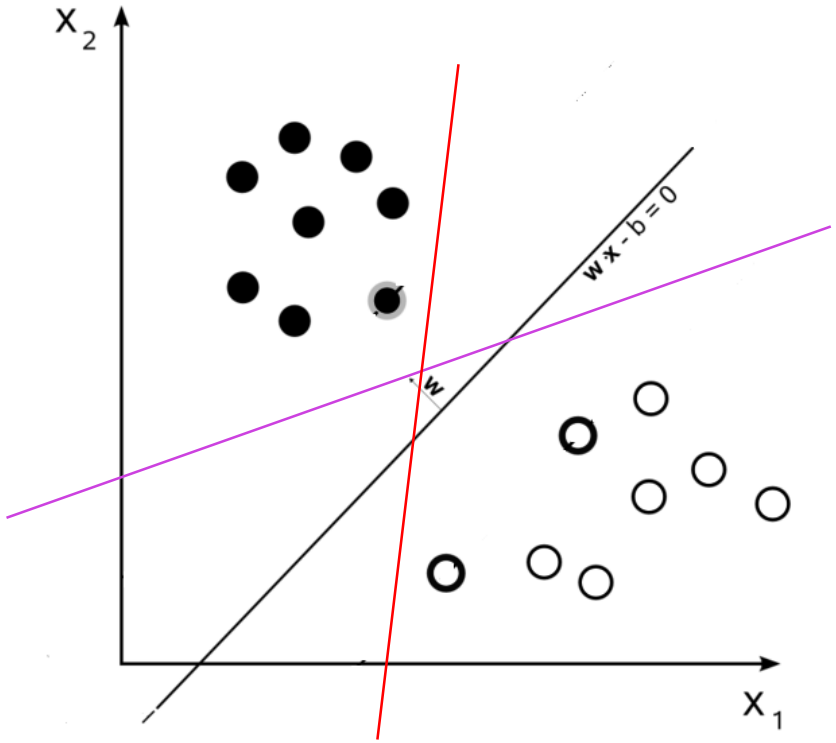
软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

支持向量机

线性可分支持向量机

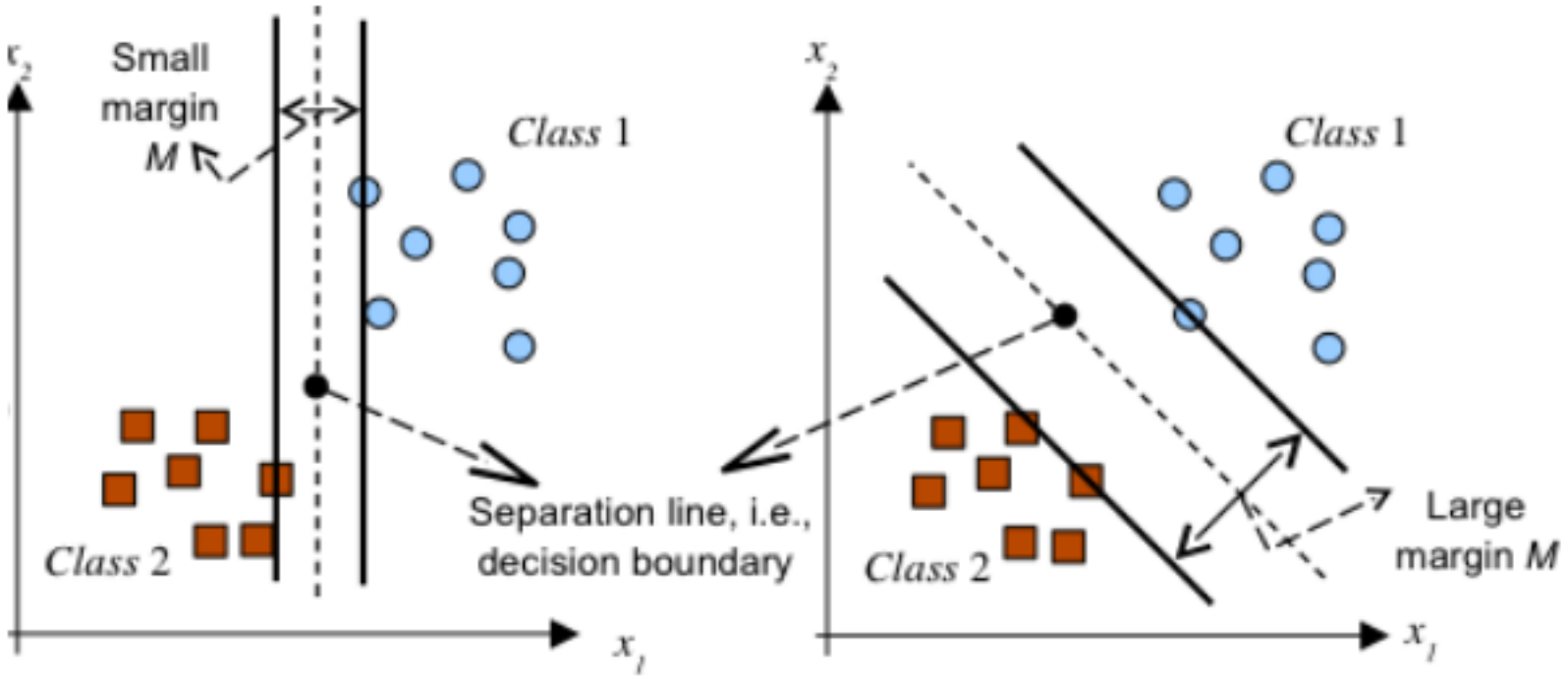
- 训练数据集
  - $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\},\ x_i \in R^n, y_i \in \{+1, -1\}$

- 二分类目标
  - 分离超平面$w^T x_i + b = 0$
  - 二分类函数$y_i = \text{sign}(w^T x_i + b)$
  - 若线性可分
    - 感知机：存在无数分离超平面
    - 线性可分支持向量机：唯一解



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。



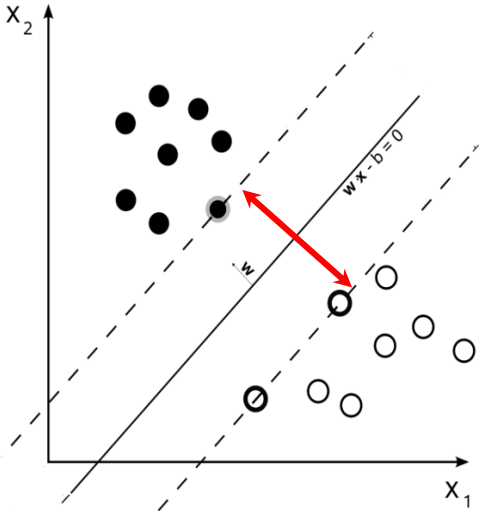
线性可分支持向量机

- 函数间隔
  - $|w^T x + b|$表示点相对分离超平面的距离，反映分类确信度
  - $y(w^T x + b)$同时反映预测与实际分类是否一致及确信度
  - 数据集T上函数间隔

$$\min_{i=1,2,\ldots,N} y_i(w^T x_i + b)$$

- 几何间隔
  - $(\mathrm{w}, \mathrm{b}) \rightarrow (2\mathrm{w}, 2\mathrm{b})$时超平面不变，函数间隔发生变化
  - 函数间隔→几何间隔，可约束$\|w\| = 1$

$$\gamma = \min_{i=1,2,\ldots,N} \frac{y_i(w^T x_i + b)}{\|w\|}$$



线性可分支持向量机

- 间隔最大化
  - 正确划分，几何间隔最大
  - 直观解释：充分大的确信度对数据进行分类
    - *最难分的点以足够大的确信度分开*
    - *对未知新数据有很好的泛化能力*
- 最大间隔分类超平面

$$\max_{w,b} \gamma \quad s.t. \quad \frac{y_i(w^T x_i + b)}{\|w\|} \geq \gamma$$

$\hat{\gamma} = \|w\|\gamma$

$$\max_{w,b} \frac{\hat{\gamma}}{\|w\|} \quad s.t. \quad y_i(w^T x_i + b) \geq \hat{\gamma}$$

$\hat{\gamma} = 1$

$$\max_{w,b} \frac{1}{\|w\|} \quad s.t. \quad y_i(w^T x_i + b) \geq 1$$

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad s.t. \quad y_i(w^T x_i + b) \geq 1$$

线性可分支持向量机

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t.\ \ y_i(w^T x_i + b) \geq 1$$

- 对偶算法
  - 拉格朗日函数：拉格朗日乘子$\alpha = (\alpha_1, \ldots, \alpha_N)^T$

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

  - 对偶问题：$\min\limits_{w,b} \max\limits_{\alpha} L(w, b, \alpha)$ -> $\max\limits_{\alpha} \min\limits_{w,b} L(w, b, \alpha)$

线性可分支持向量机

- 对偶算法
  - 对偶问题

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

(1) 求 $\min_{w,b} L(w,b,\alpha)$：对w, b分别求导

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i,$$

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

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

线性可分支持向量机

- 对偶算法
  - 对偶问题：$\max\limits_{\alpha} \min\limits_{w,b} L(w,b,\alpha)$

(2) 求 $\max\limits_{\alpha} \min\limits_{w,b} L(w,b,\alpha)$

$$\max_{\alpha} \ -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i^T x_j) + \sum_{i=1}^{N}\alpha_i$$

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

  - 最优解：由KKT条件$w^* = \sum_{i=1}^{N}\alpha_i^* y_i x_i$和$\alpha_i^*(y_i(w^T x_i + b) - 1) = 0$，存在$\alpha_i^* > 0$

$$w^* = \sum_{i=1}^{N}\alpha_i^* y_i x_i, \quad b^* = y_j - \sum_{i=1}^{N}\alpha_i^* y_i (x_i^* x_j)$$

超平面法向量w^*是支持向量的线性组合

  - 决策函数：$f(x) = \text{sign}\left(\sum_{i=1}^{N}\alpha_i^* y_i (x^T x_i) + b^*\right), \ \alpha_i^* > 0$

对偶问题

$$\begin{aligned}&\min\ f_0(x)\\ &s.t.\ f_i(x)\le 0,\quad i=1,\dots,m\\ &\quad\ h_i(x)=0,\quad i=1,\dots,p\end{aligned}$$

$$L(x,\lambda,\nu)=f_0(x)+\sum_{i=1}^{m}\lambda_i f_i(x)+\sum_{i=1}^{p}\nu_i h_i(x)$$

$$g(\lambda,\nu)=\inf_{x\in\mathcal{D}}L(x,\lambda,\nu)=\inf_{x\in\mathcal{D}}\left(f_0(x)+\sum_{i=1}^{m}\lambda_i f_i(x)+\sum_{i=1}^{p}\nu_i h_i(x)\right)$$

可以证明，Lagrange 对偶函数构成了原问题(1)最优值 p^* 的下界，

即对任意 $\lambda \succ= 0$ 和 ν 有

$$g(\lambda,\nu)\le p^*$$

KKT条件

对于具有等式和不等式约束的一般优化问题

$$
\begin{aligned}
&\min f(\mathbf{x}) \\
&s.t.\, g_j(\mathbf{x}) \le 0 (j = 1, 2, \cdots, m) \\
&h_k(\mathbf{x}) = 0 (k = 1, 2, \cdots, l)
\end{aligned}
$$

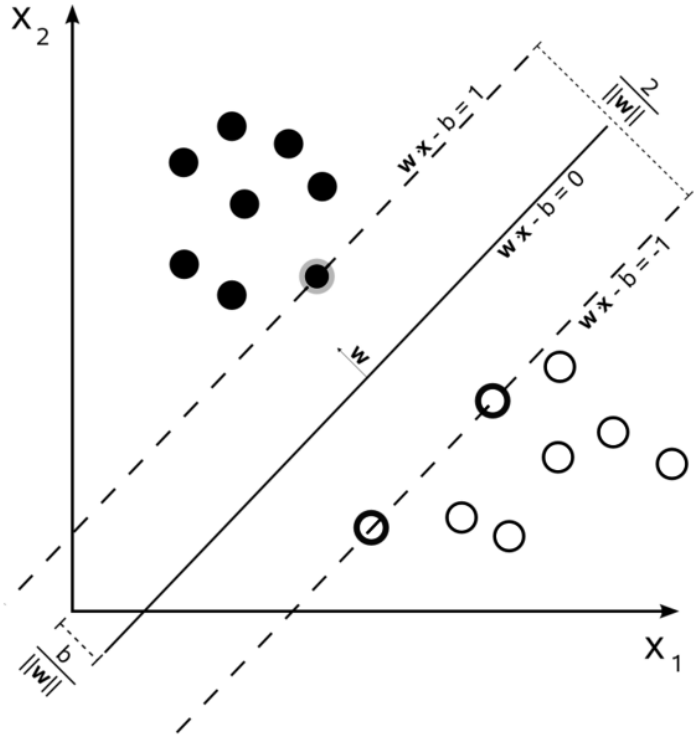
KKT条件给出了判断 $\mathbf{x}^*$ 是否为最优解的**必要条件**，即：

$$
\begin{cases}
\dfrac{\partial f}{\partial x_i} + \sum\limits_{j=1}^{m} \mu_j \dfrac{\partial g_j}{\partial x_i} + \sum\limits_{k=1}^{l} \lambda_k \dfrac{\partial h_k}{\partial x_i} = 0, \ (i = 1, 2, \ldots, n) \\
h_k\left(\mathbf{x}\right) = 0, (k = 1, 2, \cdots, l) \\
\mu_j g_j\left(\mathbf{x}\right) = 0, (j = 1, 2, \cdots, m) \\
\mu_j \ge 0.
\end{cases}
$$

线性可分支持向量机

- 支持向量
  - KKT互补条件 $\alpha_i^*(y_i(w^T x_i + b) - 1) = 0$
  - 决策函数$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i^* y_i (x^T x_i) + b^*\right),\ \alpha_i^* > 0$
  - 对应$\alpha_i^* > 0$的样例：$y_i(w^T x_i + b) = 1$，落在间隔边界

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t.\ \ y_i(w^T x_i + b) \geq 1$$



支持向量机

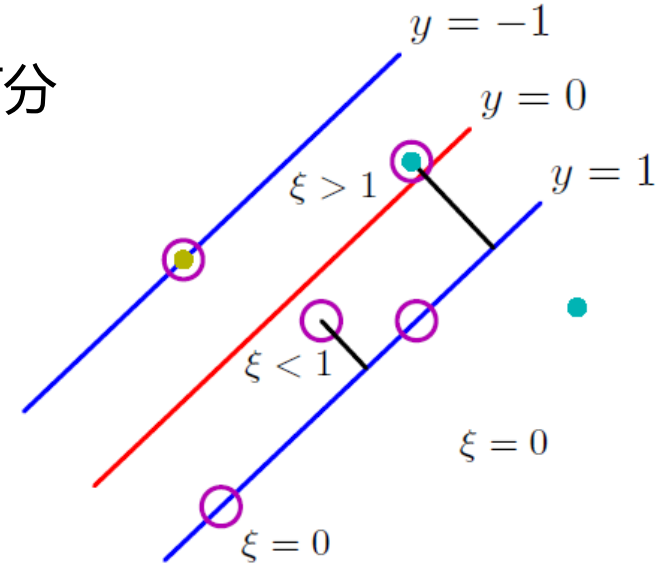
- 处理噪声和离群点
  - 求解最优分类面的时间代价大还可能导致泛化性能差。因此，对于分布有交集的数据需要有一定范围内的“错分”，又有较大分界区域的广义最优分类面。

准确性

泛化性

线性支持向量机

- 线性不可分训练数据集
  - 通常存在异常点，去除后可线性可分
  - 硬间隔无法满足



- 目标函数

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$$

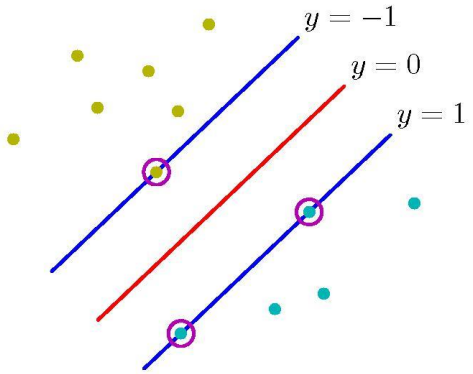
$$s.t.\ \ y_i(w^T x_i + b) \geq 1 - \xi_i$$

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。

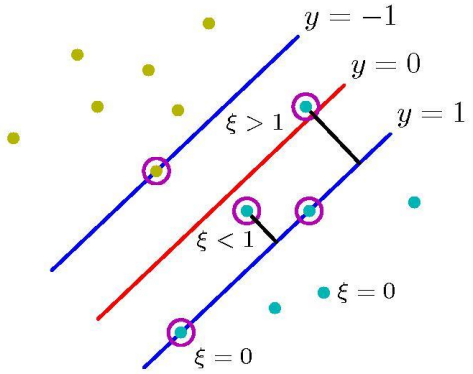
硬间隔

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



软间隔

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



线性支持向量机

- 对偶算法
  - 拉格朗日函数求$L(w,b,\xi,\alpha,\mu)$
  - 求解对偶
    - 求$\min\limits_{w,b,\xi} L(w,b,\xi,\alpha,\mu)$
    - 求$\max\limits_{\alpha} \ -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i^T x_j)+\sum_{i=1}^{N}\alpha_i$，满足$C-\alpha_i-\mu_i=0$
  - 最优解：由KKT条件$w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i$，$\mu_i^*\xi_i^*=0$，及$\alpha_i^*(y_i(w^T x_i+$

超平面法向量w^*是支持向量的线性组合

线性支持向量机

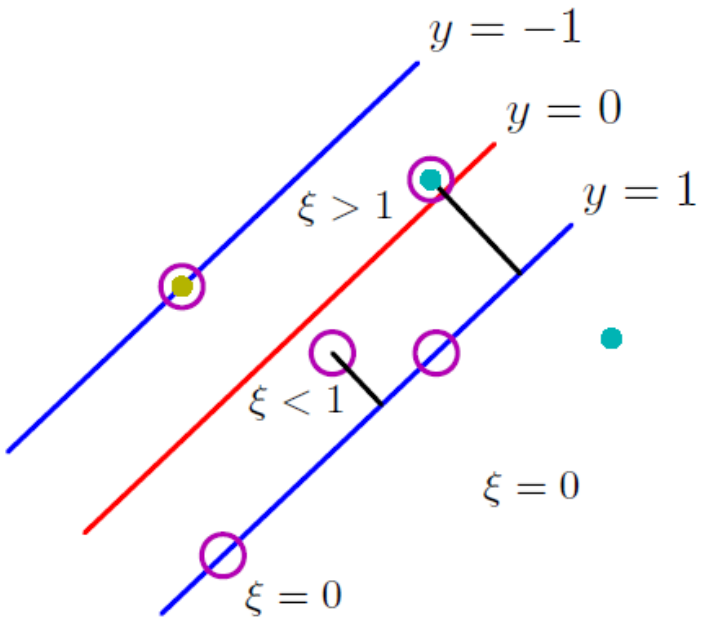
- 支持向量

$$\mu_i^*\xi_i^* = 0, \quad \alpha_i^*(y_i(w^T x_i + b) - 1 + \xi_i) = 0, \quad C - \alpha_i - \mu_i = 0$$

  - $0 < \alpha_i^* < C$: $y_i(w^T x_i + b) - 1 + \xi_i = 0$
  - $\alpha_i^* < C \rightarrow \xi_i = 0$，$x_i$落在间隔边界
  - $\alpha_i^* = C \rightarrow \xi_i < 1, \quad \xi_i = 1, \quad \xi_i > 1$

- 示例

$$C - \alpha_i - \mu_i = 0$$
$$\mu_i^*\xi_i^* = 0$$



序列最小最优化算法

- 凸二次规划对偶问题
  - 样本容量大时，直接求解效率较低
  - 序列最小最优化（SMO）算法求解

$$\max_{\alpha} \ -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_{i=1}^{N}\alpha_i$$

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

- 启发式算法
  - 所有变量满足KKT条件，选择最不满足条件的两个变量，固定其他变量，迭代求解。
  - 两变量构成二次规划问题

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。

序列最小最优化算法

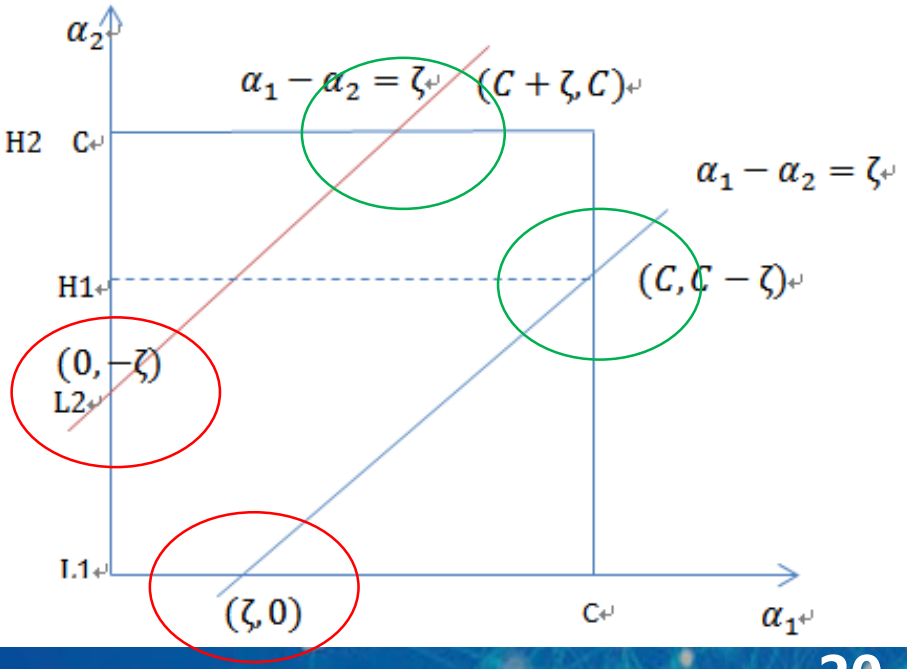
- 两变量二次规划

$$\min_{\alpha_1,\alpha_2} W(\alpha_1,\alpha_2)$$

$$s.t.\ \ \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3} y_i \alpha_i\,,\ \ C \geq \alpha_i \geq 0, i = 1,2$$

  - 迭代求解：$(\alpha_1^{old}, \alpha_2^{old})$ -> $(\alpha_1^{new}, \alpha_2^{new})$
  - $\frac{\partial W}{\partial \alpha_2} = 0 \Rightarrow \alpha_2^{new} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$

- 剪辑α_2^{new}，变量约束范围
  - y_1, y_2异号：最小值，最大值
  - y_1, y_2同号：最大值，最小值



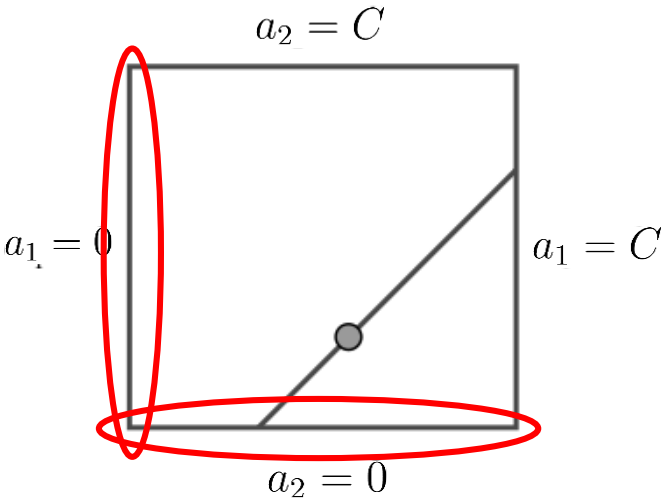
序列最小优化算法

- 两个变量二次规划的解析方法
  - 两个变量(a1, a2)的约束可用二维空间中的图形表示

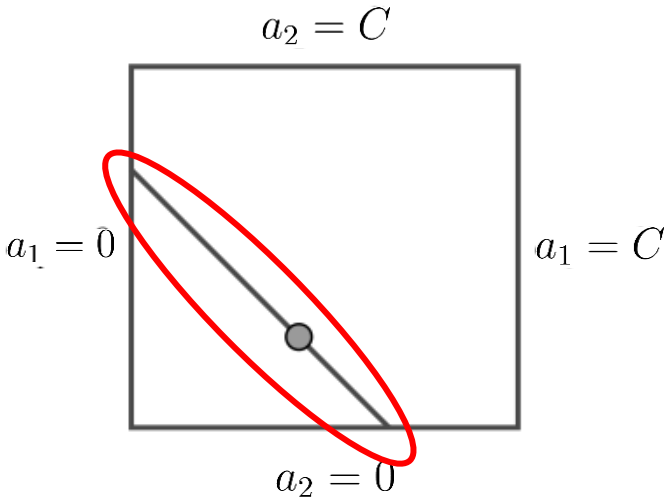
$$a_1 t_1 + a_2 t_2 = -\sum_{n=3}^{N} t_n a_n = \zeta$$

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

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



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



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

序列最小最优化算法

- 变量选取
  - 第一个变量选择（外层循环）：违反KKT条件最严重的样本点
    - $0 < \alpha_i^* < C$：$y_i(w^T x_i + b)=1$
    - $\alpha_i^* = 0$：$y_i(w^T x_i + b) \geq 1$
    - $\alpha_i^* = C$：$y_i(w^T x_i + b) \leq 1$
  - 第二个变量选择（内层循环）：使得α_2变化足够大
    - $\alpha_2^{new} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$，$E_i = [\sum_{i=1}^{N} \alpha_i^* y_i \mathrm{K}(\mathrm{x}, x_i) + b^*] - \mathrm{y_i}$
    - $E_1 - E_2$足够大
  - 更新b和E_i
  - 重复上述步骤，直到KKT条件满足

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a1, a2为两个变量， a3, a4,..., aN固定，那么：

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

  - 即a2确定，a1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a1和a2，其他变量ai (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，$K_{mn} = K(x_m, x_n), m, n = 1, 2, ...N$ ，ζ是常数。

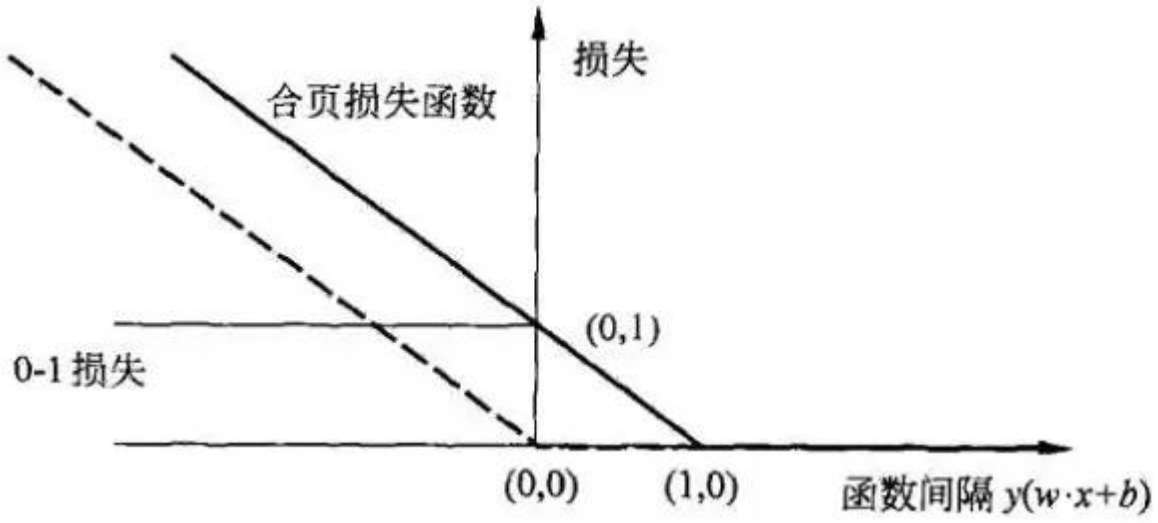
合页损失函数

- 另外一种解释

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad s.t. \quad y_i(w^T x_i + b) \geq 1$$

$$\Rightarrow \quad \min_{w,b} \sum_{i=1}^{N} [1 - y_i(w^T x_i + b)]_+ + \lambda\|w\|^2$$

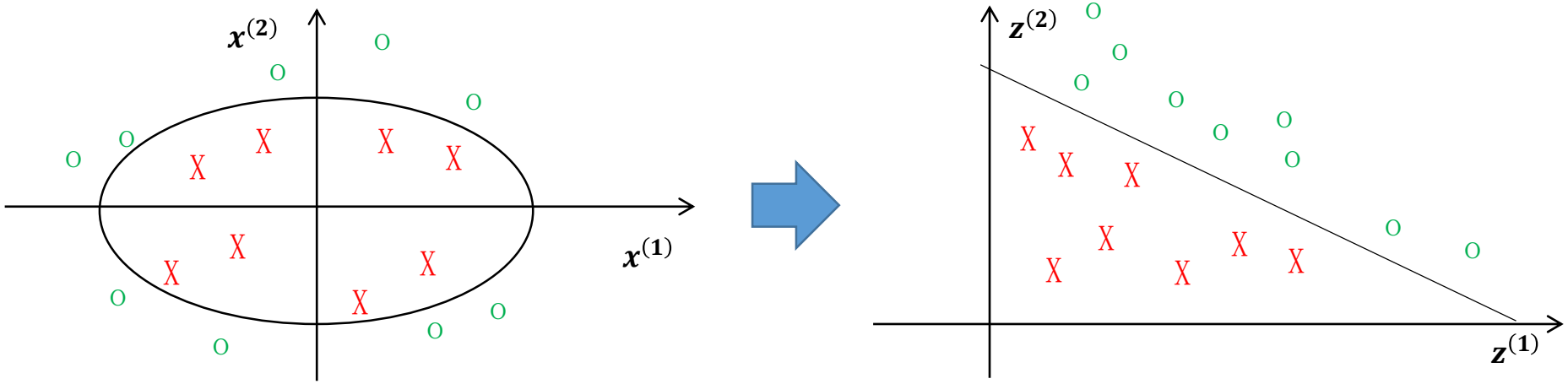
$$[z]_+ = \begin{cases} z, z > 0 \\ 0, z \leq 0 \end{cases}$$



核方法

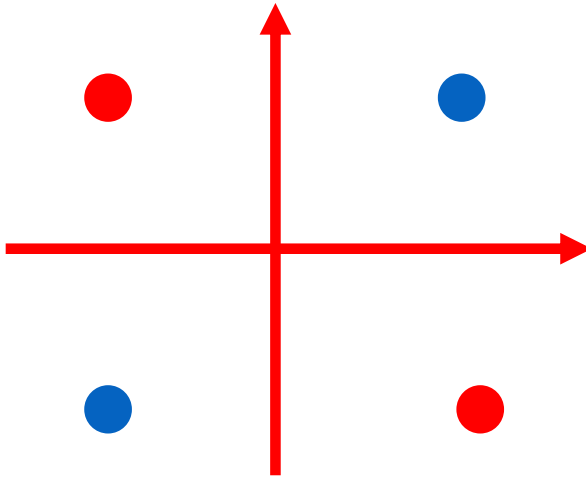
- 核技巧
  - 线性不可分，可采取非线性映射
  - 例子：线性不可分$(x^{(1)}, x^{(2)})$分类平面$w_1(x^{(1)})^2 + w_2(x^{(2)})^2 + b = 0$
  - 非线性映射$z^{(1)} = (x^{(1)})^2$，$z^{(2)} = (x^{(2)})^2$，线性可分

$$w_1 z^{(1)} + w_2 z^{(2)} + b = 0$$



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

$$\phi(x) = (x_1, x_2, x_1 x_2)$$

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

线性不可分　线性可分

核方法

- 定义：
  - 映射$\phi(x)$：输入空间 → 特征空间（希尔伯特空间）
  - 核函数$\mathrm{K}(\mathrm{x},\mathrm{z}) = \phi(x)^T\phi(z)$
- 思想
  - 显式定义核函数，而不显式定义映射函数
  - 直接计算$\mathrm{K}(\mathrm{x},\mathrm{z})$容易，$\phi(x)$通常无穷维，难以定义
- 支持向量机
  - 决策函数$f(x) = sign(\sum_{i=1}^{N} \alpha_i^* y_i (x^T x_i) + b^*)$
  - 内积$x^T x_i$由核函数代替$\mathrm{K}(\mathrm{x}, x_i)$，实际上进行了特征映射
  - 决策函数$f(x) = sign(\sum_{i=1}^{N} \alpha_i^* y_i \mathrm{K}(\mathrm{x}, x_i) + b^*)$

核方法

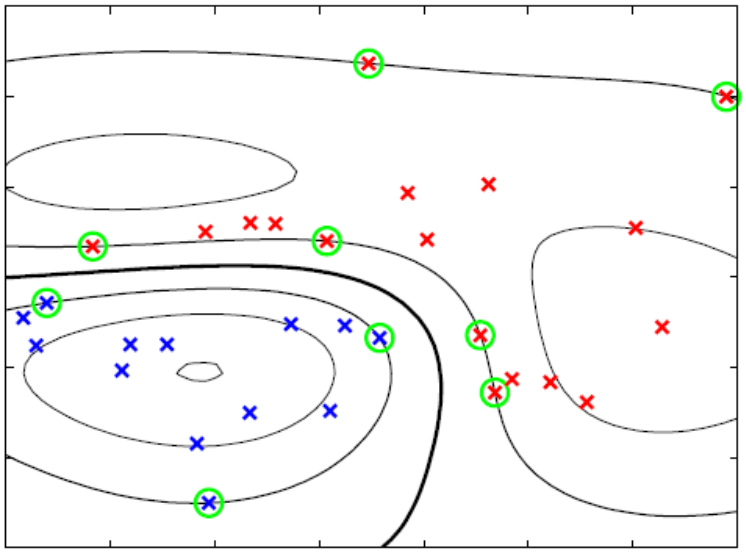
- 正定核
  - 核函数$\mathrm{K}(\mathrm{x}, z)$应该满足什么性质？→ 构造或判断
  - Gram矩阵：一组向量的内积构成矩阵的元素
  - 正定：对任意的向量v， $\mathrm{v}^{\mathrm{T}}\mathrm{Kv} \geq 0$
  - 核函数： $\mathrm{K}(\mathrm{x}, z)$正定核函数， ⇔ $\mathrm{K}(\mathrm{x}, z)$对应的Gram矩阵半正定

$$\mathrm{K} = \left[K\left(x_i, x_j\right)\right]_{m \times m}$$

    - 必要性： $\mathrm{K} = \left[K\left(x_i, x_j\right)\right]_{m \times m} = \left[\phi(x_i)^T \phi\left(x_j\right)\right]_{m \times m}$ => $\mathrm{v}^{\mathrm{T}}\mathrm{Kv} = [\sum_i v_i \phi(x_i)]^2 \geq 0$
    - 充分性： Gram矩阵可定义$\phi: \mathrm{x} \rightarrow \mathrm{K}(., \mathrm{x})$ => $\mathrm{K}(\mathrm{x}, z) = \phi(x)^T \phi(z)$
  - 正定核函数判定：
    - 对称函数
    - 对任意的$\mathrm{K} = \left[K\left(x_i, x_j\right)\right]_{m \times m}$半正定

核方法

- 常用核函数
  - 多项式核函数$\mathrm{K}(\mathrm{x}, z) = (x^T z + 1)^P$
  - 高斯核函数$\mathrm{K}(\mathrm{x}, z) = exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$
  - 字符串核函数



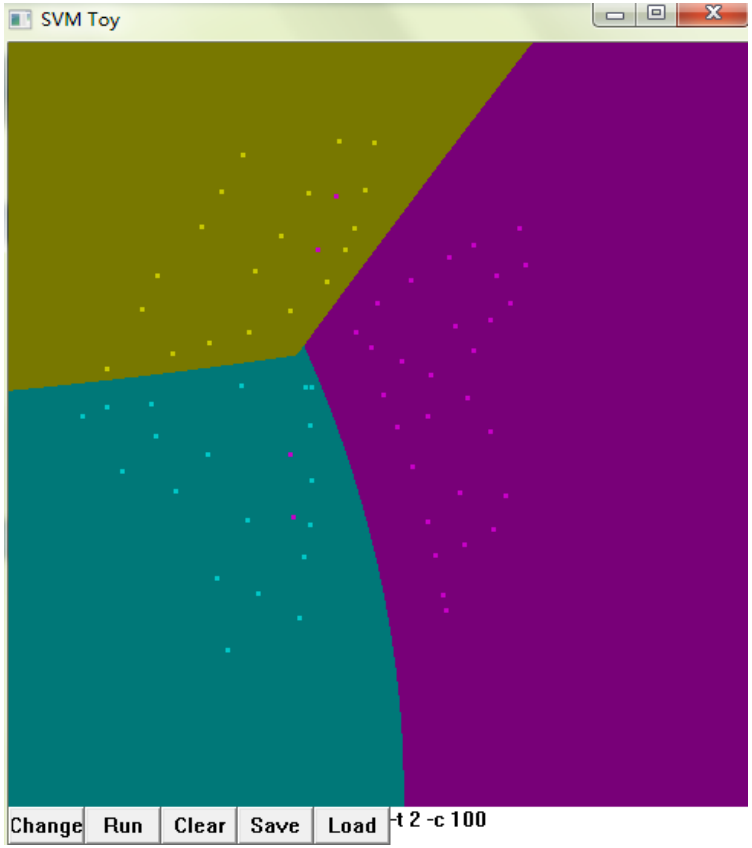
- 非线性支持向量机
  - $f(x) = \mathrm{sign}(\sum_{i=1}^{N} \alpha_i^* y_i \mathrm{K}(\mathrm{x}, x_i) + b^*)$
  - 算法

$$\max_{\alpha} \ -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) + \sum_{i=1}^{N}\alpha_i$$

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

支持向量机

- 支持向量机工具
  - LibSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/



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)

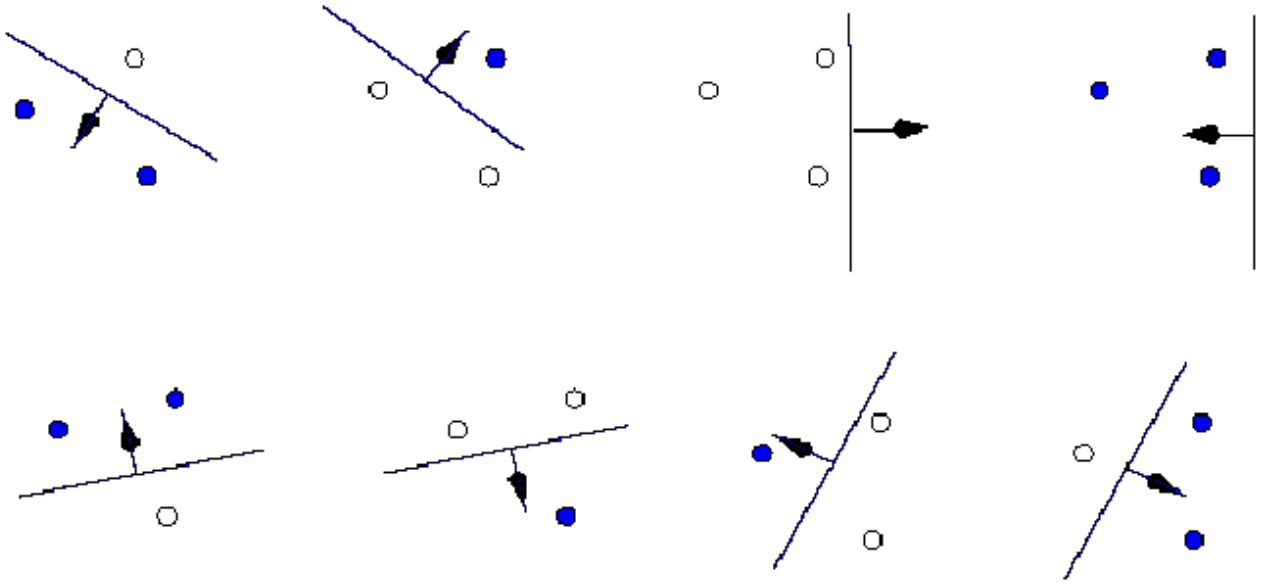
VC维定义：

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
  - 其能够以任意方式打散(分类)的最大样本数为3。

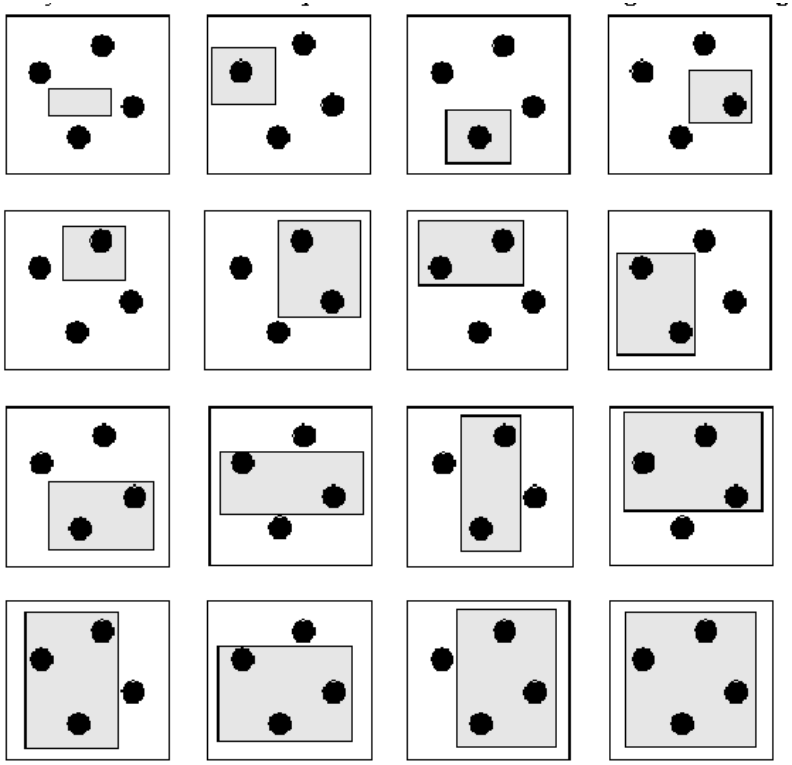
$$y = w_0 + w_1 x_1 + w_2 x_2$$



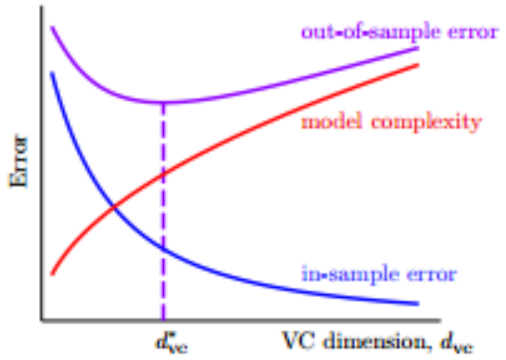
VC维

- 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
  - 的VC维为n+1。

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



VC维



- $d_{vc} \uparrow$: $E_{in} \downarrow$ but $\Omega \uparrow$
- $d_{vc} \downarrow$: $\Omega \downarrow$ but $E_{in} \uparrow$
- best d_{vc}^* in the middle

powerful $\mathcal{H}$ not always good!

$$\mathbb{P}_{\mathcal{D}}\left[|E_{in}(g) - E_{out}(g)| > \epsilon\right] \quad \leq \quad 4(2N)^{d_{vc}} \exp\left(-\tfrac{1}{8}\epsilon^2 N\right)$$

theory: $N \approx 10,000 d_{vc}$; practice: $N \approx 10 d_{vc}$

given specs $\epsilon = 0.1$, $\delta = 0.1$, $d_{vc} = 3$, want $4(2N)^{d_{vc}} \exp\left(-\tfrac{1}{8}\epsilon^2 N\right) \leq \delta$

N	bound
100	2.82×10^{7}
1,000	9.17×10^{9}
10,000	1.19×10^{8}
100,000	1.65×10^{-38}
29,300	9.99×10^{-2}

sample complexity:
need $N \approx 10,000 d_{vc}$ in theory

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。

软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

贝叶斯分类

一个例子

- 一项检查有99%的把握把患某疾病的病人鉴别出来，但对健康人也有10% 的可能性出现假阳性。若此病的发病率为2%，则当某人检查阳性时，他确实患病的概率有多大?

观点将随事实的改变而改变

贝叶斯

贝叶斯公式在实际中有很多应用，它是在观察到事件B 已发生的条件下，寻找导致B 发生的每个原因的概率。

托马斯·贝叶斯(Thomas Bayes) ，出生于英国伦敦，死于 1761 年，神学家 ，数学家，主要研究概率论。

遗世之作 一《机遇理论中 一个问题的解》(An essay towards solving a problem in the doctrine of chances) ，提出了 一种归纳推理的理论，从此贝叶斯定理诞生于世，后来的许多研究者将其不断完善， 最终发展为一种系统的统计推理方法 一 贝叶斯方法。 到 20 世纪 50 至 60 年代贝叶斯学派已“占据了数理统 计学这块领地的半壁江山，撑起了统计学的半边天 。”

贝叶斯公式

设$A_1, A_2, \cdots, A_n$是两两互斥的事件，且$P(A_i)>0$，$i=1, 2, \cdots, n$，B是另一个事件，它总是与$A_1, A_2, \cdots, A_n$之一同时发生，则

$$P(A_i \mid B) = \frac{P(A_i)P(B \mid A_i)}{\sum_{j=1}^{n} P(A_j)P(B \mid A_j)}$$

贝叶斯公式

- 解：　设A为患病，B为检查阳性，则

$$P(A|B) = \frac{P(A) \times P(B|A)}{[P(A) \times P(B|A) + P(A-) \times P(B|A-)]}$$

$$= \frac{0.02 \times 0.99}{0.02 \times 0.99 + 0.98 \times 0.10}$$

$$= \frac{0.0198}{0.1178} \approx 0.168$$

贝叶斯决策理论

- 贝叶斯决策论（Bayesian decision theory）是在概率框架下实施决策的基本方法。
  - 在分类问题情况下，在所有相关概率都已知的理想情形下，贝叶斯决策考虑如何基于这些概率和误判损失来选择最优的类别标记。

- 假设有 N 种可能的类别标记，即 $y = \{c_1, c_2, \ldots, c_N\}$，$\lambda_{ij}$ 是将一个真实标记为 c_j 的样本误分类为 c_i 所产生的损失。基于后验概率 $P\{c_i \mid \mathbf{x}\}$ 可获得将样本 $\mathbf{x}$ 分类为 c_i 所产生的期望损失（expected loss），即在样本上的“条件风险”（conditional risk）

$$R(c_i \mid \mathbf{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j \mid \mathbf{x})$$

贝叶斯决策理论

- 贝叶斯决策目标：
  - 寻找一个判定准则 $h : X \mapsto Y$ 以最小化总体风险

$$R(h) = \mathbf{E}_x \left[R(h(\mathbf{x}) \mid \mathbf{x}) \right]$$

$$R\left(c_i \mid \mathbf{x}\right) = \sum_{j=1}^{N} \lambda_{ij} P\left(c_j \mid \mathbf{x}\right)$$

贝叶斯决策理论

- 贝叶斯决策解决思路：
  - 对每个样本$\mathbf{x}$，若h能最小化条件风险 $R(h(\mathbf{x}) \mid \mathbf{x})$，则总体风险$R(h)$也将被最小化。

  - 贝叶斯判定准则（Bayes decision rule）： 为最小化总体风险，只需在每个样本上选择那个能使条件风险$R(c \mid \mathbf{x})$最小的类别标记

$$h^*(x) = \underset{c \in y}{\operatorname{argmin}} R(c \mid x)$$

贝叶斯最优分类器(Bayes optimal classifier)，与之对应的总体风险$R(h^*)$称为贝叶斯风险 (Bayes risk)

$1 - R(h^*)$反映了分类起所能达到的最好性能，即通过机器学习所能产生的模型精度的理论上限。

贝叶斯决策理论

- 分类问题：
  - 目标是最小化分类错误率，则误判损失λ_{ij}可写为

$$\lambda_{i,j}\begin{cases} 0, & \text{if } i=j; \\ 1, & \text{otherwise}, \end{cases}$$

  - 条件风险

$$R(c \mid \mathbf{x}) = 1 - P(c \mid \mathbf{x})$$

$$R\left(c_i \mid \mathbf{x}\right) = \sum_{j=1}^{N} \lambda_{ij} P\left(c_j \mid \mathbf{x}\right)$$

  - 最小化分类错误率的贝叶斯最最优类器为

$$h^*(x) = \underset{c \in y}{\operatorname{argmax}}\, P(c \mid x)$$

对每个样本$\mathbf{x}$，选择能使**后验概率** $P(c \mid \mathbf{x})$ 最大的类别标记。

后验概率

- 使用贝叶斯判定准则来最小化决策风险，首先要获得后验概率 $P(c \mid \mathbf{x})$，在现实中通常难以直接获得
- 机器学习所要实现的是基于有限的训练样本尽可能准确地估计出后验概率
- 两种策略
  - 判别式模型（discriminative models）

给定 $\mathbf{x}$，通过直接建模 $P(c \mid \mathbf{x})$，来预测 c

  - 生成式模型（generative models）

先对联合概率分布 $P(\mathbf{x}, c)$ 建模，再由此获得 $P(c \mid \mathbf{x})$

$$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})}$$

生成式模型

- 基于贝叶斯定理，$P(c \mid \mathbf{x})$可写成

先验概率
样本空间中各类样本所占的比例，可通过各类样本出现的频率估计（大数定理）

类标记 c 相对于样本 x 的“类条件概率”（class-conditional probability), 或称“似然”。

$$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})} = \frac{P(c)P(\mathbf{x} \mid c)}{P(\mathbf{x})}$$

“证据”（evidence）因子，与类标记无关

极大似然估计

- 估计类条件概率的常用策略
  - 先假定其具有某种确定的概率分布形式，再基于训练样本对概率分布参数估计
- 记关于类别 c 的类条件概率为 $P(\mathbf{x} \mid c)$ ，
  - 假设 $P(\mathbf{x} \mid c)$ 具有确定的形式被参数 $\boldsymbol{\theta}_c$ 唯一确定，我们的任务就是利用训练集 D 估计参数 $\boldsymbol{\theta}_c$
  - 概率模型的训练过程就是参数估计过程

频率主义学派 (frequentist)认为参数虽然未知，但却存在客观值，因此可通过优化似然函数等准则来确定参数值

贝叶斯学派 (Bayesian)认为参数是未观察到的随机变量、其本身也可由分布，因此可假定参数服从一个先验分布，然后基于观测到的数据计算参数的后验分布。

极大似然估计

- 令 D_c 表示训练集中第 c 类样本的组合的集合，假设这些样本是独立的，则参数 $\boldsymbol{\theta}_c$ 对于数据集 D_c 的似然

$$P(D_c \mid \boldsymbol{\theta_c}) = \prod_{\mathbf{x} \in D_c} P(\mathbf{x} \mid \boldsymbol{\theta_c})$$

  - 极大似然估计：寻找能最大化似然 $P(D_c \mid \theta_c)$ 的参数值 $\hat{\theta}_c$。直观上看，极大似然估计是试图在 θ_c所有可能的取值中，找到一个使数据出现的“可能性”最大值。

- 对数似然(log-likelihood)
  - 解决概率连乘操作易造成的下溢问题

$$LL(\boldsymbol{\theta_c}) = \log P(D_c \mid \boldsymbol{\theta_c}) = \sum_{\mathbf{x} \in D_c} \log P(\mathbf{x} \mid \boldsymbol{\theta}_c)$$

极大似然估计

- 实例：
  - 在连续属性情形下，假设概率密度函数 $p(\mathbf{x} \mid c) \sim N(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$，则参数 $\boldsymbol{\mu}_c$和$\boldsymbol{\sigma}_c^2$的极大似然估计为

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{|D_c|} \sum_{\mathbf{x} \in D_c} \mathbf{x}$$

$$\hat{\boldsymbol{\sigma}}_c^2 = \frac{1}{|D_c|} \sum_{\mathbf{x} \in D_c} (\mathbf{x} - \hat{\boldsymbol{\mu}}_c)(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}}$$

  - 通过极大似然法得到的正态分布均值就是样本均值，方差就是 $(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}}$ 的均值，这显然是一个符合直觉的结果

这种参数化的方法虽能使类条件概率估计变得相对简单，但估计结果的准确性严重依赖于所假设的概率分布形式是否符合潜在的真实数据分布。

朴素贝叶斯分类器

- 估计后验概率 $P(c \mid \mathbf{x})$ 主要困难：类条件概率 $P(\mathbf{x} \mid c)$是所有属性上的联合概率难以从有限的训练样本估计获得。

- 朴素贝叶斯分类器(Naïve Bayes Classifier)采用了“属性条件独立性假设”(attribute conditional independence assumption)：每个属性独立地对分类结果发生影响。

- 基于属性条件独立性假设，基于生成式模型的贝叶斯决策可重写为

$$P(c \mid \mathbf{x}) = \frac{P(c)P(\mathbf{x} \mid c)}{P(\mathbf{x})} = \frac{P(c)}{P(\mathbf{x})} \prod_{i=1}^{d} P(x_i \mid c)$$

  - 其中 d 为属性数目，x_i 为 $\mathbf{x}$ 在第 i 个属性上的取值。

朴素贝叶斯分类器

$$P(c \mid \mathbf{x}) = \frac{P(c)P(\mathbf{x} \mid c)}{P(\mathbf{x})} = \frac{P(c)}{P(\mathbf{x})} \prod_{i=1}^{d} P(x_i \mid c)$$

由于对所有类别来说 $P(x)$ 相同，因此贝叶斯判定准则可以简化为

$$h_{nb}(\mathbf{x}) = \underset{c \in y}{\operatorname{argmax}} P(c) \prod_{i=1}^{d} P(x_i \mid c)$$

朴素贝叶斯分类器

- 模型训练
  - 基于训练集 D 估计类先验概率 $P(c)$

令 D_c 表示训练集 D 中第 c 类样本组合的集合，若有充足的独立同分布样本，则可容易地估计出类先验概率

$$P(c) = \frac{|D_c|}{|D|}$$

  - 对每个属性估计条件概率 $P(x_i \mid c)$

对离散属性而言，令 D_{c,x_i} 表示 D_c 中在第 i 个属性上取值为 x_i 的样本组成的集合，则条件概率 $P(x_i \mid c)$ 可估计为

$$P(x_i \mid c) = \frac{|D_{c,x_i}|}{|D_c|}$$

对连续属性而言可考虑概率密度函数，假定 $p(x_i \mid c) \sim N(\mu_{c,i}, \sigma_{c,i}^2)$，其中 $\mu_{c,i}$ 和 $\sigma_{c,i}^2$ 分别是第 c 类样本在第 i 个属性上取值的均值和方差，则有

$$P(x_i \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2{\sigma_{c,i}}^2}\right)$$

朴素贝叶斯分类器

- 示例：用西瓜数据集3.0训练一个朴素贝叶斯分类器，对测试例“测1”进行分类 (p151, 西瓜数据集 p84 表4.3)

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
测 1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	?

拉普拉斯修正

$$h_{nb}(\mathbf{x}) = \underset{c \in y}{\operatorname{argmax}} P(c) \prod_{i=1}^{d} P(x_i \mid c)$$

- 若某个属性值在训练集中没有与某个类同时出现过，则直接计算会出现问题
  - 示例：“敲声=清脆”测试例，训练集中没有该样例，因此连乘式计算的概率值为0，无论其他属性上明显像好瓜，分类结果都是“好瓜=否”，这显然不合理。

- 为了避免其他属性携带的信息被训练集中未出现的属性值“抹去”，在估计概率值时通常要进行“拉普拉斯修正”（Laplacian correction）
  - 令 N表示训练集 D中可能的类别数， N_i表示第 i个属性可能的取值数，则分别先验概率和条件概率修正为

$$\hat{P}(c) = \frac{|D_c| + 1}{|D| + N} \qquad \hat{P}(x_i \mid c) = \frac{|D_{c,x_i}| + 1}{|D| + N_i}$$

软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

本节课结束