机器学习
Machine Learning

北京航空航天大学计算机学院
School of Computer Science and Engineering, Beihang University
黄 迪

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

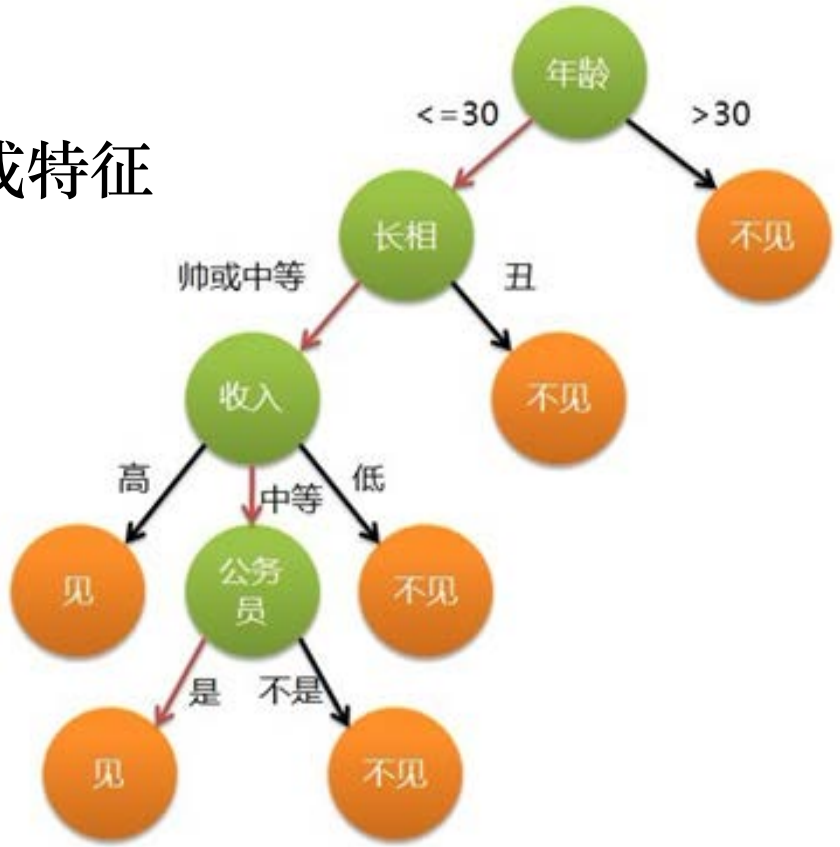
课前回顾（一）

问题背景

- **问题举例**
  - 根据症状或检查结果分类患者
  - 根据起因或现象分类设备故障
  - 根据拖欠支付的可能性分类贷款申请
- **分类问题**
  - 把样例分类到各可能的离散值对应的类别
- **问题特征**
  - 实例由“属性-值”对表示
  - 属性可以是连续值或离散值
  - 具有离散的输出值
  - 训练数据可以包含缺少属性值的实例

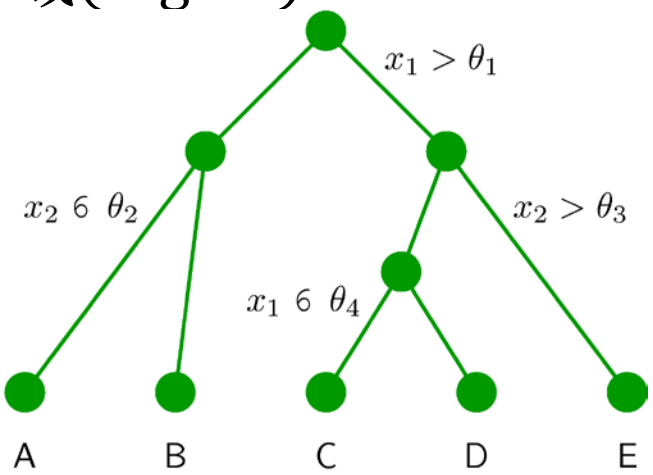
决策树定义

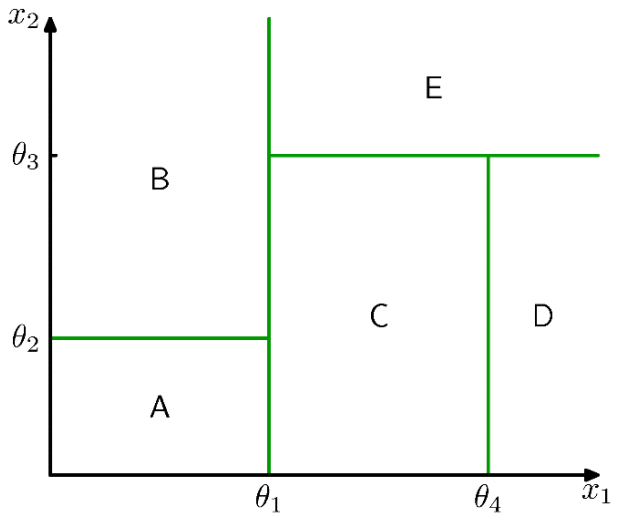
- **决策树(Decision Tree)**
  - 决策树是一种树型结构，由结点和有向边组成
  - 节点
    - 内部结点表示一个属性或特征
    - 叶结点代表一种类别
  - 有向边/分支
    - 分支代表一个测试输出



决策树算法

- **决策树(Decision Tree)**
  - **基本思想：**采用自顶向下的递归方法，以信息熵为度量构造一棵熵值下降最快的树，到叶子节点处的熵值为零，此时每个叶节点中的实例都属于同一类.
  - 决策树可以看成一个if-then的规则集合.
  - 一个决策树将特征空间划分为不相交的单元(cell)或区域(region).





算法流程

- **决策树分类的基本流程分为两步**
  - 第1步：从数据中获取知识进行机器学习

    利用训练集建立(并精化)一棵决策树，构建决策树模型.
  - 第2步：利用生成的模型对输入数据进行分类

    对测试样本，从根结点依次测试记录的属性值，直至到达某个叶结点，找到该记录所在的类别.

算法流程

- **决策树构建的基本流程**
  - Step 1：选取一个属性作为决策树的根结点，然后就这个属性所有的取值创建树的分支.
  - Step 2：用这棵树来对训练数据集进行分类：
    - 如果一个叶结点的所有实例都属于同一类，则以该类为标记标识此叶结点.
    - 如果所有的叶结点都有类标记，则算法终止.
  - Step 3：否则，选取一个从该结点到根路径中没有出现过的属性作为标记标识该结点，然后就这个属性所有的取值继续创建树的分支；重复算法步骤Step 2.

主要算法

- 建立决策树的关键，即在当前状态下选择哪个属性作为分类依据

  示例：高？富？帅？　　白？富？美？

- **目标**：每个分支节点的样本尽可能属于同一类别，即节点的“纯度”(purity)越来越高
- 根据不同目标函数，建立决策树主要有以下三种算法
  - ID3：　信息增益
  - C4.5：　信息增益率
  - CART：基尼指数

主要算法

- **ID3 (Iterative Dichotomiser 3)算法**
  - 以信息熵为度量，每次优先选取信息增益最大的属性，即能使熵值最小的属性.
  - 熵→随机变量不确定性的度量.
    - 信息熵/经验熵/条件熵/…

$$H(D) = -\sum_{c=1}^{C} p_c \log_2 p_c$$

$$= -\sum_{c=1}^{C} \frac{D_c}{D} \log_2 \frac{D_c}{D}$$

$$H(Y \mid X) = -\sum_{i=1}^{n} p_i H(Y \mid X = x_i)$$

  - 信息增益→特征对训练数据的信息增益定义为集合的经验熵与特征给定条件下集合的经验条件熵之差

$$G(D,a) = H(D) - H(D \mid a)$$

$$= H(D) + \sum_{n=1}^{N} \frac{|D^n|}{|D|} H(D^n)$$

主要算法

- 决策树的生成算法：

输入：训练数据集D，特征集A，阈值ε

输出：决策树T

（1）若D中所有实例属于同一类C_k，则T为单结点树，并将类C_k作为该结点的类标记，返回T;

（2）若$A=\emptyset$，则T为单结点树，并将D中实例数最大的类C_k作为该结点类标记，返回T;

（3）否则，计算A中各特征对D的信息增益，选择信息增益最大的特征A_g

（4）如果A_g的信息增益小于阈值ε，则置T为单结点树，并将D中样本数最大的类C_k作为该结点的类标记，返回T

（5）否则，对A_g的每一个可能值a_i，依$A_g=a_i$，奖D分割为若干非空子集D_i，将D_i中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树T，返回T

（6）对第i个子结点，以D_i为训练集，以$A-\{A_g\}$为特征集，递归的调用第（1）~（5）步，得到子树T_i，返回T_i。

主要算法

- **ID3算法特点**
  - 从一类无序、无规则的事物(概念)中推理出决策树表示的分类规则，只需要对训练实例进行较好的标注，就能够进行学习.
  - 分类模型是树状结构，简单直观，可将决策树中到达每个叶节点的路径转换为IF—THEN形式的分类规则，比较符合人类的理解方式.

主要算法

- **ID3算法局限性**
  - 信息增益偏好取值多的属性
  - 可能会受噪声或小样本影响，易出现过拟合问题
  - 无法处理连续值的属性
  - 无法处理属性值不完整的训练数据
  - 无法处理不同代价的属性

属性筛选度量标准

- **信息增益率(Information Gain Ratio)**

$$G_{ratio}(D,a)=\frac{G(D,a)}{H(a)}$$

其中

$$H(a)=-\sum_{n=1}^{N}\frac{|D_n|}{|D|}\log_2\frac{|D_n|}{|D|}$$

称为属性a的固有值

N越大，$H(a)$通常也越大；因此采用信息增益率，可缓解信息增益准则对可取值数目较多的属性的偏好.

C4.5算法就采用增益率替代了ID3 算法的信息增益

属性筛选度量标准

- **基尼指数(Gini Index)**

$$Gini(D)=\sum_{c=1}^{C}\sum_{c'\neq c}p_c p_c'=1-\sum_{c=1}^{C}p_c^2=1-\sum_{c=1}^{C}\left(\frac{|D_c|}{|D|}\right)^2$$

直观反映了从数据集中随机抽取两个样本，其类别不一致的概率；基尼指数越小，数据集的纯度越高.

- 属性A的基尼指数 $Gini(D,a)=\sum_{n=1}^{N}\frac{|D^n|}{|D|}Gini(D^n)$
- 最优属性选择 $a^*=\arg\min_{a\in A}Gini(D,a)$

CART算法就采用基尼指数替代了ID3 算法的信息增益

剪枝处理(Pruning)

- **针对过拟合问题**
  - **剪枝**是主要手段
- **基本策略**
  - **预剪枝策略(Pre-pruning)**：决策树生成过程中，对每个节点在划分前进行估计，若划分不能带来决策树泛化性能提升，则停止划分，并将该节点设为叶节点.
  - **后剪枝策略(Post-pruning)**：先利用训练集生成决策树，自底向上对非叶节点进行考察，若将该叶节点对应子树替换为叶节点能带来泛化性能提升，则将该子树替换为叶节点.

剪枝处理(Pruning)

- **预剪枝策略特点**
  - **优势：** “剪掉了”很多没必要展开的分支，降低了过拟合的风险，并且显著减少了决策树的训练时间开销和测试时间开销.
  - **劣势：**有些分支的当前划分有可能不能提高甚至降低泛化性能，但后续划分有可能提高泛化性能；预剪枝禁止这些后续分支的展开，可能会导致欠拟合.

剪枝处理(Pruning)

- **后剪枝策略特点**
  - **优势**：测试了所有分支，比预剪枝决策树保留了更多分支，降低了欠拟合的风险，泛化性能一般优于预剪枝决策树.
  - **劣势**：后剪枝过程在生成完全决策树后在进行，且要自底向上对所有非叶节点逐一评估；因此，决策树的训练时间开销要高于未剪枝决策树和预剪枝决策树.

连续值处理

- **基本思想：采用二分法(Bi-Partition)进行离散化**
  - 给定样本集D和连续属性a($a \in A$)，假定a在D上有N个不同取值，将这些值从大到小排序得$\{a_1, a_2, \ldots, a_N\}$
  - 基于划分点t，可将D分为子集D_t^+和D_t^-，其中D_t^+ (D_t^-)包含了属性值A不小(大)于t的样本子集
  - t在$[a_n, a_{n+1})$上的任意取值的划分结果都相同
  - **候选划分点集合**

$$T_a = \left\{ \frac{a_n + a_{n+1}}{2} \,\middle|\, 1 \le n \le N-1 \right\}$$

连续值处理

- 信息增益 $G(D,a)=\max_{t\in T_a} G(D,a,t)$

$$=\max_{t\in T_a}\left(H(D)-\sum_{\lambda\in\{+,-\}}\frac{\left|D_t^{\lambda}\right|}{|D|}H(D_t^{\lambda})\right)$$

其中，$G(D,A,t)$是样本集D基于划分点t二分后的信息增益. 我们需选择使 $G(D,A,t)$ 最大的划分点t.

缺失值处理

- **需要解决的问题**
  - 如何在属性值缺失的情况下进行划分属性选择(计算信息增益)?

    取属性完整的子集
  - 给定划分属性，若样本在该属性上的值缺失，如何对样本进行划分?

    让样本以不同概率划分到不同的子节点去

不同代价属性的处理

- **需要解决的问题**
  - 不同的属性测量具有不同的代价
- **解决思路**
  - 在属性筛选度量标准中考虑属性的不同代价
  - 优先选择低代价属性的决策树
  - 必要时才依赖高代价属性

课前回顾（二）

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^* 。

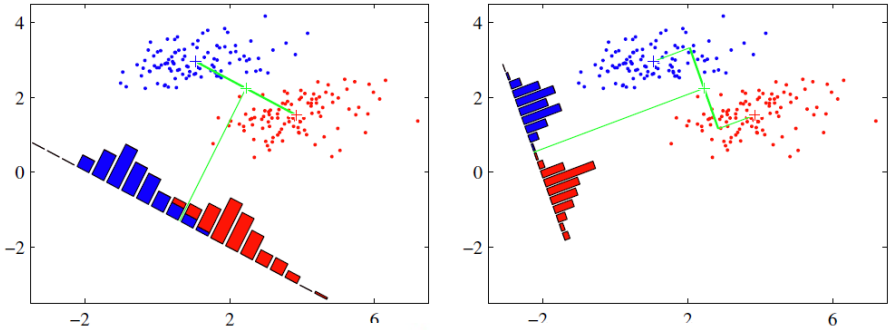
最终得到线性判别函数：$g(x) = w^{*T} x + w_0^*$ 或 $g(x) = a^{*T} y$，对于位置类别样本x_k，计算$g(x_k)$并通过决策规则判断其类别。

准则函数

- **Fisher准则**

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

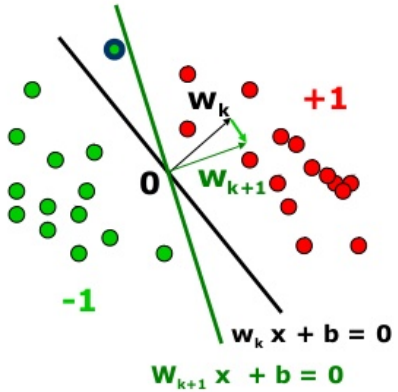
$$w^* = S_w^{-1}(m_1 - m_2)$$



- **感知机准则**

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

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

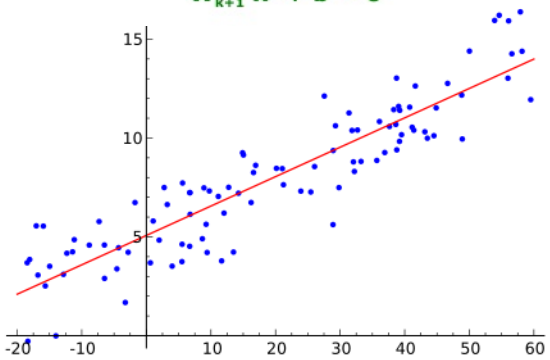


- **最小平方误差准则**

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

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

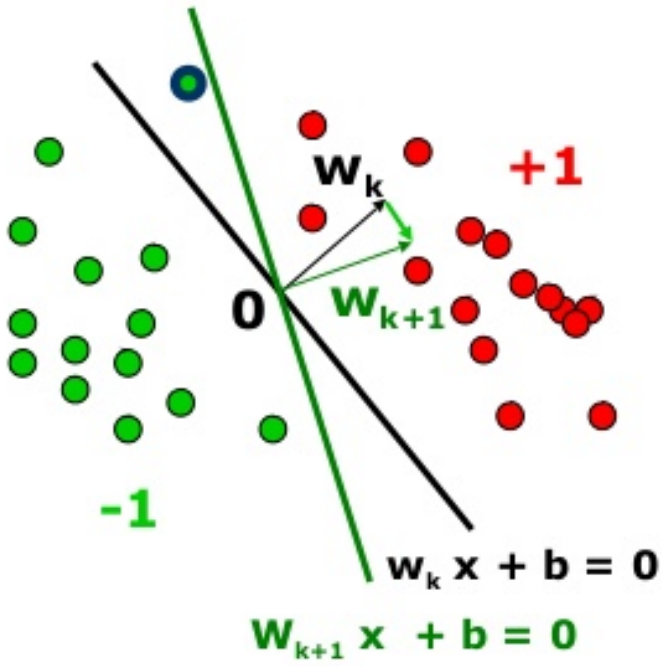
$$a(k+1) = a(k) - \rho_k Y^T (Ya - b)$$



感知机准则

- F. Rosenblatt(1950-1960年提出)

感知准则是一种自学习判别函数生成方法，由于Rosenblatt试图将其用于脑模型感知器，因此得名。该方法对随意给定的判别函数初始值，通过样本分类训练过程逐步对其修正直至最终确定。



几个基本概念

- **线性可分性**

一组容量为N的样本集$y_1, y_2, \ldots y_N$, 其中y_n为$\hat{d}$维增广样本向量, 分别来自w_1类和w_2类, 如果存在权向量a, 使得对于任何$y \in w_1$, 都有$a^T y > 0$, 而对于任何$y \in w_2$, 都有$a^T y < 0$, 则称这组样本为线性可分的, 反之亦然成立。

- **样本的规范化**

$$\begin{cases} a^T y_i > 0, y_i \in w_1 \\ a^T y_j < 0, y_j \in w_2 \end{cases}$$

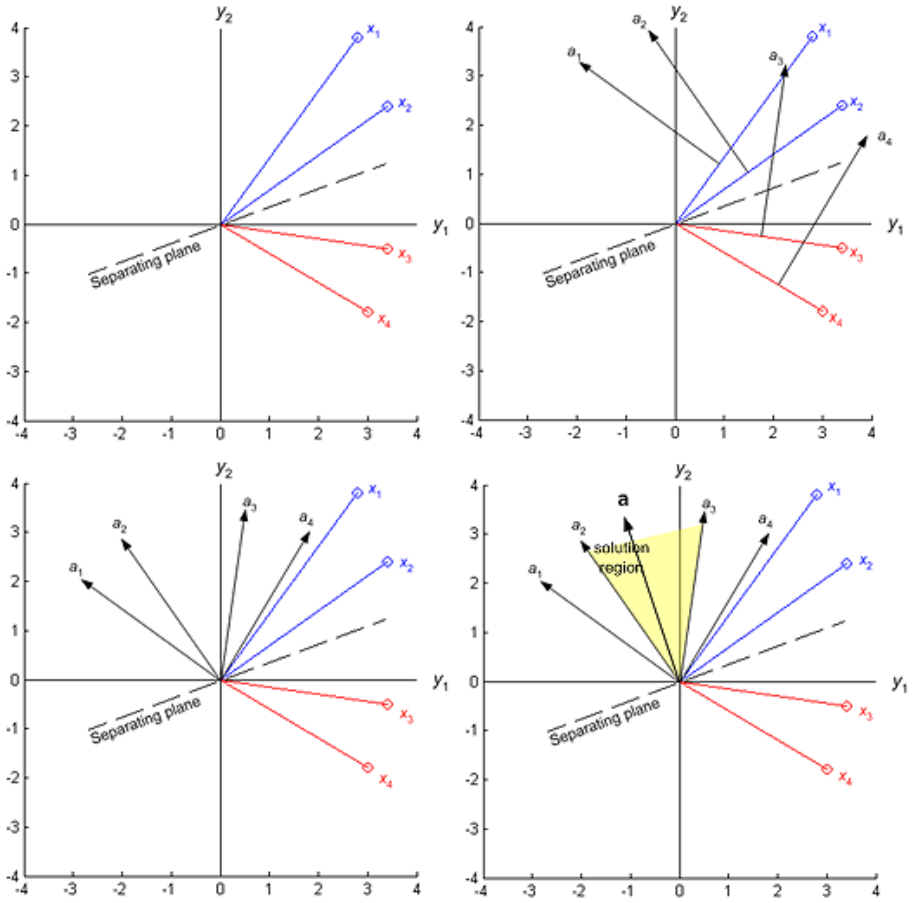
$$y_n' = \begin{cases} y_i > 0, y_i \in w_1 \\ -y_j < 0, y_j \in w_2 \end{cases}$$

规范化增广样本向量

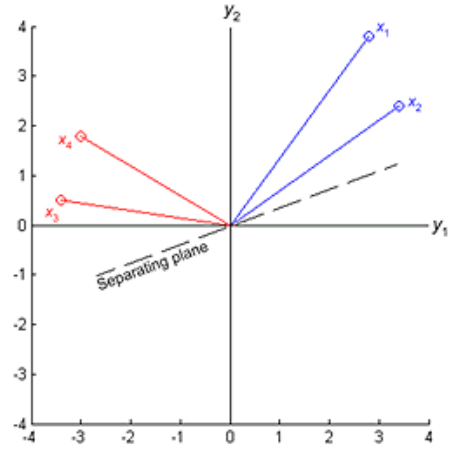
$$a^T y_n' > 0, n = 1, 2, \ldots, N$$

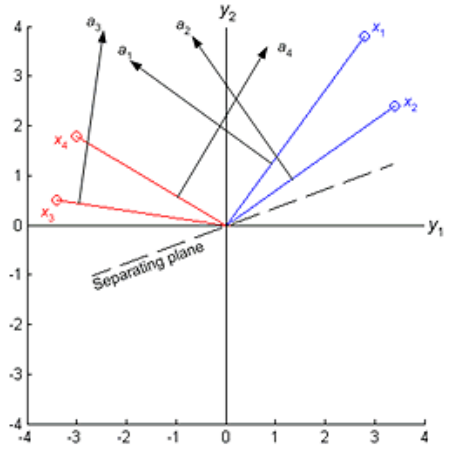
几个基本概念

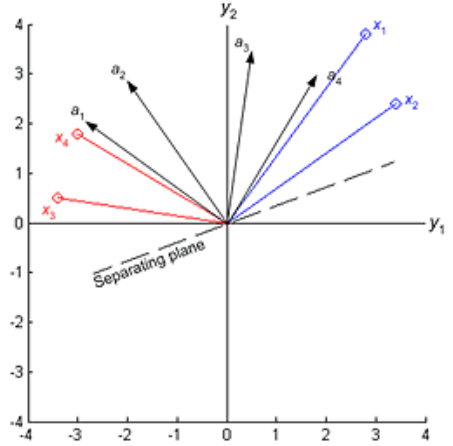
- 解向量和解区

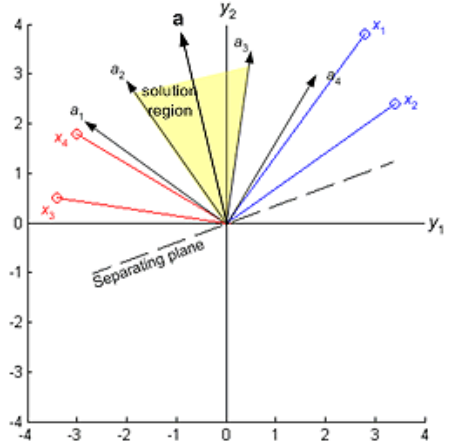


未规范化







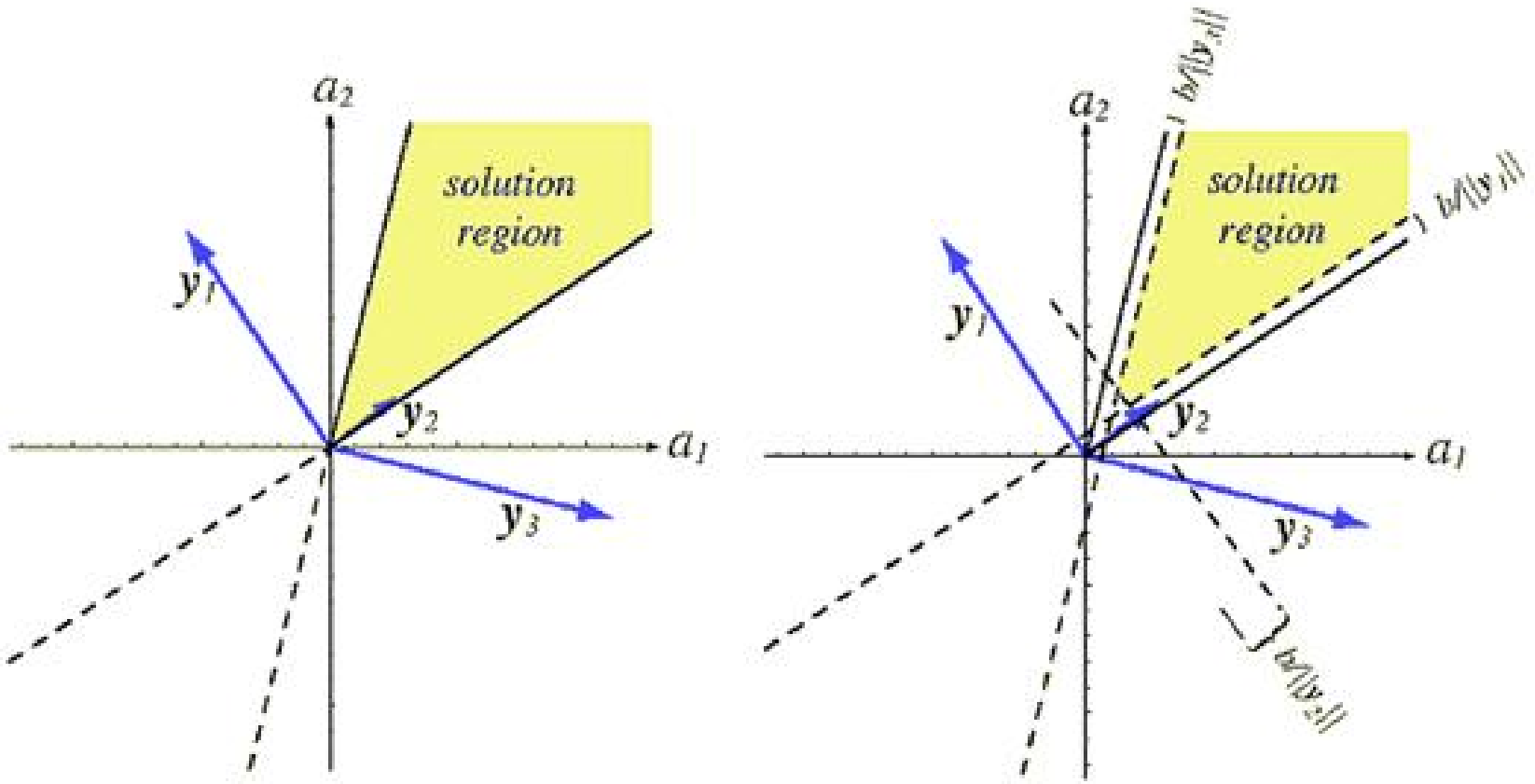


规范化

几个基本概念

- 对解区的限制

使解向量更可靠 $a^T y_n \geq b > 0$，避免解收敛到解区边界的某点上



感知机准则

- 寻找解向量a^*

对于一组样本$y_1, y_2, \ldots y_N$, 其中y_n是规范化增广样本向量, 使得:

$$a^T y_n > 0, n = 1, 2, ..., N$$

对于线性可分问题，构造准则函数 $J_P(a) = \sum_{y \in \eta^k} (-a^T y)$

其中 η^k 是被权向量a错分的样本集合，即当y被错分时，就有 $a^T y_n \leq 0$

因此 $J_P(a) \geq 0$，仅当a为解向量或在解区边界时 $J_P(a) = 0$

也就是说，当且仅当 η^k 为空集时 $J_P^*(a) = min J_P(a) = 0$

此时无错分样本，这时的a就是解向量a^* 。

感知机准则

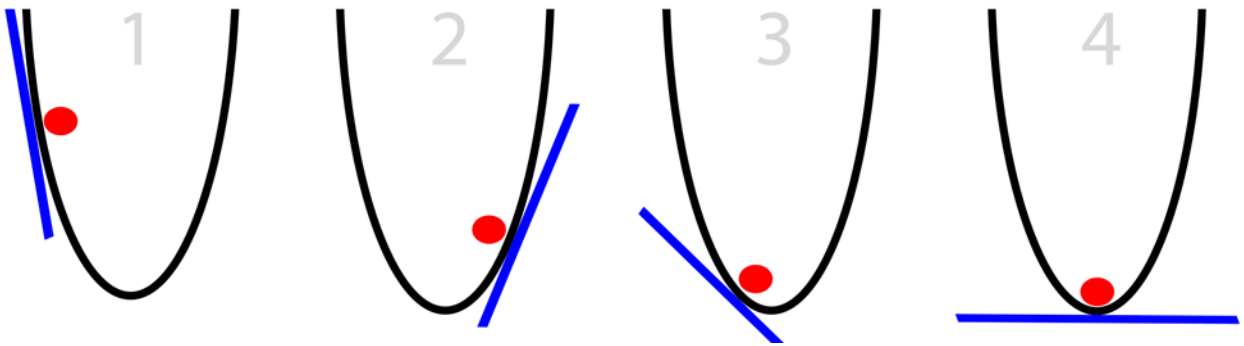
- 求使 $J_P(a)$ 达到最小值的 a^*

采用梯度下降法求解 $J_P(a) = \sum_{y \in \eta^k} (-a^T y)$

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

$$a(k+1) = a(k) - \rho_k \nabla J$$ 梯度下降法迭代公式

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



感知机准则-示例

- Sample set for two-class case
  - Class 1

$$\mathbf{x}_1 = \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \qquad \mathbf{y}_1 = \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix}, \mathbf{y}_2 = \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix}$$

  - Class 2

$$\mathbf{x}_3 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \qquad \mathbf{y}_3 = \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix}, \mathbf{y}_4 = \begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix}$$

  - Initial weight vector

$$\mathbf{a}(1) = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} \qquad \mathbf{a}(1)^t = \begin{pmatrix} 0 & 2 & 1 \end{pmatrix}$$

感知机准则-示例

- Example (Cont.)
  - Iterative procedure
    - (1) $\mathbf{y}^{(1)t} = \begin{pmatrix} 1 & -2 & 2 \end{pmatrix}$

$$\mathbf{a}(1)^t \mathbf{y}^{(1)} = \begin{pmatrix} 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix} = -2 < 0$$

$$\mathbf{a}(2)^t = \begin{pmatrix} 0 & 2 & 1 \end{pmatrix} + \begin{pmatrix} 1 & -2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 3 \end{pmatrix}$$

感知机准则-示例

- Example (Cont.)
  - Iterative procedure
    - (2) $\mathbf{y}^{(2)t} = \begin{pmatrix} 1 & -2 & -2 \end{pmatrix}$

$$\mathbf{a}(2)^t \mathbf{y}^{(2)} = \begin{pmatrix} 1 & 0 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix} = -5 < 0$$

$$\mathbf{a}(3)^t = \begin{pmatrix} 1 & 0 & 3 \end{pmatrix} + \begin{pmatrix} 1 & -2 & -2 \end{pmatrix} = \begin{pmatrix} 2 & -2 & 1 \end{pmatrix}$$

感知机准则-示例

- Example (Cont.)
  - Iterative procedure
    - (3)

$$\mathbf{y}^{(3)t} = \begin{pmatrix} -1 & -2 & -1 \end{pmatrix}$$

$$\mathbf{a}(3)^t \mathbf{y}^{(3)} = \begin{pmatrix} 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} = 1 > 0$$

$$\mathbf{a}(4)^t = \begin{pmatrix} 2 & -2 & 1 \end{pmatrix} \text{ (no chnage)}$$

感知机准则-示例

- Example (Cont.)
  - Iterative procedure
    - (4) $\mathbf{y}^{(4)t} = (-1 \quad -2 \quad 1)$

$$\mathbf{a}(4)^t \mathbf{y}^{(3)} = (2 \quad -2 \quad 1)\begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix} = 3 > 0$$

$$\mathbf{a}(5)^t = (2 \quad -2 \quad 1) \quad \text{(no chnage)}$$

感知机准则-示例

- Example (Cont.)
  - Iterative procedure
    - (5) $\mathbf{y}^{(5)t} = \begin{pmatrix} 1 & -2 & 2 \end{pmatrix}$

$$\mathbf{a}(5)^t \mathbf{y}^{(1)} = \begin{pmatrix} 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix} = 8 > 0$$

$$\mathbf{a}(6)^t = \begin{pmatrix} 2 & -2 & 1 \end{pmatrix} \qquad \text{(no change)}$$

感知机准则-示例

- Example (Cont.)
  - Iterative procedure
    - (6) $\mathbf{y}^{(6)t} = (1 \quad -2 \quad -2)$

$$\mathbf{a}(6)^t \mathbf{y}^{(2)} = (2 \quad -2 \quad 1)\begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix} = 4 > 0$$

$$\mathbf{a}(7)^t = (2 \quad -2 \quad 1)$$

感知机准则-示例

- Example (Cont.)
  - Iterative procedure
    - (7)

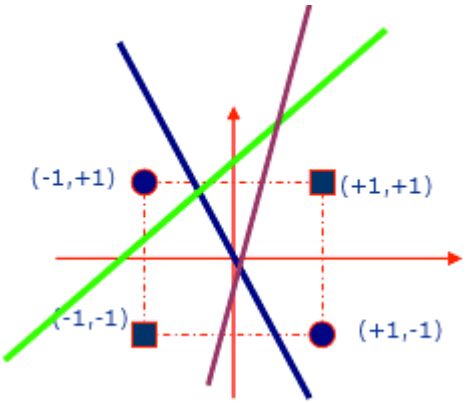
$$\mathbf{y}^{(7)t} = \begin{pmatrix} -1 & -2 & -1 \end{pmatrix}$$

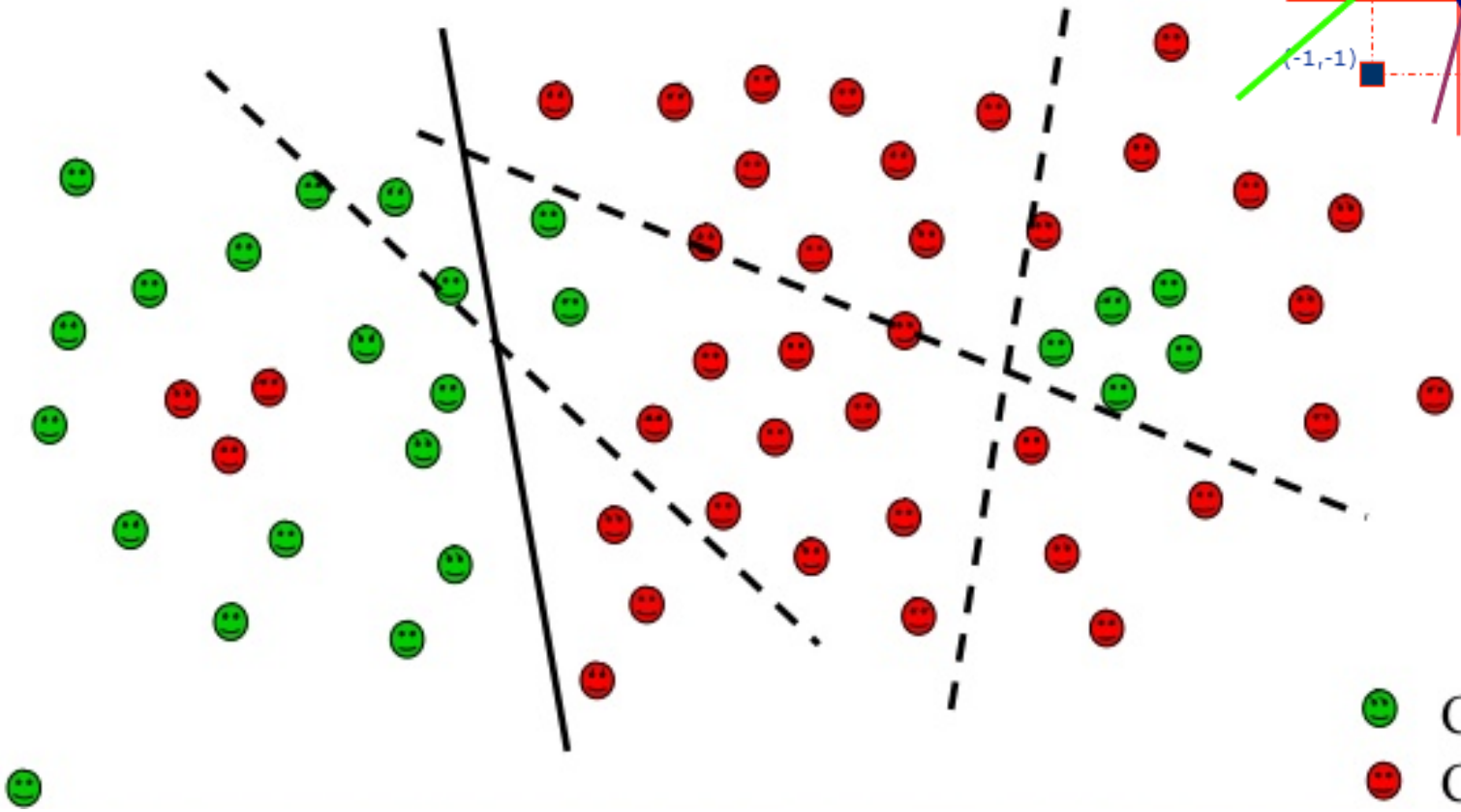
$$\mathbf{a}(7)^t \mathbf{y}^{(7)} = \begin{pmatrix} 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} = 1 > 0$$

$$\mathbf{a}(8)^t = \begin{pmatrix} 2 & -2 & 1 \end{pmatrix} \quad \text{(no chnage)}$$

感知机准则

- 结果不唯一且在线性不可分情况下不收敛

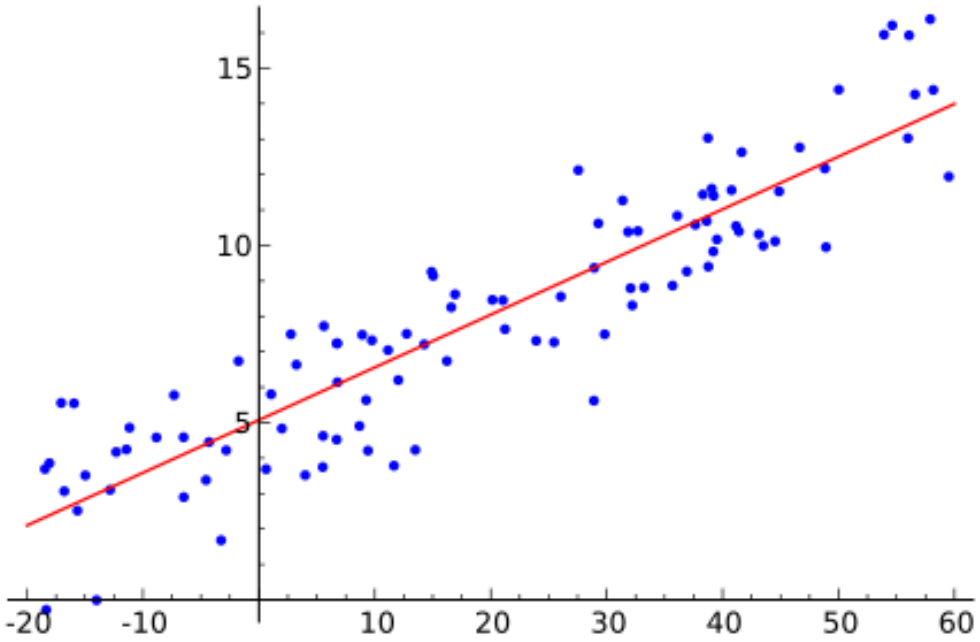




最小二乘准则

- 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：$\nabla J_S(a) = 2Y^T(Ya - b)$

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

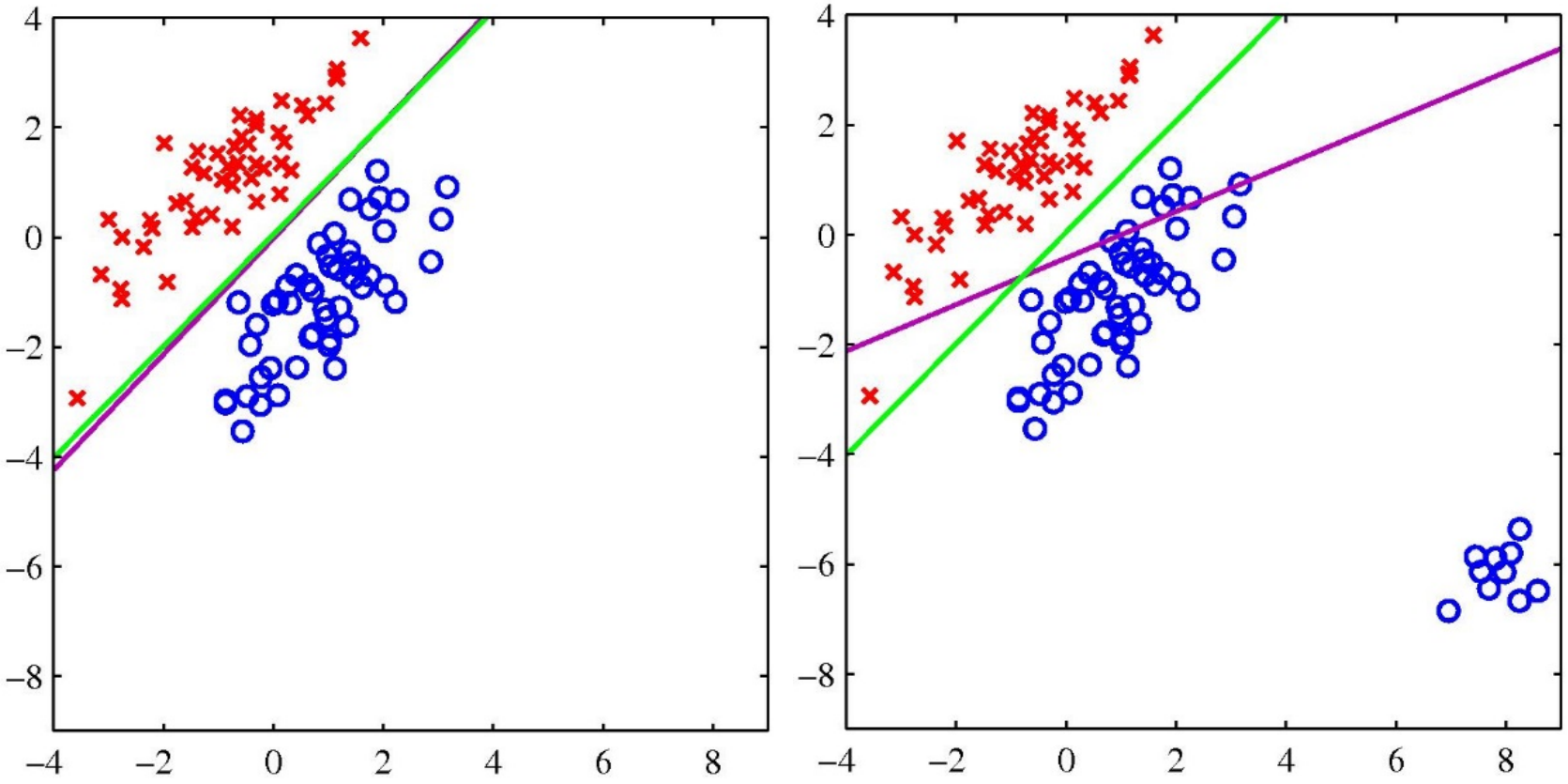
可以证明，选择 $\rho_k = \frac{\rho_1}{k}$ ，ρ_1是任意常数

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

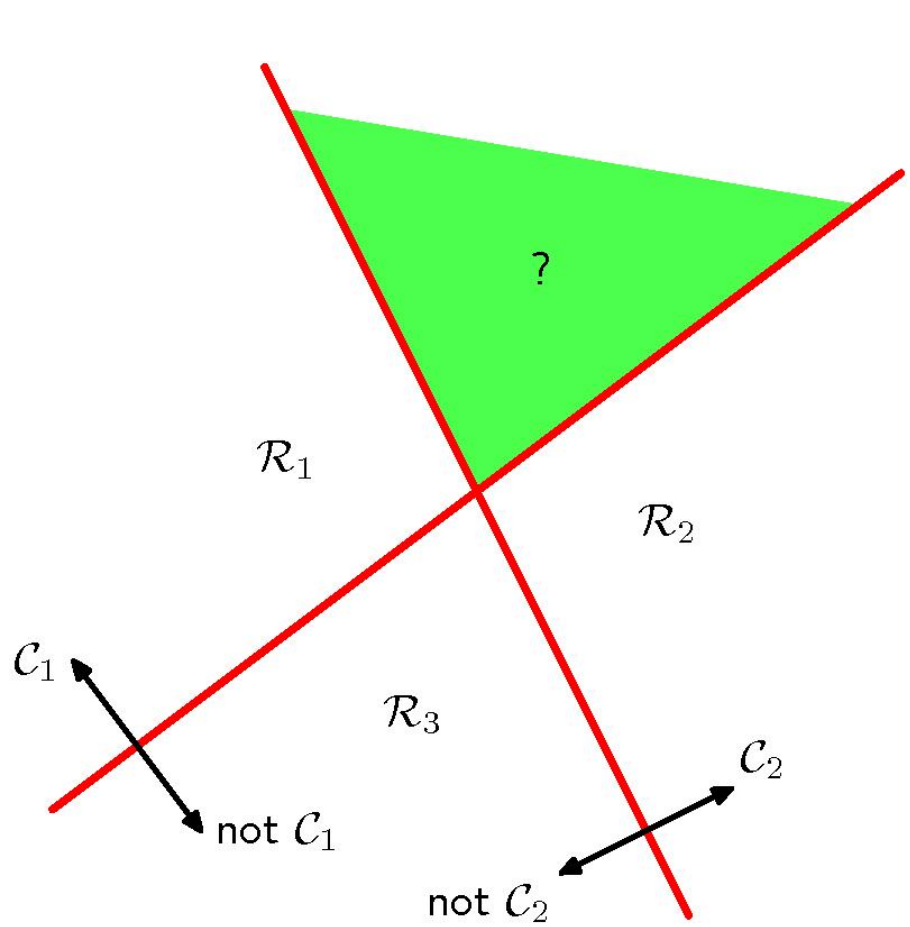
最小二乘准则

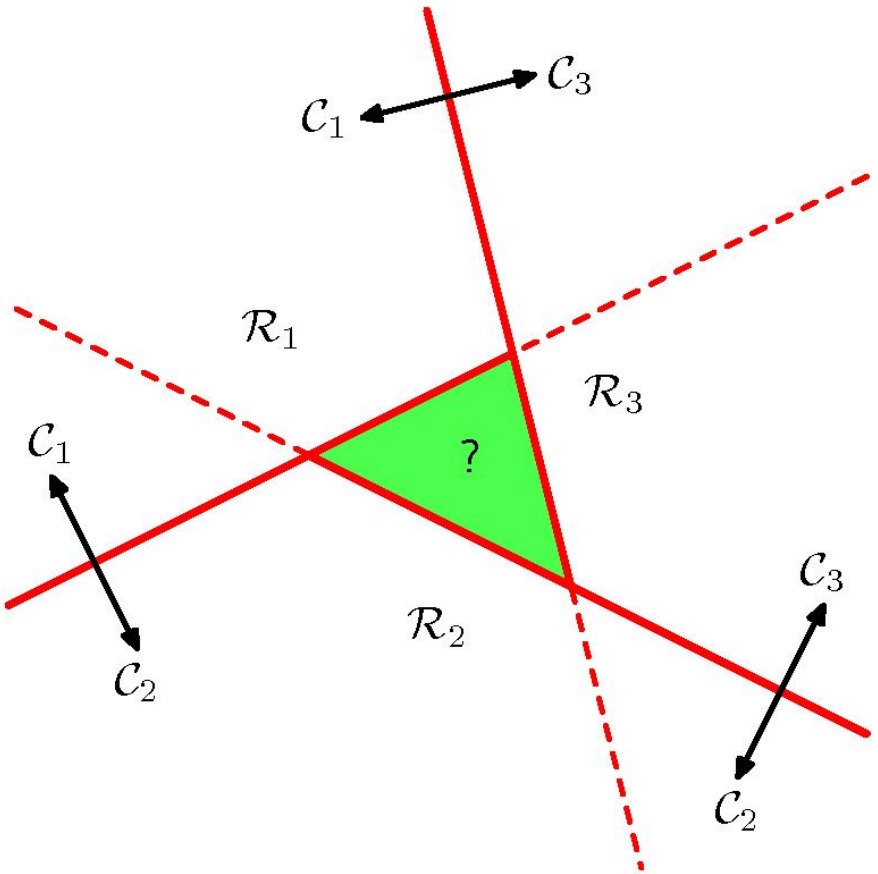
- 对于异常值(Outlier)非常敏感



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)**

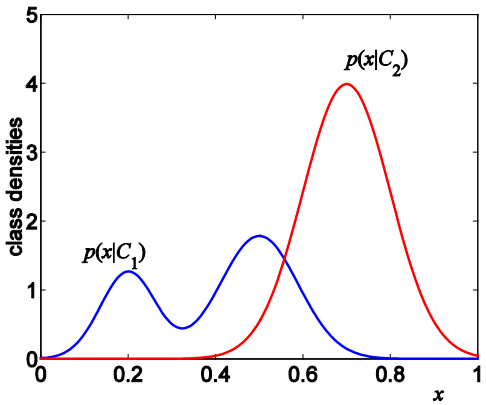
分别对各类的**类条件密度** $p(\mathbf{x}|C_k)$ 和**先验概率** $p(C_k)$ 进行建模，之后利用贝叶斯定理计算**后验概率**

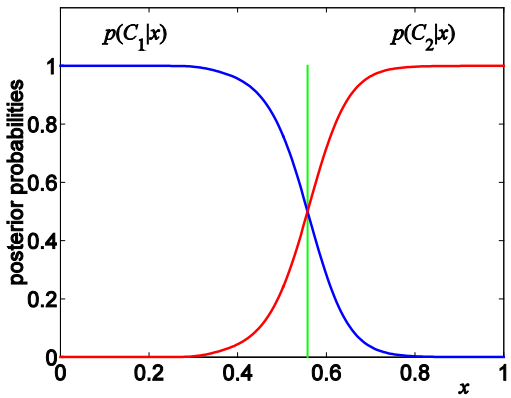
$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)}$$

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)**

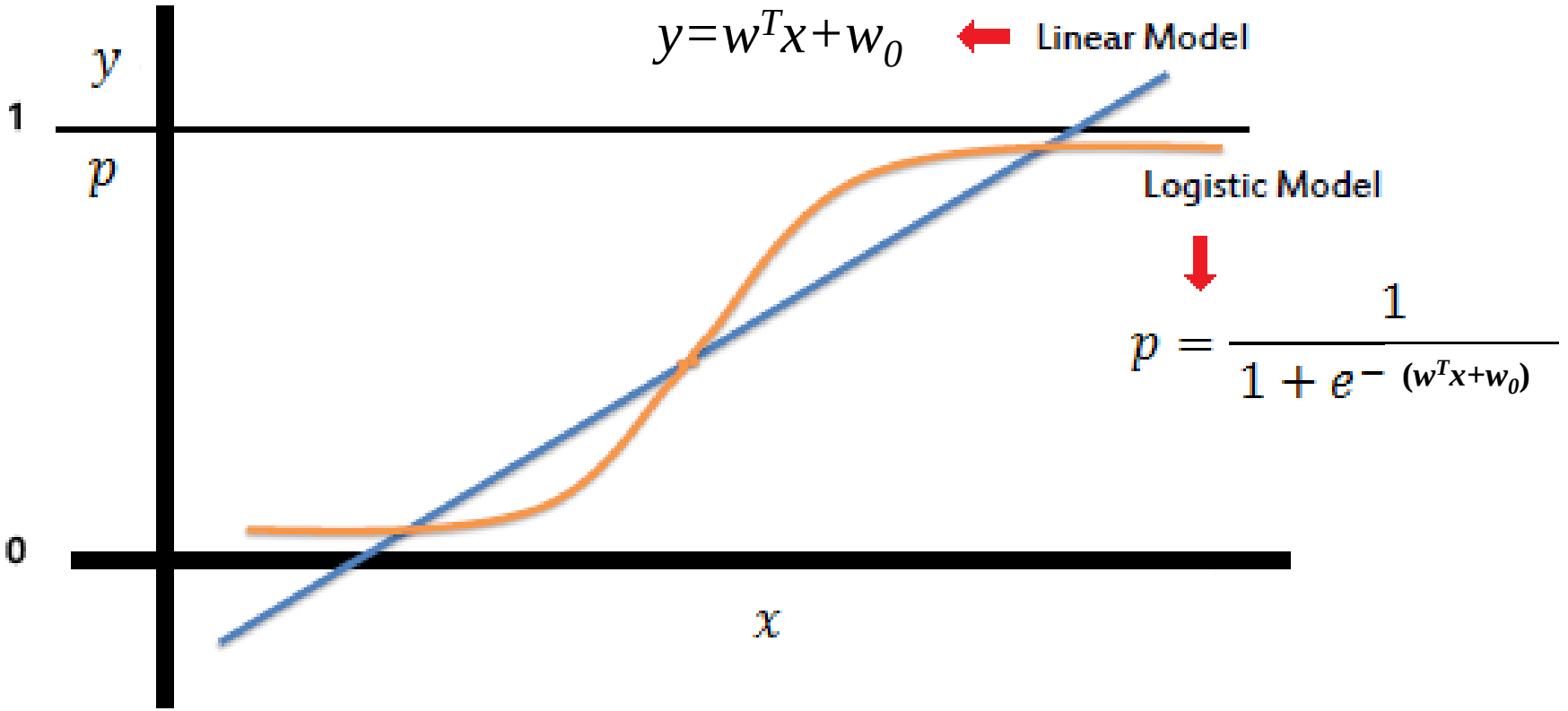
直接对**后验概率** $p(C_k|\mathbf{x})$ 建模





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。

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

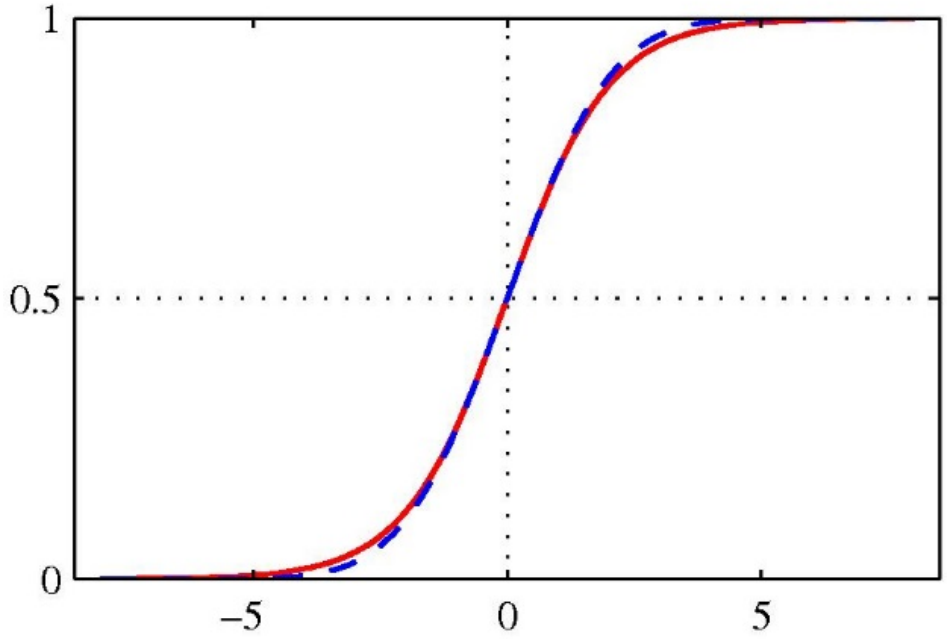
Sigmoid函数性质：

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 $a = ln(\frac{\sigma}{1-\sigma}) = ln\frac{p(C_1|x)}{p(C_2|x)}$

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)**

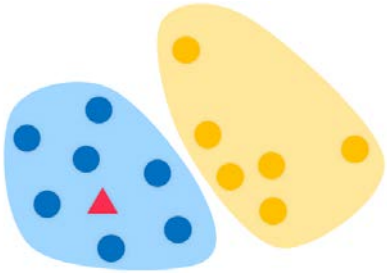
估计类条件密度并计算*a*

- **判别式模型(Discriminative Model)**

把*a*视为线性函数 $a = w^T x$ 直接估计

生成式模型和判别式模型

● 生成式模型



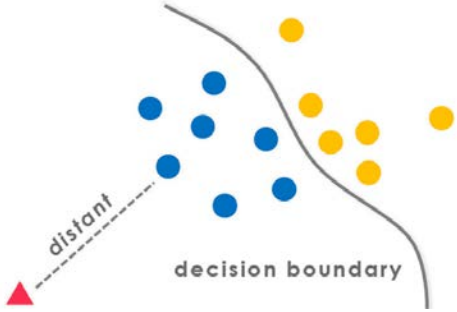
优点：

- 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

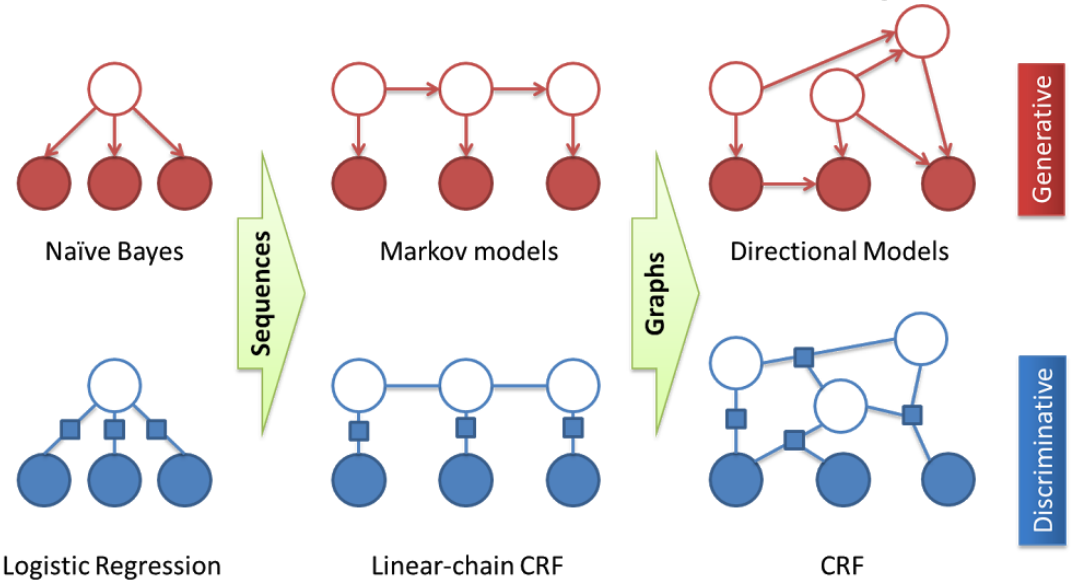
代表算法：

- 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



第五章：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

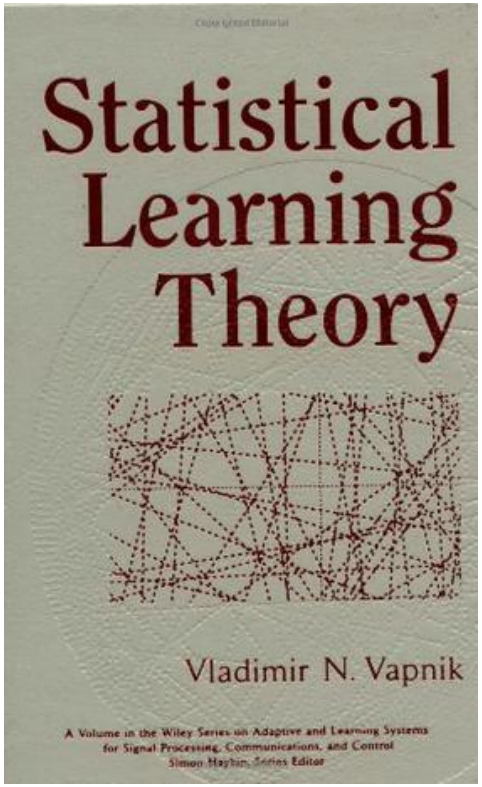
- V. Vapnik对于统计机器学习的贡献

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。

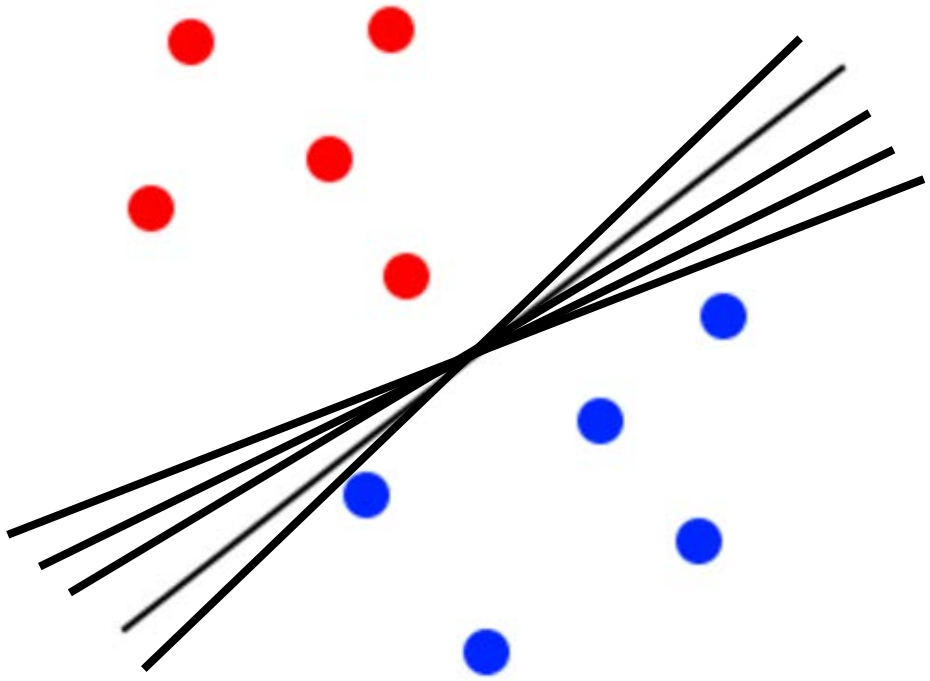
The Nature of Statistical Learning Theory 77736 * 1995
V Vapnik
Data mining and knowledge discovery

线性分类模型

- **两类样本的线性分类问题**

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

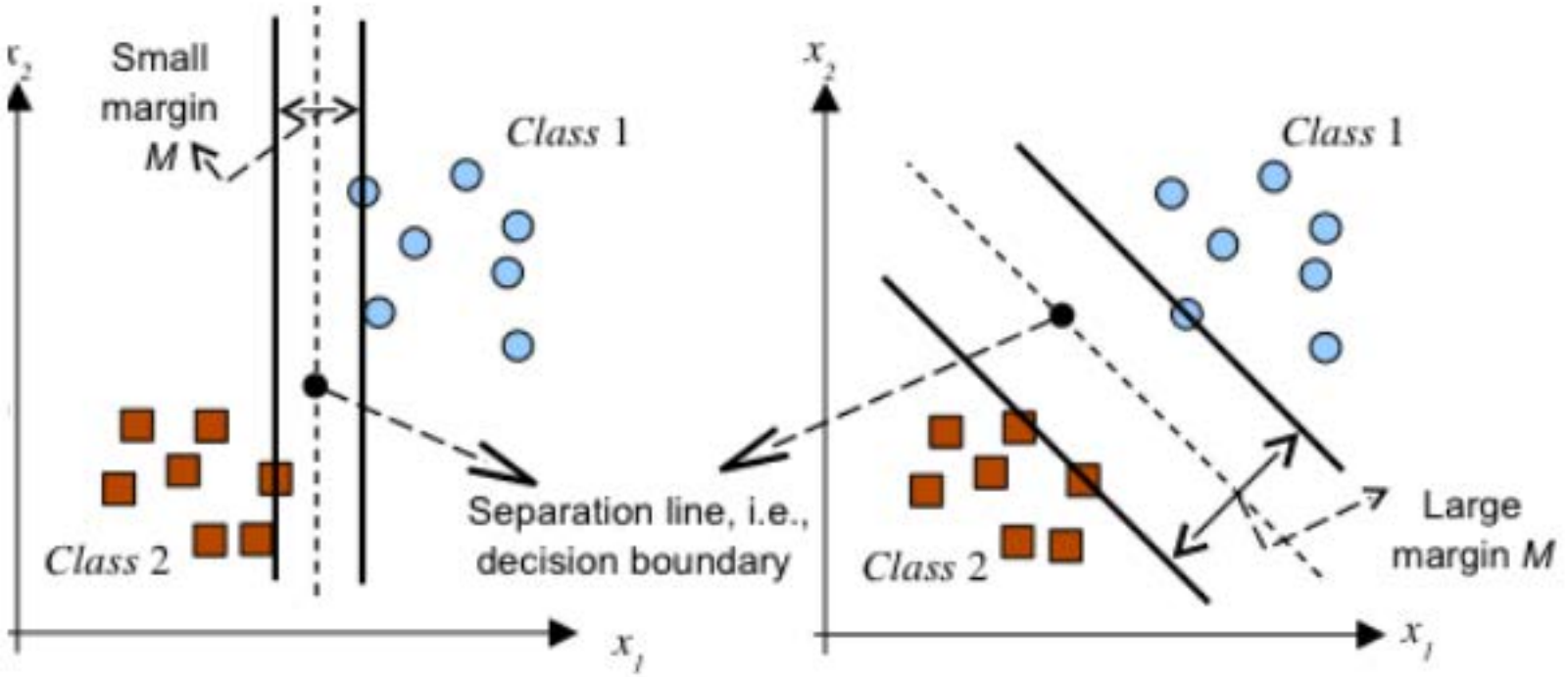
$$y(x, w) = f(\sum_{j=1}^{M} w_j x_j)$$



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**。



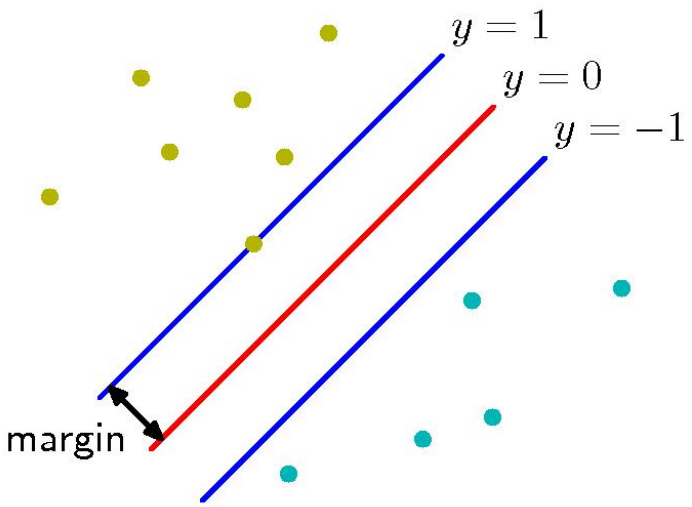
支持向量机

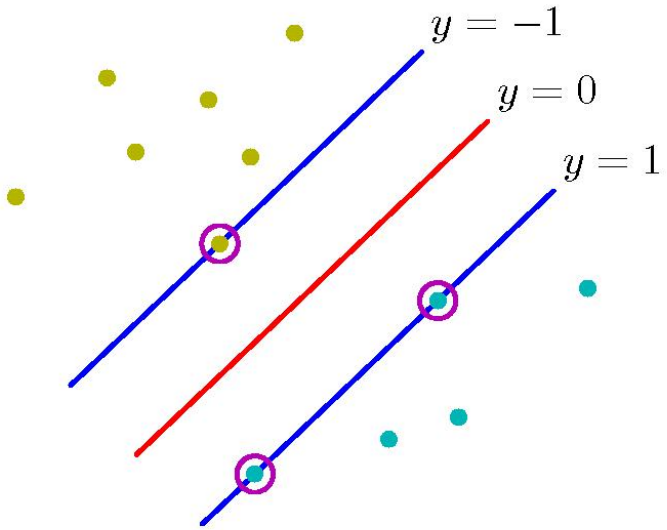
- **线性支持向量机**

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

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

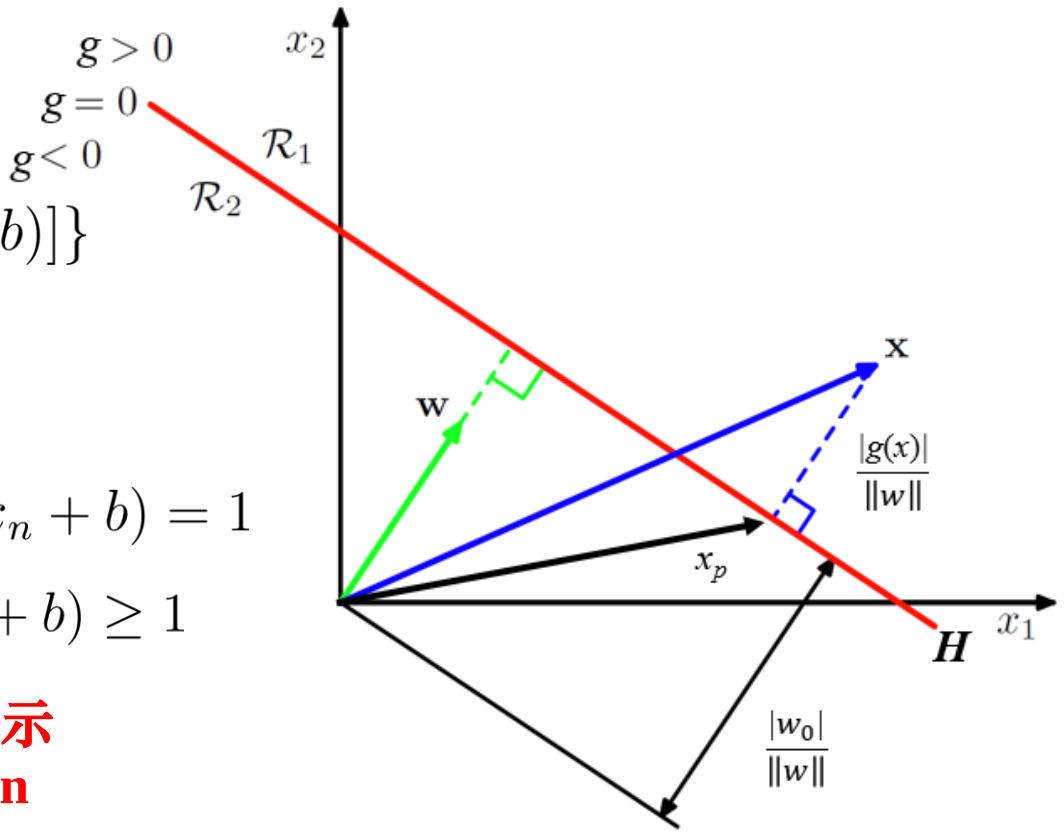
求解复杂

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

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

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

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：

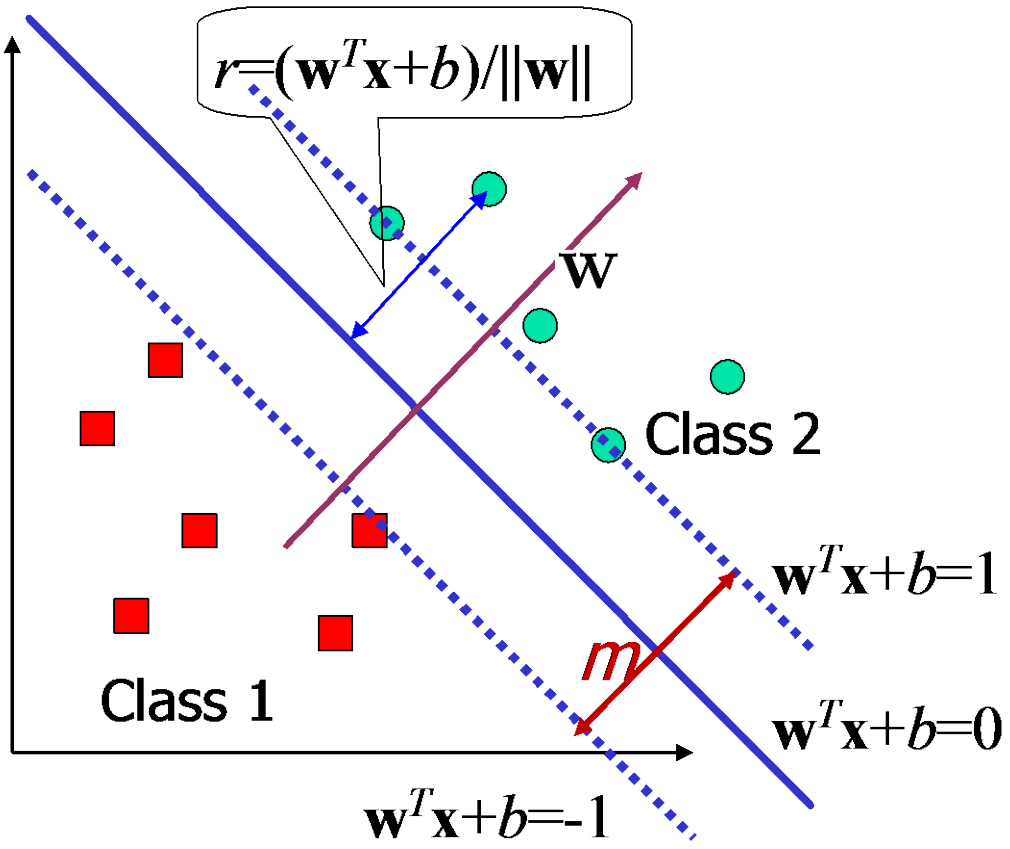
$$t_n(w^T x + b) = 1, a_n > 0$$

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