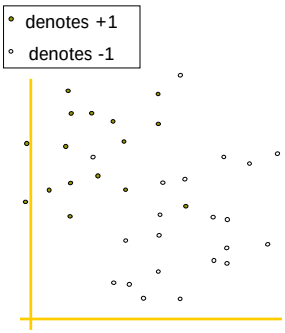
第四章 线性判别函数

2009-11-10

6. 线性支持向量机 (线性SVM)

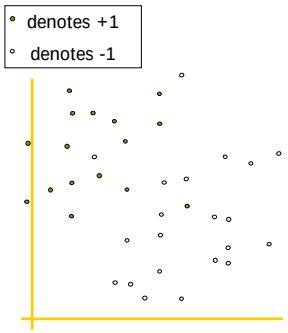
- **线性不可分情形下的广义最优线性分类面**



- 实际问题很少线性可分。虽然理论上可通过非线性映射得到线性可分的数据，但如何获得这样的映射，且避免过拟合，是一个问题。
- 更实际的策略是允许一定误差。

6. 线性支持向量机 (线性SVM)

- **线性不可分情形下的广义最优线性分类面**



- **思路一**：最小化 $\|\mathbf{w}\|^2$ 的同时最小化错分训练样本数目。

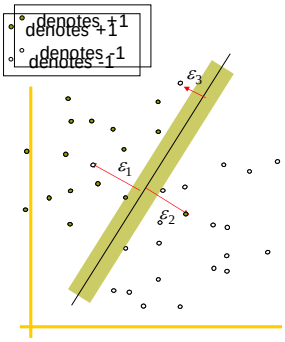
$$\min \|\mathbf{w}\|^2 + C(\#\text{training errors})$$

折衷参数

- **问题**：无法转化成二次规划问题，优化很困难；没有区分不同误差的影响。

6. 线性支持向量机 (线性SVM)

- **线性不可分情形下的广义最优线性分类面**

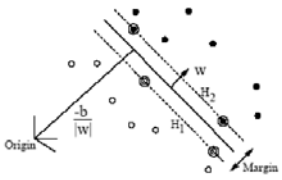


- **思路二**：同时最小化 $\|\mathbf{w}\|^2$ 和错分训练样本到正确位置的距离。
- 引入松弛项 ξ_i

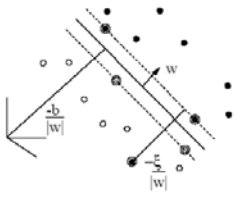
$$y_i(\mathbf{w}^T\mathbf{x}_i + \omega_0) \geq 1 - \xi_i,$$
$$i = 1, \ldots, N,$$
$$\xi_i \geq 0.$$

6. 线性支持向量机 (线性SVM)

- **Soft margin SVM**



Hard Margin（硬间隔）



Soft Margin（软间隔）

6. 线性支持向量机 (线性SVM)

- **Soft margin SVM**

折衷参数，控制惩罚错分样本的程度：C越大，惩罚越大。

$$\min_{\mathbf{w},\omega_0,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{N}\xi_i\right)^k$$
$$s.t. \quad y_i\left(\mathbf{w}^T\mathbf{x}_i + \omega_0\right) \geq 1 - \varepsilon_i,$$
$$\varepsilon_i \geq 0,\ i = 1, \cdots, N;$$

k为正整数是凸规划问题；
k=1或2是二次规划问题；
k=1有较好的对偶形式。

Primal Lagrangian:

$$\min\ L(\mathbf{w}, \omega_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left[y_i(\mathbf{w}^T\mathbf{x}_i + \omega_0) - 1 + \xi_i\right] - \sum_{i=1}^{N}\mu_i\xi_i,$$
$$s.t. \quad \forall i,\ \alpha_i \geq 0, \mu_i \geq 0.$$

6. 线性支持向量机 (线性SVM)

- **Soft margin SVM**

KKT conditions

$$
\begin{cases}
1. & \dfrac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0; \\
2. & \dfrac{\partial L}{\partial \omega_0} = -\sum_{i=1}^{N} \alpha_i y_i = 0; \\
3. & \dfrac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0; \\
4. & y_i(\mathbf{w}^T \mathbf{x}_i + \omega_0) - 1 + \xi_i \geq 0; \\
5. & \xi_i \geq 0; \\
6. & \alpha_i \geq 0; \\
7. & \mu_i \geq 0; \\
8. & \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + \omega_0) - 1 + \xi_i \right) = 0; \\
9. & \mu_i \xi_i = 0;
\end{cases}
$$

6. 线性支持向量机 (线性SVM)

- **Soft margin SVM**
  - 约简 KKT 条件，可得对偶形式 — 二次规划问题

$$
\begin{aligned}
\min_{\alpha} \quad & L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{B} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j; \\
& 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, N; \\
s.t. \quad & \sum_{i=1}^{N} \alpha_i y_i = 0.
\end{aligned}
$$

6. 线性支持向量机 (线性SVM)

- **Soft margin SVM**
  - 几何意义：超平面法向量是支持向量的线性组合。

KKT complementarity conditions:

$$
\forall i: \alpha_i \left(y_i \left(\mathbf{w}^T \mathbf{x}_i + \omega_0 \right) - (1 - \xi_i) \right) = 0;
$$

$$
(C - \alpha_i) \xi_i = 0.
$$

$\alpha_i = 0$ for non-support vectors

$\alpha_i \neq 0$ for support vectors

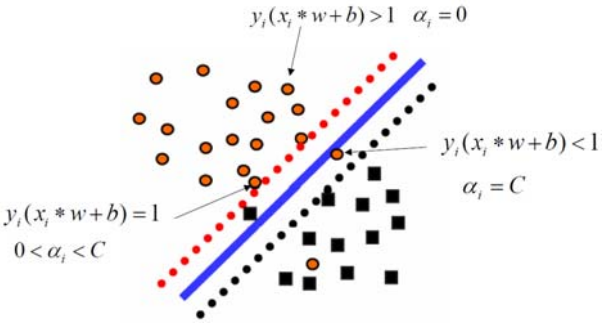
$$
0 < \alpha_i < C \Rightarrow y_i \left(\mathbf{w}^T \mathbf{x}_i + \omega_0 \right) = 1;
$$

$$
\alpha_i = 0 \Rightarrow y_i \left(\mathbf{w}^T \mathbf{x}_i + \omega_0 \right) > 1;
$$

$$
\alpha_i = C \Rightarrow y_i \left(\mathbf{w}^T \mathbf{x}_i + \omega_0 \right) < 1;
$$

6. 线性支持向量机 (线性SVM)

- **Soft margin SVM**
  - 几何意义



6. 线性支持向量机 (线性SVM)

- **序贯最小优化 (SMO) 算法**
  - 无论 Hard Margin 或 Soft Margin SVM，均可用经典的二次规划方法求解，但同时求解 N 个拉格朗日乘子涉及很多次迭代，计算开销太大，所以一般采用 Sequential Minimal Optimization (SMO) 算法 (J. Platt, 1999)。
  - 基本思路：每次只更新两个乘子，迭代获得最终解。

6. 线性支持向量机 (线性SVM)

- **序贯最小优化 (SMO) 算法**
  - 算法框架

```
for i=1:iter
    a. 根据预先设定的规则，从所有样本中选出两个
    b. 保持其他拉格朗日乘子不变，更新所选样本对应的拉格朗日乘子
end
```

  - **优点**：只有两个变量的二次规划问题存在解析解。
  - **关键技术细节**：
    - 在每次迭代中，怎样更新乘子？
    - 怎么选择每次迭代需要更新的乘子？

6. 线性支持向量机 (线性SVM)

- **序贯最小优化 (SMO) 算法**
  - 假定在某一次迭代中，需更新样本 $\mathbf{x}_1, \mathbf{x}_2$ 对应的拉格朗日乘子 α_1, α_2;
  - 这个小规模的二次规划问题可以写成

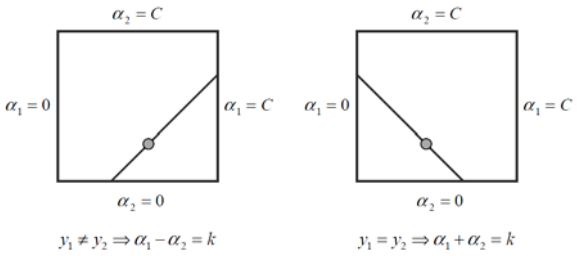
$$\max_{\alpha_1,\alpha_2} \quad L_S(\boldsymbol{\alpha}) = (\alpha_1+\alpha_2) - \frac{1}{2}\left\|\alpha_1 y_1 \mathbf{x}_1 + \alpha_2 y_2 \mathbf{x}_2 + \sum_{i=3}^{N} \alpha_i y_i \mathbf{x}_i\right\|^2,$$

$$s.t. \quad \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} y_i \alpha_i$$

$$0 \le \alpha_i \le C, i = 1, 2;$$

6. 线性支持向量机 (线性SVM)

- 序贯最小优化 (SMO) 算法
  - **约束的几何意义**



6. 线性支持向量机 (线性SVM)

- 序贯最小优化 (SMO) 算法
  - **更新拉格朗日乘子的步骤**

1. 计算 α_2^{new} 的上下界 L 和 H:

$$L = \max(0, \alpha_2^{old} - \alpha_1^{old}),\ H = \min(C, C + \alpha_2^{old} - \alpha_1^{old}),\ \text{if } y_1 \neq y_2$$

$$L = \max(0, \alpha_2^{old} + \alpha_1^{old} - C),\ H = \min(C, \alpha_2^{old} + \alpha_1^{old}),\ \text{if } y_1 = y_2$$

2. 计算 L_S 的二阶导数:

$$\eta = 2\mathbf{x}_1^T\mathbf{x}_2 - \mathbf{x}_1^T\mathbf{x}_1 - \mathbf{x}_2^T\mathbf{x}_2$$

3. 更新 L_S:

$$\alpha_2^{new} = \alpha_2^{old} - \frac{y_2(e_1 - e_2)}{\eta}$$

$$e_i = g^{old}(\mathbf{x}_i) - y_i$$

6. 线性支持向量机 (线性SVM)

- 序贯最小优化 (SMO) 算法
  - **更新拉格朗日乘子的步骤**

4. 计算（剪辑）变量 α_2:

$$\alpha_2^{temp} = \begin{cases} H, & \text{if } \alpha_2^{new} \ge H \\ \alpha_2^{new}, & \text{if } L \le \alpha_2^{new} \le H \\ L, & \text{if } \alpha_2^{new} \le L \end{cases}$$

5. 更新 α_1:

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{temp})$$

6. 线性支持向量机 (线性SVM)

- 序贯最小优化 (SMO) 算法
  - **选择需更新的乘子（启发式选择算法）**
    - 两个基本原则：**(1)** 任何乘子必须满足 KKT 条件；**(2)** 对一个不满足 KKT 条件的乘子进行更新，应能最大限度增大目标函数的值（类似于梯度下降）；
    - 步骤：**(1)** 先"扫描"所有乘子，把第一个（最）违反 KKT 条件的作为更新对象，令其为 α_2；**(2)** 在所有不违反 KKT 条件的乘子中，选择使 $|e_1 - e_2|$ 最大的对象，令其为 α_1。

6. 线性支持向量机 (线性SVM)

- 序贯最小优化 (SMO) 算法
  - 由于所有乘子可分为三种情况，当还存在违反 KKT 条件的乘子时，在扫描时可忽略在 0 和 C 之间的乘子，以加速扫描进程。

Training Set Size	SMO Time (CPU sec)	PCG Time (CPU sec)	SMO Iterations	PCG Iterations
2477	26.3	64.9	10838	1888
3470	44.1	110.4	13975	2270
4912	83.6	372.5	18978	5460
7366	156.7	545.4	27492	5274
9888	248.1	907.6	29751	5972
17188	581.0	3317.9	42026	9413
24692	1214.0	6659.7	55499	14412
49749	3863.5	23877.6	93358	24235

6. 线性支持向量机 (线性SVM)

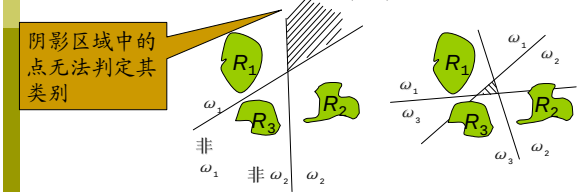
- 总结
  - 目标：求对特征空间划分的最优超平面；
  - 核心思想：最大化分类边界距离；
  - 支持向量：SVM 的训练结果，在 SVM 分类决策中起决定性作用。
    - 计算的复杂性取决于支持向量的数目，而不是样本空间的维数，这在某种意义上避免了“维数灾难”。
    - 具有较好的“鲁棒”性：增、删非支持向量样本对模型没有影响；支持向量样本集具有一定的鲁棒性。
  - 可保证解的全局最优性，不存在陷入局部极小解的问题。

总结：两类问题的线性判别函数

名称	准则	算法
Fisher	$J_F(\mathbf{w})=\frac{\tilde{S}_b}{\tilde{S}_1+\tilde{S}_2}=\frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$	$\mathbf{w}^*=S_w^{-1}(\mathbf{m}_1-\mathbf{m}_2)$
感知器	$J_P(\mathbf{a})=\sum_{\mathbf{y}\in Y^k}(-\mathbf{a}^T\mathbf{y})$	$\mathbf{a}(k+1)=\mathbf{a}(k)+r_k\sum_{\mathbf{y}\in Y^k}\mathbf{y}$
最小错分样本	$J(\mathbf{a})=\|(\mathbf{Ya}-\mathbf{b})-\lvert\mathbf{Ya}-\mathbf{b}\rvert\ \|$	共轭梯度法
MSE	$J_s(\mathbf{a})=\|Y\mathbf{a}-\mathbf{b}\|^2$	$\mathbf{a}^*=Y^+\mathbf{b}$
SVM	$\frac{1}{2}\|\mathbf{w}\|^2+C\left(\sum_{i=1}^N\xi_i\right)^k$ *s.t.* constraints	二次规划

7. 多类问题（简介）

- 思路一：两类别问题可以推广到多类别问题
  - $\omega_i/\sim\omega_i$法：将 c 类问题化为 $(c\text{-}1)$ 个两类（第i类与所有非i类）问题，按两类问题确定其判别函数与决策面方程。
  - ω_i/ω_j 法：将 c 类中的每两类别单独设计其线性判别函数，总共有 $c(c\text{-}1)/2$ 个线性判别函数。



7. 多类问题（简介）

- 思路二：**多类线性判别函数 (Linear Machine)**
  - 将特征空间确实划分为 c 个决策域，共有 c 个判别函数

$$g_i(\mathbf{x})=\mathbf{w}_i^T\mathbf{x}+\omega_{i0},\quad i=1,\ldots,c;$$

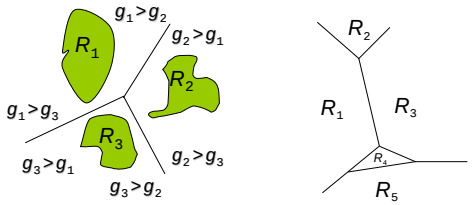
  - 决策规则：

$$j=\arg\max_i g_i(\mathbf{x}),\quad i=1,\ldots,c;$$

  - 决策域的边界由相邻决策域的判别函数共同决定，此时应有 $g_i(\mathbf{x})=g_j(\mathbf{x})$;
  - 线性分类器的决策面是凸的，决策区域单连通;
  - 多类分类器的分界面是分段线性的;

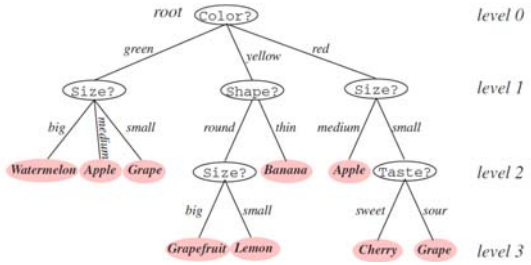
7. 多类问题（简介）

- 多类线性决策面图例



7. 多类问题（简介）

- **决策树**：一种多级分类器，它采用分级的形式，综合用多个决策规则，逐步把复杂的多类别分类问题转化为若干个简单的分类问题来解决。

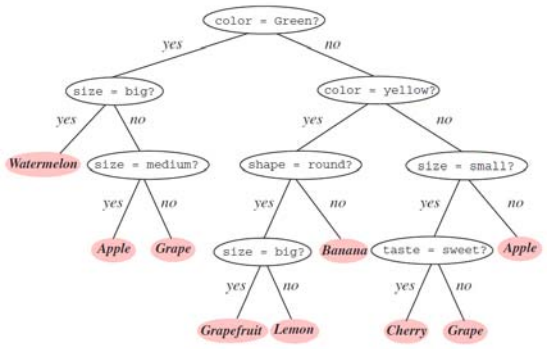


7. 多类问题（简介）

- 二叉决策树
  - 除叶节点外，决策树的每个节点n_i都有且只有两个子节点 n_{il}和 n_{ir}。
  - 二叉决策树把复杂的多类别分类问题转化为多级两类分类问题来解决。
  - 在每个节点n_i，都把样本集分成两个子集。每个子集可能仍包含多类别的样本，继续分直至仅包含单类别样本的叶节点。

7. 多类问题（简介）

- 二叉决策树



总结：线性判别函数

- 基于训练样本的直接确定判别函数方法主要包含两个步骤：
  - 确定使用的判别函数类型或决策面方程类型，如线性分类器，分段线性分类器等；
  - 在选定函数类型的条件下，确定相应的参数，从而完成整个分类器设计；
- 线性判别函数计算简单，在一定条件下能实现最优分类，经常是一种“有限合理”的选择。