

第五章 非线性判别函数

2009-11-10

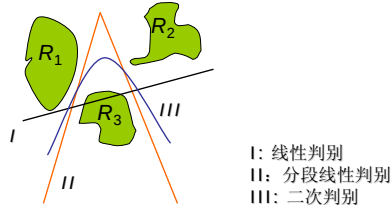
5.1 分段线性判别函数

对实际的模式识别问题来说，各类在特征空间中的分布往往比较复杂，采用线性判别函数无法取得满意的分类效果，可以采用分段线性判别或二次函数判别等方法，是最为简单的形式。

- **分段线性判别函数**：一种特殊的非线性判别函数，它的决策面是若干超平面。
- 树分类器的各节点上采用线性判别规则，即构成分段线性分类器。

5.1 分段线性判别函数

对实际的模式识别问题来说，各类在特征空间中的分布往往比较复杂，采用线性判别函数无法取得满意的分类效果，可以采用分段线性判别或二次函数判别等方法，是最为简单的形式。



5.1 分段线性判别函数

分段段数问题

- 与样本集分布有关：分段段数过少，其分类效果必然要差；但段数又要尽可能少，以免分类判别函数过于复杂，增加分类决策的计算量。
- 实际问题中，同一类样本的子类的数目就可作为确定分段段数的依据。但多数情况下样本分布及合适子类划分并不知道，聚类的方法将样本划分成相对密集的子类，然后用各种方法设计各段判别函数。
- 这一章主要讨论在样本分布及子类划分大体已定的情况下，设计分段线性判别函数的问题。

5.1 分段线性判别函数

形式定义

- 设每个类由若干个子类组成，即

$$\omega_i = \{\omega_i^1, \omega_i^2, \dots, \omega_i^{l_i}\}$$

其中 l_i 为 ω_i 类所有子类的个数。

- 对每个子类定义一个线性判别函数

$$g_i^l(\mathbf{x}) = \mathbf{w}_i^{lT} \mathbf{x} + \omega_{i0}^l, \quad l = 1, 2, \dots, l_i, \quad i = 1, 2, \dots, c;$$

其中 g_i^l 表示 ω_i 类第 l 段的线性判别函数， $\mathbf{w}_i^l$ 与 ω_{i0}^l 分别是第 l 段的权向量与阈值权。

5.1 分段线性判别函数

形式定义

- 定义 ω_i 类的判别函数

$$g_i(\mathbf{x}) = \max_{l=1,2,\dots,l_i} g_i^l(\mathbf{x}),$$

- 决策规则

$$g_j(\mathbf{x}) = \max_{i=1,2,\dots,c} g_i(\mathbf{x}), \quad \text{则 } \mathbf{x} \in \omega_j;$$

- 决策面取决于相邻的决策域：如果第 i 类的第 n 个子类与第 j 类的第 m 个子类相邻，则由它们共同决定的决策面方程是

$$g_i^n(\mathbf{x}) = g_j^m(\mathbf{x}).$$

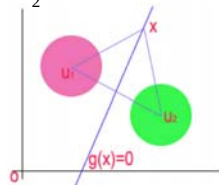
5.2 基于距离的分段线性判别函数

□ 回顾**最小距离分类器**：第二章曾讨论过正态分布条件下，两类别问题在**各特征统计独立、同方差、且先验概率相等**情况下，最小错误率贝叶斯决策就是最小距离决策，即

$$\| \mathbf{x} - \mu_1 \|^2 - \| \mathbf{x} - \mu_2 \|^2 < 0 \quad \mathbf{x} \in \omega_1$$

$$\| \mathbf{x} - \mu_1 \|^2 - \| \mathbf{x} - \mu_2 \|^2 > 0 \quad \mathbf{x} \in \omega_2$$

其中 μ_1 与 μ_2 为各类的均值。



5.2 基于距离的分段线性判别函数

□ 按距离分类的原理可以推广，即把各类别样本特征向量的均值作为各类的代表点 (prototype)，而样本的类别按它到各类别代表点的最小距离划分；决策面是两类别均值连线的垂直平分面。

■ 这种判别方法只有在各类别密集地分布在其均值附近时才有效。

■ 否则只有将各类别划分成相对密集的子类，每个子类以它们的均值作为代表点，然后按最小距离分类。

5.2 基于距离的分段线性判别函数

□ 分段线性距离分类器

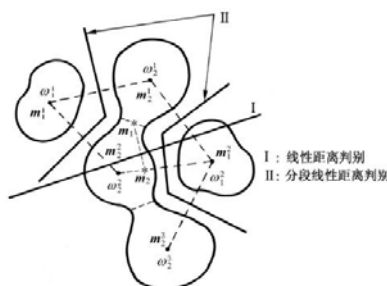
■ **判别函数**：如果对于 ω_i 类有 l_i 个子类，则有 l_i 个代表点，或者说把属于 ω_i 的决策域 R_i 分成 l_i 个子域；对于每个子区域 R_i^l 的均值用 m_i^l 表示，并以此作为该子区域的代表点，则判别函数定义为：

$$g_i(\mathbf{x}) = \min_{l=1,2,\dots,l_i} \| \mathbf{x} - m_i^l \|;$$

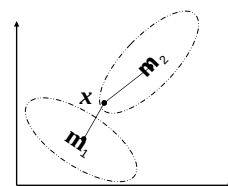
■ **判决规则**

$$\text{if } g_j(\mathbf{x}) = \min_{i=1,\dots,c} g_i(\mathbf{x}) \text{ then } \mathbf{x} \in \omega_j.$$

5.2 基于距离的分段线性判别函数



多个最小距离分类器组成分段线性分类面



最小距离分类器不适用的情况

5.3 错误修正算法

□ $\mathbf{a}_i^T \mathbf{y}$ ：是增广权向量 $\mathbf{a}_i$ 与规范化的增广样本向量 $\mathbf{y}$ 的点积（内积）。内积值比较大则表示这两个向量在方向上比较一致，即向量间的**夹角较小**。

□ 如果某一类样本比较分散，但是能用若干个增广权向量 $\mathbf{a}_i$ 表示，使得同一类样本 $\mathbf{y}$ 能够做到与代表自己一类的 $\mathbf{a}_i$ 的内积 ($\mathbf{a}_i^T \mathbf{y}$) 的最大值大于与代表其它类的增广权向量的内积值，则可以做到正确分类。因此这种算法就是要利用错误提供的信息进行迭代修正。（类似于多类线性判别函数的固定增量算法。）

5.3 错误修正算法

□ **条件**：已知每类的子类数目 l_i ，但子类划分情况未知。

□ **基本思路**：可假设初始权向量，然后由训练样本提供的错误信息进行迭代修正，直至收敛。

□ **算法步骤**

1. **初始化**：任意给定各子类的权向量：

$$\mathbf{a}_i^l(0), i=1,2,\dots,c, l=1,2,\dots,l_i;$$

其中括号中的数目表示迭代的次数。

5.3 错误修正算法

□ 算法步骤

2. **检测与修正**: 对每次迭代所得增广权向量用训练样本检测, 如发生错误分类, 则利用错分的信息进行修正;

□ 先将某一 j 类的增广样本向量 $\mathbf{y}_j$, 与该类所有增广权向量求内积 $\mathbf{a}_j^l(k)^T \mathbf{y}_j$, 找到其中的最大值

$$\mathbf{a}_j^m(k)^T \mathbf{y}_j = \max_{l=1,2,\dots,l_j} \{\mathbf{a}_j^l(k)^T \mathbf{y}_j\}$$

□ 将 $\mathbf{y}_j$ 与其它非 j 类的权向量求内积, 如

$$\mathbf{a}_j^m(k)^T \mathbf{y}_j > \mathbf{a}_i^l(k)^T \mathbf{y}_j, \quad \forall i=1,2,\dots,c, \quad i \neq j, \quad l=1,2,\dots,l_i,$$

则 $\mathbf{a}_i^l(k)$ 不影响 $\mathbf{y}_j$ 的正确分类, **不需修改**。

5.3 错误修正算法

□ 算法步骤

2. **检测与修正**

□ 如果存在某个或几个子类不满足上述条件, 即存在子类 ω_i^n 的现有权向量 $\mathbf{a}_i^n(k)$ 使得

$$\mathbf{a}_j^m(k)^T \mathbf{y}_j \leq \mathbf{a}_i^n(k)^T \mathbf{y}_j, \quad i \neq j$$

则 $\mathbf{y}_j$ 被错分类, 需修正有关权向量。

$$\text{设 } \mathbf{a}_i^n(k)^T \mathbf{y}_j = \max_{i,n} \{\mathbf{a}_i^n(k)^T \mathbf{y}_j\},$$

$$\text{则修正算法为 } \begin{cases} \mathbf{a}_j^m(k+1) = \mathbf{a}_j^m(k) + \rho_k \mathbf{y}_j \\ \mathbf{a}_i^n(k+1) = \mathbf{a}_i^n(k) - \rho_k \mathbf{y}_j \end{cases}$$

5.3 错误修正算法

□ 算法步骤

3. **迭代与停止**: 利用新的权向量值对下一个训练样本重复步骤二, 直到算法收敛或者强迫其收敛。

□ 收敛性

■ 在样本集确实能被分段线性判别函数正确划分的情况下, 迭代算法是收敛的。当该条件不满足时, 则需逐步减小 ρ_k 的数值, 迫使其“收敛”, 显然会有相应的分类错误率存在。

5.3 错误修正算法

□ 未知子类数目的情况 (简介)

■ 树状分段线性分类器: 先设计一个线性分类器, 将所有样本分成两个子类; 若子类中有错分, 则在其中再分..., 直到全部样本正确分类。

