第三章 判别域代数界面方程法

3.6 一般情况下的判别函数权矢量算法

问题：

一次准则函数及其算法（如感知器算法）只适用于线性可分的情况，如果是线性不可分的，分类过程将不收敛！

能否找到一种算法，使之能够测试出模式样本集是否线性可分，并且对线性不可分的情况也能给出“次最优”的解？

最小错分模式数目准则：

对线性不可分样本集，求一解矢量使得错分的模式数目最少。

对于两类问题，设 $n+1$ 维增广训练模式 $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N$ 已符号规范化。

如果训练模式是线性可分的，则存在权矢量 $\vec{w}$ 使不等式组 $\vec{w}'\vec{x}_i > 0 \quad (i = 1, 2, \cdots, N)$ 成立。

如果训练模式是线性不可分，不等式组没解。此时，目标⇒最少的训练模式被错分。

3.6 一般情况下的判别函数权矢量算法

将上面的不等式组写成矩阵方程形式，并引入 N 维余量矢量 $\vec{b}>\vec{\phi}$ ，于是不等式方程组变为:

$$X\vec{w} \geq \vec{b} > \vec{\phi}$$

式中 X 是 $N\times(n+1)$ 矩阵。

$$X = \begin{pmatrix} \vec{x}_1' \\ \vec{x}_2' \\ \vdots \\ \vec{x}_N' \end{pmatrix} = (\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N)'$$

3.6 一般情况下的判别函数权矢量算法

3.6.1 分段二次准则函数及共轭梯度法

- **分段二次准则函数**

$$J(\vec{w}) = \left\| (X\vec{w} - \vec{b}) - \left| X\vec{w} - \vec{b} \right| \right\|^2 \Rightarrow \min$$

- **最优解** $\vec{w}^*$

在训练模式线性可分的情况下，$J(w^*) = 0$，
即对所有模式均能正确分类。
在训练模式非线性可分的情况下，$J(w^*) > 0$，
但可能使误分模式数最少。

3.6.1 分段二次准则函数及共轭梯度法

共轭梯度法是迭代中沿梯度的共轭方向搜索，求得序列 $\boldsymbol{w}_0, \boldsymbol{w}_1, \cdots, \boldsymbol{w}^*$， 使 $J(\boldsymbol{w}_0) \geq J(\boldsymbol{w}_1) \geq \cdots \geq J(\boldsymbol{w}^*)$

共轭梯度法对二次函数 $J(\boldsymbol{w})$非常有效，最多用 $\boldsymbol{w}$维数$(n+1)$ 步就可以使$\{\boldsymbol{w}_k\}$收敛于 $J(\boldsymbol{w})$ 的极小值解 $\boldsymbol{w}^*$ 。对于其它类型的函数，由于极值点附近都可以用二次函数近似，因此也是有效的，但迭代的步数可能要多于$n+1$ 。

3.6.1 分段二次准则函数及共轭梯度法

令 $\vec{S}_k$ 为寻求 $J(\vec{w})$ 极小值点的搜索方向，

则共轭梯度法的基本公式为：

$$\vec{g}_k = \vec{g}(\vec{w}_k) = -\nabla J(\vec{w}_k)$$

$$\vec{s}_0 = \vec{g}_0 \qquad \vec{s}_{k+1} = \vec{g}_{k+1} + \beta_k \vec{s}_k$$

$$\beta_k = \frac{\vec{g}'_{k+1}\vec{g}_{k+1}}{\vec{g}'_k\vec{g}_k} = \frac{\|\vec{g}_{k+1}\|}{\|\vec{g}_k\|}$$

3.6 一般情况下的判别函数权矢量算法

3.6.1 分段二次准则函数及共轭梯度法

令 $\vec{s}_k$ 为寻求 $J(\vec{w})$ 极小值点的搜索方向，

则共轭梯度法的基本公式为：

$$\vec{w}_{k+1} = \vec{w}_k + \nu_k \vec{s}_k$$

$$\nu_k : J(\vec{w}_k + \nu_k \vec{s}_k) = \min_{\nu} J(\vec{w}_k + \nu \vec{s}_k)$$

3.6.2 最小平方误差准则及W-H算法

针对方程组 $X\vec{w}=\vec{b}$，构造方差准则函数

$$J(\vec{w})=(X\vec{w}-\vec{b})'(X\vec{w}-\vec{b})$$
$$=\sum_{i=1}^{N}(\vec{w}'\vec{x}_i-b_i)^2 \Rightarrow \min$$

对于 $\vec{w}'\vec{x}_i=b_i \quad (i=1,2,\cdots,N)$，**此时的** $J=\min J(\vec{w})=0$ ，

而对于 $\vec{w}'\vec{x}_i\neq b_i$ ，**此时的** $J(\vec{w})>0$ 。

如果方程组有唯一解，说明训练模式集是线性可分的，如果方程组无解，极小点值是最小二乘解。一般情况下使 J 极小等价于误分模式数目最少。

3.6 一般情况下的判别函数权矢量算法

3.6.2 最小平方误差准则及W-H算法

求解最佳权矢量的方法——（1）伪逆法

求 $J(\vec{w})$ 对 $\vec{w}$ 的梯度并令其为零，有:

$$\nabla J(\vec{w}) = 2X'(X\vec{w} - \vec{b}) = 0$$

可得: $X'X\vec{w} = X'\vec{b}$

当 $(X'X)^{-1}$ 存在时， $\vec{w} = (X'X)^{-1}X'\vec{b} \triangleq X^{+}\vec{b}$

$X^{+}=(X'X)^{-1}X'$ 称为X的伪逆(有的书称为广义逆或M—P逆)， $\vec{w} = X^{+}\vec{b}$ 称为 $\vec{w}$ 的伪逆解。

3.6.2 最小平方误差准则及W-H算法

求解最佳权矢量的方法——（1）伪逆法

当$(X'X)^{-1}$存在时，　$\vec{w}=(X'X)^{-1}X'\vec{b} \triangleq X^{+}\vec{b}$

$X'X$是$(n+1)\times(n+1)$矩阵，一般是非奇异的。当$(X'X)^{-1}$ 不存在时，可用广义逆法解 $\vec{w}=(X'X)^{+}X'\vec{b}$，这里$(X'X)^{+}$为$X'X$的广义逆矩阵。

求矩阵的广义逆计算量较大，引入的误差也可能很大，在实际中多采用梯度法。

3.6.2 最小平方误差准则及W-H算法

求解最佳权矢量的方法——（2）梯度法

$J(\vec{w})$的梯度为 $\nabla J(\vec{w}) = 2X'(X\vec{w} - \vec{b})$

梯度下降算法迭代公式为：

① $\vec{w}(0)$任取；

② $\vec{w}(k+1) = \vec{w}(k) - \rho_k X'(X\vec{w}(k) - \vec{b})$

可以证明，当$\rho_k = \rho_1 / k$，ρ_1为任意正的常数，则该算法使权矢量序列 $\{\vec{w}(k)\}$ 收敛于 $\vec{w}^*$；满足

$\nabla J(\vec{w}^*) = 0$　　$\vec{w}^*$也称为MSE解。

3.6.2 最小平方误差准则及W-H算法

为了减少计算量和存储量，可以仿照单样本修正法:

由于：
$$X'(X\vec{w}-\vec{b})=\sum_{k=1}^{N}(\vec{w}'\vec{x}_k-b_k)\vec{x}_k$$

于是迭代式修改为：

① $\vec{w}(0)$ 任取；

② $\vec{w}(k+1)=\vec{w}(k)+\rho_k(b_k-\vec{w}(k)'\vec{x}_k)\vec{x}_k$

此算法通常称为W—H(Widrow—Hoff)算法

3.6 一般情况下的判别函数权矢量算法

3.6.2 最小平方误差准则及W-H算法

W-H算法有两个性质

1. 当 $\vec{b}=(\underbrace{\frac{N}{N_1}\ \frac{N}{N_1}\cdots\frac{N}{N_1}}_{N_1\text{个}}\ \underbrace{\frac{N}{N_2}\cdots\frac{N}{N_2}}_{N_2\text{个}})'$ 时，MSE解 $\vec{w}$ 等价于Fisher解。

2. 令 $\vec{b}=(1,1,\cdots,1)'$，在样本数 $N\to\infty$ 时，MSE解以最小均方误差逼近贝叶斯判决函数：

$$d_B(\vec{x})=P(\omega_1|\vec{x})-P(\omega_2|\vec{x})$$

3.6.3 H-K(Ho-Kashyap)算法

前面算法中，余量矢量 $\vec{b}$ 是取定的常矢量，而这将使所得的解 $\vec{w}$ 受 $\vec{b}$ 的影响。

H-K(Ho-Kashyap)算法就是针对前面算法不足的一种改进。

H-K算法仍使用平方误差准则函数

$$J(X,\vec{w},\vec{b}) = \left\| X\vec{w} - \vec{b} \right\|^2 = \sum_{i=1}^{N} (\vec{w}'\vec{x}_i - b_i)^2 \Rightarrow \min$$

这种算法是将准则函数 $J(\bullet)$ 视为 $\vec{w}$ 和 $\vec{b}$ 的函数，在迭代过程中修正 $\vec{w}$ 的同时，也对矢量 $\vec{b}$ 进行调整。

3.6.3 H-K(Ho-Kashyap)算法

H-K算法的迭代公式为：

$$\vec{b}(k+1)=\vec{b}(k)-\rho\nabla_b J(\bullet)\equiv\vec{b}(k)+\vec{\beta}(k)$$

其中 $\nabla_b J(\bullet)=-2(X\vec{w}-\vec{b})$　　$\rho>0$

若 $X\vec{w}(k)-\vec{b}(k)\le 0$ **，则**　$\vec{\beta}(k)=0$

若 $X\vec{w}(k)-\vec{b}(k)>0$ **，则**

$$\vec{\beta}(k)=-\rho\left(\frac{\partial J}{\partial\vec{b}}\right)_{\vec{b}=\vec{b}(k)}=2\rho\left[X\vec{w}(k)-\vec{b}(k)\right]$$

3.6.3 H-K(Ho-Kashyap)算法

记误差矢量： $\vec{e}(k)=X\vec{w}(k)-\vec{b}(k)$

于是上述 $\vec{\beta}(k)$可统一写为： $\vec{\beta}(k)=\rho\left[\ \vec{e}(k)+|\vec{e}(k)|\ \right]$

由伪逆法可知： $\vec{w}=(X'X)^{-1}X'\vec{b}\equiv X^{+}\vec{b}$

于是$\vec{w}$的迭代公式为：

$$\vec{w}(k+1)=X^{+}\vec{b}(k+1)=X^{+}\vec{b}(k)+X^{+}\vec{\beta}(k)$$

$$=\vec{w}(k)+X^{+}\vec{\beta}(k)$$

H-K算法步骤

Step1. 将训练样本符号规范化，得X

求伪逆 $X^{+}=(X'X)^{-1}X'$

Step2. 置初值 $\vec{b}(1)>\vec{0}, k=1, \rho=\dfrac{1}{2};$

Step3. 计算 $\vec{w}(k)=X^{+}\vec{b}(k)$

$\vec{e}(k)=X\vec{w}(k)-\vec{b}(k)$

Step4. IF （$\vec{e}(k)$的各分量停止变为正值）

then End.（线性不可分，无解）

else IF $\vec{e}(k)\Rightarrow\vec{0}$

then 输出 $\vec{w}(k)$ End.

H-K算法步骤

Step5. $\vec{w}(k+1) = \vec{w}(k) + \rho X^{+}\left[\vec{e}(k) + \left|\vec{e}(k)\right|\right]$

$$= \vec{w}(k) + \rho X^{+}\left|\vec{e}(k)\right|$$

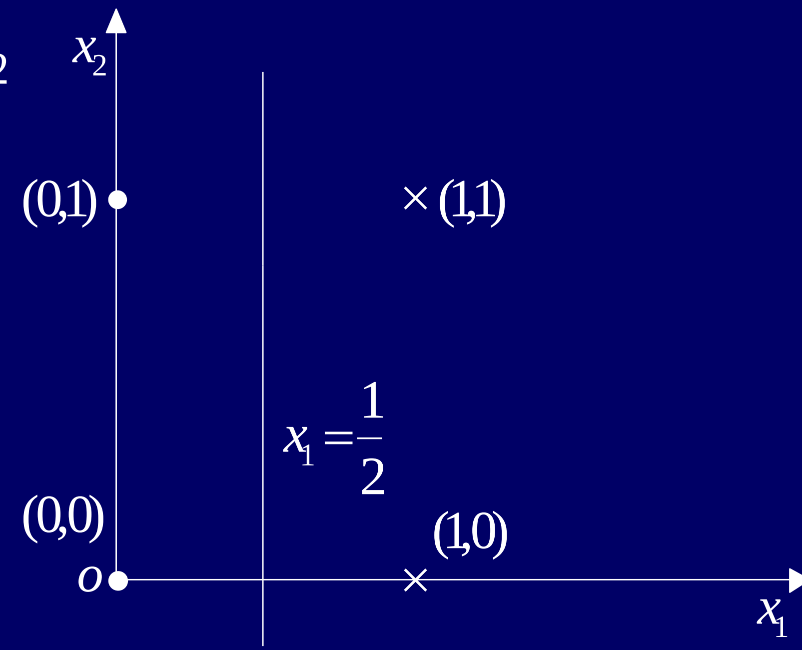
$$\vec{b}(k+1) = \vec{b}(k) + \rho(\vec{e}(k) + \left|\vec{e}(k)\right|);$$

Step6. ***k=k+1***; goto Step3;

例　已知训练模式分布如图所示，其中，

$$(0,0)',(0,1)' \in \omega_1;(1,0)',(1,1)' \in \omega_2$$

试用H—K算法进行分类器训练。



解：模式的增广矩阵为：

$$X=\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

则伪逆：

$$X^{+}=(X'X)^{-1}X'=\frac{1}{2}\begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 \\ 3/2 & 1/2 & -1/2 & 1/2 \end{pmatrix}$$

3.6 一般情况下的判别函数权矢量算法

取余量矢量$\vec{b}(0)=(1,1,1,1)'$，常数ρ=1，由算法可知:

$$\vec{w}(0)=X^{+}\vec{b}(0)=(-2,0,1)'$$

$$\vec{e}(0)=X\vec{w}(0)-\vec{b}(0)=(1,1,1,1)'-\vec{b}(0)=(0,0,0,0)'$$

所以$\vec{w}(0)$就是所求的解 $\vec{w}^{*}$，从而可得:

$$d(\vec{x})=-2x_{1}+1$$

谢谢！