



4.4 最小错分样本数准则

- 由于感知准则函数及其梯度下降法只适用于线性可分情况，对于线性不可分情况，迭代过程永远不会终结，即算法不收敛。但在实际问题中往往无法事先知道样本集是否线性可分，因此，我们希望找到既适用于线性可分情况，又适用于线性不可分情况的算法。这种算法对于线性可分问题，可以得到一个如感知准则函数那样的解向量 α^* ，使得对两类样本集作到将全部样本正确分类；而对于线性不可分问题，则得到一个使两类样本集错分数目最少的权向量 α ，也记为 α 。我们把这样的准则称为最小错分样本数准则。这里介绍两种最优化这种准则的算法。

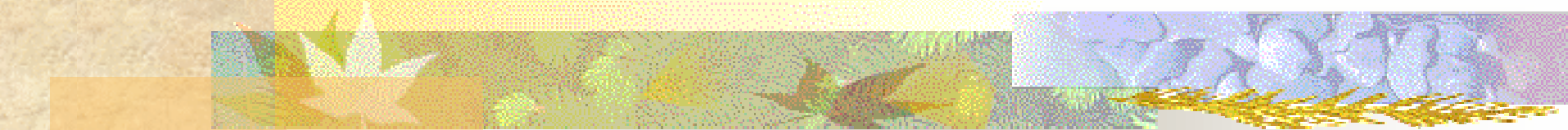


4.4.1 解线性不等式组的共轭梯度法

- 对于规范化增广样本向量，线性判别函数可写作 $g(x) = \alpha^T y$ 。如果存在权向量 α ，使得

$$\alpha^T y_n > 0, n = 1, 2, \dots, N \quad (4-44)$$

则我们说 y_n 被正确分类。因此，设计线性分类器的任务可以看成求一组 N 个线性不等式（4-44）的解的问题。若不等式组有解，即不等式组相一致的情况，说明样本集是线性可分的，我们找到这个解向量 α^* 。

- 
- 若不等式组无解，即不等式组不一致的情况，说明样本集是线性不可分的，对于任何权向量 α ，必有某些样本被错分类，这时我们可以转而寻找使最多数目的不等式得到满足的权向量 α ，把它作为问题的解。这样 α^* 分类器设计问题就转变成解线性不等式组的问题了。
 - 下面我们仍首先给出一个准则函数，然后用共轭梯度法求使准则函数取极值时的解向量 α^* 。

- 现在，我们用矩阵形式重写（4-44）式所表示的不等式组，
$$Y\alpha > 0 \quad (4-45)$$

式中

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1\hat{d}} \\ y_{21} & y_{22} & \cdots & y_{2\hat{d}} \\ \cdots & \cdots & \cdots & \cdots \\ y_{N1} & y_{N2} & \cdots & y_{N\hat{d}} \end{bmatrix} \quad (4-46)$$

Y 是 $N \times \hat{d}$ 规范化增广样本矩阵， \hat{d} 是样本 y 的维数。
为使解更可靠，引入余量 $b > 0$ ，那么（4-45）式可写成

$$Y\alpha \geq b > 0 \quad (4-47)$$

式中 b 是一个 N 维向量，

不失一般性，我们可以取

$$b = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \left. \vphantom{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}} \right\} N \text{个} 1 \quad (4-48)$$

对于 (4-47) 可以定义准则函数


$$J_{q1}(\alpha) = \| (Y\alpha - b) - |Y\alpha - b| \| ^2 \quad (4-49)$$

如果 $Y\alpha > b$ ，则 $(Y\alpha - b)$ 和 $|Y\alpha - b|$ 同号，因此，

$J_{q1}(\alpha) = 0$ ，反之，如果有某些 y_i 不满足 $\alpha^T y_i > b_i$ ，则 $(\alpha^T y_i - b_i)$ 和 $|\alpha^T y_i - b_i|$ 异号，因此， $J_{q1}(\alpha) > 0$ 。不满足的 y_i 越多， $J_{q1}(\alpha)$ 越大。显然， $J_{q1}(\alpha)$ 取极小值时的 α 为最优解 α^* ，并且在不等式组一致的情况下，

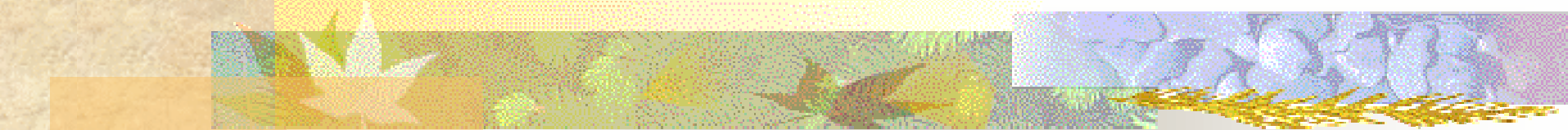
$J_{q1}(\alpha^*) = 0$ ，在不一致的情况下， $J_{q1}(\alpha^*) > 0$ 。

我们称 $J_{q1}(\alpha)$ 为 最小错分样本数准则1。

- 
- 下面讨论一下在Fletcher-Reeves共轭梯度算法的基础上求解上述问题的方法和步骤，它是由Nagaraja和Krishna提出来的。
 - 首先简单说明一下共轭梯度算法的基本概念。设 B 是一个 $d \times d$ 阶对称正定矩阵，若有两个 d 维向量 u 和 v 使 $(u, Bv) = 0$ ，则称 u 和 v 对于矩阵 B 互为共轭。显然，若 u 和 v 对于单位阵 I 互为共轭，则 u 和 v 正交。
当 x 和 y 是 B 的本征向量时，有

$$(y, Bx) = (y, \lambda x) = \lambda(y, x) = 0$$

因此，正定矩阵 B 的本征向量(x 和 y)对于矩阵 B 互为共轭。


- 
- 共轭梯度算法就是以 E^d 空间中的一组对于B互为共轭的向量为一维搜索方向，使二次正定函数

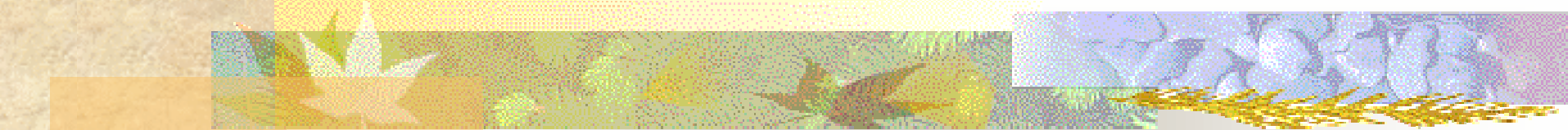
$$f(x) = b_0 + b^T x + x^T B x$$

达到极小值的最优化算法。用共轭梯度算法可以求得序列 x_0, x_1, x_2, \dots ，使得

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq \dots$$

可以证明，对于二次正定函数 $f(x)$ ，最多用 d 步，就可以使序列 $\{x_k\}$ 收敛于 $f(x)$ 的极值解 x^* 。

- 
- 由于（4-49）定义的准则函数不是一个二次正定函数，而是一个分段二次正定函数，因此，在沿 d 个（对于增广空间则为 $d+1$ 个）互为共轭的向量进行一维搜索后，有可能达不到准则函数 $J_{q1}(\alpha)$ 的最小值，即算法经过 d （或 $d+1$ 步）可能不收敛，这时就要重新开始计算，若用 r 表示重新开始的周期，则 $r=d$ （或 $d+1$ ）。



■ 在任意点 α , $J_{q1}(\alpha)$ 的负梯度方向可表示为

$$g(\alpha) = -\frac{1}{4} \nabla_{\alpha} J_{q1}(\alpha) = Y^T [|Y\alpha - b| - (Y\alpha - b)] = Y^T P \quad (4-50)$$

式中

$$P = [|Y\alpha - b| - (Y\alpha - b)] \quad (4-51)$$

令

$$\lambda = \frac{1}{\|g\|^2} \quad (4-52)$$

用 k 表示迭代步数, 用 γ 表示满足于 α 的不等式的数目,
 α^* 表示最优解。



■ 这种算法的具体步骤如下：

步骤1 置 $k=0$ ，并任意给定初始权向量 α_0 ，计算 $Y\alpha_0$ 和 γ_0 。如果 $0 < \gamma < N/2$ ，则令 $\alpha_0 = -\alpha_0$ ，

$Y\alpha_0 = -Y\alpha_0$ ， $\gamma_0 = N - \gamma_0$ ，然后继续。

步骤2 如果 $\gamma_k = N$ ，则令 $\alpha^* = \alpha_k$ ，停止；如果 $\gamma_k = 0$ ，则令 $\alpha^* = -\alpha_k$ ，停止；否则，继续。

步骤3 计算 g_k 。如果 $g_k = 0$ ，则停止；否则计算 λ_k ，然后继续。

步骤4 求 θ_k 。如果 k 为 r 的整数倍，则令 $\theta_k = 0$ ，否则 $\theta_k = 1$ ，并计算

$$S_k = \theta_k S_{k-1} + \lambda_k g_k.$$

这里 S_k 表示第 k 次搜索时的梯度下降方向。



若 α_0 表示对 α^* 的第一次逼近,则

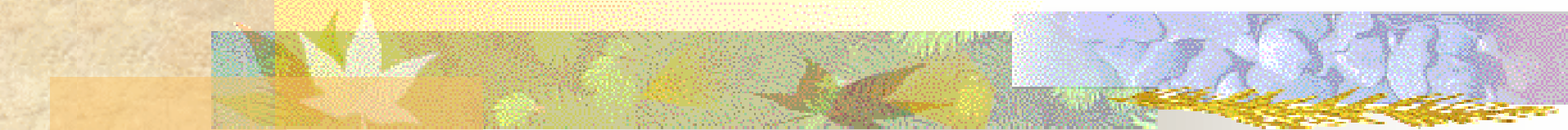
$$S_0 = \lambda_0 g_0$$

可以证明,由上述表达式所产生的 S_1, S_2, \dots , 对于二次函数中的正定矩阵是互为共轭的.

步骤5 寻找最佳步长 v_k , 即计算使 $J(\alpha_k + vS_k)$ 取极小值时的 v .

步骤6 令 $\alpha_{k+1} = \alpha_k + v_k S_k$, $Y \alpha_{k+1} = Y \alpha_k + v_k Y S_k$, 并计算 γ_{k+1} .

步骤7 令 $k=k+1$, 转向步骤 2 .


- 
- Nagaraja和Krishna证明,对于式(4-49)表示的分段二次函数,在 $Ya \geq b > 0$ 的一致条件下,上述算法可以在有限步内使序列 $\{\alpha_k\}$ 收敛于最优解 α^* .而在

$$Ya \geq b > 0$$

不一致条件下,只要适当的选择b ,使在 $J_{q1}(\alpha)$ 的唯一极小点 α^* 上,有

$$\alpha^{*T} y_i \neq b_i, (i = 1, 2, \dots, N)$$

则该算法产生的序列 $\{\alpha_k\}$ 也在有限步内收敛于 α^* .

- 
- 对于式(4-49)表示的准则函数 $J_{q1}(\alpha)$,在不等式组不一致的情况下,对某些样本,可能存在

$$0 < \alpha^T y_i < b_i \quad .$$

因此就产生了一个阈值问题.这时,由于 $\alpha^T y_i > 0$, y_i 应被正确分类;但又由于 $\alpha^T y_i < b_i$,所以在算法中是按错分类处理的.下面讨论的补充算法,就是针对这一问题而提出的.

- 
- 如果在式(4-49)中,我们假定 $\mathbf{b}=\mathbf{0}$,那么准则函数成为

$$J_{q1}(\alpha) = \|Y\alpha - |Y\alpha|\|^2 \quad (4-53)$$

很明显,上述算法可以使式(4-53)中 $J_{q1}(\alpha)$ 极小化,在一致情况下收敛于 $Y\alpha > 0$ 的解 α^* .

在不一致的情况下,由于 $J_{q1}(\alpha)$ 是严格的凸函数,其唯一极小点是 $\alpha = 0$,而且有

$$J_{q1}(\alpha) = 0.$$

因此, $\alpha^{*T} y_i \neq b_i, (i = 1, 2, \dots, N)$ 的条件不成立,所以得不到解向量 α^* .

- 
- 对于模式识别问题来说,我们可以用

$$F(\alpha) = \frac{\|Y\alpha - |Y\alpha|\|^2}{\|\alpha\|^2} \quad (4-54)$$


作为准则函数来解决上述问题.显然,这时存在下列关系

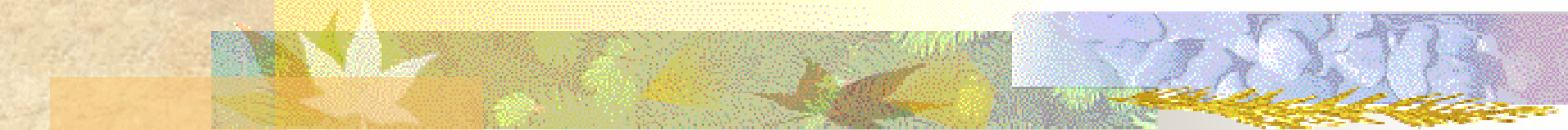
$$\nabla_a F(\alpha) = \nabla_a J_{q1}(\alpha) - 2F(\alpha) \cdot \alpha = 0 \quad (4-55)$$

也就是说,使 $F(\alpha)$ 最小,同在终止条件

$$\nabla_a J_{q1}(\alpha) = 2F(\alpha) \cdot \alpha \quad (4-56)$$

和 $\alpha \neq 0$ 下使 $J_{q1}(\alpha)$ 最小是等价的.这时需要对上述算法的步骤1和步骤4改变如下:

- 
- 步骤1' 首先通过原有算法得到一个收敛点,记为 α_s , 并以此作为补充算法的起点.
 - 步骤4' 计算 $\mu_k = -(\alpha_k^T g_k) / \|\alpha_k\|^2$ 和 $S_k = \mu_k \alpha_k + g_k$, 并且继续.
 - 可以证明,这样得到的 S_k 仍然是 $J_{q1}(\alpha)$ 的下降方向.同时可以证明,假使 $Y\alpha > 0$ 是不一致的,且在求 $J_{q1}(\alpha)$ 最小值的过程中用步骤4' 代替原算法的步骤4,若所得到的序列 $\{\alpha_k\}$ 是有限的,则序列的最后一个元素就相当于 $F(\alpha)$ 的一个局部最小值的解.

- 
- 若序列是无限的,则它趋向于 $F(\alpha)$ 的一个局部最小值的解.
 - 在进行上述计算时,由于我们使用原算法的收敛点 α_s 作为起始点,它常常是全局最优解的一个很好的逼近,故可以得到 $F(\alpha)$ 的全局最优解.



4.5 最小平方误差准则函数

■ 4.5.1 平方误差准则函数

- 在4.3节和4.4节中介绍了基于解线性不等式组的算法。他们的共同点是企图找一个权向量 \mathbf{a} ，使得得到满足的不等式 $\mathbf{a}^T \mathbf{y}_n > 0$ 的数目最大，从而使错分样本数最少。在不等式组一致的情况下，则得到一个解区中的解向量 \mathbf{a}^* 。
- 现在我们把不等式组变成如下形式：

$\mathbf{a}^T \mathbf{y}_n = b_n > 0$

其中 b_n 是任意给定的正常数。将上式写成连立方程组的形式即为

$$Y\mathbf{a} = \mathbf{b} \quad (4-62)$$

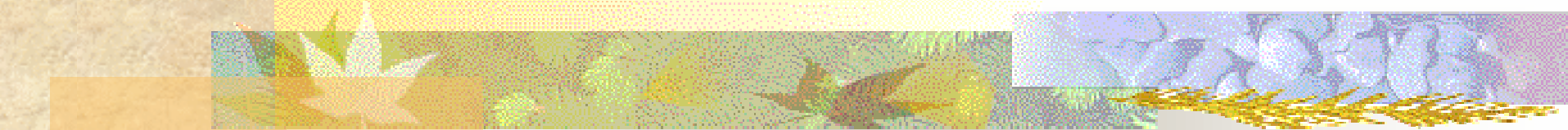
■ 式中,

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1\hat{d}} \\ y_{21} & y_{22} & \cdots & y_{2\hat{d}} \\ \cdots & \cdots & \cdots & \cdots \\ y_{N1} & y_{N2} & \cdots & y_{N\hat{d}} \end{bmatrix}$$

■ 是一个 $N \times \hat{d}$ 矩阵, y_n 是规范化增广样本向量,

$$b = [b_1, b_2, \cdots, b_N]^T$$

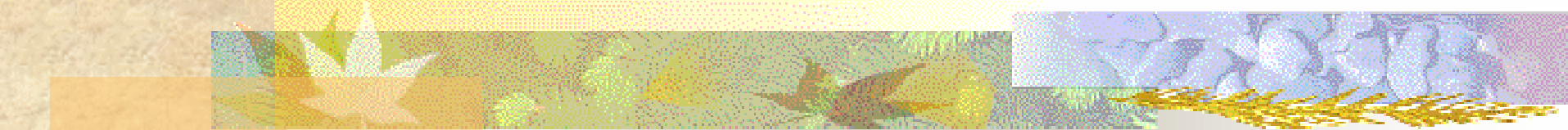
■ 是一个N维向量, $b_n > 0, n = 1, 2, \cdots, N$

- 
- 通常样本数 N 总是大于维数 \hat{d} ，因此 Y 是长方阵，一般为列满秩阵。这实际上是方程个数多于未知数的情况，因此一般为矛盾方程组，通常没有精确解存在。但我们可以定义一个误差向量

$$e = Ya - b$$

- 并定义平方误差准则函数 $J_s(a)$ 为

$$J_s(a) = \|e\|^2 = \|Ya - b\|^2 = \sum_{n=1}^N (a^T y_n - b_n)^2 \quad (4-63)$$

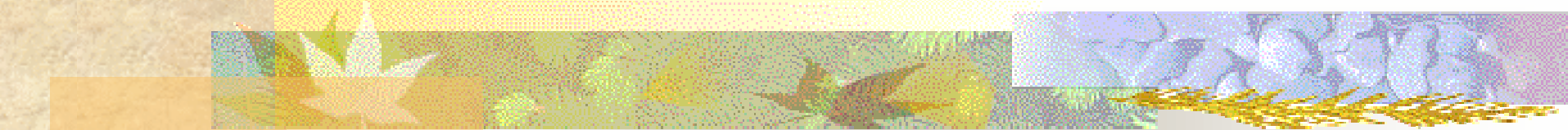
- 
- 然后找一个使 $J_s(a)$ 极小化的 a 作为问题的解，这就是矛盾方程组的最小二乘近似解，也称伪逆解或称MSE解，我们仍用 a^* 表示。式（4-63）定义的准则函数也称MSE准则函数。现在我们首先用解析法求出它的伪逆解，然后分析这种解在一些特定情况下的性质。

- 首先对式（4- 63）中的 $J_s(a)$ 求梯度，

$$\nabla J_s(a) = \sum_{n=1}^N 2(a^T y_n - b_n) y_n = 2Y^T (Ya - b) \quad (4-64)$$

- 令 $\nabla J_s(a) = 0$ ，得

$$Y^T Y a^* = Y^T b \quad (4-65)$$

- 
- 这样，求解 $Ya = b$ 的问题转化为求解 $Y^T Y a^* = Y^T b$ 的问题了。这一方程的最大优点是，矩阵 $Y^T Y$ 是 $\hat{d} \times \hat{d}$ 方阵，而且一般是非奇异的，因此可唯一地解得

$$a^* = (Y^T Y)^{-1} Y^T b = Y^+ b \quad (4-66)$$

- 式中 $(\hat{d} \times N)$ 矩阵

$$Y^+ = (Y^T Y)^{-1} Y^T \quad (4-67)$$

- 是 Y 的左逆矩阵， a^* 就是式 (4-62) 的MSE解。
- 现在的问题是向量 b 应如何选取。显然， a^* 依赖于 b 。下面我们就来说明，当 b 取某些特殊值时，MSE解将具有某些优良特性。

同Fisher线性判别的关系

- 可以证明，当取

$$b = \left[\begin{array}{c} N/N_1 \\ \vdots \\ N/N_1 \\ N/N_2 \\ \vdots \\ N/N_2 \end{array} \right] \left\{ \begin{array}{l} N_1 \text{个} \\ N_2 \text{个} \end{array} \right. \quad (4-68)$$

- 时，MSE解 a^* 等价于Fisher解。同时我们得到，

$$\omega_0^* = -m^T \omega^* \quad (4-69)$$

- 和如下决策规则，

- 
- 若 $\omega^{*T}(x - m) \geq 0$, 则决策 $x \in \begin{cases} \omega_1 \\ \omega_2 \end{cases}$ (4-70)

- 其中 m 是总样本均值,

$$m = \frac{N_1 m_1 + N_2 m_2}{N} \quad (4-71)$$

- 这和Fisher线性判别方法中取

$$y_0 = \tilde{m}$$

时的情况是一致的。

4.5.2 MSE准则函数的梯度下降算法

- 前面介绍了求MSE解的伪逆法，得到

$$a^* = Y^+ b$$

- 其中需要计算伪逆 $Y^+ = (Y^T Y)^{-1} Y^T$ 。计算 Y^+ 带来的问题有两个：其一是要求 $(Y^T Y)$ 非奇异；其二是求 Y^+ 时计算量大，同时还可能引入较大的计算误差。因此实际上往往不用这样的解析方法求MSE解，而是采用梯度下降法等最优化技术来求解。

- 如果我们采用梯度下降法，由式（4-64）可知， $J_s(a)$ 的梯度是

$$\nabla J_s(a) = 2Y^T (Ya - b)$$

- 
- 则梯度下降算法可写成

$$\begin{cases} a(1), \text{任意} \\ a(k+1) = a(k) - \rho_k Y^T (Ya - b) \end{cases} \quad (4-78)$$

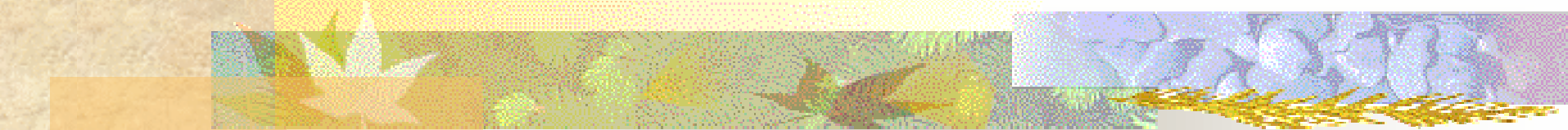
- 可以证明, 如果选择

$$\rho_k = \frac{\rho_1}{k} \quad (4-79)$$

- 式中 ρ_1 是任意正常数, 则用该算法得到的权向量序列收敛于使

$$\nabla J_s(a) = 2Y^T (Ya - b) = 0$$

的权向量 a^* , 也就是MSE解。

- 
- 无论矩阵 $(Y^T Y)$ 奇异与否，该算法总能产生一个有用的权向量，而且该算法只计算 $\hat{d} \times \hat{d}$ 方阵 $(Y^T Y)$ ，比计算 $\hat{d} \times N$ 阵 Y^+ 计算量要小得多。
 - 为了进一步减小计算量和存储量，类似于“4.3感知准则函数”中介绍的单样本修正法那样，我们可以把样本看成一个无限重复出现的序列而逐个加以考虑。这样，式（4-78）确定的算法可修改为

$$\begin{cases} a(1), \text{任意} \\ a(k+1) = a(k) + \rho_k (b_k - a(k)^T y^k) y^k \end{cases} \quad (4-80)$$

- 其中 y^k 为使 $a(k)^T y^k \neq b_k$ 的样本。

- 
- 由于 b_k 是任意给定的正常数，因此，一般说来，要使 $a(k)^T y^k = b_k$

成立几乎是不可能的，因而修正过程永远不会停止，所以必须让 ρ_k 随 k 增加而逐渐减小，以保证算法收敛。一般选择 $\rho_k = \rho_1 / k$ ，此时，式（4-80）确定的算法收敛于满意的解 a^* 。该算法是对MSE准则采用梯度下降法的一个修正算法，通常称为Widrow-Hoff算法。