模式识别的理论与方法

Pattern Recognition

裴继红

Chapter 5 (Part 2):
线性判别函数的学习：Perceptron

(Sections 5.1-5.5)

本讲内容

- 两类线性可分情况的解
- 线性分类器的实现模型：感知器
- 权空间
- 准则函数与感知器的学习
- 感知器的学习算法
- 松弛算法
- 不可分情况的处理

两类线性可分情况的解

现在假设有 n 个样本 $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$，分为两类 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$。

我们的目的是找出下面判别函数的权矢量 $\boldsymbol{a}$：

$$g(x) = \sum_{i=0}^{d} a_i y_i(x) = \mathrm{a}^t \mathrm{y}$$

若存在一个矢量 $\boldsymbol{a}^{\mathrm{t}}$，在代入 $\boldsymbol{g}(\boldsymbol{x})$ 后可以将所有不同类的样本分开，我们称该问题是线性可分的。例如，

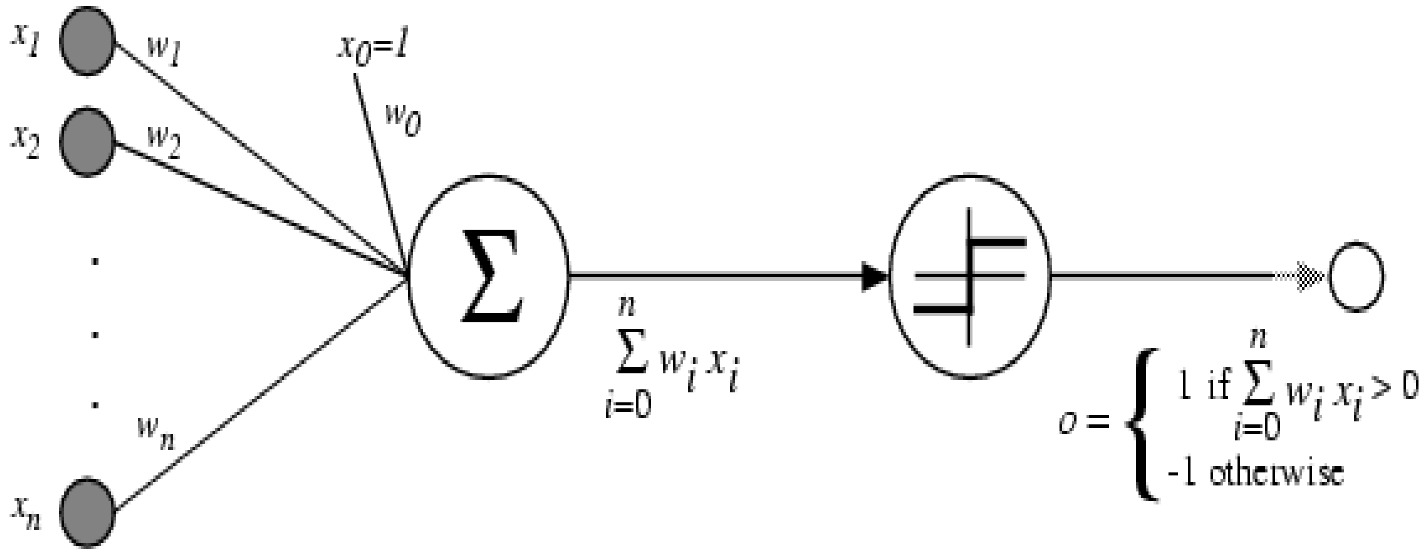
若 $\boldsymbol{y_i}$ 属于 $\boldsymbol{c}_1$ 类时有 $\boldsymbol{a}^{\mathrm{t}} \boldsymbol{y_i} > 0$，或 y_i 属于 $\boldsymbol{c}_2$ 类时 $\boldsymbol{a}^{\mathrm{t}} \boldsymbol{y_i} < 0$，则称 $\boldsymbol{y_i}$ 被正确分类了。

若对所有属于 c_2 类的 $\boldsymbol{y_i}$ 乘以一个负号，则只需要找出使所有样本的 $\boldsymbol{a}^{\mathrm{t}} \boldsymbol{y_i} > 0$ 的矢量 $\boldsymbol{a}^{\mathrm{t}}$，

该矢量 $\boldsymbol{a}^{\mathrm{t}}$ 就是解矢量。

线性分类器的实现模型

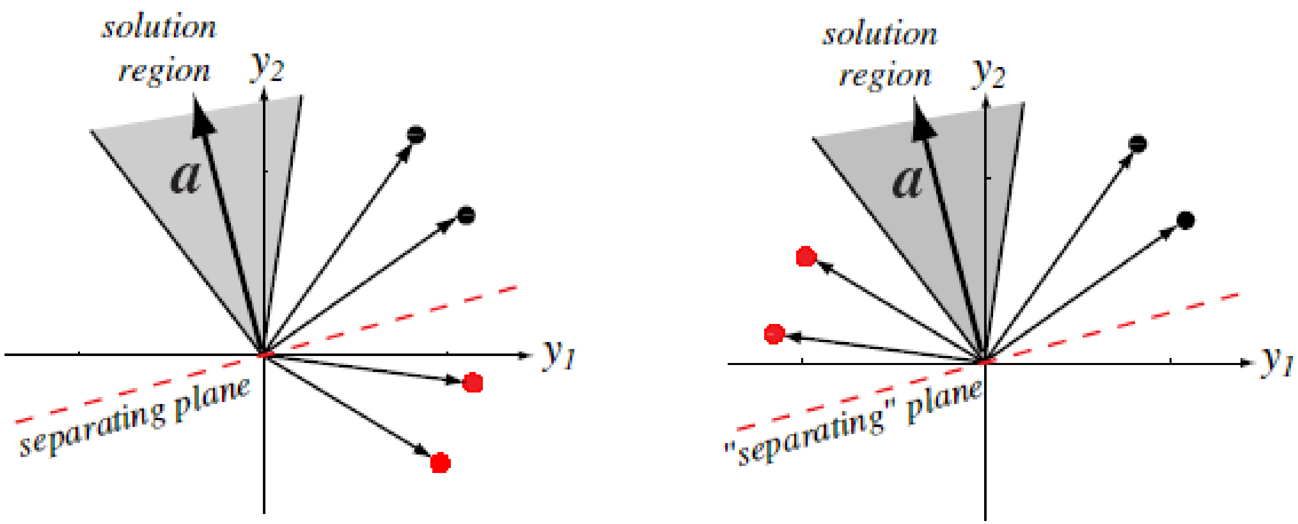
——单层人工神经网络：感知器模型



$$o(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$

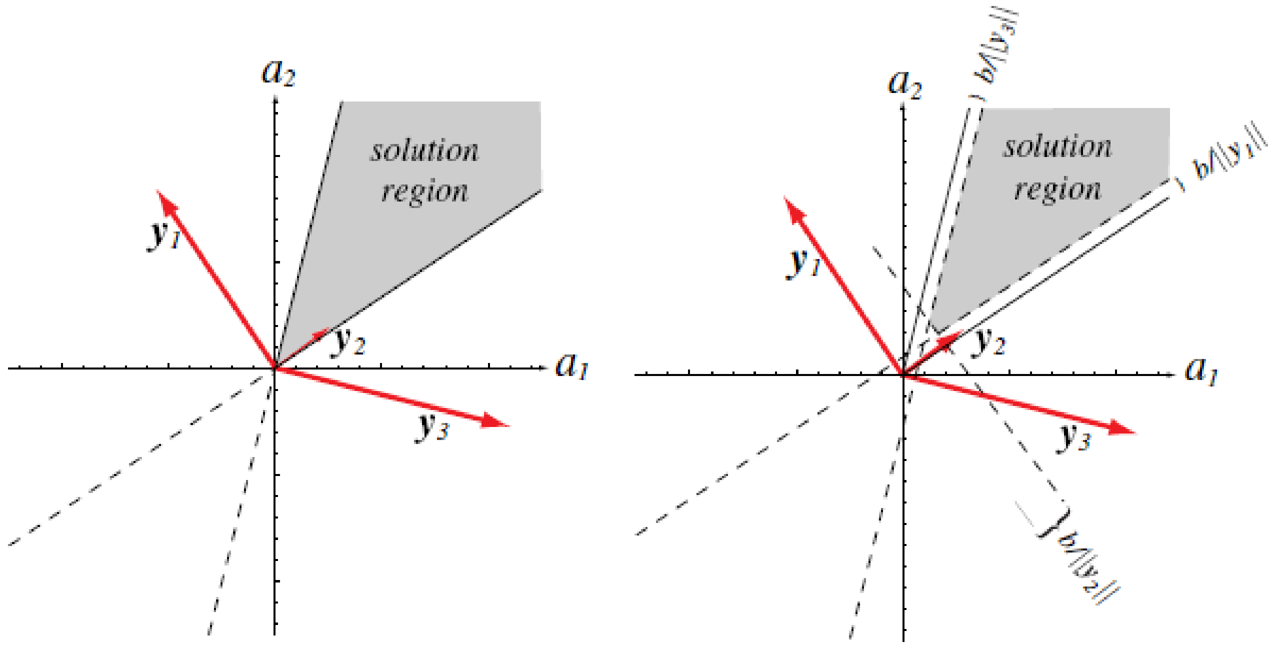
Figure5.8：权空间的几何解释

- 4个训练样本(黑色属于ω1, 红色属于ω2)及其在特征空间的解域。
  - 在左面的图中使用的原始数据，解向量导出的分类面将模式分为2类。
  - 在右图中，红色的点经过了“规范化”—改变了矢量值的符号，这样，解矢量导出的平面只需要将规范后的样本点划分到平面的同一侧。



解空间中边界裕量的作用：Figure5.9

- 左面是没有边界裕量(b= 0) 的解空间，
- 右面是边界裕量 b > 0的解空间，解区域的边界缩小了$b/||y_i||$。
- 边界裕量的存在提高了对新样本的分类性能



线性分类决策面的求解

即求解满足下面线性不等式方程的解：

$$\boldsymbol{a}^{t}\boldsymbol{y}_{i} > 0$$

该问题的解可以使用梯度下降技术求解。

求解优化问题的梯度下降技术

基本思想及步骤：

1. 定义一个准则函数（能量函数、误差函数） $\boldsymbol{J}(\boldsymbol{a})$
2. 令 $k = 1$;
3. 选取任意的初始解矢量 $\boldsymbol{a}$ 值，记为 $\boldsymbol{a}(k)$.
4. 计算 $\boldsymbol{J}(\boldsymbol{a}(k))$ 的梯度：$\nabla \boldsymbol{J}(\boldsymbol{a}(k))$
5. 沿梯度的负方向计算 $\boldsymbol{a}(k+1)$，
   (梯度的负方向是梯度最陡下降的方向).

一般地:
$$\mathrm{a}(k+1) = \mathrm{a}(k) - \eta(k) \nabla J(\mathrm{a}(k))$$

这里，η 称为学习速率。

感知器准则函数1——误分类样本代数度量

求解下面不等式的可行的准则函数

$$\boldsymbol{a}^{\mathrm{t}}\boldsymbol{y}_i > 0$$

可以仅仅简单地选择误分类的样本数。

此即为感知器准则函数:

$$J_p(a) = \sum_{\mathrm{y}\in\Upsilon}\left(-\mathrm{a}^t\mathrm{y}\right)$$

这里，和式是对所有被矢量 $\boldsymbol{a}$ 误分的样本求和。

感知器学习

注意到 $$\nabla J_p(a)=\sum_{y\in\Upsilon}(-y)$$

因此，权矢量的更新规则变为:

$$a(k+1)=a(k)+\eta(k)\sum_{y\in\Upsilon}y$$

这里，和式是对所有被矢量 $\boldsymbol{a}(k)$ 误分的样本求和。
学习规则总结如下:

新的权矢量 = 当前权矢量 + 被当前权矢量误分的所有样本之和 × 学习因子

准则函数2——模式错分总数

令样本集合为Y，则模式错分总数的准则函数可表示为

$$J_E(a)=\frac{1}{2}\sum_{\mathbf{y}\in Y}\left(1\text{-sign}\left(\mathbf{a}^t\mathbf{y}\right)\right)$$

上式中，sign(.)为符号函数

准则函数3——误差平方和

令样本集合为Y，则误差平方和准则函数可表示为

$$J_E(a) = \sum_{\mathbf{y} \in Y_E} \left(\mathbf{a}^t \mathbf{y}\right)^2$$

上式中，$\mathbf{Y_E}$为**错分样本集合**

准则函数4——带边界裕量的误差平方和

令样本集合为Y，

则带边界裕量的误差平方和准则函数可表示为

$$J_{E_2}(a) = \sum_{\mathbf{y}\in Y_E} \left(\mathbf{a}^t\mathbf{y} - b\right)^2$$

上式中，$\mathbf{Y_E}$为错分样本集合，b>0为边界裕量

不同准则下的解区域：Figure5.11

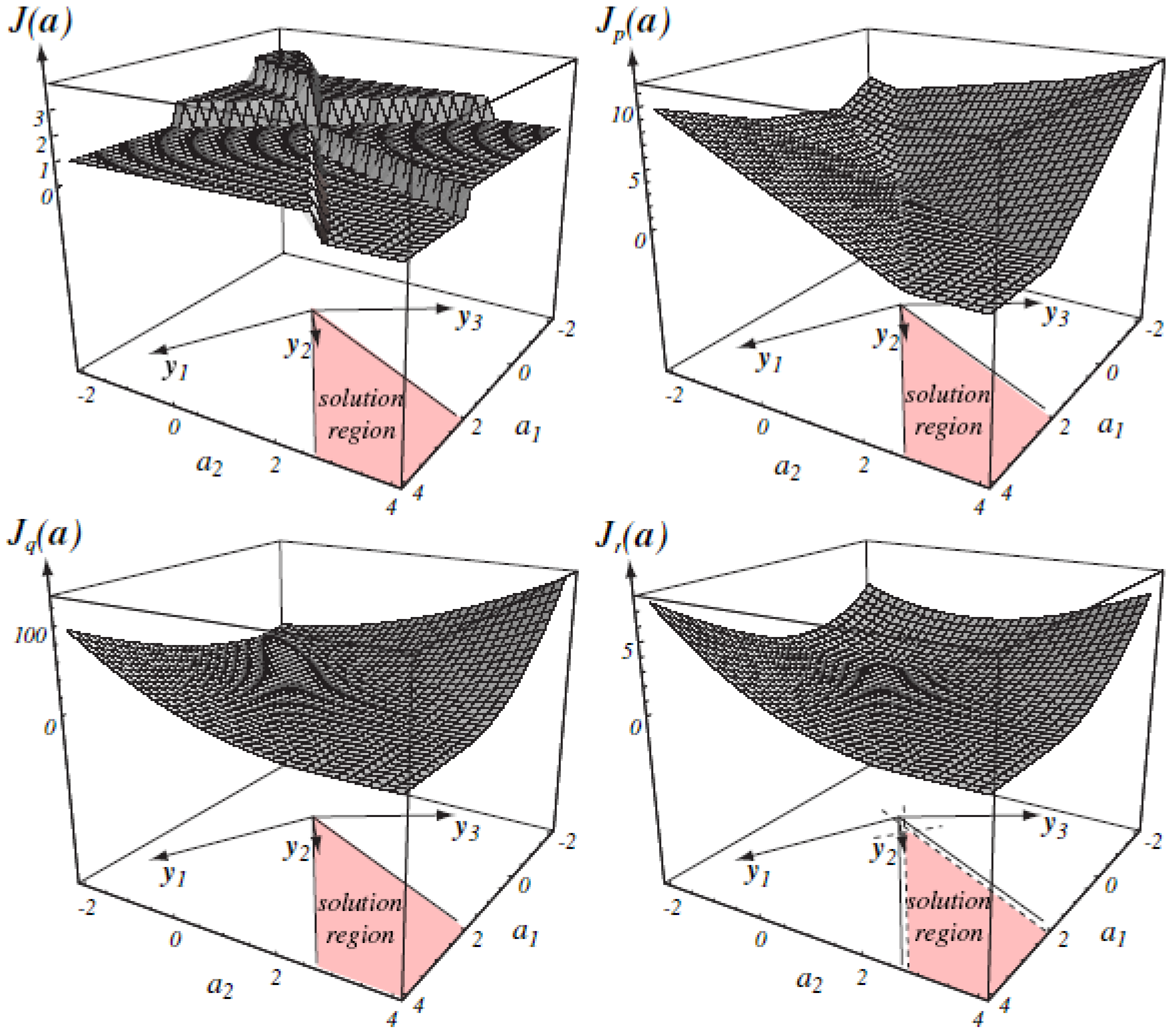
- 图中给出了在线性分类器中的4种学习准则在权空间的曲面。

左上图，**模式错分的总数**，是分段常数曲面，不适合梯度下降法。

右上图，**感知器准则 (Eq. 16)**, 是分段线性函数曲面，可以采用梯度下降法。

左下图，**平方误差函数 (Eq. 32)**, 具有很好的解析特性，即使在模式类线性不可分的情况下也可以使用。

右下图，**带有边界裕量的平方误差函数 (Eq. 33)**. 设计者可以通过调整边界裕量b 使得解矢量处于b = 0的解区域的中部，这样可以改善得到的分类器的推广性能



学习速率 η

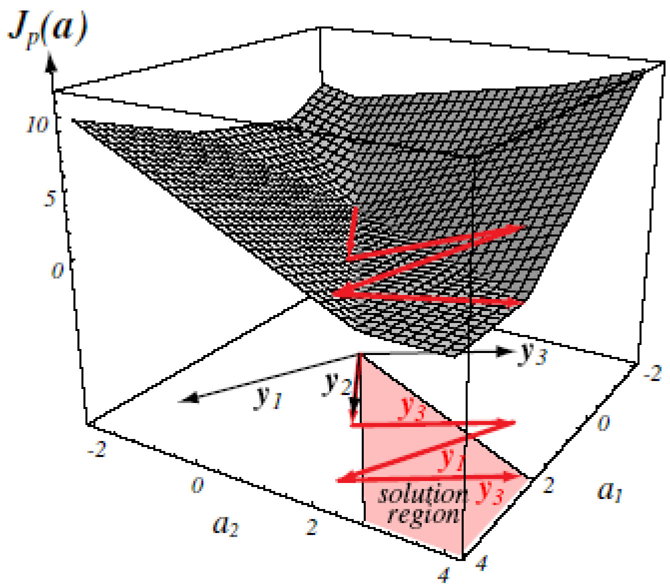
学习速率 η 是梯度下降技术中的关键参数。

a. 若η 太小，则收敛到解时计算的步数太多，收敛时间太长。

b. 若η 太大，则容易引起“过冲”，不易收敛到最优点。

权向量的学习、过冲举例：Figure5.12

- 如图是 3 个样本的感知器学习过程。图中作为 权系数 $\boldsymbol{a}_1$ 和 $\boldsymbol{a}_2$ 的函数，画出了感知器准则函数，$J_p(\boldsymbol{a})$，的曲面。
- 权矢量的初始值为 **0**。随着算法的进行，将错分模式点的规范化矢量相继加到权矢量上。
- 图中给出了以 **y**2, **y**3, **y**1, **y**3为次序进行权矢量修正的过程。最后权矢量落在解区域，迭代结束。
- 注意，第二次更新中 (通过 **y**3)，解矢量比第一次更新后的解距离解区域更远了。



感知器学习算法-1：

———批处理感知器算法

算法的实现步骤：

1. 初始化：权矢量 ***a***，学习步长$\eta(.)$，误差阈值θ，令 ***k*** ＝ 0
2. do ***k***＝***k***+1
3. $$a(k+1)=a(k)+\eta(k)\sum_{y\in Y_k} y$$
4. while $\left|\eta(k)\sum_{y\in Y_k} y\right|>\theta$
5. Return ***a***
6. 结束

- 算法的特点

感知器学习算法- 2：
——固定增量单样本感知器算法

算法的实现步骤:（训练样本已经过规范化）

1. 初始化：权矢量 $\boldsymbol{a}$，学习步长$\eta(.)$，误差阈值θ，令$\boldsymbol{k}=0$
2. do $\boldsymbol{k}=(\boldsymbol{k}+1) \bmod n$ (n为样本总数)
3. $a(k+1)=a(k)-y(k)\cdot\left(\mathrm{sign}[a^t y(k)]-1\right)/2$
4. while 存在 $\boldsymbol{y}(\boldsymbol{k})<0$, $k=1,2,\ldots n$
5. Return $\boldsymbol{a}$
6. 结束

- 算法的特点

感知器学习算法-3：

———带裕量的单样本感知器算法

算法的实现步骤：（训练样本已经过规范化）

1. 初始化：权矢量 $\boldsymbol{a}$，裕量 $\boldsymbol{b}$，$\eta(.)$，误差阈值θ，令$\boldsymbol{k}$= 0
2. do $\boldsymbol{k}$=（$\boldsymbol{k}$+1）mod n （n为样本总数)
3. $$a(k+1)=a(k)-\eta(k)\cdot y(k)\cdot\left(\mathrm{sign}[a^t y(k)-b]-1\right)/2$$
4. while 存在 $\boldsymbol{a}^t\boldsymbol{y}(\boldsymbol{k})<\boldsymbol{b}$, k=1,2,…n
5. Return $\boldsymbol{a}$
6. 结束

- 算法的特点

感知器算法的松弛过程

以前的准则函数 ***J***(.) 定义为:

$$J_p(\mathrm{a}) = \sum_{\mathrm{y}\in\Upsilon} \mathrm{a}^t \mathrm{y}$$

上面准则函数的问题为:

1. 该函数过于平滑。 解矢量很可能正好落在边界上.
2. 函数的解在很大程度上受到模值大的样本的支配

感知器规则可以被推广为 “松弛过程”.

准则函数4——松弛算法

感知器松弛算法是对下面函数进行求解:

$$J_r(\mathrm{a}) = \frac{1}{2}\sum_{y\in\Upsilon}\frac{\left(\mathrm{a}^t\mathrm{y}-b\right)^2}{\|\mathrm{y}\|^2}$$

这里，是对所有满足下面关系的样本集合进行求和:

$$\mathrm{a}^t\mathrm{y} \le b$$

上面的准则函数总是非负。当且仅当在所有 $\mathbf{a}^t\,\mathbf{y} > \boldsymbol{b}$ 时，函数值为0

松弛算法：梯度、迭代公式

➢ 准则函数的梯度

$$\nabla J_r = \sum_{y \in \Upsilon} \frac{a^t y - b}{\|y\|^2} y$$

➢ 批处理裕量迭代公式

$$a(k+1) = a(k) + \eta(k) \sum_{y \in \Upsilon} \frac{a^t y - b}{\|y\|^2} y$$

➢ 单样本裕量迭代公式

$$a(k+1) = a(k) + \eta(k) \frac{a^t y(k) - b}{\|y(k)\|^2} y(k)$$

松弛算法的实质

- 使用规范化样本进行学习，即学习中主要利用**样本在特征空间的方向信息**进行学习

$$\nabla J_r = \sum_{y \in \Upsilon} \frac{a^t y - b}{\|y\|^2} y = \sum_{y \in \Upsilon} \left(\frac{a^t y - b}{\|y\|} \cdot \frac{y}{\|y\|} \right)$$

$$a(k+1) = a(k) + \eta(k) \sum_{y \in \Upsilon} \left(\frac{a^t y - b}{\|y\|} \cdot \frac{y}{\|y\|} \right)$$

$$a(k+1) = a(k) + \eta(k) \frac{a^t y(k) - b}{\|y(k)\|} \cdot \frac{y(k)}{\|y(k)\|}$$

感知器算法：不可分的情况的处理

感知器准则对线性可分情况是有效的，可以寻找到一个无错误的解 。

在针对线性不可分问题时，一个解决的办法是使用可变学习率

$\eta(k)$ 在 $\boldsymbol{k}$ 趋近于无穷时，可以令其逐渐趋近于0.

具体做法为:

- 在性能改善时，逐渐减小 $\eta(k)$

例如可以定义一个函数： $\eta(k)=\eta(1)/k$