固定增量单样本感知器

Algorithm 4 (Fixed-increment single-sample Perceptron)

```
begin initialize  a, k = 0
        do  k ← (k + 1) mod n
            if  y_k is misclassified by a  then  a ← a − y_k
        until all patterns properly classified
    return a
end
```

感知准则函数

- **梯度下降算法**求解：对（迭代）向量沿某函数的负梯度方向修正，可较快到达该函数极小值。

$$\nabla J_p(\mathbf{a}) = \frac{\delta J_p(\mathbf{a})}{\delta \mathbf{a}} = \sum_{\mathbf{y}\in Y^k} (-\mathbf{y});$$

$$\mathbf{a}(k+1) = \mathbf{a}(k) - r_k \nabla J_p(\mathbf{a})$$
$$= \mathbf{a}(k) + r_k \sum_{\mathbf{y}\in Y^k} \mathbf{y}.$$

线性支持向量机 (线性SVM)

- 目标函数

$$\begin{array}{ll} \text{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} & 0 \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, \ell \end{array}$$

线性支持向量机 (线性SVM)

- 梯度上升算法求解

$W(\boldsymbol{\alpha})$梯度的第 i 个成分是：

$$\frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_i} = 1 - y_i \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

因此可以一个固定的学习率 η 使用简单的迭代更新规则：

$$\alpha_i \leftarrow \alpha_i + \eta \frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_i}$$

最大化 $W(\boldsymbol{\alpha})$，并且在一个正象限内同时更新所有 α_i 的值。

线性支持向量机 (线性SVM)

- 梯度上升算法求解

表 7.1　具有一阶软间隔的简单在线算法

```
给定训练集S和学习率η ∈ (R+)^ℓ
α ← 0
重复
    for i = 1 to ℓ
        α_i ← α_i + η_i (1 − y_i Σ_{j=1}^{ℓ} α_j y_j K(x_i, x_j))
        if α_i < 0 then α_i ← 0
        else
        if α_i > C then α_i ← C
    end for
直到满足停止条件
返回α
```