# 感知器 Perceptron

bxf_hit@163.com

2017 年 3 月 24 日

## 1 基本概念

感知器模型是 Rosenblatt(1962) 提出的一个线性判别模型，它在模式识别算法历史上占有重要的意义；感知器对应一个 2 分类的模型，在这个模型中，输入向量 X 先是使用一个固定的非线性变化得到另一个向量 $\phi$(x), 这个向量然后被用于构造一个一般的线性模型，形式为

$$y(x) = f(w^T \phi(x)) \tag{1}$$

其中，这里的 $\phi$(x)为n+1 维，$w^T\phi(x)$ 可以写成 $a^T x + b$型，非线性激活函数 f(t) 是一个阶梯函数，形式为

$$f(t) = \begin{cases} +1, t \geq 0 \\ -1, t < 0 \end{cases} \tag{2}$$

在之前对于二分类问题的讨论中, 我们对于目标变量的表示方法为 $f(t)\in 0, 1$, 这对于概率模型来说是很合适的。然而, 对于感知器来说, 更方便的做法是使用 $t = +1$ 表示 C1 , 使用 $t = -1$ 表示 C2 , 这与激活函数的选择相匹配。

感知器的目的就是找到一个解向量 w，可以正确划分所有正例和反例。也就是在空间中寻找一个超平面 $a^T x + b = 0$, 使之可以划分空间中的两类点。

## 2 感知器准则函数

为了求解解向量 w, 我们需要定义一个误差函数，误差函数的 一个自然的选择是误分类的模式的总数。但是, 这样做会使得学习算法不会很简单,

因为这样做会使误差函数变为 w 的分段常函数, 从而当 w 的变化使得决策边界移过某个数据点时, 这个 函数会不连续变化。这样做还使得使用误差函数改变 w 的方法无法使用, 因为在几乎所有的地方梯度都等于零。

因此我们考虑一个另外的误差函数, 被称为感知器准则 (perceptron criterion)。为了推导这个函数, 我们注意到我们正在做的是寻找一个权向量 w 使得对于类别 C1 中的模式 $x_n$ 都有 $w^T\varphi(x_n) > 0$, 而对于类别 C2 中的模式 $x_n$ 都有 $w^T\varphi(x_n) < 0$。使用 $t \in -1, +1$ 这种目标变量的表示方法, 我们要做的就是使得所有的模式都满足 $w^T\varphi(x_n)t_n > 0$。对于正确分类的模式, 感知器准则赋予零误差, 而对于误分类的模式 $x_n$, 它试着最小化 $-w^T\varphi(x_n)t_n$ 。因此, 感知器准则为

$$EP(w) = -\sum_{n\in M} w^T\varphi(x_n)t_n \tag{3}$$

其中, M 表示所有误分类的元素 x 所组成的集合。当且仅当 $EP(w^*) = minEP(w) = 0$ 时, $w^*$ 是解向量。这就是 Rosenblatt 提出的感知器（Perceptron）准则函数

对于感知器的准则函数，其实还有另一种理解：采用误分类点到超平面的距离（可以自己推算一下，这里采用的是几何间距，就是点到直线的距离）:

$$EP(w) = -\frac{1}{\parallel a \parallel}\sum_{n\in M}(a^T\phi(x_n) + b)t_n \tag{4}$$

之前有提到，$w^T\phi(x)$ 可以写成 $a^Tx + b$型，再加上 $\frac{1}{\|a\|}$ 可以归一化, 就可以得到与公式（3）相同的感知器准则函数了。

我们现在对这个误差函数（3）使用随机梯度下降算法，可以得到:

$$w(t+1) = w(t) - \eta\nabla EP(w) \tag{5}$$

其中，t 为迭代次数，$\eta$ 为调整的步长。即下一次迭代的权向量是把当前时刻的权向量向目标函数的负梯度方向调整一个修正量, 修正量可以表示为:

$$\nabla EP(w) = \frac{\partial EP(w)}{\partial w} = -\sum_{n\in M}\varphi(x_n)t_n \tag{6}$$

由（4）（5）可得:

$$w(t+1) = w(t) + \eta\sum_{n\in M}\varphi(x_n)t_n \tag{7}$$

我们反复对于训练模式进行循环处理, 对于每个模式 $x_i$ 我们计算感知器函数, 如果模式正确分类, 那么权向量保持不变, 而如果模式被错误分类, 那么对于类别 C1 , 我们把向量 $\varphi(x_i)$ 加到当前对于权向量 w 的估计值上, 而对于类 别 C2, 我们从 w 中减掉向量 $\varphi(x_i)$(这里做了假设 $\eta = 1$)。

感知器算法的伪代码可以简单描述如下:

**Algorithm 1 (Basic gradient descent)**

```
1 begin initialize  a, criterion θ, η(·), k = 0
2   do k ← k + 1
3       a ← a − η(k)∇J(a)
4   until η(k)∇J(a) < θ
5 return a
6 end
```

通常情况，一次将所有错误样本进行修正不是效率最高的做法，更常用是每次只修正一个样本或一批样本的固定增量法:

**Algorithm 4 (Fixed-increment single-sample Perceptron)**

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

## 3 感知器收敛定理

感知器收敛定理 (perceptron convergence theorem) 表明, 如果存在一个精确的解 (即, 如果训练数据线性可分), 那么感知器算法可以保证在有限步骤内找到一个精确解。这个定理的证明可以参考 Rosenblatt(1962)、Block(1962)、Nilsson(1965)、Minsky and Papert( 1969)、Hertz et al.( 1991) 以及 Bishop(1995a)。

# 4 关于感知器

- 需要注意的是, 感知器达到收敛状态所需的步骤数量可能非常大, 并且在实际应用中, 在达到收敛状态之前, 我们不能够区分不可分问题与缓慢收敛问题。
- 即使数据集是线性可分的, 也可能有多个解, 并且最终哪个解会被找到依赖于参数的初始化以及数据点出现的顺序。此外, 对于线性不可分的数据集, 感知器算法永远不会收敛。
- 除了学习算法的这些困难之处以外, 感知器算法无法提供概率形式的输出, 也无法直接推广 到 $K > 2$ 个类别的情形。然而, 最重要的局限性是它基于固定基函数的线性组合

关于感知器算法更多的局限性, 可以参考 Minsky and Papert( 1969) 和 Bishop(1995a)

# 5 参考文章

1. Pattern Recognition And Machine Learning,chapter 4
2. http://blog.csdn.net/xiaowei_cqu/article/details/9004101
3. 统计学习方法-李航