统计学习方法——感知机

create：2021/10/28 广州

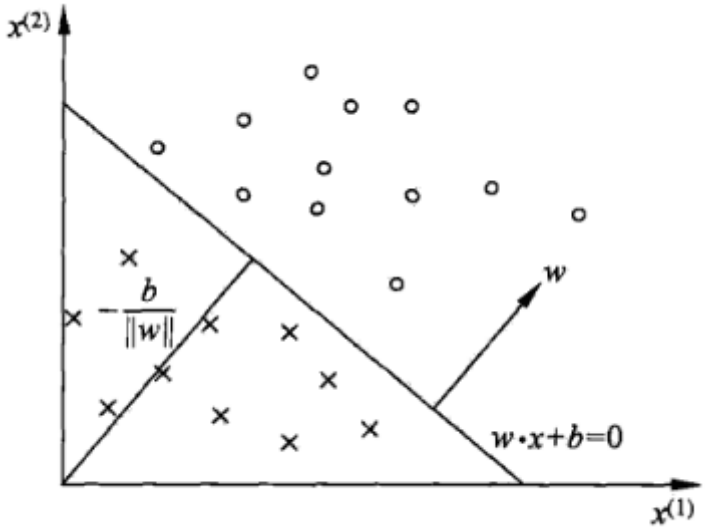

图 2.1　感知机模型

什么是感知机

感知机是一个二分类线性判别模型，假设输入$x \in \mathbb{R}^n$，输出$y \in -1, +1$，感知机为如下函数：

$$f(x) = sign(w^T x + b),$$
$$sign(z) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

其中，w叫做权重，是分类超平面的法向量；b叫做偏置，是超平面的截距。

设数据集线性可分，感知机的损失函数为所有误分类点到分类超平面的函数间隔，即：

$$L(w, b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b)$$

几何间隔和函数间隔的区别我没有提到，在文中下面给出的b站视频中，有详细的解释

- 为什么感知机不使用几何间隔

由于感知机的前提是原数据集线性可分，这意味着必须存在一个正确的超平面。那么，不管几何距离还是函数距离，损失函数最后都要等于0，因此感知机并不关心点到超平面之间的间隔，关心的是误分类的点的个数。采用几何间隔并不会带来什么好处，反而会使学习过程复杂化。

感知机的原始形式

给定一个训练的数据集$T=(x_1,y_1),(x_2,y_2),..,(x_N,y_N)$，其中$x_i\in\mathbb{R}^n$，$Y_i\in-1,1$，$i=1,2,3....N$。感知机$sing(w\cdot x+b)$的损失函数为:

$$\min_{w,b}L(w,b)=-\sum_{x_i\in M}y_i(w\cdot x_i+b)$$

其中M为误分类点的集合，有了损失函数分别对w和b求偏导就可以得到梯度了，损失函数$L(w,b)$的梯度为:

$$\nabla_wL(w,b)=-\sum_{x_i\in M}y_ix_i$$
$$\nabla_bL(w,b)=-\sum_{x_i\in M}y_i$$

接着就可以随机选取一个失误分类(x_i,y_i)，对w,b进行更新:

$$w\leftarrow w+\eta y_ix_i$$
$$b\leftarrow b+\eta y_i$$

其中$\eta(0<\eta\leq1)$是步长，又称为学习率，在本文中如果不做详细说明一般取1

可以得到如下的算法了:

输入：训练数据$T=(x_1,y_1),(x_2,y_2),..,(x_N,y_N)$，其中$x_i\in\mathbb{R}^n$，$Y_i\in-1,1$；学习率$\eta\in(0,1]$

输出：w,b；感知机模型$f(x)=sign(w^Tx+b)$

- 随机任选一个超平面w_0,b_0，一般都初始化为0
- 在训练集中选取数据(x_i,y_i)
- 如果$y_i(w^Tx_i+b)\leq0$，则更新w和b：
$$w=w+\eta y_ix_i$$
$$b=b+\eta y_i$$
- 转至第二步，直到训练集中没有误分点

这个算法的流程简单来说就是：刚开始初始化超平面为0，然后选取第一个数据，如果结果小于等于0的话，那么就更新w和b，然后用更新后的新模型再次遍历所有数据，碰到错误情况就重复：更新w和b，遍历所有数据这个操作。直到找到合适的w和b满足所有的数据。

具体的例子可看统计学习方法第29页

感知机的对偶性质

~~假设样本点(x_i,y_i)在更新的过程中被使用了n_i次，因此，从原始形式的学习过程可以得到，最后学习到的w和b可以分别表示为：~~

$$w=\sum_{i=1}^{N}n_i\eta y_ix_i$$
$$b=\sum_{i=1}^{N}n_i\eta y_i$$

n_i**的含义：**

如果n_i的值越大，那么意味着这个样本点经常为误分。说明这个点就是离超平面很近的点。超平面稍微移动一点点，这个点就会被归结到别的类别中。

那么此时感知机的公式就可以写为：

$$f(x) = sign(w \cdot x + b) = sign(\sum_{j=1}^{N} n_j \eta y_j x_j \cdot x_i + \sum_{i=j}^{N} n_j \eta y_j)$$

此时，学习的目标就不再是w**和**b**，而是**$n_i, i = 1, 2, 3 \ldots, N$

当然为了和书本同一，我们可以定义$\alpha_i = n_i \eta$，也可以上式中的第三项换为b，都不影响。

替换过后，w和b的式子为：

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$b = \sum_{i=1}^{N} \alpha_i y_i$$

具体算法为：

输入：训练数据集$T = (x_1, y_1), (x_2, y_2), .., (x_N, y_N)$，其中$x_i \in \mathbb{R}^n$，$Y_i \in -1, 1$；学习率$\eta \in (0, 1]$

输出：α,b；感知机模型更新为：

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i x_i \cdot x + b)$$

其中$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$

- 令$\alpha = \mathbf{0}, b = 0$
- 在训练集中选取数据(x_i, y_i)
- 如果$y_i(\sum_{j=1}^{N} \alpha_j y_j x_j \cdot x_i + b) \leq 0$，则更新$\alpha$和$b$.

$$\alpha_i = \alpha_i + \eta$$
$$b = b + \eta y_i$$

- 转至第三步，直到训练集中没有误分点

再次强调，α_i**是第i个数据在更新过程中被使用到的次数（也就是第i个数据误分的次数）和**η**的乘积，而本文一直将**η**设置为1**

这个算法流程简单来说就是：先初始化$\alpha = \mathbf{0}, b = 0$，然后将第1个数据带入，如果误分，然后更新参数$\alpha_1 = \alpha_1 + \eta$，$b = b + \eta y_1$，接着再次遍历数据，如果碰到误分就执行$\alpha_i = \alpha_i + \eta$，$b = b + \eta y_i$操作，直到没有误分。

具体的例子可看统计学习方法第34页

对偶性这里参考了知乎上的回答（**前面写的都是狗屎，直接看这个回答**）

关于感知机学习算法的对偶形式

我们假设样本点(x_i, y_i)在更新过程中被使用了n_i次。因此，从原始形式的学习过程可以得到，最后学习到的w和b可以分别表示为：

$$w = \sum_{i=1}^{N} n_i \eta \, y_i x_i \tag{1}$$

$$b = \sum_{i=1}^{N} n_i \eta \, y_i \tag{2}$$

考虑n_i的含义：如果n_i的值越大，那么意味着这个样本点经常被误分。什么样的样本点容易被误分？很明显就是离超平面很近的点。超平面稍微一点点移动，这个点就从正变为负，或者从负变为正。如果学过SVM就会发现，这种点很可能就是支持向量。

代入式(1)和式(2)到原始形式的感知机模型中，可得：

$$f(x) = sign(w \cdot x + b) = sign(\sum_{j=1}^{N} n_j \eta \, y_j x_j \cdot x + \sum_{j=1}^{N} n_j \eta \, y_j) \tag{3}$$

此时，学习的目标就不再是w和b，而是n_i，$i = 1, 2, \ldots, N$。

相应地，训练过程变为：

1. 初始时$\forall n_i = 0$。
2. 在训练集中选取数据(x_i, y_i)
3. 如果 $y_i(\sum_{j=1}^{N} n_j \eta \, y_j x_j \cdot x_i + \sum_{j=1}^{N} n_j \eta \, y_j) \leq 0$，更新：$n_i \leftarrow n_i + 1$
4. 转至2直至没有误分类数据。

可以看出，其实对偶形式和原始形式并没有本质区别，那么对偶形式的意义在哪里呢？从式(3)中可以看出，样本点的特征向量以内积的形式存在于感知机对偶形式的训练算法中，因此，如果事先计算好所有的内积，也就是Gram矩阵，就可以大大地加快计算速度。

这个回答和李航书中讲的角度不一样，但是解释的非常好！而且训练过程也有不同的地方，因为没有用α_i，因为这个回答的所有操作都是对n_i进行的，在更新时，是将$n_i + 1$，这个公式和书中的$\alpha_i = \alpha_i + \eta$本质是一样的。

因为$\alpha_i = n_i \eta$，所以$\alpha_i =$ （$n_i + 1$）$\eta = n_i \eta + \eta = \alpha_i + \eta$

注意

本文写的比较笼统，具体的可以看

统计学习方法第二章

b站视频