感知机算法

杨航锋

感知机算法是应用于线性可分样本数据的二分类算法，本文在推导感知机算法过程中遵循的顺序是，首先提出假设函数、其次构造出损失函数、然后求解损失函数得出假设函数中的参数值、最后给出感知机算法的对偶形式。弄清楚感知机算法是学好支持向量机和神经网络的基础 。更多文章见GitHub地址

感知机的假设函数

假设输入空间是 $X \in \mathbb{R}^n$,输出空间是 $Y \in \{+1, -1\}$ 。输入 $x \in X$ 表示实例的特征向量，对应于输入空间的点；输出 $y \in Y$ 表示实例的类别。由输入空间到输出空间的如下函数：

$$\begin{aligned} \psi(x) &= rsign(\omega_1 x_1 + \omega_2 x_2 + \cdots + w_n x_n + b) \\ &= rsign(\omega^T x + b) \end{aligned}$$

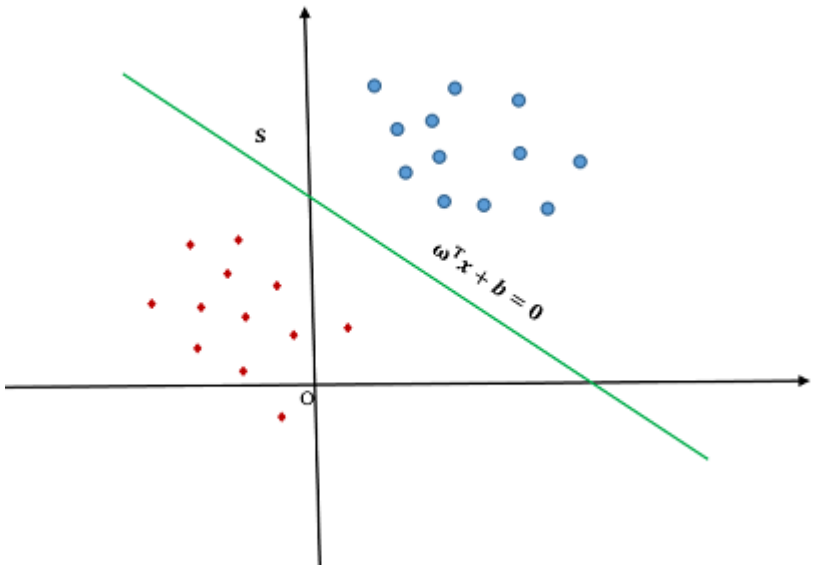
称为感知机，其中 $\omega \in \mathbb{R}^n$，$b \in \mathbb{R}$ 为感知机算法的参数值，$rsign$ 为反对称的符号函数，即：

$$rsign(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

因此可知感知机的假设函数为 $\psi(x)$,通过输入特征向量x即可判断其所属类别。

感知机的损失函数

感知器的训练过程其实就是求解 ω 和 b 的过程。正确的 ω 和 b 所构成的超平面 $\omega^T x + b = 0$ 恰好将两类数据点分割在这个平面的两侧。为了找出这样的超平面，即确定感知机模型参数 ω、b ，需定义损失函数并将损失函数最小化。不妨假设含有m个线性可分的样本数据($x^{(1)}$,$y^{(1)}$)、($x^{(2)}$,$y^{(2)}$)、$\cdots$、($x^{(m)}$,$y^{(m)}$)，$y^{(i)} \in Y$ 。



损失函数使用误分类点到超平面 S 的总距离，因此输入空间 X 中任何一点 $x^{(i)}$ 到超平面 S 的距离为(类比点到直线的距离)：

$$\frac{1}{\|\omega\|_2}|\omega^T x^{(i)} + b|$$

不妨假设误分类的点集为M，则任意误分类点($x^{(i)}$, $y^{(i)}$)$\in M$，当$w^T x^{(i)} + b > 0$时，有$y^{(i)} = -1$；当$w^T x^{(i)} + b < 0$时，有$y^{(i)} = +1$。所以对任意误分类点($x^{(i)}$,$y^{(i)}$)有如下不等式成立:

$$y^{(i)}(w^T x^{(i)} + b) < 0$$

则误分类点($x^{(i)}$,$y^{(i)}$)到超平面 S 的距离为(去掉绝对值符号)：

$$-\frac{y^{(i)}(\omega^T x^{(i)} + b)}{\|\omega\|_2}$$

故误分类点集M中所有点到超平面 S 的距离之和为

$$-\frac{1}{\|\omega\|_2}\sum_{(x^{(i)},y^{(i)})\in M} y^{(i)}(w^T x^{(i)} + b)$$

所以求解最小化距离之和可知

$$\begin{aligned}\arg\min_{\omega,b} L(\omega,b) &\Leftrightarrow \arg\min_{\omega,b} -\frac{1}{\|\omega\|_2}\sum_{(x^{(i)},y^{(i)})\in M} y^{(i)}(w^T x^{(i)} + b) \\ &\Leftrightarrow \arg\min_{\omega,b} -\sum_{(x^{(i)},y^{(i)})\in M} y^{(i)}(w^T x^{(i)} + b)\end{aligned}$$

综上可知，感知机算法的损失函数为 $L(\omega,b)=-\sum\limits_{(x^{(i)},y^{(i)})\in M}y^{(i)}(w^Tx^{(i)}+b)$ 。

求解感知机的损失函数

最小化损失函数采用梯度下降算法，在感知机中梯度下降算法的迭代格式为

$$\omega:=\omega-\alpha\nabla_\omega L(\omega,b)$$
$$b:=b-\alpha\nabla_b L(\omega,b)$$

其中 $\alpha\in(0,1]$ 代表感知机算法的学习率，为了实现该算法首先要求出损失函数 $L(\omega,b)$ 分别对参数ω, b 的梯度:

$$\nabla_\omega L(\omega,b)=-\sum_{(x^{(i)},y^{(i)})\in M}x^{(i)}y^{(i)}$$
$$\nabla_b L(\omega,b)=-\sum_{(x^{(i)},y^{(i)})\in M}y^{(i)}$$

因此随机选取 M 中的一点 $(x^{(i)},y^{(i)})$ 更新 ω、b 直到感知机的损失函数 $L(\omega,b)$ 下降到 0 为止

$$\omega:=\omega+\alpha x^{(i)}y^{(i)}$$
$$b:=b+\alpha y^{(i)}$$

如果样本数据是线性可分的，那么数学上可以证明这种算法一定是收敛的，即经过有限步的迭代后可以求出能够正确分类样本数据的参数 ω,b 。

感知机算法的对偶形式

感知机算法对偶形式的目的是降低运算量，但是并不是在任何情况下都能降低运算量，而是在特征空间的维度很高时才能发挥作用。 每次感知机梯度下降算法的迭代都是选择一个误分类样本数据来更新 ω、b参数。最终经过若干次的迭代后得到最终的结果。对于从来都没有误分类过的样本，它被选择参与 ω、b 迭代的次数是 0 ，假设样本点 $(x^{(i)},y^{(i)})\in M$ 在迭代更新 ω、b 时被使用了 k_i 次，因此在原始感知机算法中，算法最后收敛时的 ω、b 为:

$$\omega=\alpha\sum_{i=1}^{K}k_i x^{(i)}y^{(i)}$$
$$b=\alpha\sum_{i=1}^{K}k_i y^{(i)}$$

把上述 ω、b 回代到原始感知机算法的假设函数中可得

$$\psi(x)=rsign(\alpha\sum_{j=1}^{K}k_jx^{(j)}y^{(j)}\cdot x+\alpha\sum_{j=1}^{K}k_jy^{(j)})$$

此时的参数不再是 ω、b 而是 k_i ,$i=1,2,\cdots,K$。因此原始问题由求解参数 ω、b 就转化为怎么求解参数 k_i

只要 k_i 求出来了，那么原始问题的 ω、b 就求出来了。综上可以给出对偶形式下的参数迭代格式

$$\begin{cases}k_i=k_i+1, & \text{if } y^{(i)}\left(\alpha\sum\limits_{j=1}^{K}k_jy^{(j)}(x^{(i)},x^{(j)})+\alpha\sum\limits_{j=1}^{K}k_jy^{(j)}\right)\le 0\\ i=i+1, & \text{otherwise}\end{cases}$$

更一般的把判断误分类点的不等式向量化

$$\begin{aligned}&y^{(i)}\left(\alpha\sum_{j=1}^{K}k_jy^{(j)}(x^{(i)},x^{(j)})+\alpha\sum_{j=1}^{K}k_jy^{(j)}\right)\\&=\alpha y^{(i)}k^Ty(x^{(i)},x^{(j)})+\alpha y^{(i)}k^Ty\end{aligned}$$

其中 $(x^{(i)},x^{(j)})$ 表示做内积运算，迭代更新至无误分类样本数据即可。在迭代更新时可以发现当**某样本数据点在算法更新时使用次数越多，意味着它距离分离超平面越近，也就越难以正确分类，换言之，这类样本数据点对感知机算法学习的效果影响最大 。**

总结

假设特征空间是 $\mathbb{R}^n$，样本数据为N， $N\ll n$。当考虑原始形式的感知机算法时，每轮迭代中至少要判断某个输入样本数据点是不是误分类点，既对于 $(x^{(i)},y^{(i)})$,是否有 $y^{(i)}(\omega^Tx^{(i)}+b)\le 0$ 。这里的运算量主要集中在求输入特征 $x^{(i)}$ 和权值向量 ω 的内积上，时间复杂度为 $\Theta(n)$ ，当特征空间维度非常高时，原始感知机算法效率很低；而在对偶形式的感知机算法中，对于输入数据点 $(x^{(i)},y^{(i)})$ 是否为误分类点的条件转化为 $y^{(i)}\left(\alpha\sum\limits_{j=1}^{K}k_jy^{(j)}(x^{(i)},x^{(j)})+\alpha\sum\limits_{j=1}^{K}k_jy^{(j)}\right)\le 0$ 。注意到这里所有输入样本数据都仅仅以内积的形式出现，所以我们可以预先计算出输入样本数据两两之间的内积，得到所谓的 $Gram$ 矩阵 $G=[(x^{(i)},x^{(j)})]_{N\times N}$。这样一来每次做误分类点检测时候只需要在 $Gram$ 矩阵里查找就能获取内积 $(x^{(i)},x^{(j)})$ ，所以这个误分类点检测的时间复杂度是 $\Theta(1)$ 。也就是说，对偶形式的感知机算法，把每轮迭代的时间复杂度从特征空间维度 n 转移到了样本数据数量 N 上，这是一件很奇妙的事情。