感知机推导

空间 $\mathbb{R}^n$ 中有一个超平面 $w\cdot x+b=0$；则该空间中任一点 x_0 到超平面的距离为 $\frac{1}{\|w\|}|w\cdot x_0+b|$ ……①

∴ 所有误分类点到超平面的距离为 $-\frac{1}{\|w\|}\sum_{x_i\in M} y_i(w\cdot x_i+b)$

其中 M 表示误分类点的集合.

$\forall y_i\in\{-1,1\}$. 利用误分类点的 y 与 $wx+b$ 异号的条件去掉①内的 abs 符号

如果忽略 $\|w\|$ 这个正数. 则感知机 $sign(wx+b)$ 的 Loss 定义为:

$$L(w,b)\triangleq -\sum_{x_i\in M} y_i(wx_i+b)\xrightarrow[\text{定义}\theta=(b,w)]{\text{增广}x} -\sum_{x_i\in M}(\theta\cdot x_i)\cdot y_i$$

易知 $\nabla_\theta L=-\sum_{x_i\in M} y_i\cdot x_i$ ……② 即感知机的原始形式

算法的收敛性见小蓝书 P31 分析. 略.

观察②发现本质上就是误分类点的 x 和 y 乘积的线性组合.

假每个样本在②式的迭代中参与了 α_i 轮. 则 $\theta=\sum_{i=1}^{N}\alpha_i y_i x_i$

模型可以改写为 $f(x)=sign(\theta x)=sign\left(\sum_{i=1}^{N}\alpha_i y_i x_i\cdot x\right)$ ……③

式③即感知机的对偶形式. 直接求 $\frac{\partial L}{\partial \alpha_i}$ 的结果不直观, 不好优化

只好回归 α_i 的定义, 即样本 i 参与了 α_i 次迭代. 每当发现第 i 个样本被误分类就 α_i += 1 即可.

注: 这里我求出来的 $\frac{\partial L}{\partial \alpha_j}=\frac{\partial -\sum_{x_i\in M}\sum_{j=1}^{N}\alpha_j y_j\vec{x}_j\cdot\vec{x}_i\cdot y_i}{\partial\ \alpha_j}$

$=-\sum_{x_i\in M} y_i y_j\cdot\vec{x}_i\cdot\vec{x}_j$

然而并不等于 1.

求大牛解释.

根据上页推导的原始/对偶形式迭代规则，代码如下：

```
class Perceptron:
    # __init(self, dual=True). 初始化θ, α, dual, history. 省略
    def fit(self, x, y, η=1.0, n_epoch=1000):
        x = append_bias(atleast_2d(x))
        m, n = x.shape
        self.θ = zeros(n)
        if self.dual:
            G = x @ x.T  # Gram矩阵
            self.α = zeros(m)  注意形状
        for _ in range(n_epoch):
            if self.dual:
                h = sign(self.α * y @ G)   G中每列对应一个样本
                grad = -(h != y).astype(int)
                self.α -= η * grad
            else:
                h = self.predict(x)
                grad = -(h != y).astype(int) * y @ x
                self.θ -= η * grad
            if (grad == 0).all():  # Early stop 省略.
        if self.dual: self.θ = self.α * y @ x  将α转成θ，方便预测
    def predict(self, x):
        return sign(atleast_2d(x) @ self.θ)
```

以上です。

koyo922@qq.com

2018.4.20