感知机

> - **感知机**是二分类的*线性*分类模型，输入为实例的特征向量，输出为实例的类别，取+1和-1二值；
> - 感知机对应于输入空间中将实例划分为正负两类的分离超平面，属于判别模型；

2.1 感知机模型

- **感知机**（Perceptron）：由输入空间到输出空间定义了如下函数

$$f(x) = \operatorname{sign}(w \cdot x + b) \tag{1}$$

其中$w \in \mathbb{R}^n$和b为模型参数，叫做权值（weight）或者权值向量（weight vector），和偏置（bias），sign是符号函数，表示为

$$\operatorname{sign}(x) = \begin{cases} +1, x \geq 0 \\ -1, x < 0 \end{cases}$$

- **几何解释**：线性方程$w \cdot x + b = 0$对应于特征空间中一个超平面S，其中w是超平面的法向量，b是超平面的截距；这个超平面将特征空间划分为两个部分，位

于不同区域的点被分为正负类；因此S也叫分离超平面（separating hyperplane）；

2.2 学习策略

2.2.1 数据集的线性可分性

- **数据集的线性可分性**：给定数据集T，如果存在某个超平面$S : w \cdot x + b$，能够将数据集的正实例点和负实例点完全正确划分到超平面两侧，则称T为线性可分数据集（linearly separable data set），否则线性不可分；

2.2.2 学习策略

- **损失函数直观选择**：误分类*点的总数*，但是该损失函数不是参数的连续可导函数，难以优化；
- **距离损失函数**：误分类*点到超平面的距离*，其中某个误分类点到超平面的距离

$$-\frac{1}{\|w\|_2} y_i (w \cdot x_i + b) \tag{2}$$

- **感知机采用的损失函数**：误分类点到超平面的总距离（省去归一化）

$$L(w, b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b) \tag{3}$$

其中M表示误分类点的集合，这也就是感知机的经验风险函数；

- 损失函数$L(w, b)$是非负的，如果没有误分类点，值为0；

2.3 学习算法

> 感知机的学习问题转化为损失函数的最优化问题，方法是随机梯度下降法。

2.3.1 原始形式

- **损失函数极小化问题**：

$$\min_{w,b} L(w, b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b) \tag{4}$$

- 算法流程，采用随机梯度下降法（stochastic gradient descent）：
    - 任意选取一个超平面w_0, b_0；
    - 采用梯度下降法不断极小化目标函数（不是一次性使M所有点梯度下降，而是一次选一个点），损失函数的梯度为：

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

    - 随机选一个误分类点(x_i, y_i)，对参数进行更新：

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

  其中$\eta(0 \leq \eta \leq 1)$是步长，又称为学习率（learning rate）；
    - 重复上述步骤，直到训练集中没有误分类点；
    - 输出模型：$f(x) = \text{sign}(w \cdot x + b)$；
- 该学习算法由于采用不同的初值或选取不同误分类点，解可以不一样；

2.3.2 算法收敛性

- 当训练数据线性可分时，学习算法原始形式是收敛的；
- 但是感知机学习算法存在多解，既依赖于初值的选择，也依赖于迭代中误分类点的选择顺序；
- 当训练数据集线性不可分时，算法不收敛，迭代结果发生震荡；

2.3.3 对偶形式

- **基本想法**：将w和b表示为实例(x_i, y_i)的线性组合的形式，通过求解其系数求得w和b；假设初值$w_0 = b_0 = 0$，对误分类点(x_i, y_i)通过梯度更新n_i次后的增量分别是$\alpha_i y_i x_i$和$\alpha_i y_i$，其中$\alpha_i = n_i \eta$，因此学习到的参数为

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

- 算法流程：

- 初始化$\alpha=(\alpha_1,\ldots,\alpha_N)^T$和$b$为0；
- 在训练集中选取数据(x_i,y_i)，如果$y_i\left(\sum_{j=1}^N \alpha_j y_j x_j \cdot x_i + b\right) \le 0$，那么：

$$
\begin{aligned}
\alpha_i &\leftarrow \alpha_i + \eta \\
b &\leftarrow b + \eta y_i
\end{aligned}
$$

- 重复上述步骤直到没有误分类点；
- 输出模型：$f(x)=\mathrm{sign}(\sum_{j=1}^N \alpha_j y_j x_j \cdot x + b)$；

- 对偶形式中训练实例以*<u>内积的形式</u>*出现，可以预先计算并以矩阵存储，即Gram矩阵：

$$G=[x_i \cdot x_j]_{N\times N} \tag{5}$$