3.线性分类

3.1任务设定

输入：x1，x2....

输出：离散空间中的y，由k个类别构成。

二分类：y={-1,+1} or y={0,1}

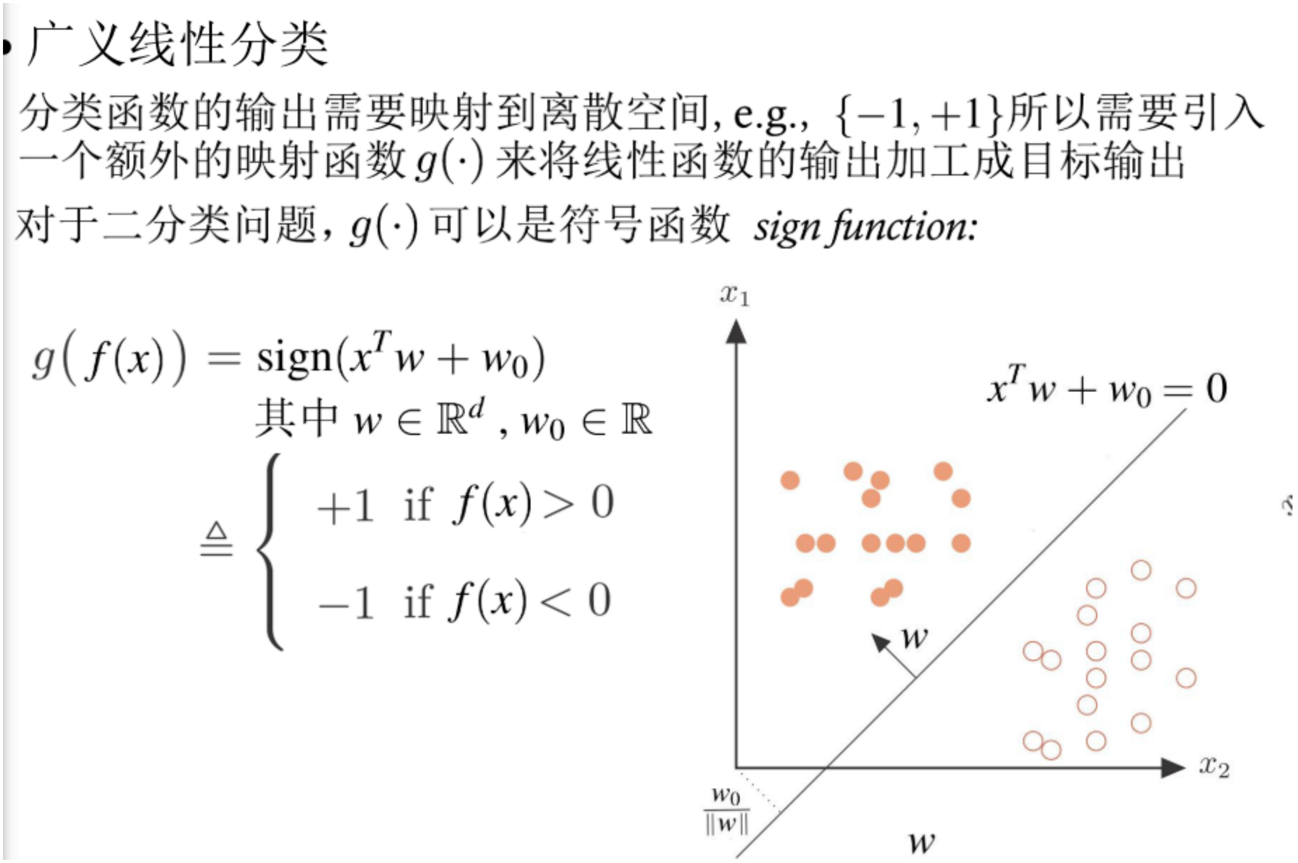
多分类：y={1,2...K}对应K个类别

线性可分

如果满足下述条件，即为线性可分。

$$x^T w + w_0 \begin{cases} > 0 & \text{if x} \in \text{正类} \\ < 0 & \text{if x} \in \text{负类} \end{cases}$$

广义线性分类：



可以使用最小二乘回归进行线性分类，但受到离群点的影响很大。

3.2感知机算法

– 假设：
数据是线性可分的，即存在一个线性分类器能够将所有样本都分对

– 决策函数

$$y = f(x) = \text{sign}(x^T w)$$

– 损失函数

I：indicate function，括号中成立为1，不成立为0

$$\mathcal{L} = -\sum_{i=1}^{n} (y_i \cdot x_i^T w) \mathbb{1}\{y_i \neq \text{sign}(x_i^T w)\}$$

注意到 $y \in \{-1, +1\}$, 所以有：

$$y_i \cdot x_i^T w \ \text{满足} \begin{cases} > 0 \ \text{if} \ y_i = \text{sign}(x_i^T w) \\ < 0 \ \text{if} \ y_i \neq \text{sign}(x_i^T w) \end{cases}$$

可见最小化 $\mathcal{L}$ 对应于我们希望能够将样本全部分类正确.

难以解出损失函数导数为0的解，所以采用梯度下降法，用一个合适的步长迭代。

– 算法：

输入: 训练数据 $(x_1, y_1), \ldots, (x_n, y_n)$ 以及步长参数 η

1. 令 $w_{(1)} = \mathbf{0}$
2. **For** $t = 1, 2, \ldots$ **do**
   a) 遍历训练样本 $(x_i, y_i) \in \mathcal{D}$ 查看是否有错分即满足 $y_i \neq \text{sign}({x_i}^T w^{(t)})$ 的样本
   b) 如果错分样本存在，则挑选其中的一个样本 (x_i, y_i) 并进行如下更新

$$w^{(t+1)} = w^{(t)} + \eta y_i x_i,$$

如果不存在错分样本，则返回能够分对所有训练样本的 $w^{(t)}$

限制：

1. 对于一个线性可分的训练集，感知器无法对能够正确分类的（无数多个）决策面进行质量上的区分（只能分开，不能保证分好）
2. 对于线性不可分的训练集，感知器的学习过程无法收敛

逻辑回归

https://blog.csdn.net/weixin_39445556/article/details/83930186讲的很明白

将对数几率同线性函数对应起来

$$\ln \frac{P(y=+1|x)}{P(y=-1|x)} = x^T w + w_0 \quad p(y=-1|x) = 1 - p(y=1|x)$$

$$p(y=1|x) = \frac{exp\{x^T w + w_0\}}{1 + exp\{x^T w + w_0\}} = \sigma(X^T w + w_0)$$

使用对数函数原因（单个数据的损失函数）

逻辑回归的损失函数是 log loss，也就是**对数似然函数**，函数公式如下：

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

公式中的 y=1 表示的是真实值为1时用第一个公式，真实 y=0 用第二个公式计算损失。为什么要加上log函数呢？可以试想一下，当真实样本为1是，但h=0概率，那么log0=∞，这就对模型最大的惩罚力度；当h=1时，那么log1=0，相当于没有惩罚，也就是没有损失，达到最优结果。所以数学家就想出了用log函数来表示损失函数。

最后按照梯度下降法一样，求解极小值点，得到想要的模型效果。

我们可以得到观察变量 $y_1, \ldots, y_n$ 的联合似然为（记 $\sigma_i(w) = \sigma(x_i^T w)$）

$$\begin{aligned} p(y_1, \ldots, y_n | x_1, \ldots, x_n, w) &= \prod_{i=1}^{n} p(y_i | x_i, w) \\ &= \prod_{i=1}^{n} \sigma_i(w)^{\mathbb{1}(y_i=+1)} (1 - \sigma_i(w))^{\mathbb{1}(y_i=-1)} \end{aligned}$$

我们可以推导出：

$$\underbrace{\frac{e^{y_i x_i^T w}}{1 + e^{y_i x_i^T w}}}_{\sigma_i(y_i \cdot w)} = \Bigg(\underbrace{\frac{e^{x_i^T w}}{1 + e^{x_i^T w}}}_{\sigma_i(w)}\Bigg)^{\mathbb{1}(y_i=+1)} \Bigg(\underbrace{1 - \frac{e^{x_i^T w}}{1 + e^{x_i^T w}}}_{1-\sigma_i(w)}\Bigg)^{\mathbb{1}(y_i=-1)}$$

这能够令我们将似然公式写的更漂亮：

$$p(y_1, \ldots, y_n | x_1, \ldots, x_n, w) = \prod_{i=1}^{n} \sigma_i(y_i \cdot w)$$

记最大似然 *maximum likelihood* 下w的解为

$$\begin{aligned} w_{\mathrm{ML}} &= \arg\max_w \sum_{i=1}^{n} \ln \sigma_i(y_i \cdot w) \\ &= \arg\max_w \mathcal{L} \end{aligned}$$

$$\nabla_w \mathcal{L} = \sum_{i=1}^{n} (1 - \sigma_i(y_i \cdot w))\, y_i x_i$$

输入: 训练集$(x_1, y_i), \ldots, (x_n, y_n)$和步长$\eta > 0$

1. 令$w_{(1)} = \mathbf{0}$
2. **For step** $t = 1, 2, \ldots$ **do**

   更新 $w^{(t+1)} = w^{(t)} + \eta \sum_{i=1}^{n} (1 - \sigma_i(y_i \cdot w))\, y_i x_i$