Logistic 回归模型

上一节课我们学习了简单的线性回归模型，这一次课中，我们会学习第二个模型，Logistic 回归模型。

Logistic 回归是一种广义的回归模型，其与多元线性回归有着很多相似之处，模型的形式基本相同，虽然也被称为回归，但是其更多的情况使用在分类问题上，同时又以二分类更为常用。

模型形式

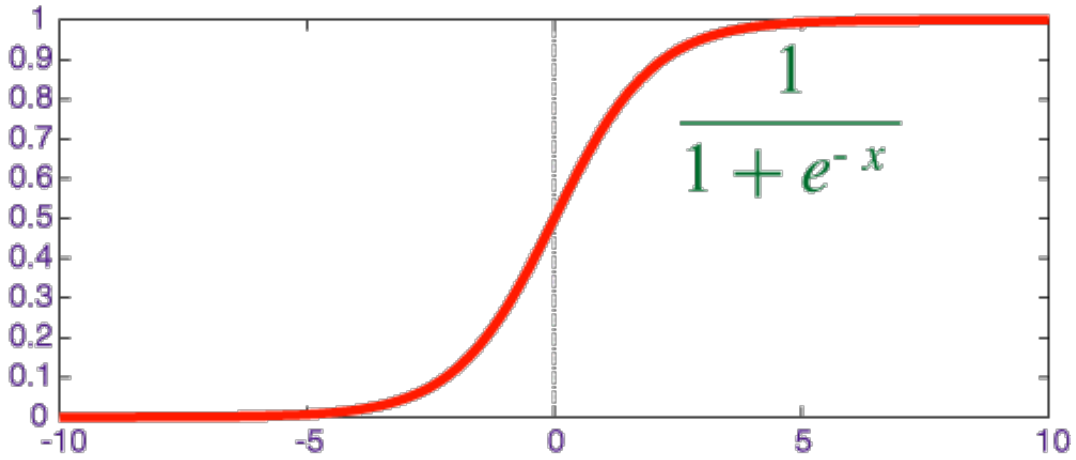
Logistic 回归的模型形式和线性回归一样，都是 y = wx + b，其中 x 可以是一个多维的特征，唯一不同的地方在于 Logistic 回归会对 y 作用一个 logistic 函数，将其变为一种概率的结果。 Logistic 函数作为 Logistic 回归的核心，我们下面讲一讲 Logistic 函数，也被称为 Sigmoid 函数。

Sigmoid 函数

Sigmoid 函数非常简单，其公式如下

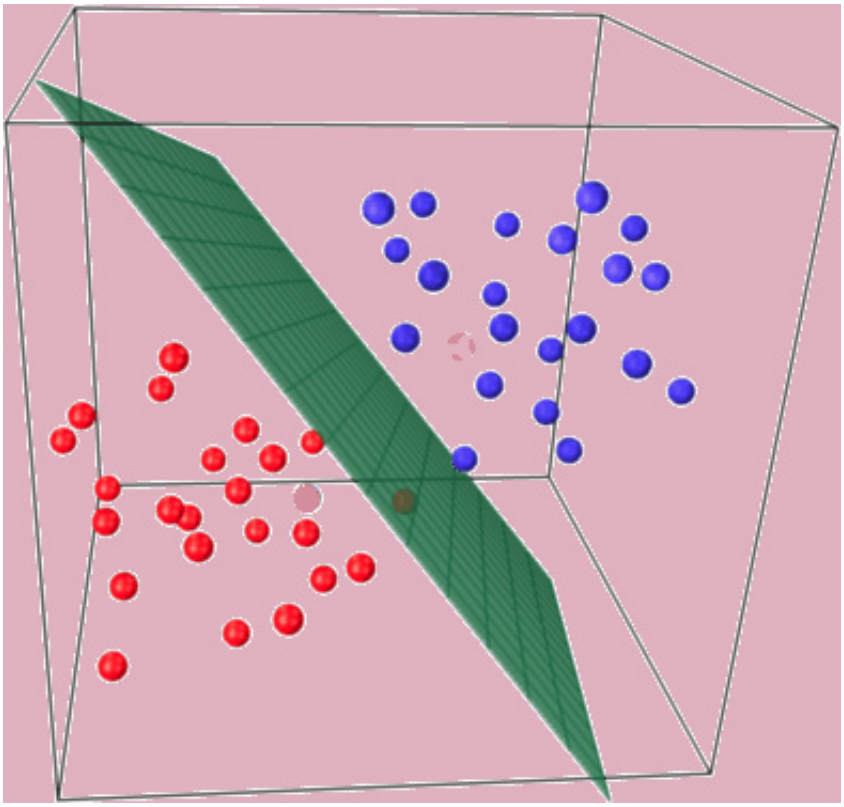
$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Sigmoid 函数的图像如下



可以看到 Sigmoid 函数的范围是在 0 ~ 1 之间，所以任何一个值经过了 Sigmoid 函数的作用，都会变成 0 ~ 1 之间的一个值，这个值可以形象地理解为一个概率，比如对于二分类问题，这个值越小就表示属于第一类，这个值越大就表示属于第二类。

另外一个 Logistic 回归的前提是确保你的数据具有非常良好的线性可分性，也就是说，你的数据集能够在一定的维度上被分为两个部分，比如



可以看到，上面红色的点和蓝色的点能够几乎被一个绿色的平面分割开来

回归问题 vs 分类问题

Logistic 回归处理的是一个分类问题，而上一个模型是回归模型，那么回归问题和分类问题的区别在哪里呢?

从上面的图可以看出，分类问题希望把数据集分到某一类，比如一个 3 分类问题，那么对于任何一个数据点，我们都希望找到其到底属于哪一类，最终的结果只有三种情况，{0, 1, 2}，所以这是一个离散的问题。

而回归问题是一个连续的问题，比如曲线的拟合，我们可以拟合任意的函数结果，这个结果是一个连续的值。

分类问题和回归问题是机器学习和深度学习的第一步，拿到任何一个问题，我们都需要先确定其到底是分类还是回归，然后再进行算法设计

损失函数

前一节对于回归问题，我们有一个 loss 去衡量误差，那么对于分类问题，我们如何去衡量这个误差，并设计 loss 函数呢?

Logistic 回归使用了 Sigmoid 函数将结果变到 0 ~ 1 之间，对于任意输入一个数据，经过 Sigmoid 之后的结果我们记为 $\hat{y}$，表示这个数据点属于第二类的概率，那么其属于第一类的概率就是 $1-\hat{y}$。如果这个数据点属于第二类，我们希望 $\hat{y}$ 越大越好，也就是越靠近 1 越好，如果这个数据属于第一类，那么我们希望 $1-\hat{y}$ 越大越好，也就是 $\hat{y}$ 越小越好，越靠近 0 越好，所以我们可以这样设计我们的 loss 函数

$$loss = -(y * log(\hat{y}) + (1 - y) * log(1 - \hat{y})) \tag{2}$$

其中 y 表示真实的 label，只能取 {0, 1} 这两个值，因为 $\hat{y}$ 表示经过 Logistic 回归预测之后的结果，是一个 0 ~ 1 之间的小数。如果 y 是 0，表示该数据属于第一类，我们希望 $\hat{y}$ 越小越好，上面的 loss 函数变为

$$loss = -(log(1 - \hat{y})) \tag{3}$$

在训练模型的时候我们希望最小化 loss 函数，根据 log 函数的单调性，也就是最小化 $\hat{y}$，与我们的要求是一致的。

而如果 y 是 1，表示该数据属于第二类，我们希望 $\hat{y}$ 越大越好，同时上面的 loss 函数变为

$$loss = -(log(\hat{y})) \tag{4}$$

我们希望最小化 loss 函数也就是最大化 $\hat{y}$，这也与我们的要求一致。

所以通过上面的论述，说明了这么构建 loss 函数是合理的。