Machine Learning - Week 3

赵燕

目录

1 Classification and Representation

1.1 Classification

To attempt classification, one method is to use linear regression and map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0. However, this method doesn't work well because classification is not actually a linear function.

在分类问题中，需要预测的变量y是离散的值，引出要学习的逻辑回归算法（Logistic Regression），这是目前最流行使用的一种学习算法。

分类问题举例：

（1）判断一封电子邮件是否是垃圾邮件；

（2）判断一次金融交易是否是欺诈；

（3）判断肿瘤是 良性还是恶性；

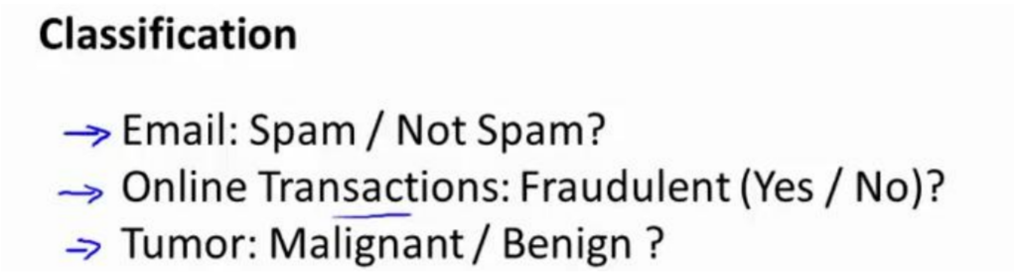

图 1: 分类问题举例

从二元的问题开始讨论：

将因变量（dependent variable）可能属于两个类分别称为负向类（negative class）和正向类（positive class），则因变量$y \in \{0, 1\}$，其中0表示负向类，1表示正向类。

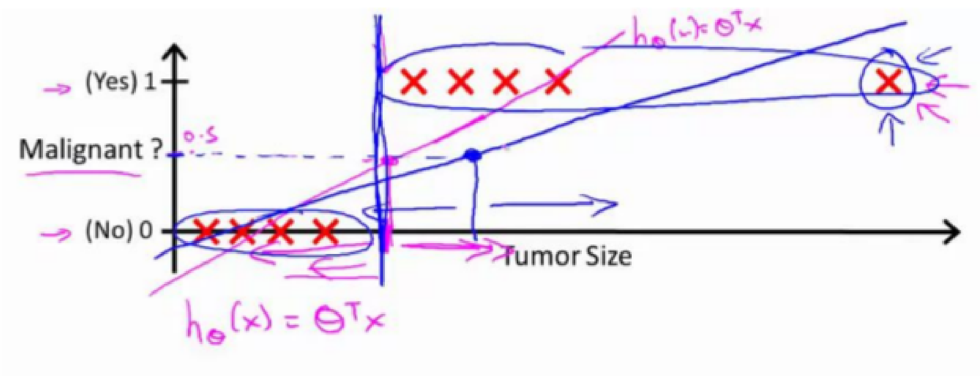

图 2: 图示

The classification problem is just like the regression problem, except that the values we now want to predict take on only a small number of discrete values. For now, we will focus on the binary classification problem in which y can take on only two values, 0 and 1. (Most of what we say here will also generalize to the multiple-class case.) For instance, if we are trying to build a spam classifier for email, then x(i) may be some features of a piece of email, and y may be 1 if it is a piece of spam mail, and 0 otherwise. Hence, y ∈ 0,1. 0 is also called the negative class, and 1 the positive class, and they are sometimes also denoted by the symbols "-" and "+." Given x(i), the corresponding y(i) is also called the label for the training example.

如果我们要用线性回归算法来解决一个分类问题,对于分类,y 取值为 0 或者 1,但如果你使用的是线性回归,那么假设函数的输出值可能远大于 1,或者远小于 0,即使所有训练样本的标签 y 都等于 0 或 1。尽管我们知道标签应该取值 0 或者 1,但是如果算法得到的值远大于 1 或者远小于 0 的话,就会感觉很奇怪。所以我们在接下来的要研究的算法就叫做逻辑回归算法,这个算法的性质是:它的输出值永远在 0 到 1 之间。

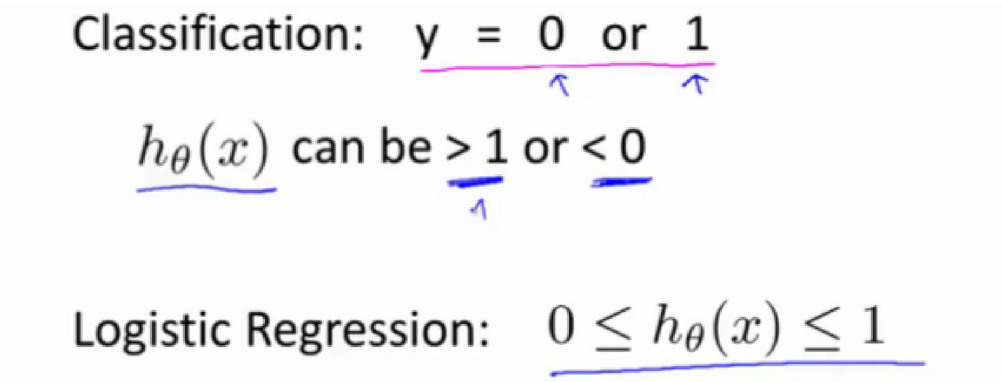

图 3: 逻辑回归算法

逻辑回归算法是分类算法，我们将它作为分类算法使用，有时候可能因为这个算法的名字中出现了"回归"使你感到困惑，但逻辑回归算法实际上是一种分类算法，它适用于标签y取值离散的情况下，如：1 ，0，0， 1 。

1.2 Hpothesis Representation

假设函数表达式:

我们希望分类器的输出在0和1之间，因此，要想出一个满足某个性质的假设函数，这个性质是它的预测值要在0和1之间。

回顾肿瘤分类问题：可以用线性回归的方法求出一条适合数据的一条直线:

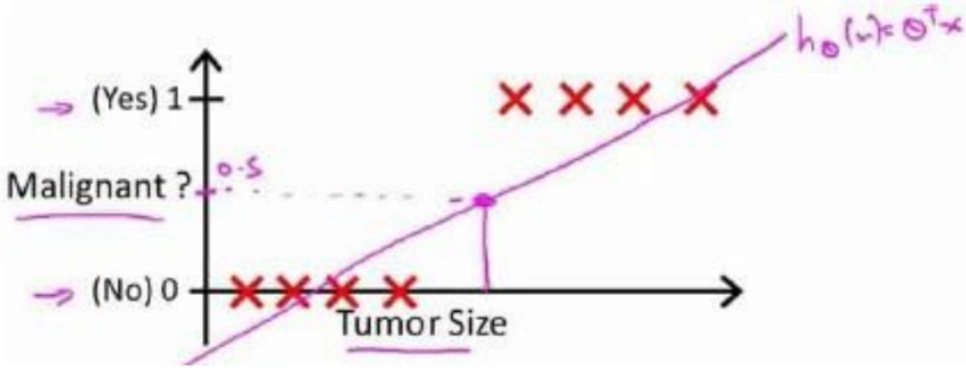

图 4: 肿瘤分类线性回归

根据线性回归我们只能预测连续的值，然而对于分类问题，我们要输出0或1，我们可以预测：（1）当h_θ大于等于 0.5 时,预测 y=1；（2）当 h_θ 小于 0.5 时,预测 y=0 对于上图所示的数据,这样的一个线性模型似乎能很好地完成分类任务。假使我们又观测到一个非常大尺寸的恶性肿瘤,将其作为实例加入到我们的训练集中来,这将使得我们获得一条新的直线。

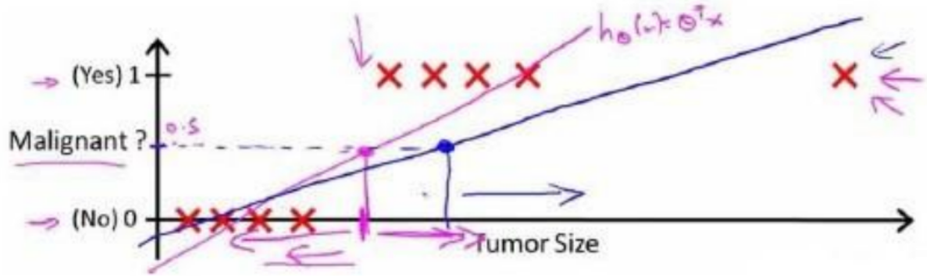

图 5: 肿瘤问题线性回归超范围

这时，再使用 0.5 作为阀值来预测肿瘤是良性还是恶性便不合适了。可以看出线性回归模型，因为其预测的值可以超越[0,1]的范围，并不适合解决这样的问题。

We could approach the classification problem ignoring the fact that y is discrete-valued, and use our old linear regression algorithm to try to predict y given x. However, it is easy to construct examples where this method performs very poorly. Intuitively, it also doesn't make sense for h θ (x) to take values larger than 1 or smaller than 0 when we know that y ∈ 0, 1. To fix this, let's change the form for our hypotheses h θ (x) to satisfy 0⩽h θ (x)⩽1. This is accomplished by plugging θ Tx into the Logistic Function.

引入一个新的模型，逻辑回归，该模型的输出变量范围始终在 0 和 1 之间。

逻辑回归模型的假设是:

$$h_\theta(x) = g(\theta^T x) \tag{1}$$

其中:

x:代表特征向量；

g:代表逻辑函数(logistic function)，是一个常用的逻辑函数为 S 形函数(Sigmoid function)，公式为:

$$g(z) = \frac{1}{1+e^{-z}} \tag{2}$$

该函数的图像为:

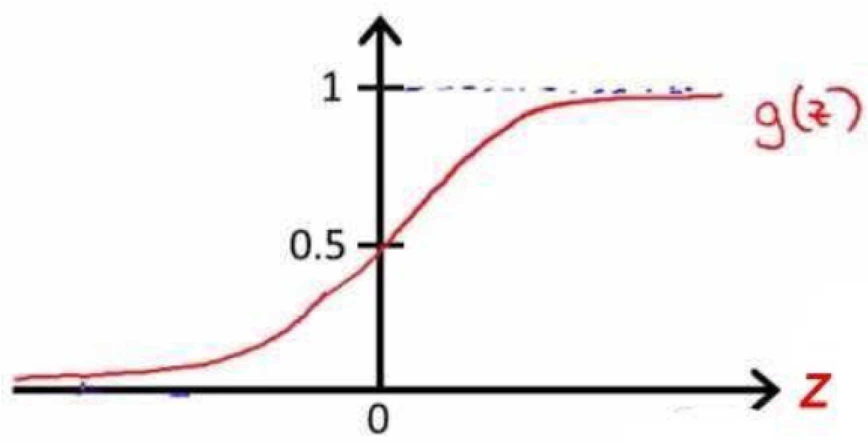

图 6: 逻辑函数（S函数）图像

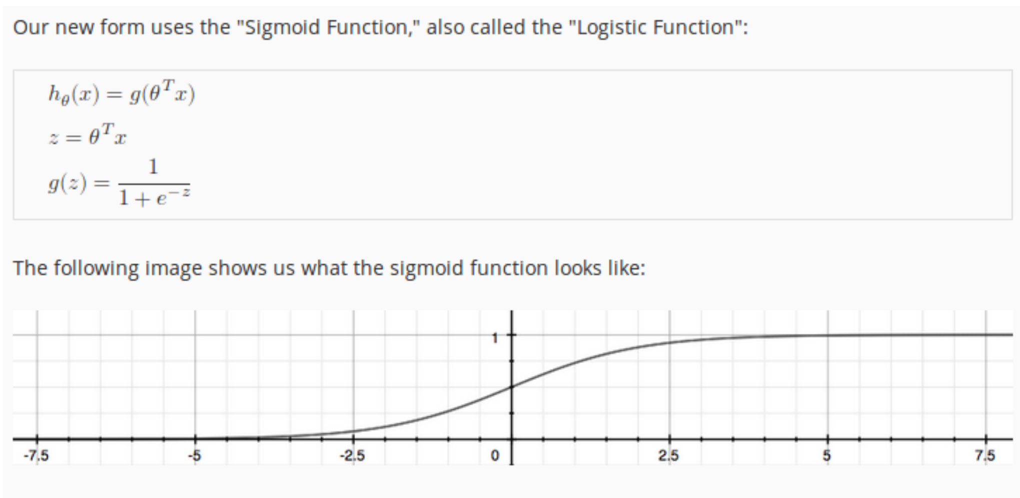

图 7: 逻辑函数表达式及图像

The function g(z), shown here, maps any real number to the (0, 1) interval, making it useful for transforming an arbitrary-valued function into a function better suited for classification.

逻辑回归模型的假设:

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \tag{3}$$

$h_\theta(x)$的作用是,对于给定的输入变量,根据选择的参数计算输出变量=1 的可能性(estimated probablity)即

$$h_\theta(x) = P(y=1|x;\theta) \tag{4}$$

$h_\theta(x)$will give us the probability that our output is 1.

$$h_\theta(x) = P(y=1|x;\theta) = 1 - P(y=0|x;\theta) \tag{5}$$

$$P(y=0|x;\theta) + P(y=1|x;\theta) = 1 \tag{6}$$

例如，如果对于给定的 x，通过已经确定的参数计算得出$h_\theta(x) = 0.7$,则表示有 70%的几率 y 为正向类,相应地 y 为负向类的几率为 1-0.7=0.3。

1.3 Decision Boundary

决策边界（Decision Boundary）可以更好的帮助我们理解假设函数在计算什么

首先回顾一下逻辑回归中假设函数的表达式:

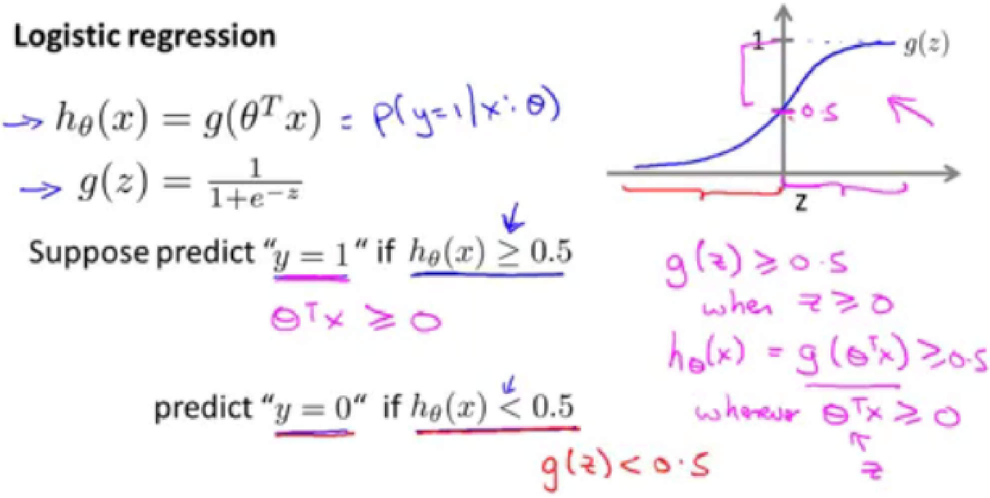

图 8: 逻辑函数回顾

根据上面绘制的S形函数的图像，我们可以知道:

当z=0时，g(z)=0.5;

当z>0时，g(z)>0.5;

当z<0时，g(z)<0.5;

又因为

$$z = \theta^T x \tag{7}$$

所以:

$\theta^T x \geq 0$时，预测y=1;

$\theta^T x < 0$时，预测y=0;

假设有一个模型:

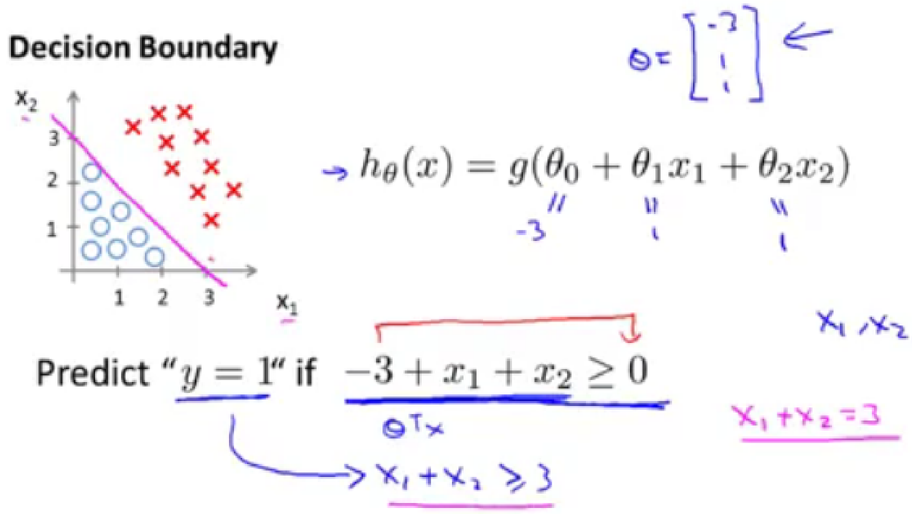

图 9: 线性决策边界

绘制直线$x_1 + x_2 = 3$,这条直线便是我们的模型的分界线，将预测y=1的区域和预测y=0的区分隔开，如图所示:

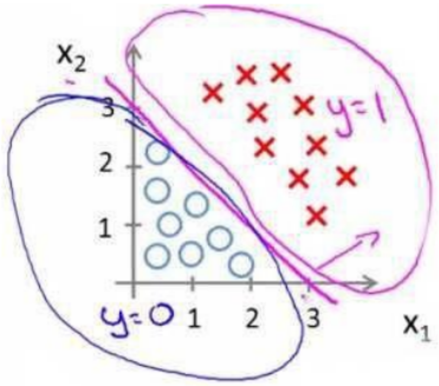

图 10: 线性决策边界图

另一种情况:

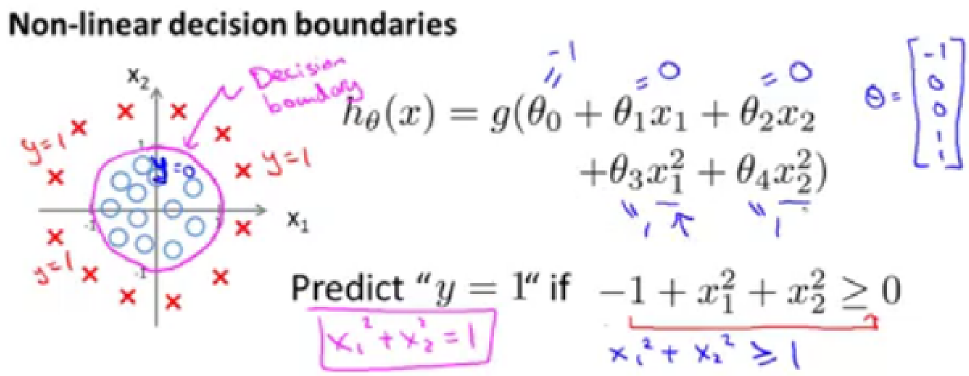

图 11: 非线性决策边界

需要二次方特征，得到的决策边界（判定边界）恰好是在原点且半径为1的圆形。

我们可以用非常复杂的模型来适应非常复杂的判定边界。 例如:

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + ...) \tag{8}$$

我们可能会得到比椭圆等更复杂的判定边界:

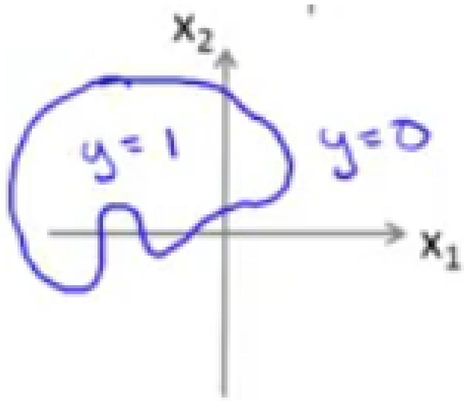

图 12: 更复杂的情况

2 Logistic Regression Model

2.1 Cost Function

2.2 Simplified Cost Function and Gradient Descent

2.3 Advanced Optimization

3 Multiclass Classification

3.1 Multiclass Classification:One-vs-all