机器学习实战

Stephen CUI[1]

February 17, 2023

[1]cuixuanStephen@gmail.com

Contents

Part I

分类

前两部分主要探讨监督学习（supervised learning）。在监督学习的过程中，我们只需要给定输入样本集，机器就可以从中推演出指定目标变量的可能结果。

监督学习一般使用两种类型的目标变量：标称型和数值型。标称型目标变量的结果只在有限目标集中取值，如真与假、动物分类集合 { 爬行类、鱼类、哺乳类、两栖类 } ；数值型目标变量则可以从无限的数值集合中取值，如 0.100、42.001、1000.743 等。数值型目标变量主要用于回归分析。

Chapter 1

Logistic回归

假设现在有一些数据点，我们用一条直线对这些点进行拟合（该线称为最佳拟合直线），这个拟合过程就称作回归。利用Logistic回归进行分类的主要思想是：根据现有数据对分类边界线建立回归公式，以此进行分类。这里的“回归”一词源于最佳拟合，表示要找到最佳拟合参数集。

1.1 基于 Logistic 回归和 Sigmoid 函数的分类

Logistic回归

优点：计算代价不高，易于理解和实现。
缺点：容易欠拟合，分类精度可能不高。
适用数据类型：数值型和标称型数据。

我们想要的函数应该是，能接受所有的输入然后预测出类别。例如，在两个类的情况下，上述函数输出0或1。或许你之前接触过具有这种性质的函数，该函数称为**海维塞德阶跃函数**（Heaviside step function），或者直接称为单位阶跃函数。然而，海维塞德阶跃函数的问题在于：该函数在跳跃点上从0瞬间跳跃到1，这个瞬间跳跃过程有时很难处理。幸好，另一个函数也有类似的性质[1]，且数学上更易处理，这就是Sigmoid函数。Sigmoid函数具体的计算公式如下：

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

为了实现Logistic回归分类器，我们可以在每个特征上都乘以一个回归系数，然后把所有的结果值相加，将这个总和代入Sigmoid函数中，进而得到一个范围在0 1之间的数值。任何大于0.5的数据被分入1类，小于0.5即被归入0类。所以，Logistic回归也可以被看成是一种概率估计。

1.2 基于最优化方法的最佳回归系数确定

Sigmoid函数的输入记为z，由下面公式得出：

$$z = w_0x_0 + w_1x_1 + w_2x_3 + \cdots + w_nx_n$$

[1]Sigmoid函数是一种阶跃函数（step function）。在数学中，如果实数域上的某个函数可以用半开区间上的指示函数的有限次线性组合来表示，那么这个函数就是阶跃函数。而数学中指示函数（indicator function）是定义在某集合X上的函数，表示其中有哪些元素属于某一子集A。

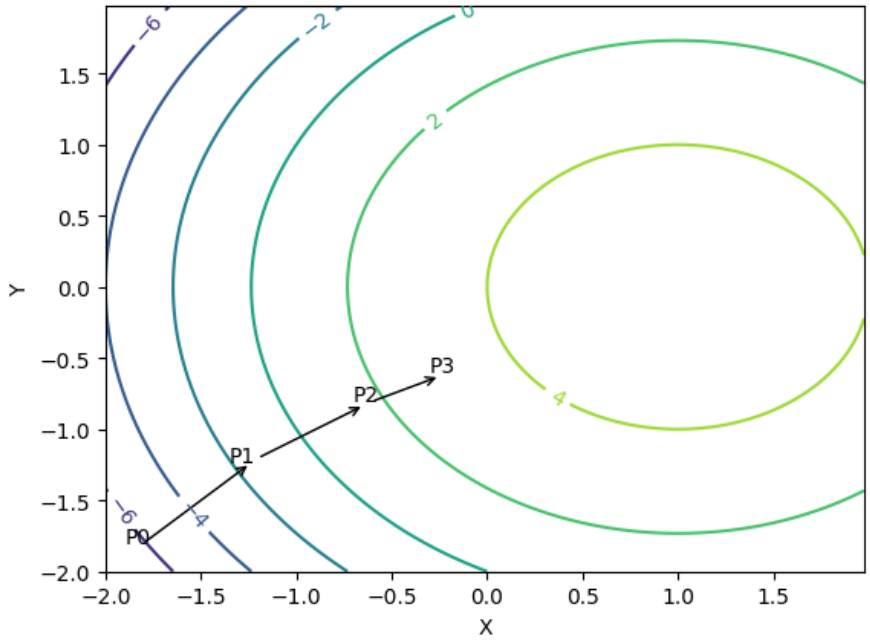

Figure 1.1: Gradient descent

如果采用向量的写法，上述公式可以写成$z = \boldsymbol{w}^T\boldsymbol{x}$，它表示将这两个数值向量对应元素相乘然后全部加起来即得到z值。其中的向量$\boldsymbol{x}$是分类器的输入数据，向量$\boldsymbol{w}$也就是我们要找到的最佳参数（系数），从而使得分类器尽可能地精确。

1.2.1 梯度上升法

梯度上升法基于的思想是：要找到某函数的最大值，最好的方法是沿着该函数的梯度方向探寻。如果梯度记为∇，则函数$f(x,y)$的梯度由下式表示：

$$\nabla f(x,y) = \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

这个梯度意味着要沿x的方向移动$\frac{\partial f(x,y)}{\partial x}$，沿$y$的方向移动$\frac{\partial f(x,y)}{\partial y}$。其中，函数$f(x,y)$必须要在待计算的点上有定义并且可微。

Figure 1.1中的梯度上升算法沿梯度方向移动了一步。可以看到，梯度算子总是指向函数值增长最快的方向。这里所说的是移动方向，而未提到移动量的大小。该量值称为步长，记做α。用向量来表示的话，梯度上升算法的迭代公式如下：

$$\boldsymbol{w} := \boldsymbol{w} + \alpha\nabla_{\boldsymbol{w}} f(w)$$

梯度下降

梯度下降算法，它与这里的梯度上升算法是一样的，只是公式中的加法需要变成减法。因此，对应的公式可以写成

$$\boldsymbol{w} := \boldsymbol{w} - \alpha\nabla_{\boldsymbol{w}} f(w)$$

梯度上升算法用来求函数的最大值，而梯度下降算法用来求函数的最小值。

1.2.2 训练算法：使用梯度上升找到最佳参数

梯度上升法的伪代码如下：

Algorithm 1: 梯度上升法

每个回归系数初始化为1;

repeat

  计算整个数据集的梯度;

  使用alpha × gradient更新回归系数的向量;

until *R*次;

return 回归系数

Chapter 2

树回归