**logistic回归**

1. **logistic回归概述：**

**Logistic**回归又称logistic回归分析，主要在流行病学中应用较多，比较常用的情形是探索某疾病的危险因素，根据危险因素预测某疾病发生的概率，等等。例如，想探讨胃癌发生的危险因素，可以选择两组人群，一组是胃癌组，一组是非胃癌组，两组人群肯定有不同的体征和生活方式等。这里的因变量就是是否胃癌，即“是”或“否”，为两分类变量，[自变量](https://baike.baidu.com/item/%E8%87%AA%E5%8F%98%E9%87%8F" \t "https://baike.baidu.com/item/Logistic%E6%A8%A1%E5%9E%8B/_blank)就可以包括很多了，例如年龄、性别、饮食习惯、幽门螺杆菌感染等。自变量既可以是连续的，也可以是分类的。通过logistic回归分析，就可以大致了解到底哪些因素是胃癌的危险因素。

1. **用途：**
2. 寻找危险因素，正如上面所说的寻找某一疾病的[危险](https://baike.baidu.com/item/%E5%8D%B1%E9%99%A9" \t "https://baike.baidu.com/item/Logistic%E6%A8%A1%E5%9E%8B/_blank)因素等。
3. 预测，如果已经建立了logistic回归模型，则可以根据模型，预测在不同的自变量情况下，发生某病或某种情况的概率有多大。
4. 判别，实际上跟预测有些类似，也是根据logistic模型，判断某人属于某病或属于某种情况的[概率](https://baike.baidu.com/item/%E6%A6%82%E7%8E%87" \t "https://baike.baidu.com/item/Logistic%E6%A8%A1%E5%9E%8B/_blank)有多大，也就是看一下这个人有多大的可能性是属于某病。
5. **预备知识：**

Logistic回归理解起来也是比较简单的。当然了，你的具备一些高等数学（如：导数和偏导数）、线性代数（向量和矩阵）、最优化理论（梯度下降算法，梯度上升算法）等方面的知识。

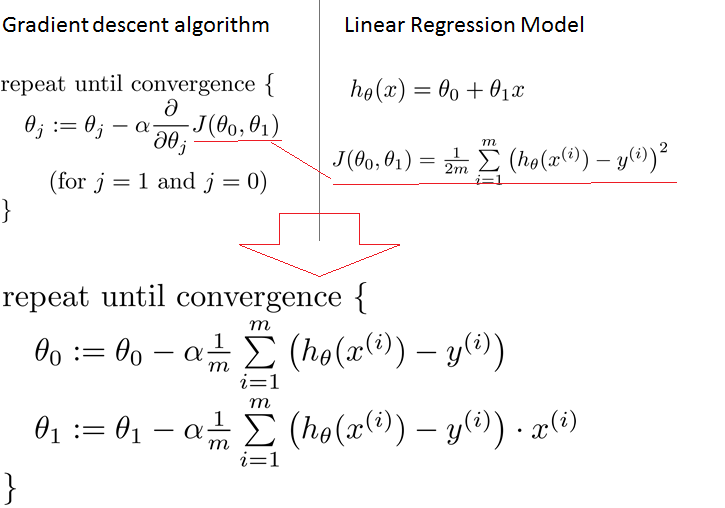
对于分类问题来说，我们首先需要假设我们的拟合函数，然后根据给定的训练数据集；来求出我们的拟合函数。然后将拟合函数作为Sigmoid函数的输入值，将我们的输入值映射到0到1之间。然后根据映射值得大小来进行分类。

如果映射值大于0.5，属于某一类，如果映射值小于0.5，属于另一类。

上面描述起来很简单，但是，实际操作起来还是有些难度的。难度在于去寻找我们的拟合函数。

梯度下降法：

迭代求出最优值：



很简单，不多说了。

对梯度下降的补充：

## **4.梯度下降法的简单应用**

### **4.1求**f(x)=x2f(x)=x2**的极小值**

f(x)=x2的梯度为： ∇f(x)=2x

步长设置为0.1，选取自变量从3开始，则计算过程如下

| **迭代次数(n)** | **自变量(**xx**)** | **梯度(**2x2x**)** | **步长(**αα**)** | **因变量(**x2x2**)** |
| --- | --- | --- | --- | --- |
| 0 | 3 | 6 | 0.1 | 9 |
| 1 | 2.4 | 4.8 | 0.1 | 5.76 |
| 2 | 1.92 | 3.84 | 0.1 | 3.69 |
| 3 | 1.536 | 3.072 | 0.1 | 2.36 |
| 10 | 0.32 | 0.64 | 0.1 | 0.10 |
| 20 | 0.03 | 0.06 | 0.1 | 0.0009 |
| … | … | … | … | … |

可以看到随着迭代次数的增加，该函数越来越接近极小值点(0,0)(0,0)，依据该方法一定可以找到精度允许范围内的极小值点。

以下是迭代count次的代码：

if \_\_name\_\_ == "\_\_main\_\_":

x = 3

y = x \* x

alpha = 0.1

count = 3

while (count > 0):

x = x - alpha \* 2 \* x

y = x \* x

count = count - 1

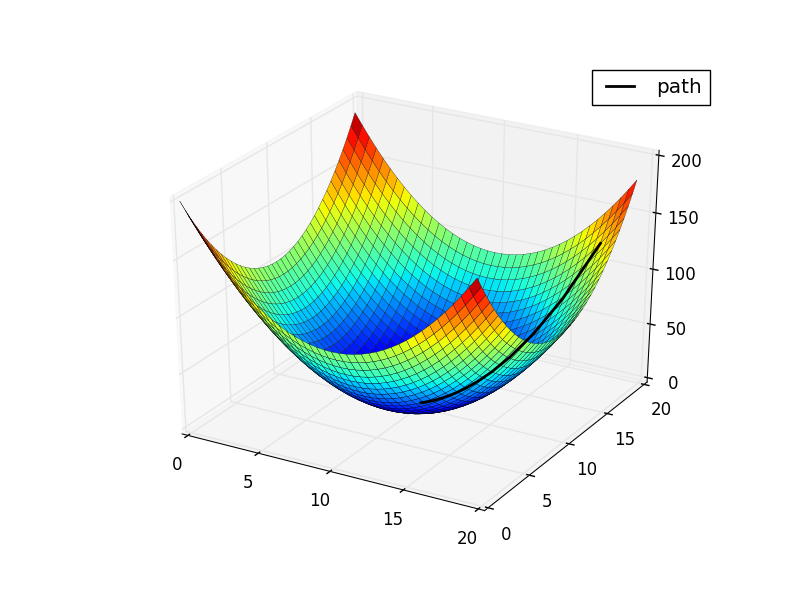
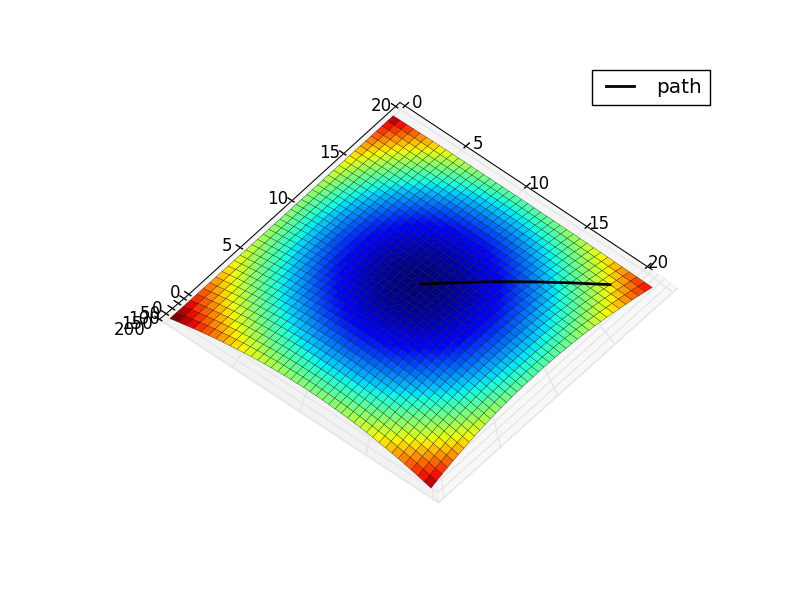
print x, y

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

### **4.2求**f(x,y)=(x−10)2+(y−10)2f(x,y)=(x−10)2+(y−10)2**的极小值**

f(x,y)=(x−10)2+(y−10)2f(x,y)=(x−10)2+(y−10)2的梯度为：

∇f(x,y)=(2(x−10),2(y−10))∇f(x,y)=(2(x−10),2(y−10))

步长设置为0.1，选择初始点为(20,20)(20,20)，这次以图形表示计算过程，图中的黑色曲线即为梯度下降法下降时的轨迹，效果非常好。   
斜视图：   
   
俯视图：   


## **5.小结**

* 梯度下降法求的是****极小值****，而不是****最小值****
* 梯度下降法常常用来求凸函数的最小值，例如机器学习中各种代价函数的最小值
* 步长的选取很关键，步长过长达不到极值点甚至会发散，步长太短导致收敛时间过长
* 斯坦福的机器学习视频中建议按照[0.001,0.003,0.01,0.03,…]的顺序尝试设置步长，同时观察函数值选择收敛最快的步长
* 步长也可以设置为非固定值，根据迭代的情况变化
* 下降的初始点一般设置为从原点开始