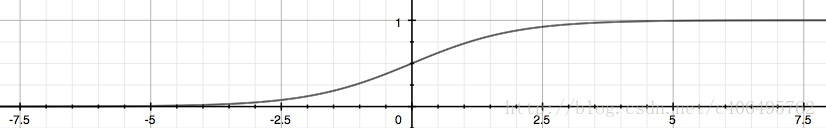
## Logistic回归与梯度上升算法

Logistic回归是众多分类算法中的一员。通常，Logistic回归用于二分类问题，例如预测明天是否会下雨。当然它也可以用于多分类问题，不过为了简单起见，本文暂先讨论二分类问题。首先，让我们来了解一下，什么是Logistic回归。

1 Logistic回归

Logistic回归一种二分类算法，它利用的是Sigmoid函数阈值在[0,1]这个特性。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。其实，Logistic本质上是一个基于条件概率的判别模型(Discriminative Model)。

我们在线性回归中可以得到一个预测值，再将该值映射到Sigmoid函数中，这样就完成了有值到概率的转换，也就是分类任务



所以要想了解Logistic回归，我们必须先看一看Sigmoid函数 ，我们也可以称它为Logistic函数。它的公式如下：

整合成一个公式，就变成了如下公式：

z是一个矩阵，θ是参数列向量(要求解的)，x是样本列向量(给定的数据集)。θ^T表示θ的转置。g(z)函数实现了任意实数到[0,1]的映射，这样我们的数据集([x0,x1,…,xn])，不管是大于1或者小于0，都可以映射到[0,1]区间进行分类。hθ(x)给出了输出为1的概率。比如当hθ(x)=0.7，那么说明有70%的概率输出为1。输出为0的概率是输出为1的补集，也就是30%。

如果我们有合适的参数列向量θ([θ0,θ1,…θn]^T)，以及样本列向量x([x0,x1,…,xn])，那么我们对样本x分类就可以通过上述公式计算出一个概率，如果这个概率大于0.5，我们就可以说样本是正样本，否则样本是负样本。

举个例子，对于"垃圾邮件判别问题"，对于给定的邮件(样本)，我们定义非垃圾邮件为正类，垃圾邮件为负类。我们通过计算出的概率值即可判定邮件是否是垃圾邮件。

那么问题来了！如何得到合适的参数向量θ?

根据sigmoid函数的特性，我们可以做出如下的假设：

上式即为在已知样本x和参数θ的情况下，样本x属性正样本(y=1)和负样本(y=0)的条件概率。理想状态下，根据上述公式，求出各个点的概率均为1，也就是完全分类都正确。但是考虑到实际情况，样本点的概率越接近于1，其分类效果越好。比如一个样本属于正样本的概率为0.51，那么我们就可以说明这个样本属于正样本。另一个样本属于正样本的概率为0.99，那么我们也可以说明这个样本属于正样本。但是显然，第二个样本概率更高，更具说服力。我们可以把上述两个概率公式合二为一：

似然函数：

对数似然函数：

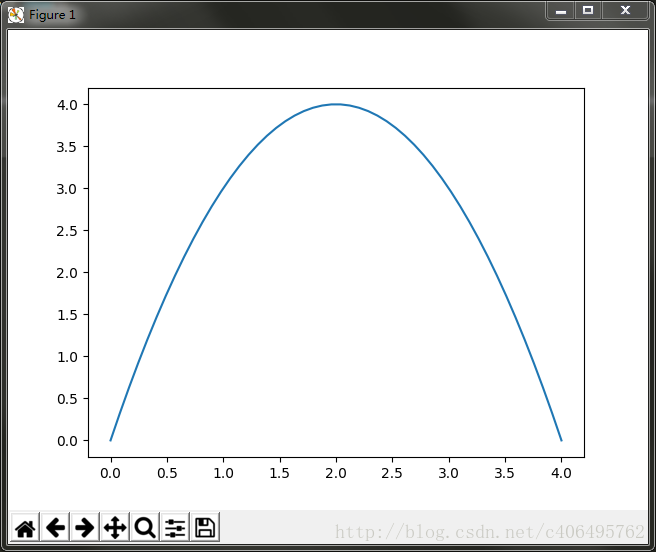
其中，m为样本的总数，y(i)表示第i个样本的类别，x(i)表示第i个样本，需要注意的是θ是多维向量，x(i)也是多维向量。综上所述，满足J(θ)的最大的θ值即是我们需要求解的模型。

怎么求解使J(θ)最大的θ值呢？因为是求最大值，所以我们需要使用梯度上升算法。如果面对的问题是求解使J(θ)最小的θ值，那么我们就需要使用梯度下降算法。面对我们这个问题，如果使J(θ) := -J(θ)，那么问题就从求极大值转换成求极小值了，使用的算法就从梯度上升算法变成了梯度下降算法，它们的思想都是相同的，学会其一，就也会了另一个。本文使用梯度上升算法进行求解。

2 梯度上升算法

说了半天，梯度上升算法又是啥？J(θ)太复杂，我们先看个简单的求极大值的例子。一个看了就会想到高中生活的函数：

来吧，做高中题。这个函数的极值怎么求？显然这个函数开口向下，存在极大值，它的函数图像为：

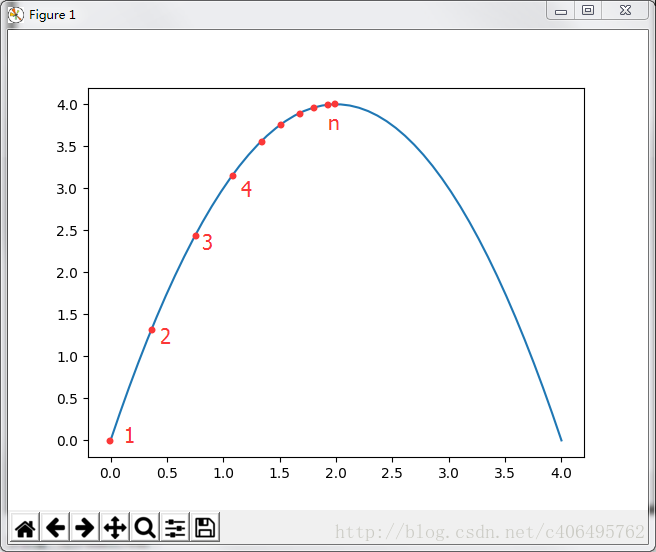


求极值，先求函数的导数：

令导数为0，可求出x=2即取得函数f(x)的极大值。极大值等于f(2)=4

但是真实环境中的函数不会像上面这么简单，就算求出了函数的导数，也很难精确计算出函数的极值。此时我们就可以用迭代的方法来做。就像爬坡一样，一点一点逼近极值。这种寻找最佳拟合参数的方法，就是最优化算法。爬坡这个动作用数学公式表达即为：

其中，α为步长，也就是学习速率，控制更新的幅度。效果如下图所示：



比如从(0,0)开始，迭代路径就是1->2->3->4->…->n，直到求出的x为函数极大值的近似值，停止迭代。这一过程，就是梯度上升算法。那么同理，J(θ)这个函数的极值，也可以这么求解。公式可以这么写：

由上小节可知J(θ)为：

sigmoid函数为：

那么，现在我只要求出J(θ)的偏导，就可以利用梯度上升算法，求解J(θ)的极大值了。

那么现在开始求解J(θ)对θ的偏导，求解如下(数学推导)：

其中：

再由：

可得：

接下来，就剩下第三部分：

综上所述：

因此，梯度上升迭代公式为：

其中：

知道了，梯度上升迭代公式，我们就可以自己编写代码，计算最佳拟合参数了。

3.全批量梯度下降和随机梯度下降

