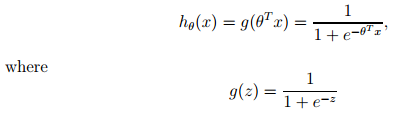
Logistic Regression

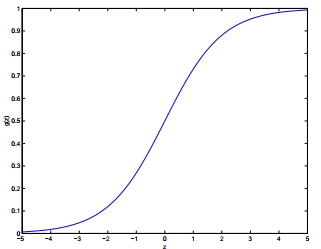
分类与回归的不同点在于分类的y值是离散值，而回归是连续值。

# 1.Logistic regression (逻辑回归)

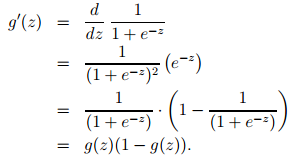
 逻辑回归中的假设（hypotheses）为:



 该函数称为逻辑函数或者S型函数，函数具有如下图特点：



 并且该函数的导数有如下特点：



    既然有了假设函数（hypotheses）,如何得到C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(3).png，进行如下的一些推导：

    \*假设

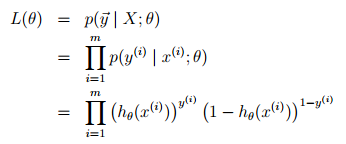
C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(4).png

      h = estimated probability that y = 1 on input x.

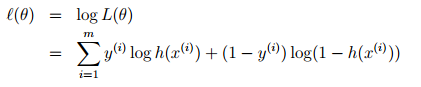
      该式可写成：

C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(5).png

       假设m个训练集是相互独立的，这样参数的可能性可以写成：

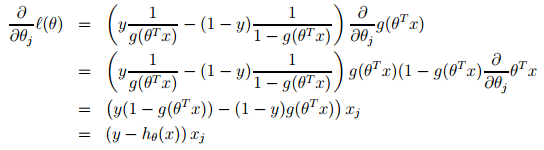


       然后写成对数的形式，这样最大化比较容易：



      这样可得到逻辑回归的cost function C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(10).png

      如何来最大化这个函数呢，与前面最小化线性回归相似，可以使用梯度上升（gradient ascent）的形式,每一次迭代过程中改变C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(11).png值，C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(12).png。通过对一个训练例子，可以得到梯度上升的更新规则：



C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(14).png

C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(15).png(向量化)

# 2.最小化的（- C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(16).png）其它算法（逻辑回归中）

   -- Gradient descent

  -- Conjugate gradient

   -- BFGS

  -- L- BFGS

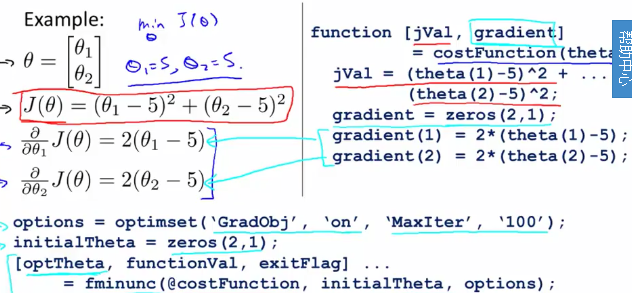
 Advantages:

  -- No need to manually pick a

  -- often faster than gradient descent

 Disadvantages:

 -- More complex



# 3.Multiclass Classification: one vs all

Train a logistic regression classifier C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(24).png for each class i to predict the probability that y = i.

On a new input x, to make a prediction, pick the class i that maximizes

C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(25).png

4.Regularized Logistic Regression  
  
 Cost Function:  
