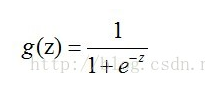
**作业3.3**

1. **实验要求**

编程实现对率回归，并给出西瓜数据集3.0a上的结果。

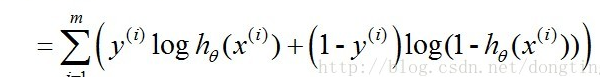
1. **实验原理**

使用对数几率函数去将真实标记y与线性回归的预测值联系起来，其是一个sigmoid函数，如下所示：



将z转化成一个接近0或1的值。Z=x+b，利用g(z)去拟合标记值，即可得到线性回归方程。

经过一系列推导，得到优化目标为:



令为l(),则优化目标为argmin(l()),求解得出相应的w,b即可。

1. **实验过程与代码**

本人使用开源工具库tensorflow用于解决该题，主要过程如下：

1. 将西瓜数据集a转换成矩阵的形式。
2. 设置w,b等各种参数，以及训练样本和标记。
3. 定义优化目标（代码中的loss），并利用tensorflow对其进行最优化求解。
4. 利用tensorflow中的会话功能进行训练求解。
5. 解出的w,b代入原线性方程，得到两个属性之间的数学关系表达式。
6. 对原数据进行画图，对拟合出的两个属性之间关系线进行画图。

相关代码如下：（https://github.com/CHNnwpuer/machine-learning-homework-zhou-）

|  |  |
| --- | --- |
|  | from numpy import \* |
|  | import matplotlib.pyplot as plt |
|  | Import TensorFlow as tf |
|  | data = mat([[0.697,0.460,1], |
|  | [0.774,0.376,1], |
|  | [0.634,0.264,1], |
|  | [0.608,0.318,1], |
|  | [0.556,0.215,1], |
|  | [0.403,0.237,1], |
|  | [0.481,0.149,1], |
|  | [0.437,0.211,1], |
|  | [0.666,0.091,0], |
|  | [0.243,0.267,0], |
|  | [0.245,0.057,0], |
|  | [0.343,0.099,0], |
|  | [0.639,0.161,0], |
|  | [0.657,0.198,0], |
|  | [0.360,0.370,0], |
|  | [0.593,0.042,0], |
|  | [0.719,0.103,0]]) |
|  | #数据类型转换 |
|  | x\_train = data[:,0:2].astype(float32) |
|  | y\_train = data[:,2].astype(float32) |
|  | #定义计算w,b的参数 |
|  | y\_train = mat(y\_train) |
|  | w = tf.Variable(tf.zeros([2, 1])) |
|  | b= tf.Variable(tf.zeros([1, 1])) |
|  | y = 1 / (1 + tf.exp(-(tf.matmul(x\_train, w) + b))) |
|  | loss = tf.reduce\_mean(- y\_train.reshape(-1, 1) \* tf.log(y) - (1 - y\_train.reshape(-1, 1)) \* tf.log(1 - y)) |
|  | train = tf.train.GradientDescentOptimizer(0.5).minimize(loss) |
|  | #运行并显示w,b的运行结果 |
|  | init = tf.initialize\_all\_variables() |
|  | sess = tf.Session() |
|  | sess.run(init) |
|  | for step in range(10000): |
|  | sess.run(train) |
|  | print(step, sess.run(w).flatten(), sess.run(b).flatten()) |
|  | #画图 |
|  | X0 = array(data[:8]) |
|  | X1 = array(data[8:]) |
|  | ax = plt.subplot(111) |
|  | ax.scatter(X0[:,0],X0[:,1],c='r',label='+') |
|  | ax.scatter(X1[:,0],X1[:,1],c='b',label='-') |
|  | x1 = arange(-0.2,1.0,0.1) |
|  | y1 = (-b-w[0]\*x1)/w[1] |
|  | #print(shape(sess.run(y1).flatten())) |
|  | plt.sca(ax) |
|  | plt.plot(x1,sess.run(y1).flatten()) |
|  | plt.xlabel('density') |
|  | plt.ylabel('ratio\_sugar') |
|  | plt.legend() |
|  | plt.show() |
|  |  |

1. **实验结果**

得到w =[  3.15144444  12.49511814],b= [-4.41981125]。

回归图线如下：

