

Augster的博客

目录视图

摘要视图

RSS 订阅

个人资料



秋水长天q

关注

发私信

访问：32291次

积分：637

等级：BLOG > 3

排名：千里之外

原创文章：34篇

转载：0篇

译文：0篇

评论：25条

文章搜索

文章分类

机器学习 (15)

Java基础 (6)

Hadoop (6)

Python (1)

线性代数 (1)

Leetcode (4)

数据结构 (1)

JavaWeb (1)

文章存档

2017年09月 (1)

2017年07月 (1)

2017年04月 (2)

2017年03月 (4)

2017年02月 (1)

展开

阅读排行

神经网络之BP神经网络 (Pyt... (12732)

迁移学习算法之TrAdaBoost (6125)

图灵赠书——程序员11月书单

【思考】Python这么厉害的原因竟然是！

感恩节赠书：《深度学习》等异步社区优秀图书和作译者评选启动！

每周荐书：京东架构、Linux内核、Python全栈

线性回归

标签：机器学习

2016-12-30 16:34

341人阅读

评论(0)

收藏

举报

分类：

机器学习 (14)

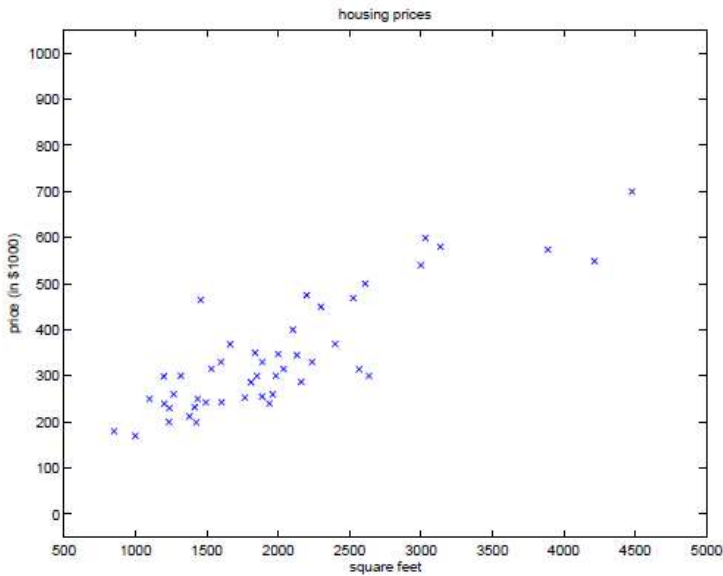
版权声明：本文为博主原创文章，未经博主允许不得转载。

1.前言

时间如白驹过隙，转眼之间就到了2016年的年末，近来回想自己这一年的收获，感慨颇多。翻看之前的学习，觉得自己依旧存在着很多的问题，上学期大四的最后日子，做完毕设悠悠哉哉了很久，暑假回家找了个兼职，其中的经历个中滋味只有自己知道。下学期在实验室，刚开始的时候一度迷茫和抱怨过，后来渐渐的也想明白了，既然事实依然如此，不如就躺下来享受吧。最近想着把以前做的工作好好的总结一下，做一个回顾。

2 线性回归

线性回归问题是机器学习中常见的基本问题之一，问题模型如下：给定一系列的数据样例和标记，用一个线性的方程来表示这些数据。简单点也就是给出点，求拟合曲线。在Andrew Ng的课程里，给出一系列房屋的面积和售价，求二者之间的线性关系。



基于用户的协同过滤推荐一实...	(1062)
范数汇总	(789)
SVD矩阵奇异值分解	(785)
决策树学习	(747)
迁移学习概述	(722)
集体智慧编程第三章之发现群组	(629)
集体智慧编程第二章之提供推荐	(581)
MapReduce实现KNN	(540)

评论排行	
迁移学习算法之TrAdaBoost	(14)
神经网络之BP神经网络 (Pyt...	(6)
范数汇总	(3)
MapReduce实现KNN	(1)
k-近邻算法	(0)
决策树学习	(0)
贝叶斯分类算法	(0)
SSM框架整合	(0)
equals和hashCode方法	(0)
Java线程	(0)

二手玛莎拉蒂

最新评论	
神经网络之BP神经网络 (Python实现) 香香的七仔 : 使用√ (输入层节点数×输出层节点数) 作为隐藏层节点数 (88个) 可以将正确率提升到大概95%	
范数汇总 yujianglan : 你好, 请问F范数和21范数之间怎么转化呢? 谢谢	
迁移学习算法之TrAdaBoost confina : 想问下什么样本的权重怎么和基本的学习器结合啊 哪一种学习器会考虑样本的权重呢	
范数汇总 秋水长天q : @aoqiulei7117机器学习中的很多地方都会有范数的应用, 比如说公式中的一些约束之类的:	
范数汇总 aoqiulei7117 : 范数有啥应用呢	
神经网络之BP神经网络 (Python实现) weixin_39518878 : 博主你好, 请问这段代码的迭代次数在哪里啊	
神经网络之BP神经网络 (Python实现) Coding_ForFun : @Fengming1220:用matlab就可以转化为mat格式了~	
神经网络之BP神经网络 (Python实现) Fengming1220 : 你好, 请问原始数据怎么转化为.mat格式文件?	
MapReduce实现KNN binyet : 您好, 可以看一下您的Instance这	

这是一个二维的问题, 用一条直线就可以进行拟合。对于多维的情况, 用

powered by
CODECOGS[®]
Please obtain a licence
from www.CodeCogs.com
❗ Equation Quota Exceeded

The World's #1 Online Equation Editor表示多维的数据, 用Y表示数据的标记, 对于线性回归, 是要找到这样的一

powered by
CODECOGS[®]
Please obtain a licence
from www.CodeCogs.com
❗ Equation Quota Exceeded

个The World's #1 Online Equation Editor使得:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

powered by
CODECOGS[®]
Please obtain a licence
from www.CodeCogs.com
❗ Equation Quota Exceeded

成立, 如果令The World's #1 Online Equation Editor=1, 则下面的式子成立:

$$h(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x,$$

有了线性回归的模型, 定义下面的代价函数:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2.$$

powered by
CODECOGS[®]
Please obtain a licence
from www.CodeCogs.com
❗ Equation Quota Exceeded

现在要求取The World's #1 Online Equation Editor使得代价函数J最小, 求J最小的方法可以采用梯度下降算法和最小二乘法来实现。

3 梯度下降算法

梯度下降算法是典型的利用搜索算法来求取最优化的问题, 每次求取函数的负梯度方向, 沿着函数的负梯度方向进行搜索, 知道找到函数的最有解为止。对J进行求导, 过程如下:

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_{\theta}(x) - y)^2 \\ &= 2 \cdot \frac{1}{2} (h_{\theta}(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_{\theta}(x) - y) \\ &= (h_{\theta}(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^n \theta_i x_i - y \right) \\ &= (h_{\theta}(x) - y) x_j \end{aligned}$$

月嫂价格表



所以对于 [The World's #1 Online Equation Editor](#) , 可以进行如下迭代：

$$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)} \quad (\text{for every } j).$$



可以看出，梯度下降算法在每次对 [The World's #1 Online Equation Editor](#) 进行迭代的时候都要计算所有样本的一个梯度值，在样本数字比较小的时候，可以在较短的时间内找到全局的最优解，当样本的个数比较多时候，这样的方法是很费时间的。这种方法叫做批梯度下降算法，与批梯度下降算法对应的是随机梯度下降算法。

4 随机梯度下降算法



随机梯度下降算法每次在对 [The World's #1 Online Equation Editor](#) 进行迭代的时候选择一个样本，利用一个样



本的梯度值对 [The World's #1 Online Equation Editor](#) 来进行该进，所以在时间性能上优于批梯度下降算法。批梯度下降方法在每次迭代的时候总是可以向着最优化的方向进行，但是批梯度下降算法每次并不一定是向着最优化的方向进行，但是整体的效果是向着最优化的方向。

$$\begin{aligned} &\text{for } i=1 \text{ to } m, \{ \\ &\quad \theta_j := \theta_j + \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)} \quad (\text{for every } j). \\ &\} \end{aligned}$$

5.最小二乘法

用最小二乘法来求取最优解，上述的问题可以简化为下述方程问题

$$X\theta = Y$$



其中X表示样本，Y表示标记，[The World's #1 Online Equation Editor](#) 为求取的变量，现在方程组中方程的个数多于变量的个数，所以只能求取最优解，由线性代数的内容可以知道该方程的最有解可以通过 在方程两边同时乘以X的转置来实现。所以：

$$X^T X \theta = X^T \vec{y}$$

月嫂价格表

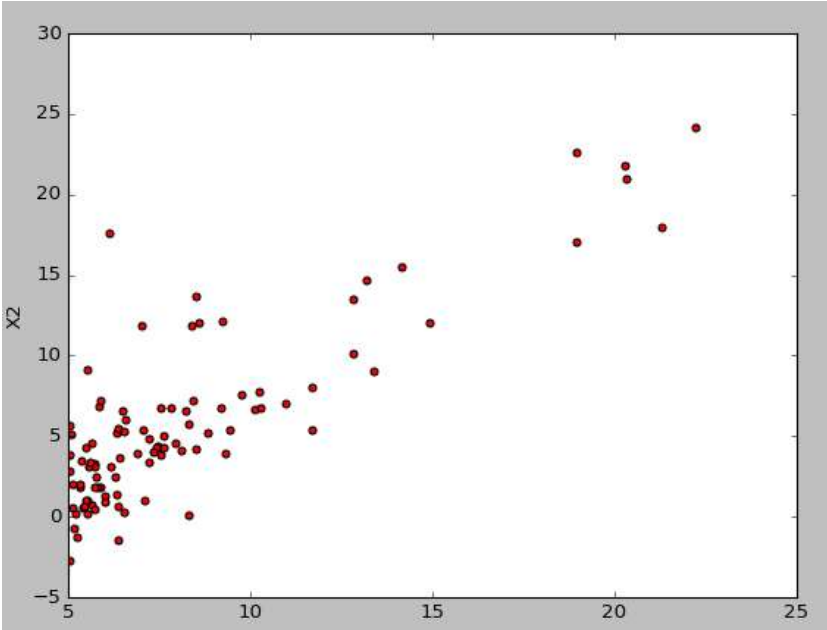


最终求得的

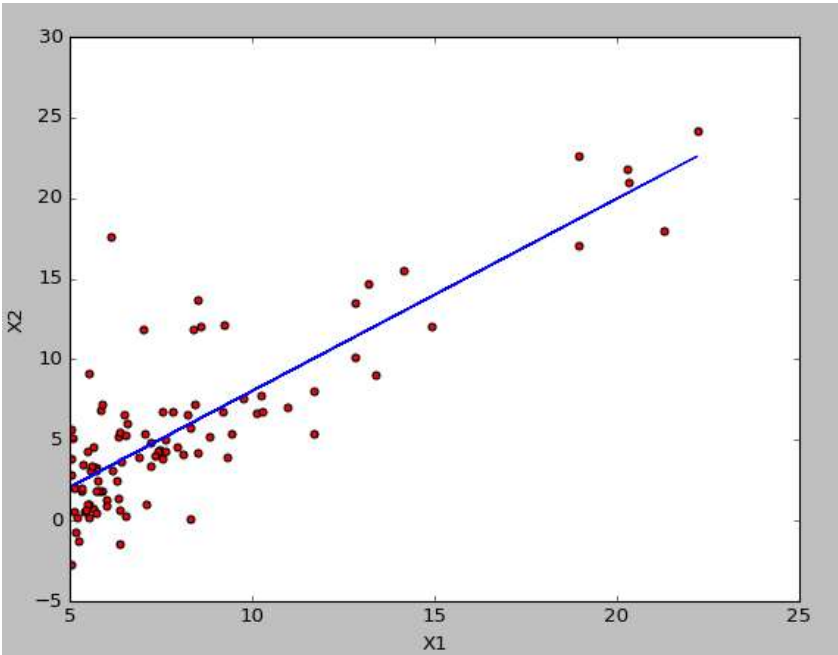
$$\theta = (X^T X)^{-1} X^T \vec{y}.$$

6 实验实现

实验采取的数据的散点图如下：



最终求得的结果如下所示：



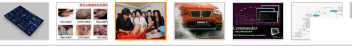
所有的代码如下：

[python] view plain copy print ?

```
01. <span style="font-size:14px;">##简单线性回归
02. import numpy as np
03. import matplotlib.pyplot as plt
04. #导入数据
05. def ReadFile(filename):
06.     data=[]
07.     label=[]
08.     lines=open(filename,encoding='utf-8').readlines()
09.     for i in range(len(lines)):
10.         xy=lines[i].strip().split(',')
```



月嫂价格表



```

11.         data.append([1,float(xy[0])])
12.         label.append(float(xy[1]))
13.     return data,label
14.
15. #定义代价函数
16. def costfunction(X,Y,thea):
17.     m=np.shape(Y)[0]
18.     return sum((X.dot(thea)-Y)**2)/(2*m)
19.
20. #批梯度下降函数
21. def batchgrad(data,label):
22.     X = np.array(data)
23.     m, n = np.shape(X)
24.     Y = np.array(label).reshape(m,1)
25.     alpha=0.01                #定义步长
26.     iterations = 1500;         #定义迭代次数
27.     thea=np.ones((2,1))        #定义初始的点
28.     for k in range(iterations):
29.         H = X.dot(thea)
30.         T = np.zeros((2, 1))
31.         for i in range(m):
32.             T=T+((H[i]-Y[i])*X[i]).reshape(2,1)
33.         thea = thea - (alpha * T)/m
34.     return thea
35.
36. #随机梯度下降函数
37. def Randomgrad(data,label):
38.     X = np.array(data)
39.     m, n = np.shape(X)
40.     Y = np.array(label).reshape(m,1)
41.     alpha=0.01                #定义步长
42.     thea=np.ones((2,1))        #定义初始的点
43.     for i in range(m):
44.         thea=thea+alpha*((Y[i]-X[i].dot(thea))*X[i]).reshape(2,1)
45.     return thea
46.
47. #最小二乘法
48. def Leastsquares(data,label):
49.     X = np.array(data)
50.     m, n = np.shape(X)
51.     Y = np.array(label).reshape(m, 1)
52.     return np.linalg.inv(X.T.dot(X)).dot(X.T).dot(Y)
53.
54. if __name__=="__main__":
55.     data,label=ReadFile('ex1data1.txt')
56.     X = np.array(data)
57.     #thea=batchgrad(data,label)
58.     #thea=Randomgrad(data,label)
59.     thea=Leastsquares(data,label)
60.     print(thea)
61.     Y=X.dot(thea)
62.     fig=plt.figure()
63.     ax=fig.add_subplot(111)
64.     x=[]
65.     y1=[]
66.     y2=[]
67.     for i in range(len(data)):
68.         x.append(data[i][1])
69.         y1.append(label[i])

```



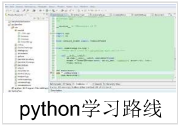
```
70.         y2.append(Y[i])
71.     ax.scatter(x,y1,c='red')
72.     ax.plot(x,y2)
73.     plt.xlim(5,25)
74.     plt.xlabel("X1")
75.     plt.ylabel("X2")
76.     plt.show()
77.
78. </span>
```

顶 0 踩 0

- 上一篇 集体智慧编程第三章之发现群组
- 下一篇 分类和Logistic回归

相关文章推荐

- 线性回归、logistic回归
- MySQL在微信支付下的高可用运营--莫晓东
- 线性回归、logistic回归、一般线性模型回归
- 容器技术在58同城的实践--姚远
- SparkML之回归(一)线性回归
- SDCC 2017之容器技术实战线上峰会
- 线性回归tensorflow例子，完整python代码
- SDCC 2017之数据库技术实战线上峰会
- 多元线性回归java实现
- 腾讯云服务器服务架构实现介绍--董晓杰
- 多元线性回归程序Fortran2013版
- 微博热点事件背后的数据库运维心得--张冬洪
- 贝叶斯线性回归
- 线性回归python实现(含数据集)
- 线性回归例子
- 机器学习-线性回归预测



查看评论

暂无评论

发表评论

用户名： wujuxKkoolerter
评论内容：



月嫂价格表



提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 

