个人资料

moodytong

访问： 279692次

积分： 3930

等级： BLOG 5

排名： 第4349名

原创： 115篇 转载： 24篇

译文： 6篇 评论： 50条

文章搜索

学院APP首次下载，可得50C币！ 欢迎来帮助开源“进步” 当讲师？爱学习？投票攒课吧 CSDN 2015博客之星评选结果公布

机器学习经典算法8-树回归

2013-08-19 09:18 6536人阅读 评论(6) 收藏 举报

分类： 统计分析与数据挖掘（16）

版权声明：本文为博主原创文章，未经博主允许不得转载。

1.简单介绍

线性回归方法可以有效的拟合所有样本点（局部加权线性回归除外）。当数据拥有众多特征并且特征之间关系十分复杂时，构建全局模型的想法一个是困难一个是笨拙。此外，实际中很多问题为非线性的，例如常见到的分段函数，不可能用全局线性模型来进行拟合。

树回归将数据集切分成多份易建模的数据，然后利用线性回归进行建模和拟合。这里介绍较为经典的树回归CART(classification and regression trees，分类回归树)算法。

2.分类回归树基本流程

构建树：

1.找到[最佳待切分特征]

2.若不能再切分，则将该节点存为[叶子节点]并返回

3.按照最佳待切分特征将数据集切分成左右子树（这里为了方便，假设大于特征值则为左，小于则归为右）

4.对左子树进行[构建树]

5.对右子树进行[构建树]

最佳待切分特征：

1.遍历特征

1.1遍历特征所有特征值

1.1.1计算按该特征值进行数据集切分的[误差]

2.选择误差最小的特征及其相应值作为最佳待切分特征并返回

基于回归树的预测：

1.判断当前回归树是否为叶子节点，如果是则[预测]，如果不是则执行2

2.将测试数据相应特征上的特征值与当前回归树进行比较，如果测试数据特征值大，则判别当前回归树的左子树是否为叶子节点，如果不是叶子节点则进行[基于回归树的预测]，如果是叶子节点，则[预测]；反之，判别当前回归树的右子树是否为叶子节点，如果不是叶子节点则进行[基于回归树的预测]，如果是叶子节点，则[预测]

3.分类回归树的实践说明

误差、叶子节点和预测三者有相关的关联关系，一种相对简单的是误差采用的是y值均方差，叶子节点相应的建立为该节点下所有样本的y值平均值，预测的时候根据判断返回该叶子节点下y值平均值即可。

在进行最佳待切分特征选取的时候，一般还有两个参数，一个是允许的误差下降值，一个是切分最小样本数。对于允许误差下降值，在实际过程中，需要在分割之后其误差减少应该至少大于该bound；对于切分最小样本数，也就是说切分后的子树中包含的样本数应该多于该bound。其实这两种策略都是为了避免过拟合。

4树剪枝

机器学习经典算法12-SVI (5655)
网络科学论坛纪要-2012 (5632)

Online Courses

Coursera
edX
MITOPENCOURSEWARE

Academic Search Engine

微软学术
Google Scholar
CiteSeer
Ohio Thesis
arxiv

评论排行

利用matlab进行简单的贝 (10)
机器学习经典算法8-树回 (6)
关于dom4j读取xml文件自 (5)
networkx使用笔记(三)之 (4)
机器学习经典算法7-线性 (4)
机器学习经典算法12-SVI (3)
利用mencoder进行视频 (3)
octopy的MapReduce编程 (2)
社交标签技术的研究 (2)
关于机器学习课程的小记 (2)

文章存档

2015年02月 (1)
2013年11月 (1)
2013年08月 (11)
2013年07月 (3)
2013年06月 (1)

展开

推荐文章

*App竞品技术分析 （6）热修复
*架构设计：系统间通信（17）——服务治理与Dubbo 中篇（分析）
*你的计划为什么执行不下去？怎么破？
*图解堆算法、链表、栈与队列（多图预警）
*【android】仿360手机卫士的简易设计思路及源码
*Android平台Camera实时滤镜实现方法探讨(九)--磨皮算法探讨(一)

关闭

最新评论

机器学习经典算法8-树回归
rururur: @chenybh110:我感觉这块也有错，他目的是想把dataSet分成两部分

机器学习经典算法8-树回归
rururur: @t2396156:那你是怎么解决的？我也遇到同样的问题了

机器学习经典算法8-树回归
柳枫大人: 第52行应该是有问题的，dataSet里面的元素是matrix类型的，无法用set

机器学习经典算法8-树回归
ourarewe: 请问楼主用的是什么数据，那个ex00之类又是什么

关于机器学习课程的小记
moodytong: @longyou1243:链接:http://pan.baidu.com/s/1i3J6pfv

通过在最佳待切分特征选取时进行参数设定来避免过拟合，这其实是一种预剪枝的行为；而在回归树建立后，再进行剪枝，则是一种后剪枝的行为。

后剪枝的过程如下：

如果存在任一子集是一棵树，则在该子集中递归剪枝

计算当前两个叶子节点合并后的误差

计算不合并的误差

比较合并前后误差，如果合并后的误差降低，则对叶子节点进行合并

5模型树

之前讲到误差、叶子节点和预测三者具备关联关系，当建立叶子节点是基于模型的，则构建了相应的模型树。这里可以使用之前的线性回归模型，建立相应的叶子节点。这样误差计算采用的将是线性回归中的误差，而预测则是基于该叶子节点拟合其样本后的参数。

6编程实现

这里createTree负责进行树的构建；chooseBestSplit函数负责进行最佳带切特征的选取，而ops参数则是进行了两个bound的设定；prune进行了相关后剪枝。

这里regErr、regEval是基于简单均值计算的误差、叶子节点和预测；而modelErr、modelLeaf和modelTreeSolve）则是基于线性回顾模型的误差、叶子节点和预测。

数据集链接:http://pan.baidu.com/share/link?shareid=3744521160&uk=973467359 密码:9ivd

```
[python] view plain copy print ?
from numpy import *
def loadDataSet(fileName):
    dataMat = []
    fr = open(fileName)
    for line in fr.readlines():
        curLine = line.strip('\n').split('\t')
        fltLine = map(float, curLine)
        dataMat.append(fltLine)
    fr.close()
    return dataMat
def regLeaf(dataSet):
    return mean(dataSet[:,-1])
def regErr(dataSet):
    return var(dataSet[:,-1])*shape(dataSet)[0]
def regTreeEval(model, inDat):
    return float(model)
def linearSolve(dataSet):
    m,n=shape(dataSet)
    X = mat(ones((m,n)))
    Y = mat(ones((m,1)))
    X[:,1:n] = dataSet[:,0:n-1]
    Y = dataSet[:,-1]
    xTx = X.T*X
    if linalg.det(xTx) == 0.0:
        raise NameError('This matrix is singular, cannot do inverse, \
the second value of ops')
    ws = xTx.I * (X.T * Y)
    return ws,X,Y
def modelLeaf(dataSet):
    ws,X,Y = linearSolve(dataSet)
    return ws
def modelErr(dataSet):
    ws,X,Y = linearSolve(dataSet)
    yHat = X * ws
    return sum(power(Y-yHat,2))
def modelTreeEval(model, inDat):
    n=shape(inDat)[1]
    X = mat(ones((1,n+1)))
    X[:,1:n+1]=inDat
    return float(X*model)
def chooseBestSplit(dataSet, leafType=regLeaf, errType=regErr, ops=(1,4)):
    tolS = ops[0]
    tolN = ops[1]
    if len(set(dataSet[:,-1].T.tolist()[0])) == 1:
        return None, leafType(dataSet)
    m,n=shape(dataSet)
    S = errType(dataSet)
    bestS = inf
    bestIndex = 0
    bestValue = 0
```

密...

关于机器学习课程的小记
longyou1243: 资料的连接不能用了，楼主更新一下连接可否？

机器学习经典算法10-Apriori
拾毅者: 36和37行还是不太明白，看《机器学习实战》这两行有点不明白，能解释下么
L1=list(Lk) L...

社交标签技术的研究
ebruce: 博主对“标签”理解太深了，彻底佩服。“标签”几乎可以在所有平台看到，只是形态上有些区别。最近“图片+...

机器学习经典算法7-线性回归
月光下的夜曲: 代码有错误，害人啊

networkx使用笔记(三)之好汉篇M
moodytong: @novalist:在前面要申明下import numpy as np注意安装numpy库。此外，有...

```
    for featIndex in range(n-1):
        for splitVal in set(dataSet[:,featIndex]):
            mat0, mat1 = binSplitDataSet(dataSet, featIndex, splitVal)
            if(shape(mat0)[0]<tolN) or (shape(mat1)[0]<tolN):
                continue
            newS = errType(mat0)+errType(mat1)
            if newS < bestS:
                bestIndex = featIndex
                bestValue = splitVal
                bestS = newS
    if (S-bestS)<tolS:
        return None, leafType(dataSet)
    mat0, mat1 = binSplitDataSet(dataSet, bestIndex, bestValue)
    if(shape(mat0)[0]<tolN) or (shape(mat1)[0]<tolN):
        print "Not enough nums"
        return None, leafType(dataSet)
    return bestIndex, bestValue
def binSplitDataSet(dataSet, feature, value):
    mat0 = dataSet[nonzero(dataSet[:, feature]>value)[0],:][0]
    mat1 = dataSet[nonzero(dataSet[:, feature]<=value)[0],:][0]
    return mat0, mat1
def createTree(dataSet, leafType=regLeaf, errType=regErr, ops=(1,4)):
    feat, val = chooseBestSplit(dataSet, leafType, errType, ops)
    if feat == None:
        return val
    retTree={}
    retTree['spInd'] = feat
    retTree['spVal'] = val
    lSet, rSet = binSplitDataSet(dataSet, feat, val)
    retTree['left']=createTree(lSet, leafType, errType, ops)
    retTree['right']=createTree(rSet, leafType, errType, ops)
    return retTree
def isTree(obj):
    return (type(obj).__name__=='dict')
def getMean(tree):
    if isTree(tree['right']):
        tree['right'] = getMean(tree['right'])
    if isTree(tree['left']):
        tree['left'] = getMean(tree['left'])
    return (tree['left']+tree['right'])/2.0
def prune(tree, testData):
    if shape(testData)[0] == 0:
        return getMean(tree)
    if(isTree(tree['right']) or isTree(tree['left'])):
        lSet, rSet = binSplitDataSet(testData, tree['spInd'],tree['spVal'])
    if isTree(tree['left']):
        tree['left']=prune(tree['left'],lSet)
    if isTree(tree['right']):
        tree['right']=prune(tree['right'],rSet)
    if not isTree(tree['right']) and not isTree(tree['left']):
        lSet, rSet = binSplitDataSet(testData, tree['spInd'], tree['spVal'])
        errorNoMerge = sum(power(lSet[:,-1]-tree['left'],2))+\
                       sum(power(rSet[:,-1]-tree['right'],2))
        treeMean = (tree['left']+tree['right'])/2.0
        errorMerge = sum(power(testData[:,-1]-treeMean,2))
        if errorMerge < errorNoMerge:
            print "Merging"
            return treeMean
        else:
            return tree
    else:
        return tree
def treeForeCast(tree, inData, modelEval=regTreeEval):
    if not isTree(tree):
        return modelEval(tree, inData)
    if inData[tree['spInd']]>tree['spVal']:
        if isTree(tree['left']):
            return treeForeCast(tree['left'], inData, modelEval)
        else:
            return modelEval(tree['left'],inData)
    else:
        if isTree(tree['right']):
            return treeForeCast(tree['right'], inData, modelEval)
        else:
            return modelEval(tree['right'], inData)
def createForeCast(tree, testData, modelEval=regTreeEval):
    m=len(testData)
    yHat = mat(zeros((m,1)))
    for i in range(m):
```

```
        yHat[i,0]=treeForeCast(tree, mat(testData[i]), modelEval)
    return yHat
'''''
myData2 = loadDataSet(r"ex2.txt")
myMat2 = mat(myData2)
tree2 = createTree(myMat2, ops=(0,1))
print tree2
myData2Test = loadDataSet(r"ex2test.txt")
myMat2Test = mat(myData2Test)
print prune(tree2, myMat2Test)
'''
trainMat = mat(loadDataSet('bikeSpeedVsIq_train.txt'))
testMat = mat(loadDataSet('bikeSpeedVsIq_test.txt'))
myregTree=createTree(trainMat, ops=(1,20))
mymodTree=createTree(trainMat, modelLeaf, modelErr, (1,20))
yregHat=createForeCast(myregTree, testMat[:,0])
ymodHat=createForeCast(mymodTree, testMat[:,0], modelTreeEval)
regCo = corrcoef(yregHat, testMat[:,1], rowvar=0)[0,1]
modCo = corrcoef(ymodHat, testMat[:,1], rowvar=0)[0,1]
print "reg", regCo
print "model", modCo
```

上一篇 机器学习经典算法7-线性回归
下一篇 机器学习经典算法9-k-means

顶 0 踩 0

我的同类文章

统计分析与数据挖掘（16）

- R的基本使用(1)
- 关于机器学习课程的小记
- 机器学习经典算法3-朴素贝叶斯
- 机器学习经典算法4-logistic回归
- 机器学习经典算法7-线性回归
- 利用matlab进行简单的贝叶斯网络构建
- 机器学习经典算法2-决策树
- 机器学习经典算法5-支持向量机SVM
- 机器学习经典算法6-AdaBoost

更多

主题推荐 机器学习 算法

猪你在找

有趣的算法（数据结构）
数据结构和算法
Winform数据库编程:ADO.NET入门
《C语言/C++学习指南》加密解密篇（安全相关算法）
Java经典算法讲解
机器学习经典算法详解及Python实现--线性回归
机器学习经典算法8-树回归
机器学习经典算法详解及Python实现--CART分类决
机器学习经典算法7-线性回归
机器学习经典算法详解及Python实现---Logistic回


小米笔记本

app外包

小米笔记本电脑

手游回合制游戏

超薄笔记本

查看评论

3楼 柳枫大人 2015-11-18 15:48发表

第52行应该是有问题的，dataSet里面的元素是matrix类型的，无法用set

Re: rururur 2015-11-26 11:17发表

回复柳枫大人：那你是怎么解决的？我也遇到同样的问题了

2楼 ourarewe 2015-11-09 22:44发表

请问楼主用的是什么数据，那个ex00之类又是什么

1楼 upc2whu 2014-03-16 21:36发表

请问楼主程序都运行了吗？为什么我运行有错误。

```
def binSplitDataSet(dataSet, feature, value):
mat0 = dataSet[nonzero(dataSet[:, feature]>value)[0],:][0]
mat1 = dataSet[nonzero(dataSet[:, feature]<=value)[0],:][0]
return mat0, mat1
```

这一段代码中，mat0 和 mat1对应的feature列是不是应该去掉？否则会无限递归下去

Re: rururur 2015-11-26 11:20发表

回复upc2whu：我感觉这块也有错，他目的是想把dataSet分成两部分

Re: abnormal_zzb 2014-12-17 19:43发表

回复upc2whu：你运行是什么错误？

您还没有登录,请[登录]或[注册]

* 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

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker
OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC
WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML
LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra
CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App
SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP
HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 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 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved