"练习一万小时成天才！" - 摘自《异类》
昵称：bourneli
园龄：4年1个月
粉丝：75
关注：9
+加关注

< 2013年3月 >

日	一	二	三	四	五	六
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索
谷歌搜索

最新随笔
1. Spark随机深林扩展—OOB错误评估和变量权重
2. Spark随机森林实现学习
3. RDD分区2GB限制
4. Spark使用总结与分享
5. Spark核心—RDD初探
6. 机器学习技法--学习笔记04--Soft SVM
7. 机器学习技法--学习笔记03--Kernel技巧
8. 机器学习基石--学习笔记02--Hard Dual SVM
9. 机器学习基石--学习笔记01--linear hard SVM
10. 特征工程(Feature Enginnering)学习记要

我的标签
coursera(4)
C/C++(3)

决策树学习笔记整理

本文目的

最近一段时间在Coursera上学习Data Analysis，里面有个assignment涉及到了决策树，所以参考了一些决策树方面的资料，现在将学习过程的笔记整理记录于此，作为备忘。

算法原理

决策树（Decision Tree）是一种简单但是广泛使用的分类器。通过训练数据构建决策树，可以高效的对未知的数据进行分类。决策数有两大优点：1）决策树模型可以读性好，具有描述性，有助于人工分析；2）效率高，决策树只需要一次构建，反复使用，每一次预测的最大计算次数不超过决策树的深度。

如何预测

先看看下面的数据表格：

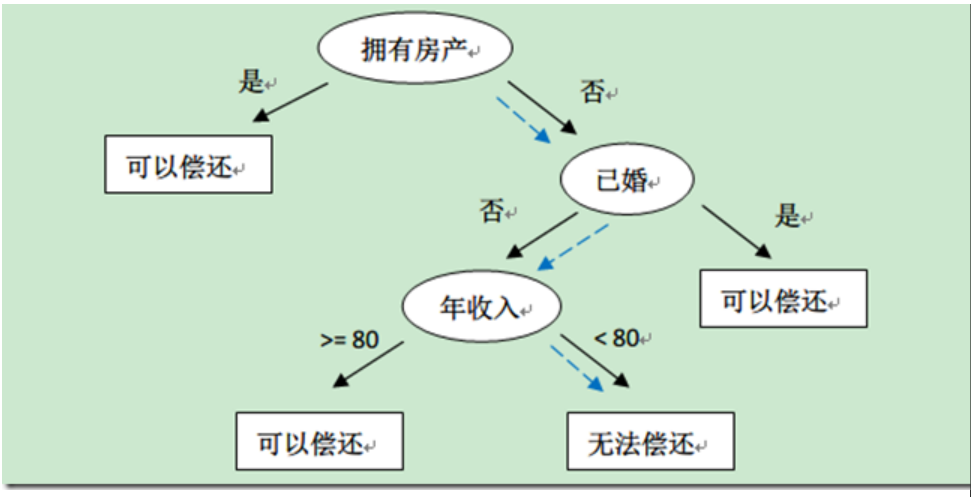
ID	拥有房产（是/否）	婚姻情况（单身，已婚，离婚）	年收入（单位：千元）	无法偿还债务（是/否）
1	是	单身	125	否
2	否	已婚	100	否
3	否	单身	70	否
4	是	已婚	120	否
5	否	离婚	95	是
6	否	已婚	60	否
7	是	离婚	220	否
8	否	单身	85	是
9	否	已婚	75	否
10	否	单身	90	是

上表根据历史数据，记录已有的用户是否可以偿还债务，以及相关的信息。通过该数据，构建的决策树如下：

决策树(2)
data analysis(2)
dlopen(1)
jsoncpp(1)
k fold(1)
MapReduce(1)
MOOC(1)
singleton(1)
更多

随笔分类(92)
C/C++(4)
R(7)
Web前端开发(26)
大数据(7)
机器学习(6)
数据分析&挖掘(42)

随笔档案(119)
2015年5月 (2)
2015年4月 (2)
2015年3月 (1)
2015年1月 (4)
2014年11月 (1)
2014年9月 (1)
2014年8月 (1)
2014年4月 (1)
2014年3月 (1)
2013年10月 (1)
2013年9月 (1)
2013年8月 (3)
2013年7月 (1)
2013年6月 (1)
2013年4月 (3)



比如新来一个用户：无房产，单身，年收入55K，那么根据上面的决策树，可以预测他无法偿还债务（蓝色虚线路径）。从上面的决策树，还可以知道是否拥有房产可以很大的决定用户是否可以偿还债务，对借贷业务具有指导意义。

基本步骤

决策树构建的基本步骤如下：

1. 开始，所有记录看作一个节点

2. 遍历每个变量的每一种分割方式，找到最好的分割点

3. 分割成两个节点N_1和N_2

4. 对N_1和N_2分别继续执行2-3步，直到每个节点足够"纯"为止

决策树的变量可以有两种：

1） 数字型（Numeric）：变量类型是整数或浮点数，如前面例子中的"年收入"。用">="，">","<"或"<="作为分割条件（排序后，利用已有的分割情况，可以优化分割算法的时间复杂度）。

2） 名称型（Nominal）：类似编程语言中的枚举类型，变量只能重有限的选项中选取，比如前面例子中的"婚姻情况"，只能是"单身"，"已婚"或"离婚"。使用"="来分割。

如何评估分割点的好坏？如果一个分割点可以将当前的所有节点分为两类，使得每一类都很"纯"，也就是同一类的记录较多，那么就是一个好分割点。比如上面的例子，"拥有房产"，可以将记录分成了两类，"是"的节点全部都可以偿还债务，非常"纯"；"否"的节点，可以偿还贷款和无法偿还贷款的人都有，不是很"纯"，但是两个节点加起来的纯度之和与原始节点的纯度之差最大，所以按照这种方法分割。构建决策树采用贪心算法，只考虑当前纯度差最大的情况作为分割点。

量化纯度

前面讲到，决策树是根据"纯度"来构建的，如何量化纯度呢？这里介绍三种纯度计算方法。如果记录被分为n类，每一类的比例P(i)=第i类的数目/总数目。还是拿上面的例子，10个数据中可以偿还债务的记录比例为P(1) = 7/10 = 0.7，无法偿还的为P(2) = 3/10 = 0.3，N = 2。

Gini不纯度

$$Gini = 1 - \sum_{i=1}^{n} P(i)^2$$

熵（Entropy）

$$Entropy = -\sum_{i=1}^{n} P(i) * \log_2^{P(i)}$$

错误率

$$Error = 1 - \max\{p(i) | i \text{ in } [1, n$$

上面的三个公式均是值越大，表示越"不纯"，越小表示越"纯"。三种公式只需要取一种即可，实践证明

2013年3月 (4)
2013年2月 (1)
2013年1月 (4)
2012年12月 (4)
2012年11月 (17)
2012年10月 (12)
2012年9月 (10)
2012年8月 (5)
2012年7月 (4)
2012年6月 (3)
2012年5月 (7)
2012年4月 (10)
2012年3月 (1)
2012年2月 (3)
2012年1月 (4)
2011年12月 (6)

文章分类(23)

apache
C/C++
gbk(1)
gtest&gmock
js(2)
LAMP(2)
Linux(4)
mysql(3)
php(2)
shell(4)
单元测试
多进程(1)
工作感悟(1)
开源软件
设计模式

三种公司的选择对最终分类准确率的影响并不大，一般使用熵公式。

纯度差，也称为信息增益（Information Gain），公式如下：

$$\Delta = I(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} * I(v_j)$$

其中，I代表不纯度（也就是上面三个公式的任意一种），K代表分割的节点数，一般K = 2。v_j表示子节点中的记录数目。上面公式实际上就是当前节点的不纯度减去子节点不纯度的加权平均数，权重由子节点记录数与当前节点记录数的比例决定。

停止条件

决策树的构建过程是一个递归的过程，所以需要确定停止条件，否则过程将不会结束。一种最直观的方式是当每个子节点只有一种类型的记录时停止，但是这样往往会使得树的节点过多，导致过拟合问题（Overfitting）。另一种可行的方法是当前节点中的记录数低于一个最小的阀值，那么就停止分割，将max(P(i))对应的分类作为当前叶节点的分类。

过渡拟合

采用上面算法生成的决策树在事件中往往会导致过滤拟合。也就是该决策树对训练数据可以得到很低的错误率，但是运用到测试数据上却得到非常高的错误率。过渡拟合的原因有以下几点：

- 噪音数据：训练数据中存在噪音数据，决策树的某些节点有噪音数据作为分割标准，导致决策树无法代表真实数据。
- 缺少代表性数据：训练数据没有包含所有具有代表性的数据，导致某一类数据无法很好的匹配，这一点可以通过观察混淆矩阵（Confusion Matrix）分析得出。
- 多重比较（Mulitple Comparition）：举个列子，股票分析师预测股票涨或跌。假设分析师都是靠随机猜测，也就是他们正确的概率是0.5。每一个人预测10次，那么预测正确的次数在8次或8次以上的概率为 $(C_{10}^{8} + C_{10}^{9} + C_{10}^{10})/2^{10} = 0.0547$，只有5%左右，比较低。但是如果50个分析师，每个人预测10次，选择至少一个人得到8次或以上的人作为代表，那么概率为 $1 - (1 - 0.0547)^{50} = 0.9399$，概率十分大，随着分析师人数的增加，概率无限接近1。但是，选出来的分析师其实是打酱油的，他对未来的预测不能做任何保证。上面这个例子就是多重比较。这一情况和决策树选取分割点类似，需要在每个变量的每一个值中选取一个作为分割的代表，所以选出一个噪音分割标准的概率是很大的。

优化方案**1**：修剪枝叶

决策树过渡拟合往往是因为太过"茂盛"，也就是节点过多，所以需要裁剪（Prune Tree）枝叶。裁剪枝叶的策略对决策树正确率的影响很大。主要有两种裁剪策略。

前置裁剪 在构建决策树的过程时，提前停止。那么，会将切分节点的条件设置的很苛刻，导致决策树很短小。结果就是决策树无法达到最优。实践证明这中策略无法得到较好的结果。

后置裁剪 决策树构建好后，然后才开始裁剪。采用两种方法：1）用单一叶节点代替整个子树，叶节点的分类采用子树中最主要的分类；2）将一个字数完全替代另外一颗子树。后置裁剪有个问题就是计算效率，有些节点计算后就被裁剪了，导致有点浪费。

优化方案**2**：**K-Fold Cross Validation**

首先计算出整体的决策树T，叶节点个数记作N，设i属于[1,N]。对每个i，使用K-Fold Validataion方法计算决策树，并裁剪到i个节点，计算错误率，最后求出平均错误率。这样可以用具有最小错误率对应的i作为最终决策树的大小，对原始决策树进行裁剪，得到最优决策树。

优化方案**3**：**Random Forest**

Random Forest是用训练数据随机的计算出许多决策树，形成了一个森林。然后用这个森林对未知数据进行预测，选取投票最多的分类。实践证明，此算法的错误率得到了经一步的降低。这种方法背后的原理可以用"三个臭皮匠定一个诸葛亮"这句谚语来概括。一颗树预测正确的概率可能不高，但是集体预测正确的概率却很高。

数据库事务

字符编码(3)

文章档案(17)

2013年8月 (1)

2013年1月 (1)

2012年10月 (2)

2012年9月 (2)

2012年8月 (3)

2012年7月 (5)

2012年6月 (2)

2012年4月 (1)

友情链接

R博客（英文）

R中文网

TOWER -- 思想，智慧，人生

DM，Hadoop

数据科学与R语言

统计之都

统计学中文论坛

小虫织网

郑纪blog

积分与排名

积分 - 155984

排名 - 1049

最新评论

1. Re:Spark使用总结与分享

楼主，spark sql 通过自定义bean来实现创建表，需要序列化，而这个序列化应该是十分消耗内存的，有没有什么方法解决。

--骁枫

2. Re:MySQL全文检索初探

@conqweal是分词，英文可以按

准确率估计

决策树T构建好后，需要估计预测准确率。直观说明，比如N条测试数据，X预测正确的记录数，那么可以估计acc = X/N为T的准确率。但是，这样不是很科学。因为我们是通过样本估计的准确率，很有可能存在偏差。所以，比较科学的方法是估计一个准确率的区间，这里就要用到统计学中的置信区间（Confidence Interval）。

设T的准确率p是一个客观存在的值，X的概率分布为X ~ B(N,p)，即X遵循概率为p，次数为N的二项分布（Binomial Distribution），期望E(X) = N*p，方差Var(X) = N*p*(1-p)。由于当N很大时，二项分布可以近似有正太分布（Normal Distribution）计算，一般N会很大，所以X ~ N(np,n*p*(1-p))。可以算出，acc = X/N的期望E(acc) = E(X/N) = E(X)/N = p，方差Var(acc) = Var(X/N) = Var(X) / N^2 = p*(1-p) / N，所以acc ~ N(p,p*(1-p)/N)。这样，就可以通过正太分布的置信区间的计算方式计算执行区间了。

正太分布的置信区间求解如下：

1） 将acc标准化，即 $z = (acc - p) / \sqrt{p*(1-p)/N}$

2） 选择置信水平α= 95%，或其他值，这取决于你需要对这个区间有多自信。一般来说，α越大，区间越大。

3） 求出 α/2和1-α/2对应的标准正太分布的统计量 $Z_{\alpha/2}$和$Z_{1-\alpha/2}$（均为常量）。然后解下面关于p的不等式。acc可以有样本估计得出。即可以得到关于p的执行区间

$$-Z_{\alpha/2} <= (acc - p) / \sqrt{p*(1-p)/N} <= Z_{1-\alpha/2}$$

参考资料

[1] 《数据挖掘导论》Chapter 4 Classification:Basic Concepts, Decision Trees, and Model Evaluation，Pang-Ning Tan & Micheal Steinbach & Vipin Kumar著

[2] Data Analyis, Lectures in Week 6,7 at Coursera

[3] 《集体智慧编程》Chapter 7 Modeling with Decision Tree，Toby Segaran著

[4] 《Head First Statistics》 Chapter 12 置信区间的构造, Dawn Griffiths 著

分类：数据分析&挖掘

标签： 决策树

bourneli
关注 - 9
粉丝 - 75
+加关注

6 0

(请您对文章做出评价)

« 上一篇：假设检验的学习和理解
» 下一篇：C++STL学习之algorithm库函数

posted @ 2013-03-15 15:44 bourneli 阅读(42554) 评论(1) 编辑 收藏

评论列表

#1楼 2015-09-01 12:29 李可以

受教了，但是我用你的例子，用《机器学习实战》里面的决策树算法，得出的决策树和你的不一样哦。

支持(0) 反对(0)

空格分词，中文就比较麻烦了...

--抠脚大汉

3. Re:kmeans聚类理论篇

请问附件在哪里呢？谢谢

--fanfan123

4. Re:Linux上使用gtest

亲测可行！

--www点elesos点com

5. Re:数据挖掘学习08 - 实验：使用R评估kmeans聚类的最优K

谢谢博主！
这个方法挺好的，回头也找这本书看看
不过# 遍历计算kmeans的SSE这里，pam的运行速度好慢，是kmeans的好几倍。。。

--尾巴AR

阅读排行榜

1. 决策树学习笔记整理(42553)
2. kmeans聚类理论篇(10540)
3. Django静态文件配置(9543)
4. PHP判断键值数组是否存在，使用empty或isset或array_key_exists(8462)
5. Apache alias目录配置(7544)

评论排行榜

1. PHP多进程处理并行处理任务实例(5)
2. 数据挖掘学习05 - 使用R对文本进行hierarchical cluster并验证结果(4)
3. Linux上使用gtest(4)
4. PHPUnit学习05---Mock使用进阶(3)
5. MySQL全文检索初探(3)

推荐排行榜

1. 决策树学习笔记整理(6)
2. 5分钟回忆正则表达式(3)

最新知识库文章:
· Git协作流程
· 企业计算的终结
· 软件开发的核心
· Linux概念架构的理解
· 从涂鸦到发布——理解API的设计过程
» 更多知识库文章...