**决策树算法**

***一、算法描述***

每次只根据一个特征进行分类，再在子类中在根据某一特征进行分类直到该类中所有样本均属同一类，递归遍历完所有分之。

**难点**：

**1：每次分类时如何选择特征?**

使信息增益最大。

**2：每次分类时，如果选择的特征不止0，1两个值，如何确定所有可取的特征值？是否需要归一化处理或是对特征值进行分段取值？如过不归一化或分段取值处理，是否需要根据每个特征值都进行分类？特征值选择顺序？**

A：把训练样本按矩阵形式进行存储，取某一特征对应的列，遍历序列，把所有可取的特征值作为字典的健存储；

B：是否需要归一化处理现在还想不清楚，但是我觉得可以进行分段取值，比如说分成三段，取每段的平均值为特征值，这样判别时的条件为不等式，可以减少判别次数；

C：特征值选择顺序是否对应上述字典中按值降序排列对应的键；（可通过信息增益验证，或需要分类的次数）

D：是不是需要对每个特征值进行分类与该特征对整体分类的影响有关。

**3：参考《Machine Learning in Action》中的代码，每次分类时，得到的子类中就不含用来判别的特征(page43)，此处想不通两个问题：**

**(1)如何记录特征选择的顺序**，很麻烦,没有找到规律，但是通过构造特殊的训练数据实现该目的。

取训练样本为lxm。

n (1,2,3,...,m-1)为分类次数；

a 记录每次用来判别的特征在没进行分类之前在样本中的位置；

b 记录每一次分类时所用特征在此次样本（每分类一次，样本维度减小一维）中的位置；

c 每次分类时所用的特征值。

举例：若m=20，第4次分类时的特征向量为17维，此时若选第3个特征来判断，则b4=2,a4则根据具体情况判定。

以下是构造的测试算法，可以记录10次分类的特征取值顺序。

import random,matplotlib.pyplot as plt

dataSet = [[0,1,2,3,4,5,6,7,8,9,'a'],

[10,11,12,13,14,15,16,17,18,19,'b'],

[20,21,22,23,24,25,26,27,28,29,'c'],

[30,31,32,33,34,35,36,37,38,39,'d'],

[40,41,42,43,44,45,46,47,48,49,'a'],

[50,51,52,53,54,55,56,57,58,59,'b'],

[60,61,62,63,64,65,66,67,68,69,'c'],

[70,71,72,73,74,75,76,77,78,79,'d'],

[80,81,82,83,84,85,86,87,88,89,'a'],

[90,91,92,93,94,95,96,97,98,99,'b']]

#标签列可有可无

#行大于等于列数相等是为了保证在第10次分类时，还有列表不为空

#不同样本的同一特征的值个位相同且都为未分类时的索引

def splitData(dataSet,axis,value):

leftDataSet = []

for featVec in dataSet:

if featVec[axis] != value:

reduceFeatVec = featVec[:axis]

reduceFeatVec.extend(featVec[axis+1:])

leftDataSet.append(reduceFeatVec)

return leftDataSet

def recurse(dataSet,n):

a = []

b=[]

c=[]

while n >= 0 :

axis = random.randint(0,n)

value = dataSet[0][axis]

c.append(value)

a.append(value%10)

temResult = splitData(dataSet,axis,value)

dataSet = temResult

b.append(axis)

n -= 1

return b,c,a

result = recurse(dataSet,9)

print result

plt.plot(result[2],result[0])

plt.xlabel('a')

plt.ylabel('b')

plt.show()

得到结果如下：

b = [4, 6, 4, 1, 2, 2, 0, 0, 1, 0]

a = [4, 7, 5, 1, 3, 6, 0, 2, 9, 8])

c = [4, 17, 25, 31, 43, 56, 60, 72, 89, 98]

**(2)如何判断子集不再需要分类？**

A：子集中所有样本标签一致

B：子集中样本的特征数量为1

C：以上两个条件取或

**4:《Machine Learning in Action》的作者选择每次分类时，在子集中剔除用来判别的特征，是否是出于运行速率的考虑？**

通过对比不剔出和剔出特征的执行速度比较。

**5:如果两个样本特征向量相同，但标签不一致，算一类还是多类?**

***二、代码实现***

dataSet = [[1,1,'maybe'],

[1,1,'yes'],

[1,0,'yes'],

[0,1,'maybe'],

[0,0,'no']]

1:按第二个特征进行分类

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | maybe | yes | no | sum |
| overal | 2 | 2 | 1 | 5 |
| axis[1]=1 | 1 | 1 | 1 | 3 |
| axis[1]=0 | 0 | 1 | 1 | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| axis[0]=1 | 1 | 2 | 0 | 3 |
| axis[0]=0 | 1 | 0 | 1 | 2 |

整体熵：E(s)=

为Script