决策树实验报告

201250205 郭慕凡

决策树实验报告

1.数据的分析和处理

1.1 数据集分析

1.2 数据集处理

1.2.1 用pandas库读取csv文件

1.2.2 数据清洗

2.设计原理和核心代码

2.1 决策树算法：ID3

2.2 具体代码

2.3 改进：由于药物类别属性取值很多，故对该属性使用增益比计算

3. 验证集评估结果

4.存在问题反思

1.数据的分析和处理

1.1 数据集分析

csv数据集中包含有关药品的各种属性：

- recordID：药品编号
- drugName：药品名称
- condition：使用情况，可以理解为药品类别
- reviewComment：评论
- date：可能是药品上市时间
- usefulCount：有效评价人数
- sideEffect：副作用，分为No Side Effect，Mild Side Effect，Moderate Side Effect，Severe Side Effect和Extremely Side Effect这从低到高的五级
- rating：评级，也是我们要学习和预测的属性。

根据以上对药品属性的分析，我们可以得出数据清洗的标准：

1. 药品编号，药品名称与药品质量无关，上市时间这条属性很多药品没有，也是脏数据，且也与药品质量无关
2. reviewComment数据无法利用（除非进行文本情感分析），因此去除
3. usefulCount为评价人数，虽然与药物质量无关，但也间接反映市场反响或者适用性以及rating的可信度，保留。**视作连续数据**
4. condition与sideEffect与rating核心相关，保留。**视作离散数据**

1.2 数据集处理

1.2.1 用pandas库读取csv文件

```
training_data = pd.read_csv("./dataset/training.csv")
validation_data = pd.read_csv("./dataset/validation.csv")
test_data = pd.read_csv("./dataset/testing.csv")
```

1.2.2 数据清洗

1. 去掉不需要的列
2. 将字符串数据转换为整数数据（非必要）

```python
def pre_processing(data):
    # 清洗掉不需要的列
    data = data.drop(columns=['recordId', 'drugName', 'reviewComment', 'date'])
    # 将字符串转成数字，便于后续处理
    condition_dict = {k: i for i, k in enumerate(data['condition'].unique())}
    data['condition'] = data['condition'].map(condition_dict)
    sideeffect_dict = {k: i for i, k in enumerate(data['sideEffects'].unique())}
    data['sideEffects'] = data['sideEffects'].map(sideeffect_dict)
    data['rating'] = data['rating'].astype(int)
    # 分开attributes和label
    attributes = data[['condition', 'usefulCount', 'sideEffects']].values.tolist()
    labels = data['rating'].values.tolist()

    return attributes, labels
```

2.设计原理和核心代码

2.1 决策树算法：ID3

使用信息熵增量作为选择最优划分的依据。

训练步骤可以概括为：

1. 生成节点
2. 计算信息熵，如果为0，说明是叶节点，将node标记为该类别节点
3. 利用信息增益选择最优划分属性
4. 递归生成子节点（子节点的最优划分属性与父节点不同）

2.2 具体代码

说明：对于连续属性usefulCount(col = 2),我将其离散化为区间：[0], [0,5], [5,10], [10, 20], [20, 50], [50, 100], [100,)

类的定义：

```python
class DecisionNode:
    def __init__(self, col=-1, results=None, children=None):
        self.col = col           # 切分属性的列索引
        self.results = results   # 叶节点中的分类结果
        self.children = children  # 子节点
```

数据划分：

```
# 对数据集按照属性列col进行分类
def divideSet(rows, col, value):
    splittingFunction = None
    if col == 1:  # 处理数值型数据
        splittingFunction = lambda row: row[col] >= value
    else:  # 处理字符串型数据
        splittingFunction = lambda row: row[col] == value
    # 将数据集划分成两个集合，并返回
    set1 = [row for row in rows if splittingFunction(row)]
    set2 = [row for row in rows if not splittingFunction(row)]
    return (set1, set2)
```

计算信息熵：（使用匿名函数lambda表达式）

```
# 计算熵
def entropy(rows):
    log2 = lambda x: math.log(x) / math.log(2)
    results = uniqueCounts(rows)
    ent = 0.0
    for r in results.keys():
        p = float(results[r]) / len(rows)
        ent -= p * log2(p)
    return ent
```

构建树的代码较多，故在实验报告中略去，详见提交的代码。总体上与2.1所述步骤相同

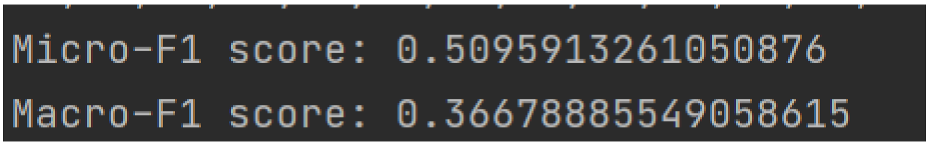
预测（递归）：

```
def classify(observation, tree):
    if tree.results != None: # 叶节点
        return tree.results
    else:
        v = observation[tree.col]
        branch = None
        if tree.col == 1:
            if v >= 100:
                branch = tree.children[100]
            elif v >= 50:
                branch = tree.children[50]
            elif v >= 20:
                branch = tree.children[20]
            elif v >= 10:
                branch = tree.children[10]
            elif v >= 5:
                branch = tree.children[5]
            elif v >= 0:
                branch = tree.children[0]
        else:
            try:
                branch = tree.children[v]
            except:
                return default_class
        return classify(observation, branch)
```

2.3 改进：由于药物类别属性取值很多，故对该属性使用增益比计算

```
"""更新增益律的值"""
    log2 = lambda x: math.log(x) / math.log(2)
    tmp = uniqueCounts(rows, 0)
    for key in tmp.keys():
        p = tmp[key]/len(rows)
        IV[0] += -p * log2(p)
```

3. 验证集评估结果



```
Micro-F1 score: 0.5095913261050876
Macro-F1 score: 0.36678885549058615
```

4.存在问题反思

可见，以上的结果不能称之为优秀，我经过debug后发现可能原因有一下几点：

1. 抛弃Comment带来的问题，比如下面这行：

188604	ParaGard	Birth Control	"I had paragard for a year and a half and had cramping the whole time with terrible long heavy periods. The last couple months I had it I was in so much pain I could hardly walk. Went to my doc and she said nothing seemed wrong. Then the next day I went to the ER and had it removed. It had slipped a little and was causing a lot of discomfort. I will never get an IUD again. I'm lucky I didn't need surgery like some women"	#######	4	No Side Effects	1

No sideeffects, 最终药物评级只有1 这是令人疑惑的

然而翻译一下Comment:

> "I had paragard for a year and a half and had cramping the whole time with terrible long heavy periods. The last couple months I had it I was in so much pain I could hardly walk. Went to my doc and she said nothing seemed wrong. Then the next day I went to the ER and had it removed. It had slipped a little and was causing a lot of discomfort. I will never get an IUD again. I'm lucky I didn't need surgery like some women"

> 我曾经使用Paragard长达一年半时间，期间一直有痉挛和严重的长期大量月经。最后几个月我疼得几乎无法行走。去看医生，但她说一切都正常。然后第二天我去了急诊室，并将其拆除。它稍微滑动了一点，引起了很多不适。我再也不会使用宫内节育器了。我很幸运没有像一些女性那样需要手术。

可见，该医用器械的反响并不好，然而如果只用副作用去衡量，显然不能得到药物评级为1的结果

2. 个人认为部分数据标注比较迷惑，因为在使用副作用进行划分后，信息熵竟然会增加，即信息增益为负，这是令人疑惑的。
3. 个人代码可能存在一定的错误，但是在逻辑上没有debug出来