C4.5决策树生成算法示例程序

本程序演示了如何使用Hadoop来实现C4.5决策树生成算法。

注意：本程序只演示了原始决策树的生成算法实现，剪枝过程需要感兴趣的读者自己实现。另外，本程序目前只能处理离散属性，对于连续属性，还请读者自己修改程序完成。算法实现的基本思路和原理已经在书中得到详细阐述，请读者同时参考书中的原理介绍。算法的正确性在Car数据集和C4.5参考实现程序上验证过。在使用C4.5程序验证时，通过它输出的未剪枝的树的调试信息来比照，确定MapReduce输出的树结构和C4.5生成的是一致的。

文件组织结构

```
src - 源代码目录
  datatype - 数据结构
     NodeStatisticInfo.java 节点统计信息
     Rule.java 决策树节点规则描述
     StatisticRecord.java 一条统计信息
  program - 主要程序
     DecisionTreeDriver.java 驱动类
     DecisionTreeMapper.java Mapper类
     DecisionTreeReducer.java Reducer类
test-data - 书上的例子数据
     names 属性值域信息
     data  训练集数据
     model 示例模型输出
run.sh - 示例运行脚本
```

使用方法

编译

将src下面的代码导入Eclipse或NetBeans等IDE中（或者编写Makefile等），将Hadoop相关库加入工程的依赖条件中进行编译，打包成dtree.jar这样的JAR包即可。

运行

将数据准备好（格式等见数据准备部分），然后以如下命令运行程序：

```
hadoop jar dtree.jar program.DecisionTreeDriver metaFile dataSetPath workingDir modelFile
```

参数含义如下：

- metaFile 记录属性值域信息的元文件的路径，
- dataSetPath 数据集文件（夹）所在路径，
- workingDir 临时的工作空间文件夹，
- modelFile 模型文件的输出路径

数据准备

输入数据

训练集元文件 包含了各个属性的值域信息，文件的每一行代表一个属性的信息。每个属性的各个可能取值，使用逗号分隔。例如在书上的例子中，第一个属性是年龄Age信息。那么在元文件的第一行内容为：

```
 youth,middle,senior
```

注意 每一行的信息中不包括类标签，逗号左右不能有空格。

最后一个属性将被认为是类标签。示例文件请见`test-data/names`。

训练集文件（夹） 描述了参与训练的数据。训练集文件（夹）即可以是单个HDFS上的文件，也可以是HDFS上的一个文件夹。对于文件夹，程序会处理其中所有的文件。训练文件的每一行都代表了一个元组（或者说是训练样本），该元组中的各个字段使用`','`分隔。比如以书上给出的示例训练集为例，其第一个元组在文件中的表示如下：

```
youth,high,no,fair,no
```

它表示了该元组的5个属性（age、income、student、credit、buy）的取值依次是youth、high、no、fair、no。其中最后一个

属性buy是类标签。

注意 逗号左右不能有空格。示例文件见`test-data/data`。

程序输出

临时的工作空间文件夹 程序运行中会产生一系列的临时HDFS文件，都将放与于该文件夹下。运行前删除上一次运行时产生的文件夹内容，以确保该文件夹在运行前不存在。

模型文件 程序将把决策树模型输出为一组规则集。规则集中的每任意一条规则都对应于原决策树中的一个叶子节点。通过将元组与规则进行匹配，就知道该元组最终会落入哪个叶子节点。C4.5决策树生成算法本身的性质保证了规则集的一致性，即对于任意一条有效的元组，其最多符合规则集中的一条规则，并且至少满足规则集中的一条规则。在文件中，每一条规则占据一行，其具有如下的形式：

<规则>:类标签

其中“规则”部分的结构是：**属性ID，属性取值&属性ID2，属性取值2&...**，比如

`1,youth&3,yes:yes`

就表示了(age == youth) && (student == yes) -> buy = yes 这条规则。

注意 分隔符左右不能有空格。示例文件见`test-data/model`。