数据挖掘—ID3算法Matlab实现

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temperature | Humidity | Wind | Playtennis |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

首先，对表格中的状态进行“数值化”，具体修改如下：

Outlook：Sunny-0;Overcast-1;rain-2

Temperature:hot-2,midd-1,cool-2

Humidity:high-0,nomal-1

Wind:week-0,strong-1

Playtennis:no-0,yes-1

源文件

【数据预处理】

clc;

clear all;

close all;

%% 导入数据

data = [0,2,0,0,0;

0,2,0,1,0;

1,2,0,0,1;

2,1,0,0,1;

2,0,1,0,1;

2,0,1,1,0;

1,0,1,1,1;

0,1,0,0,0;

0,0,1,0,1;

2,1,1,0,1;

0,1,1,1,1;

1,1,0,1,1;

1,2,1,0,1;

2,1,0,1,0];

【决策树生成】

%% 生成决策树

make\_tree(data);

function make\_tree(train\_data)

%input train\_data 训练数据

%output

[m,n] = size(train\_data);

disp('original data');

disp(train\_data);

class\_list = train\_data(:,n);

class\_first = 1;

for i = 2:m

if train\_data(i,n) == class\_list(1,:)

% if strcmp(train\_data(i,n),class\_list(1,:))

class\_first = class\_first + 1;

end

end

%% 退出程序条件

if class\_first == m || n == 1

disp('final data');

disp(train\_data);

return;

end

%% 建立决策树

bestfeat = choose\_bestfeat(train\_data);

disp(['bestfeature:',num2str(bestfeat)]);

featvalue = unique(train\_data(:,bestfeat));

featvalue\_num = length(featvalue);

for i = 1:featvalue\_num

make\_tree(splitData(train\_data,bestfeat,featvalue(i,:)));

disp('--------------------------------------------');

end

end

【splitData.m】

function [subSet] = splitData(data, j, value)

subSet = data;

subSet(:,j) = [];

k = 0;

for i = 1:size(data,1)

if data(i,j) ~= value

subSet(i-k,:) =[];

k = k + 1;

end

end

【calc\_entropy.m】

function [entropy] = calc\_entropy(train\_data)

%input train\_data 训练数据

%output entropy 熵值

[m,n] = size(train\_data);

%% 得到类的项并统计每个类的个数

label\_value = train\_data(:,n);

label = unique(label\_value);

label\_number = zeros(length(label),2);

label\_number(:,1) = label';

for i = 1:length(label)

label\_number(i,2) = sum(label\_value == label(i));

end

%% 计算熵值

label\_number (:,2) = label\_number(:,2) ./ m;

entropy = 0;

entropy = sum(-label\_number(:,2).\*log2 (label\_number(:,2)));

end

【choose\_bestfeat.m】

function [best\_feature] = choose\_bestfeat(data)

%input data 输入数据

%output bestfeature 选择特征值

[m,n] = size(data);

feature\_num = n - 1;

baseentropy = calc\_entropy(data);

best\_gain = 0;

best\_feature = 0;

%% 挑选最佳特征位

for j =1:feature\_num

feature\_temp = unique(data(:,j));

num\_f = length(feature\_temp);

new\_entropy = 0;

for i = 1:num\_f

subSet = splitData(data, j, feature\_temp(i,:));

[m\_s,n\_s] = size(subSet);

prob = m\_s./m;

new\_entropy = new\_entropy + prob \* calc\_entropy(subSet);

end

inf\_gain = baseentropy - new\_entropy;

if inf\_gain > best\_gain

best\_gain = inf\_gain;

best\_feature = j;

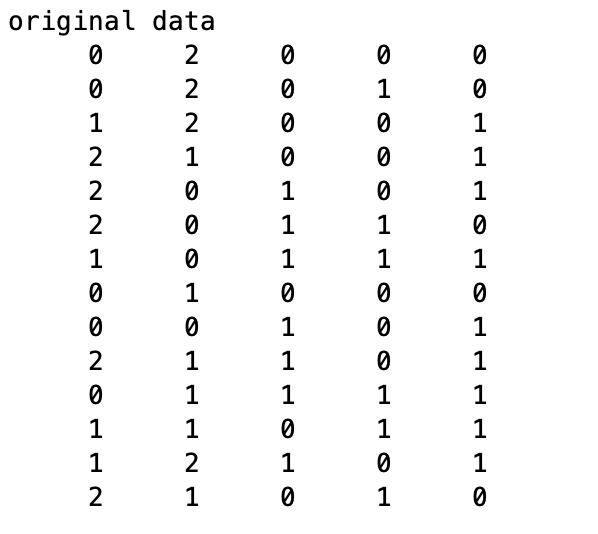
end

end

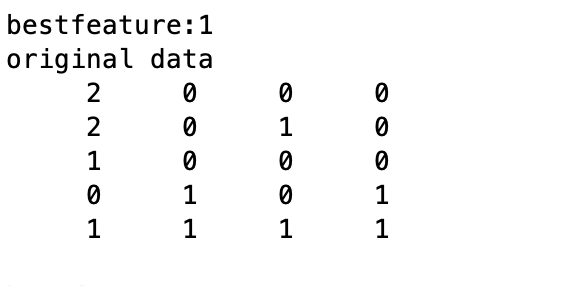
End

输出结果

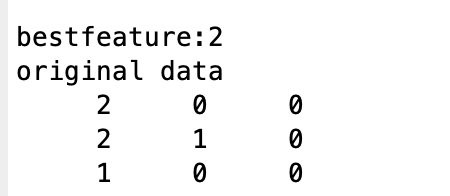
初始数据如下图：



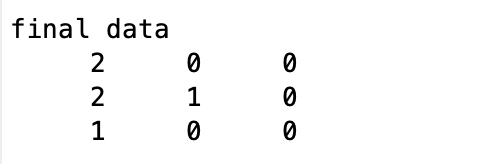
得到第一个属性为根节点，以第一个属性的第一个标签划分子数据，如下图所示



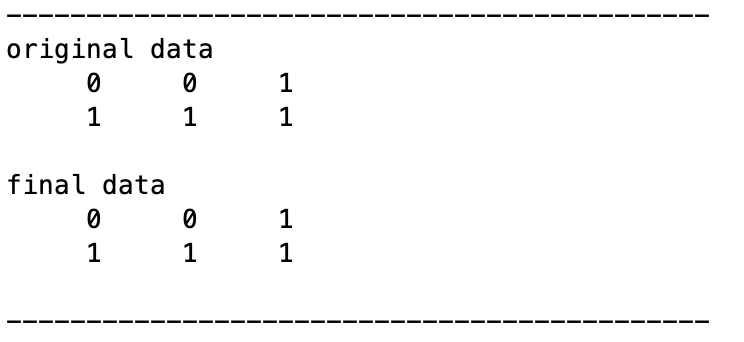
第二个属性为第一个属性的子节点，如下图所示



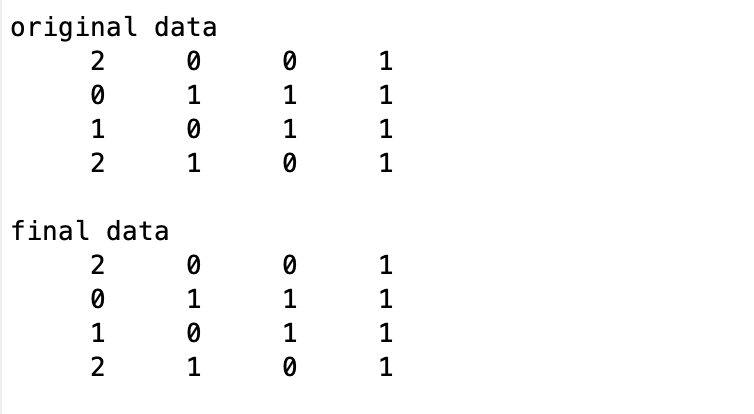
由于类别中只有一种标签，类别为0，分类结束，属性2为叶子结点



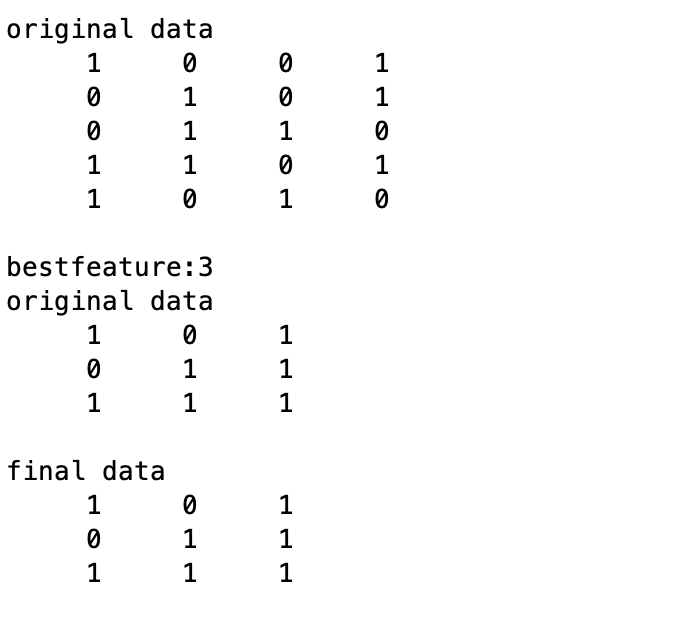
该支路结束，该支路对应属性1的第一个标签（0），该支路仅有一个标签，分类为1，即Yes，分类结束。



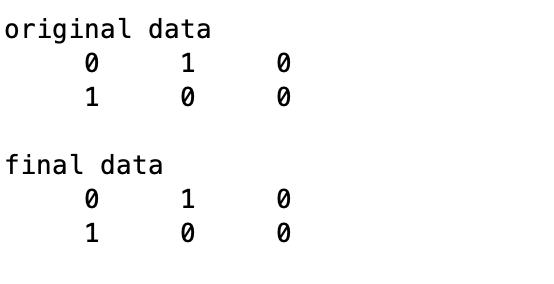
以第一个属性的第二个标签划分子数据



以第一个属性的第三个标签划分子数据



该支路仅有一个标签，分类为0，分类结束



最终得到的决策树描绘如下：

