山东大学 计算机科学与技术 学院

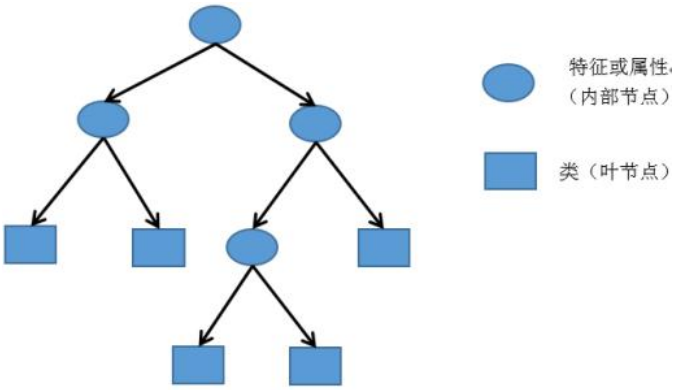
机器学习与模式识别 课程实验报告

学号：201600301304	姓名：贾乘兴	班级： 人工智能 16
实验题目：决策树		
实验学时：2 小时	实验日期： 2018/11/16	
实验目的：在给定的数据集下，实现连续数值下的决策树，解决二分类问题，并可视化决策树		
硬件环境： 16 GB 内存		
软件环境：mac os，matlab 2017b		

实验步骤与内容：

一. 决策树算法及其 matlab 实现

1. 决策树是十分常见的分类算法，其结构包括为根节点、若干的内部节点和若干个叶结点，根节点与内部节点都对应了一个属性测试，根节点是所有分类标准的出发点，将样本集合根据该属性分类到不同的分支，而内部节点也是具有相同的作用，而叶结点在分类问题中对应着最终的类别决策



2. 但是如何合理的选择属性的顺序是决策树构建的关键，我们基于的想法是，经过该属性分类后，各类里的混乱程度最小，所以我们使用信息熵代表各类的混乱程度，信息熵的定义如下

$$Ent(D) = -\sum_{k=1}^{|y|} p_k \log_2 p_k$$

在定义了信息熵之后，我们定义了信息增益来描述在经过该分类后信息的纯度，信息增益越大，我们得到的新的分类的纯度越高，信息增益的定义如下

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

3. 但是为了防止出现类别过多等问题导致出现过拟合等问题，我们用信息增益率来描述，定义了固有值如下

$$IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

得到的信息增益率为

$$Gain\_ratio(D,a)=\frac{Gain(D,a)}{IV(a)}$$

这样可以有效避免过拟合，但本次实验为二分类二标签问题，故并没有太大的作用，类似的指数还有基尼指数，本次实验选取了信息增益率作为标准

4. 预剪枝与后剪枝

剪枝是解决过拟合的较好方法，在学习过程中将一些“冗余”的分支去除，从而得到泛化更高的正确率，即测试集更高的正确率，预剪枝是在训练过程中的剪枝，我们可以基于阈值，基于显著性检验，基于信息增益的大小来进行判断；后剪枝是基于生成了决策树后，通过比较验证集的精度提升来进行剪枝，本次实验实现了基于阈值的预剪枝

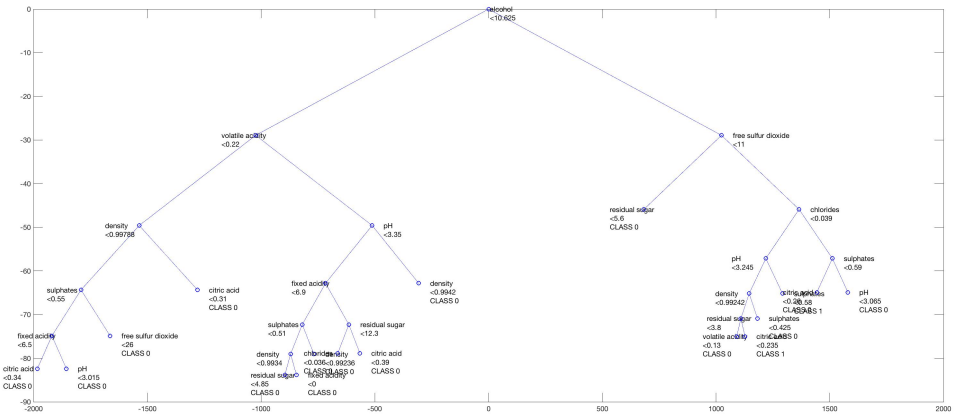
5. 连续值 t 值的确定，在连续值的情况下，我们将确定一个分割点 t，分为大于 t 和小于 t 的两部分，t 的确定为，取最大的信息增益，而这一点一般在间断值处取得

二．K 折交叉验证

将数据集随机分为 k 份，每次留出一份作为验证集，经过 k 次训练后，充分利用数据，得到的结果取平均值进行评估

三．决策树可视化

最后对得到的决策树进行可视化



可见经过预剪枝的决策树深度等较为合适

四．结果展示与分析

最终得到十次的正确率如下

0. 8078　0. 8041　0. 8020　0. 7898　0. 8057

0. 7878　0. 7898　0. 7939　0. 8020　0. 7796

平均后得到的正确率为0. 7962

五．各部分代码功能说明

1. 信息增益计算

```
%% 信息增益

function [g,class,th,index,rate]=gain(D,M)

m=length(M);

n=length(D);

pos=length(M(M(:,D(n))==1))/m;

neg=length(M(M(:,D(n))==0))/m;

Ent=-(pos*log(pos)+neg*log(neg));
```

```
%
for i=1:n-1
    pos=D(i);
    Ms=sortrows(M,pos);
    Enti(pos)=1;
    for j=1:m-1
        if(Ms(j,n)~=Ms(j+1,n))
            Ms1=[];
            Ms1=Ms(1:j,:);
            pos1=length(Ms1(Ms1(:,D(n))==1))/(j);
            neg1=length(Ms1(Ms1(:,D(n))==0))/(j);
            Ent1=-(pos1*log(pos1)+neg1*log(neg1));
            Ms2=[];
            Ms2=Ms(j+1:m,:);
            pos2=length(Ms2(Ms2(:,D(n))==1))/(m-j);
            neg2=length(Ms2(Ms2(:,D(n))==0))/(m-j);
            Ent2=-(pos2*log(pos2)+neg2*log(neg2));
            e=(j/m)*Ent1+(1-j/m)*Ent2;
            if e<Enti(pos)
                Enti(pos)=e;
                t(pos)=(Ms(j,pos)+Ms(j+1,pos))/2;
                y(pos)=j;
            end
        end
    end
end
gs=1;
class=1;
index=1;
th=0;
for i=1:n-1
    pos=D(i);
    if Enti(pos)<gs
        gs=Enti(pos);
        class=pos;
        index=y(pos);
        th=t(pos);
    end
end
g=Ent-gs;
rate=g/Ent;
end
```

2. 生成树

```
while(j~=length(t))
        j=j+1;
```

```
        if length(t(j).num)==0
            t(j).num=0;
            continue;
        elseif(t(j).num~=0&&t(j).rate>=p)
            Mt=t(j).M;
            Mi=t(j).index;
            Md=t(j).D;
            if(Mi>number&&Mi<length(Mt)-number)
                Ml=Mt(1:Mi,:);
                t(j*2)=buildtree(Md,Ml);
                Mr=Mt(Mi+1:length(Mt),:);
                t(j*2+1)=buildtree(Md,Mr);
            else
                t(j*2).num=0;
                t(j*2+1).num=0;
            end
        elseif(t(j).num~=0&&t(j).rate<p)
            t(j*2).num=0;
            t(j*2+1).num=0;
        end
    end

%% build tree
function [tree]=buildtree(D,M)
[g,class,th,index,rate]=gain(D,M);
M0=sortrows(M,class);
D0=setdiff(D,[class]);

tree.num=class;
tree.threshold=th;
tree.rate=rate;
tree.gain=g;
tree.D=D0;
tree.M=M0;
tree.index=index;
end
```

3. 绘制树图像

```
%% plot tree
function []=plotree(t)
l=length(t);
px(1)=0;
py(1)=0;
plot(px(1),py(1),'bo');
text(px(1)+5,py(1),L(t(1).num));
text(px(1)+5,py(1)-2,"<"+num2str(t(1).threshold));
```

```
for k=2:l
    if t(k).num~=0
        fm=floor(k/2);
        if(mod(k,2)==0)
            px(k)=px(fm)-2^(10-log2(fm));
            py(k)=py(fm)-1.4^(10-log2(fm));
            text(px(k)-150,py(k),L(t(k).num));
            text(px(k)-150,py(k)-2,"<"+num2str(t(k).threshold));
            if length(t(k).label)~=0
                text(px(k)-150,py(k)-4,"CLASS "+num2str(t(k).label));
            end
        else
            px(k)=px(fm)+2^(10-log2(fm));
            py(k)=py(fm)-1.4^(10-log2(fm));
            text(px(k)+50,py(k),L(t(k).num));
            text(px(k)+50,py(k)-2,"<"+num2str(t(k).threshold));
            if length(t(k).label)~=0
                text(px(k)+50,py(k)-4,"CLASS "+num2str(t(k).label));
            end
        end
        hold on
        plot(px(k),py(k),'bo');
        hold on;
        plot([px(fm):(px(k)-px(fm))/10000:px(k)],[py(fm):(py(k)-py(fm))/10000:py(k)],'b-');
    end
end
end
```

4. k 折交叉验证

```
data=crossvalind('Kfold',m,k);


    itest=(data==i);
    itrain=(data~=i);
    dtest=M(itest,:);
    dtrain=M(itrain,:);
```

结论分析与体会： 通过本次实验，更加了解了决策树的生成，同时对编程语言的使用更加灵活

附录：程序源代码

ex6.m

```
clear,clc;
L={'fixed acidity','volatile acidity','citric acid','residual
sugar','chlorides','free sulfur dioxide','total sulfur
dioxide','density','pH','sulphates','alcohol','quality'};
M=csvread('ex6Data.csv',1);
m=length(M);
k=10;
N=length(L);
data=crossvalind('Kfold',m,k);
p=0.0001;
number=100;

for i=1:k
    itest=(data==i);
    itrain=(data~=i);
    dtest=M(itest,:);
    dtrain=M(itrain,:);
    D0=[1:N];
    M0=dtrain;
    j=0;
    clear t;
    t(1)=buildtree(D0,M0);
    while(j~=length(t))
        j=j+1;
        if length(t(j).num)==0
            t(j).num=0;
            continue;
        elseif(t(j).num~=0&&t(j).rate>=p)
            Mt=t(j).M;
            Mi=t(j).index;
            Md=t(j).D;
            if(Mi>number&&Mi<length(Mt)-number)
                Ml=Mt(1:Mi,:);
                t(j*2)=buildtree(Md,Ml);
                Mr=Mt(Mi+1:length(Mt),:);
                t(j*2+1)=buildtree(Md,Mr);
            else
                t(j*2).num=0;
                t(j*2+1).num=0;
            end
        elseif(t(j).num~=0&&t(j).rate<p)
            t(j*2).num=0;
            t(j*2+1).num=0;
```

```
        end
    end
    for k=1:j
        if(t(k).num~=0&&k*2<=j&&t(k*2).num==0)
            Ml=t(k).M;
            pos=length(Ml(Ml(:,N)==1));
            neg=length(Ml(Ml(:,N)==0));
            if(pos>neg)
                t(k).label=1;
            else
                t(k).label=0;
            end
        end
    end
    a=length(dtest);
    sum1=0;
    for k=1:a
        g=1;
        while(t(g*2).num~=0)
            num=t(g).num;
            thres=t(g).threshold;
            if dtest(k,num)<=thres
                g=g*2;
            else
                g=g*2+1;
            end
        end
        if t(g).label==dtest(k,N)
            sum1=sum1+1;
        end
    end
    ac(i)=sum1/length(dtest)
end
rac=sum(ac)/k;
plotree(t);

%% plot tree
function []=plotree(t)
l=length(t);
px(1)=0;
py(1)=0;
plot(px(1),py(1),'bo');
text(px(1)+5,py(1),L(t(1).num));
text(px(1)+5,py(1)-2,"<"+num2str(t(1).threshold));
for k=2:l
    if t(k).num~=0
```

```
        fm=floor(k/2);
        if(mod(k,2)==0)
            px(k)=px(fm)-2^(10-log2(fm));
            py(k)=py(fm)-1.4^(10-log2(fm));
            text(px(k)-150,py(k),L(t(k).num));
            text(px(k)-150,py(k)-2,"<"+num2str(t(k).threshold));
            if length(t(k).label)~=0
                text(px(k)-150,py(k)-4,"CLASS "+num2str(t(k).label));
            end
        else
            px(k)=px(fm)+2^(10-log2(fm));
            py(k)=py(fm)-1.4^(10-log2(fm));
            text(px(k)+50,py(k),L(t(k).num));
            text(px(k)+50,py(k)-2,"<"+num2str(t(k).threshold));
            if length(t(k).label)~=0
                text(px(k)+50,py(k)-4,"CLASS "+num2str(t(k).label));
            end
        end
        hold on
        plot(px(k),py(k),'bo');
        hold on;
        plot([px(fm):(px(k)-px(fm))/10000:px(k)],[py(fm):(py(k)-
py(fm))/10000:py(k)],'b-');
    end
end
end

%% build tree
function [tree]=buildtree(D,M)
[g,class,th,index,rate]=gain(D,M);
M0=sortrows(M,class);
D0=setdiff(D,[class]);

tree.num=class;
tree.threshold=th;
tree.rate=rate;
tree.gain=g;
tree.D=D0;
tree.M=M0;
tree.index=index;
end

%% 信息增益
function [g,class,th,index,rate]=gain(D,M)
m=length(M);
n=length(D);
```

```
pos=length(M(M(:,D(n))==1))/m;
neg=length(M(M(:,D(n))==0))/m;
Ent=-(pos*log(pos)+neg*log(neg));
%
for i=1:n-1
    pos=D(i);
    Ms=sortrows(M,pos);
    Enti(pos)=1;
    for j=1:m-1
        if(Ms(j,n)~=Ms(j+1,n))
            Ms1=[];
            Ms1=Ms(1:j,:);
            pos1=length(Ms1(Ms1(:,D(n))==1))/(j);
            neg1=length(Ms1(Ms1(:,D(n))==0))/(j);
            Ent1=-(pos1*log(pos1)+neg1*log(neg1));
            Ms2=[];
            Ms2=Ms(j+1:m,:);
            pos2=length(Ms2(Ms2(:,D(n))==1))/(m-j);
            neg2=length(Ms2(Ms2(:,D(n))==0))/(m-j);
            Ent2=-(pos2*log(pos2)+neg2*log(neg2));
            e=(j/m)*Ent1+(1-j/m)*Ent2;
            if e<Enti(pos)
                Enti(pos)=e;
                t(pos)=(Ms(j,pos)+Ms(j+1,pos))/2;
                y(pos)=j;
            end
        end
    end
end
gs=1;
class=1;
index=1;
th=0;
for i=1:n-1
    pos=D(i);
    if Enti(pos)<gs
        gs=Enti(pos);
        class=pos;
        index=y(pos);
        th=t(pos);
    end
end
g=Ent-gs;
rate=g/Ent;
end
```