数据挖掘大作业

——————————ID3 决策树

学号：02105111

姓名：张旭

一. 决策树算法

Ⅰ.决策树的基本概念

机器学习中，决策树是一个预测模型；它代表的是对象属性值与对象值之间的一种映射关系。树中每个节点表示某个对象，每个分叉路径则代表的某个可能的属性值，而每个叶结点则对应具有上述属性值的子对象。决策树仅有单一输出；若需要多个输出，可以建立独立的决策树以处理不同输出。

从数据产生决策树的机器学习技术叫做决策树学习，通俗说就是决策树。决策树学习也是数据挖掘中一个普通的方法。在这里，每个决策树都表述了一种树型结构，它由它的分支来对该类型的对象依靠属性进行分类。每个决策树可以依靠对源数据库的分割进行数据测试。这个过程可以递归式的对树进行修剪。当不能再进行分割或一个单独的类可以被应用于某一分支时，递归过程就完成了。另外，随机森林分类器将许多决策树结合起来以提升分类的正确率。

决策树同时也可以依靠计算条件概率来构造。决策树如果依靠数学的计算方法可以取得更加理想的效果。

决策树一般可归纳为 2 类：分类与预测。本文着重关于其分类的作用，并以此来构建一个完整的决策树。

Ⅱ.决策树分类器的优点

以此次用的 ID3 算法为例，以此算法产生的决策树分类器具有很多优点：决策树的构造不需要任何领域知识或参数设置，因此适合于探测式知识发现；决策树可以处理高维数据，推理过程完全依赖于属性变量的取值特点，可自动忽略目标变量没有贡献的属性变量，也为判断属性变量的重要性，减少变量的数目提供参考，同时对噪声数据具有很好的健壮性；决策树归纳的学习和分类步骤是简单和快速的，推理过程可以表示成 If Then 形式，并且具有很好的准确率；获取的知识用树的形式表示是直观的，并且容易被人理解。因而，决策树归纳分类是目前应用最广泛的归纳推理算法之一，在数据挖掘中受到研究者的广泛关注。

但是其缺点也是很多的，如：信息增益的计算依赖于特征数目较多的特征，而属性取值最多的属性并不一定最优。ID3 是非递增算法。ID3 是单变量决策树(在分枝节点上只考虑单个属性)，许多复杂概念的表达困难，属性相互关系强调不够，容易导致决策树中子树的重复或有些属性在决策树的某一路径上被检验多次。抗噪性差，训练例子中正例和反例的比例较难控制。

二. ID3 算法

ID3 算法主要针对属性选择问题，是决策树学习方法中最具影响和最为典型的算法。ID3 采用贪心方法，其中决策树以自顶向下递归的分治方式构造。大多数决策树归纳算法都沿用这种自顶向下的方法，从训练元组集和它们的相关联的类标号开始构造决策树。随着树的构建，训练集递归地划分成较小的子集。

ID3 算法中关键的一步是属性选择度量，即选择分裂准则。其中的三种度量方法分别是信息增益、增益率和 Gini 指标。（示例算法选择了第一种方法）。当获取信息时，将不确定的内容转为确定的内容，因此信息伴着不确定性。

算法的基本策略如下：

1.选择一个属性放置在根节点，为每个可能的属性值产生一个分支

2.将样本划分成多个子集，一个子集对应于一个分支
3.在每个分支上递归地重复这个过程，仅使用真正到达这个分支的样本
4.如果在一个节点上的所有样本拥有相同的类别，即停止该部分树的扩展

此次问题在选择属性值时采用启发式标准，其内容为：

只跟本身与其子树有关，采取信息理论用熵来量度。属性选择度量是一种选择分裂准则，将给定的类标记的训练元组的数据划分 D“最好”地分成个体类的启发式方法。如果我们要根据分裂准则的输出将 D 划分成较小的划分，理想地，每个划分是“纯”的，即，落在给定划分的所有元组都属于相同的类。从概念上讲，最好的划分准则是导致最接近这种情况的划分。此次问题采用一种流行的属性选择度量——信息增益。

信息增益度量基于 Claude Shannon 在研究消息的值或“信息内容”的信息论方面的先驱工作。设节点 N 代表或存放划分 D 的元组。选择具有最高信息增益的属性作为节点 N 的分裂属性。该属性使结果划分中的元组分类所需的信息量最小，并反映这些划分中的最小随机性或“不纯性”。这种方法使对给定元组分类所需的期望测试数目最小，并确保找到一棵简单的树。

熵是选择事件时选择自由度的量度，其计算方法为：P=freq(Cj,S)/|S|；Exp(S)=-SUM(P*LOG(P))；SUM()函数是求 j 从 1 到 n 的和。Entropy(X)=SUM((|Ti|/|T|)*Exp(X)；Gain(X)=Exp(X)- Entropy(X)；

为保证生成的决策树最小，ID3 算法在生成子树时，选取使生成的子树的熵(即 Gain(S))最小的特征来生成子树。

三. 实验内容

实验目的：研究糖尿病数据（diabetes 数据集），构造一颗决策树。

实验数据：Title: Pima Indians Diabetes Database

For Each Attribute: (all numeric-valued)

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)

Class Value	Number of instances
0	500
1	268

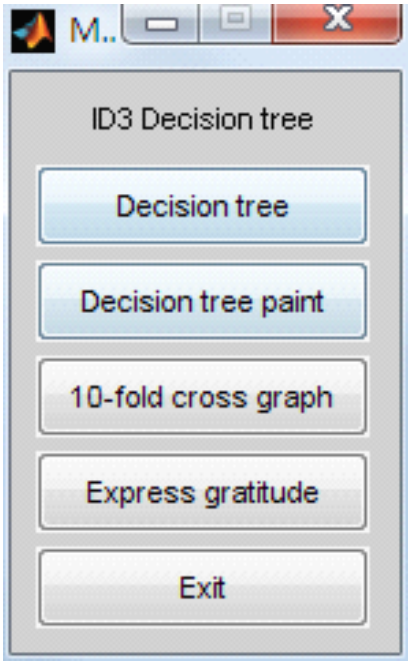
实验代码：

```
%*****************************************
%%目录
%*****************************************

close all
s=menu('ID3 Decision tree','Decision tree','Decision tree paint','10-fold cross
```

```
graph','Express gratitude','Exit');
switch s
    case 1, clc;clear all;close all hidden;decisiontree();IDmenu
    case 2, clc;clear all;close all hidden ;show_tree();IDmenu;
    case 3, clc;clear all;close all hidden ;errorrate();IDmenu;
    case 4, clc;clear all;close all hidden ;disp('Thanks for everyone who helped
me in this programming period');IDmenu;
    case 5, clc;clear all;close all hidden;clc;
    otherwise clc;clear all;close all hidden;disp('Error!')
end

%******************************************
%%构建一个决策树
%******************************************
function decisiontree()
[S1,S2,S3,S4,S5,S6,S7,S8,classity]=textread('train.txt','%f %f %f %f %f %f %f %f
%s');
D=[S1 S2 S3 S4 S5 S6 S7 S8];
AttributName={ 'preg','plas','pres','skin','insu','mass','pedi','age'};
t=classregtree(D,classity,'names',AttributName);
t=prune(t,'level',5);
disp(t);
end

%******************************************
%%绘制一个决策树
%******************************************
function show_tree()
[S1,S2,S3,S4,S5,S6,S7,S8,classity]=textread('train.txt','%f %f %f %f %f %f %f %f
%s');
D=[S1 S2 S3 S4 S5 S6 S7 S8];
AttributName={ 'preg','plas','pres','skin','insu','mass','pedi','age'};
t=classregtree(D,classity,'names',AttributName);
t=prune(t,'level',8);
view(t);
end

%******************************************
%%计算错误率并绘制成曲线
%******************************************
function errorrate()
[S1,S2,S3,S4,S5,S6,S7,S8,classity]=textread('train.txt','%f %f %f %f %f %f %f %f
%s');
D=[S1 S2 S3 S4 S5 S6 S7 S8];
```

```
AttributName={ 'preg','plas','pres','skin','insu','mass','pedi','age'};
t=classregtree(D,classity,'names',AttributName);
t=prune(t,'level',5);costsum=zeros(10,1);
for k=1:10
cost=test(t,'cross',D,classity);
costsum=costsum+cost;
end
costsum=costsum/10;
i=1:10;
plot(i,costsum,'-o');xlabel('交叉次数');ylabel('错误率');
title('决策树 k 倍交叉错误率曲线');
end
```

实验结果：



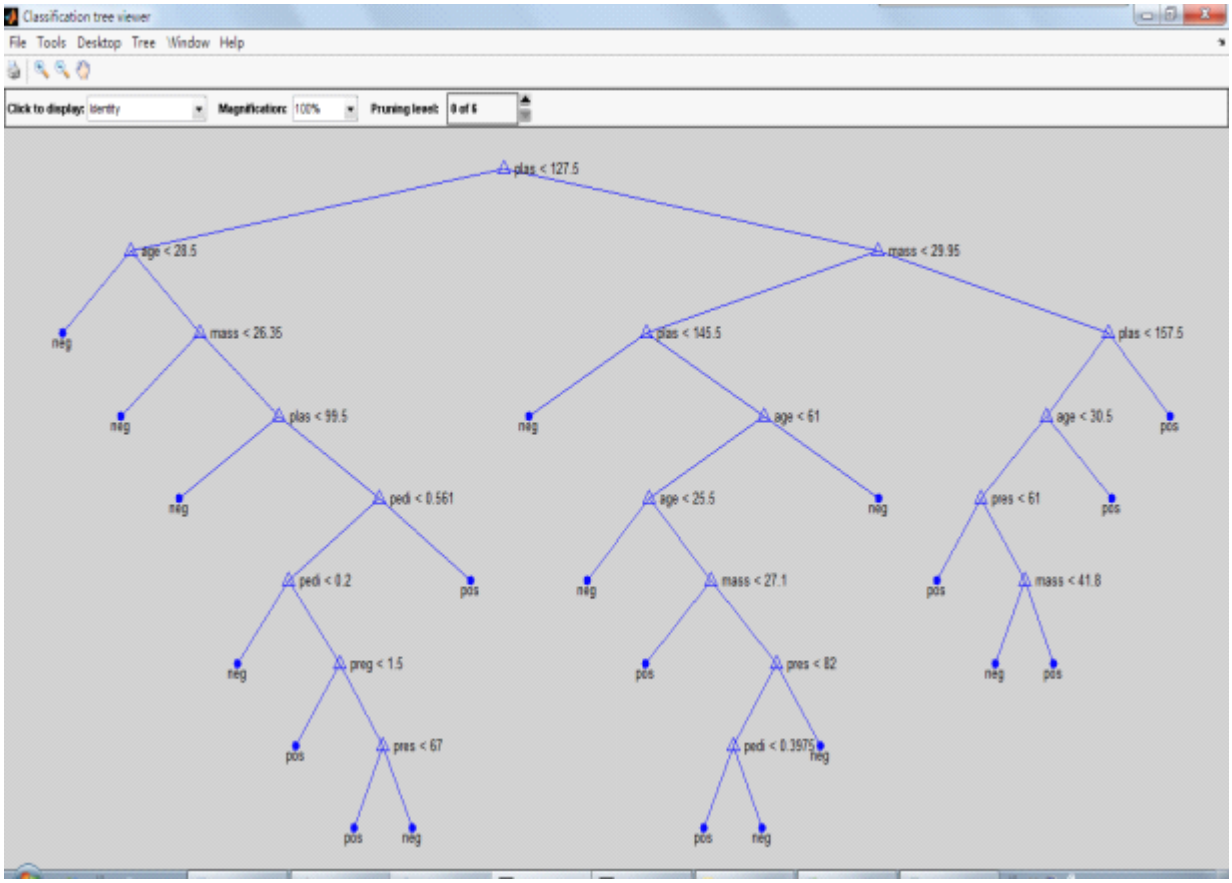
Decsion tree：

```
Decision tree for classification
 1  if plas<127.5 then node 2 else node 3
 2  if age<28.5 then node 4 else node 5
 3  if mass<29.95 then node 6 else node 7
 4  if mass<45.4 then node 8 else node 9
 5  if mass<26.35 then node 10 else node 11
 6  if plas<145.5 then node 12 else node 13
 7  if plas<157.5 then node 14 else node 15
 8  class = neg
 9  class = pos
10  if mass<9.65 then node 16 else node 17
11  if plas<99.5 then node 18 else node 19
12  class = neg
13  if age<61 then node 20 else node 21
```

```
14  if age<30.5 then node 22 else node 23
15  class = pos
16  class = pos
17  class = neg
18  if plas<28.5 then node 24 else node 25
19  if pedi<0.561 then node 26 else node 27
20  if age<25.5 then node 28 else node 29
21  class = neg
22  if pres<61 then node 30 else node 31
23  if pedi<0.4295 then node 32 else node 33
24  class = pos
25  class = neg
26  if pedi<0.2 then node 34 else node 35
27  if preg<6.5 then node 36 else node 37
28  class = neg
29  if mass<27.1 then node 38 else node 39
30  class = pos
31  if mass<41.8 then node 40 else node 41
32  if mass<45.55 then node 42 else node 43
33  class = pos
34  class = neg
35  if preg<1.5 then node 44 else node 45
36  if insu<120.5 then node 46 else node 47
37  class = pos
38  class = pos
39  if pres<82 then node 48 else node 49
40  if pedi<1.1415 then node 50 else node 51
41  class = pos
42  if pres<92 then node 52 else node 53
43  class = pos
44  class = pos
45  if pres<67 then node 54 else node 55
46  if age<34.5 then node 56 else node 57
47  class = pos
48  if pedi<0.3975 then node 58 else node 59
49  class = neg
50  class = neg
51  class = pos
52  if pedi<0.1365 then node 60 else node 61
53  class = pos
54  class = pos
55  if mass<34.45 then node 62 else node 63
56  class = neg
57  class = pos
```

58 class = pos
59 class = neg
60 class = pos
61 class = neg
62 if pres<83 then node 64 else node 65
63 class = neg
64 if plas<120 then node 66 else node 67
65 class = neg
66 class = pos
67 if pedi<0.239 then node 68 else node 69
68 class = pos
69 class = neg

Decision tree paint:

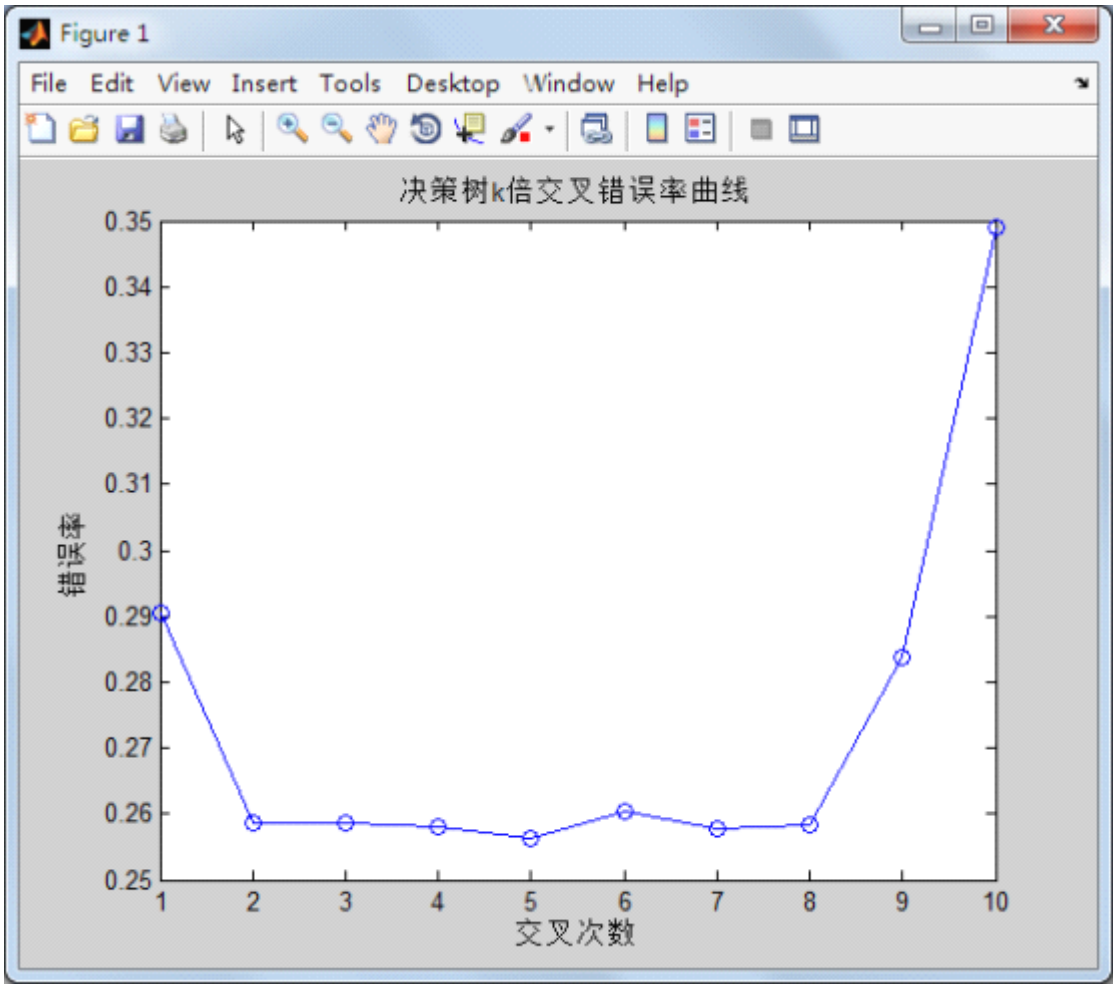


10-fold cross graph:

costsum =

    0.2904
    0.2587
    0.2587
    0.2579
    0.2564
    0.2604
    0.2577

0.2585
0.2837
0.3490



四. 总结与分析

此次 ID3 决策树算法的设计虽然较上次 PCA 算法困难了许多，但是随着对 Matlab 软件的熟悉，已经多方面查询资料，最终还是成功的将其做出，在无数次的失败与总结的过程中，我学会了更多课堂上和课本上无法学到的东西。