ICT4Health Lab 5 - Hierarchical Clustering and Classification

Table of Contents

In this lab, hierarchical clustering is implemented on dataset of chronic kidney disease patients to identify two clusters of patients who either 'have ckd' or 'dont have ckd'. Important features are identified from decision tree obtained after classification.

Data Source: https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease

Data Preparation

```
close all;
clear all;
clc;

% loading data matrix
load('ckd.mat');

% Pre processing data
a = chronickidneydisease;

% changing nominal feature values to meaningful numeric values
keylist={'normal','abnormal','present','notpresent','yes',...
    'no','good','poor','ckd','notckd','?',''};
keymap=[0,1,0,1,0,1,0,1,2,1,NaN,NaN];
[N,F] = size(a);
for i=1:N
    for j=1:F
        c = strtrim(a{i,j});
        check = strcmp(c,keylist);

        if sum(check)==0
            b(i,j)=str2num(a{i,j});% from text to numeric
        else
            ii=find(check==1);
            b(i,j)=keymap(ii);% use the lists
        end
    end
end
dec = b(:,end);
b(:,end)=[]; % Excluding last column
```

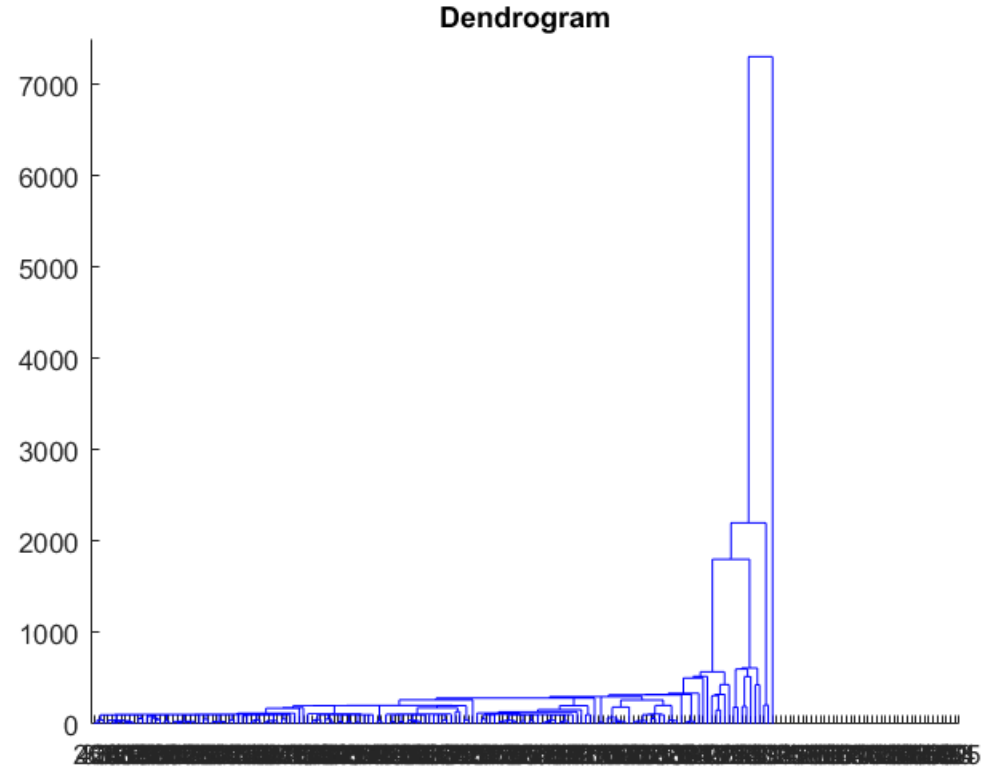
Hierarchical Clustering

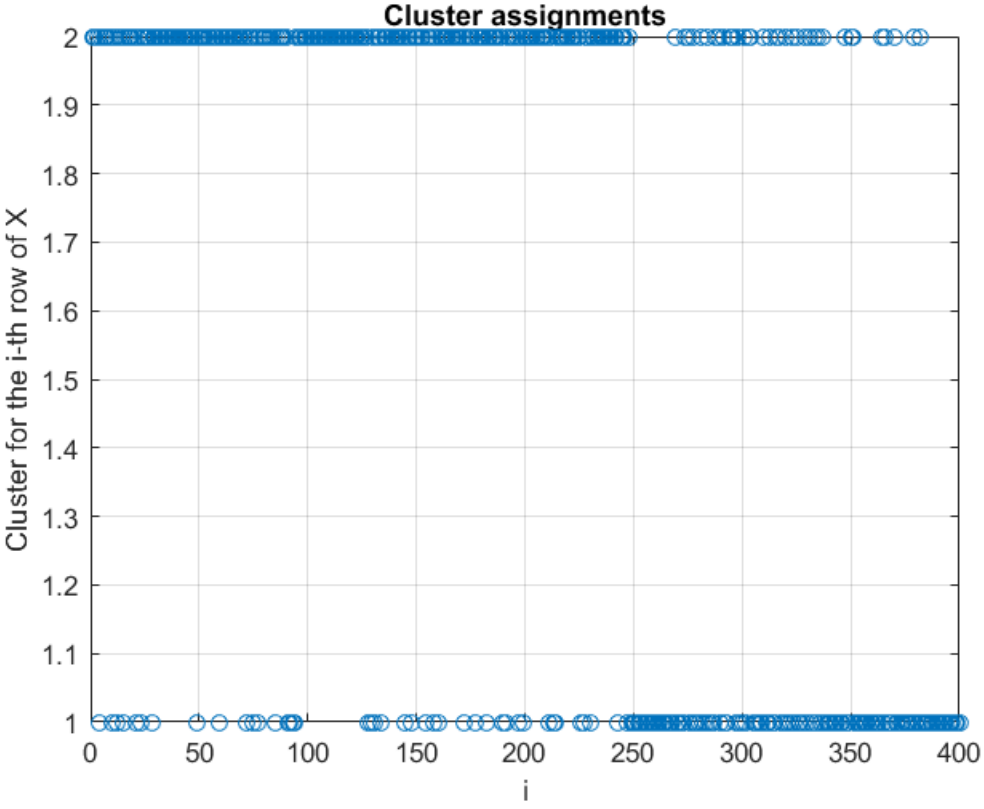
```
d = pdist(b);
s = squareform(d);
Tree = linkage(d);
c = cluster(Tree,'maxclust',2);

figure
p=0;% p=0 means that all the leaves must be included in the plotted
 tree
dendrogram(Tree,p);
%leafOrder = optimalleaforder(Tree,d);
%dendrogram(Tree,'Reorder',leafOrder)
title('Dendrogram')
axis([0 200 0 7500])

figure
nn=[1:N];
plot(nn,c,'o'),grid on
xlabel('i')
ylabel('Cluster for the i-th row of X')
title('Cluster assignments')

% calculating error probabilities
N1=sum(c==1); N2=sum(c==2);
false_pos=sum((dec==2)&(c==1))/N1; %0.2739
true_pos=sum((dec==2)&(c==2))/N2; %0.8519
false_neg=sum((dec==1)&(c==2))/N2; %0.1481
true_neg=sum((dec==1)&(c==1))/N1; %0.7261
```

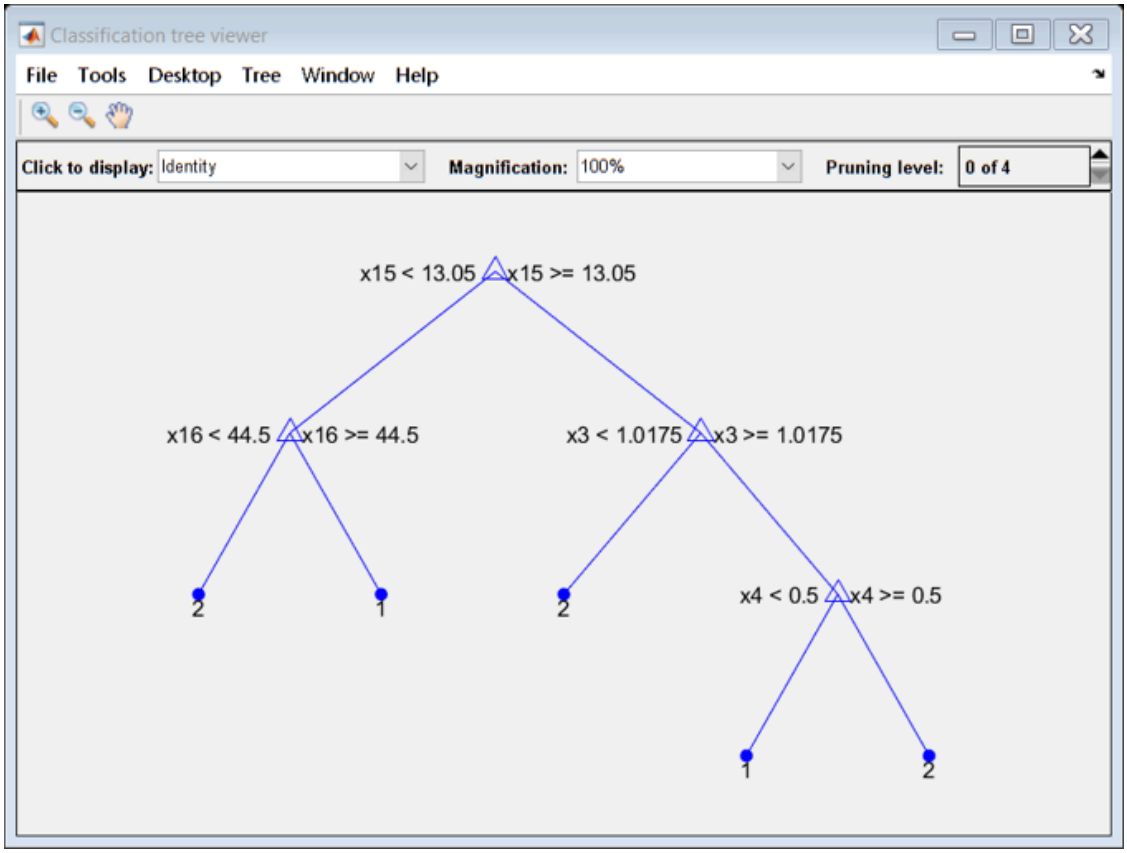

Dendrogram
7000
6000
5000
4000
3000
2000
1000
0


Cluster assignments
Cluster for the i-th row of X
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0
50
100
150
200
250
300
350
400
i

```
clc;
close all;
clear all;
% loading data matrix
load('ckd.mat');
% Pre processing data
a = ckdData1;
% changing nominal feature values to meaningful numeric values
keylist={'normal','abnormal','present','notpresent','yes',...
'no','good','poor','ckd','notckd','?',''};
keymap=[0,1,0,1,0,1,0,1,2,1,NaN,NaN];
[N,F] = size(a);
for i=1:N
for j=1:F
c = strtrim(a{i,j});
check = strcmp(c,keylist);
if sum(check)==0
b(i,j) = str2num(a{i,j});% from text to numeric
else
ii=find(check==1);
b(i,j)=keymap(ii);% use the lists
end
end
end
dec = b(:,end);
b(:,end)=[];

% replacing NaN values with mean values of each column
% bmean = nanmean(b);
% X=b-ones(N,1)*bmean;
% rng('default')
% for i=1:N
% noise = randn(1,F)*0.05;
% indx = isnan(X(i,:));
% X(i,indx) = bmean(indx)+noise(indx);
% end
% diagonalizing matrix
% R=X'*X/N;
% [U,D]=eig(R);
% Y=X*U*sqrt(inv(D));
% obtaining decision tree
tc = fitctree(b,dec);
view(tc,'Mode','graph')
% making new decisions using most important features and theor values
% obtained from decision tree before
c1 = find(((b(:,15) < 13.05)&(b(:,16) < 44.5))
 | ((b(:,15)>=13.05)&(b(:,15)<1.0175)) |
 ((b(:,15)>=13.05)&(b(:,15)>=1.0175)&(b(:,4)>=0.5)));
b(c1,end+1) = 2;
dec_new = b(:,end);
dec_new(dec_new==0) = 1;
% calculating error probabilities
```

```
N1=sum(dec==1); N2=sum(dec==2);
false_pos_new=sum((dec_new==2)&(dec==1))/N1; %0.0
true_pos_new=sum((dec_new==2)&(dec==2))/N2; %0.696
false_neg_new=sum((dec_new==1)&(dec==2))/N2; %0.3040
true_neg_new=sum((dec_new==1)&(dec==1))/N1; %1
```



Published with MATLAB® R2018b