CPS 571 HW2

Shengxin Qian

1. ROC and AUC

y	Linear decision	Nonlinear decision
0	-1.7	0.1545
1	0.7	0.6682
1	3.55	0.9721
1	1.1	0.7503
1	0.65	0.6570
0	-1.65	0.1611
0	-1.55	0.1751
0	0.55	0.6341
1	3.65	0.9747
0	-1.85	0.1359

Chart. 1 decision statistics of two classifiers

(a). The required threshold range is [0.55, 0.65], the corresponding confusion matrix is:

	Predicted yes	Predicted no
Actual yes	5	0
Actual no	0	5

Chart. 2 confusion matrix of classifier g(x)

(b). The required threshold range is [0.6341, 0.6570], we can choose any threshold in this range with no misclassification error. The corresponding information is:

Confusion matrix:

	Predicted yes	Predicted no
Actual yes	5	0
Actual no	0	5

Chart. 3 confusion matrix of classifier f(x)

Precision: 1

Recall: 1

F1 score: 1

(c). As we can see in Figure 1, the AUC = 1

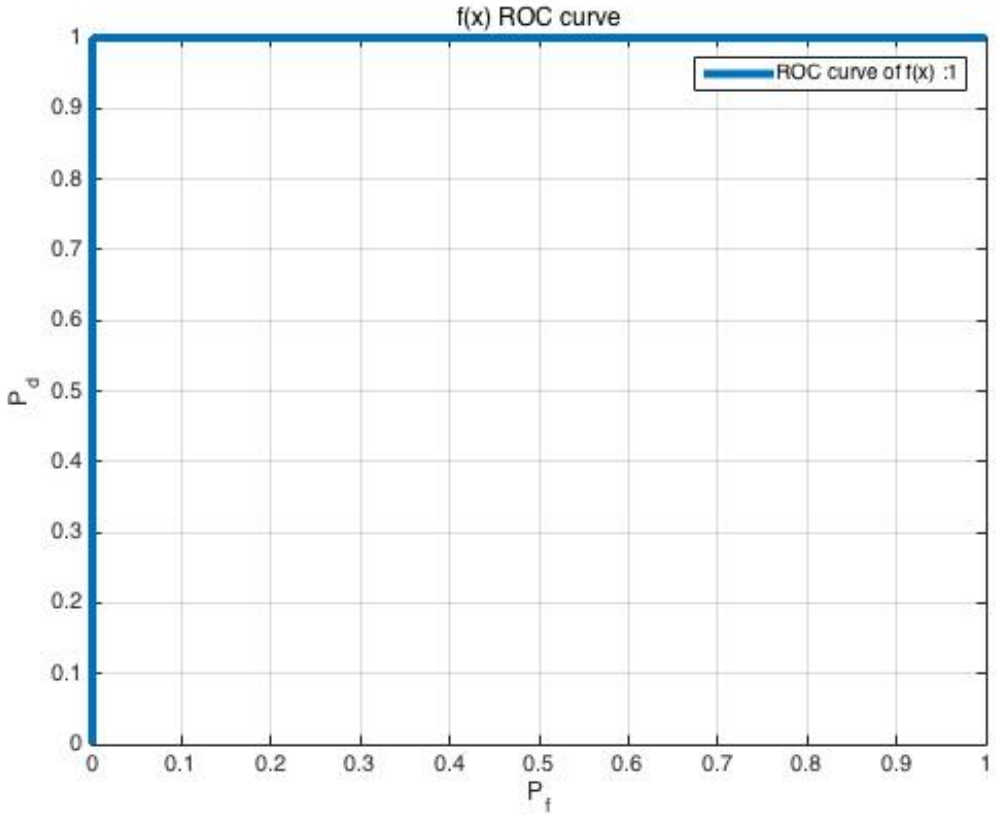

Figure 1. ROC curve of classifier f(x)

(d). There would no change between the ROC curve of f(x) and h(f(x)). Because the order of each confidence will not change. You can always find a threshold to obtain the same separation.

2. Decision Tree

(a). **Initial Gini index**: $2 * \frac{5}{10} * \frac{5}{10} = \frac{1}{2}$

X_1 Gini index: $2 * \frac{4}{10} * 0 * 1 + 2 * \frac{6}{10} * \frac{5}{6} * \frac{1}{6} = \frac{1}{6}$

X_2 Gini index: $2 * \frac{5}{10} * \frac{3}{5} * \frac{2}{5} + 2 * \frac{5}{10} * \frac{3}{5} * \frac{2}{5} = \frac{12}{25}$

X_3 Gini index: $2 * \frac{6}{10} * \frac{4}{6} * \frac{2}{6} + 2 * \frac{4}{10} * \frac{3}{4} * \frac{1}{4} = \frac{5}{12}$

X_1 Gini index reduction: $\frac{1}{2} - \frac{1}{6} = \frac{1}{3}$

X_2 Gini index reduction: $\frac{1}{2} - \frac{12}{25} = \frac{1}{50}$

X_3 Gini index reduction: $\frac{1}{2} - \frac{5}{12} = \frac{1}{12}$

Obviously, we should choose $\mathbf{X_1}$ feature.

(b). Obviously, we should choose $\mathbf{X_1}$ feature.

X_1 Information Gain: $\mathrm{H}(\frac{1}{2}, \frac{1}{2}) - \frac{4}{10} * \mathrm{H}(0,1) - \frac{6}{10} * H\left(\frac{5}{6}, \frac{1}{6}\right) = 0.61$

X_2 Information Gain: $\mathrm{H}(\frac{1}{2}, \frac{1}{2}) - \frac{5}{10} * \mathrm{H}(\frac{3}{5}, \frac{2}{5}) - \frac{5}{10} * H\left(\frac{3}{5}, \frac{2}{5}\right) = 0.029$

X_3 Information Gain: $\mathrm{H}(\frac{1}{2}, \frac{1}{2}) - \frac{6}{10} * \mathrm{H}(\frac{4}{6}, \frac{2}{6}) - \frac{4}{10} * H\left(\frac{3}{4}, \frac{1}{4}\right) = 0.1245$

2.1. Decision Trees

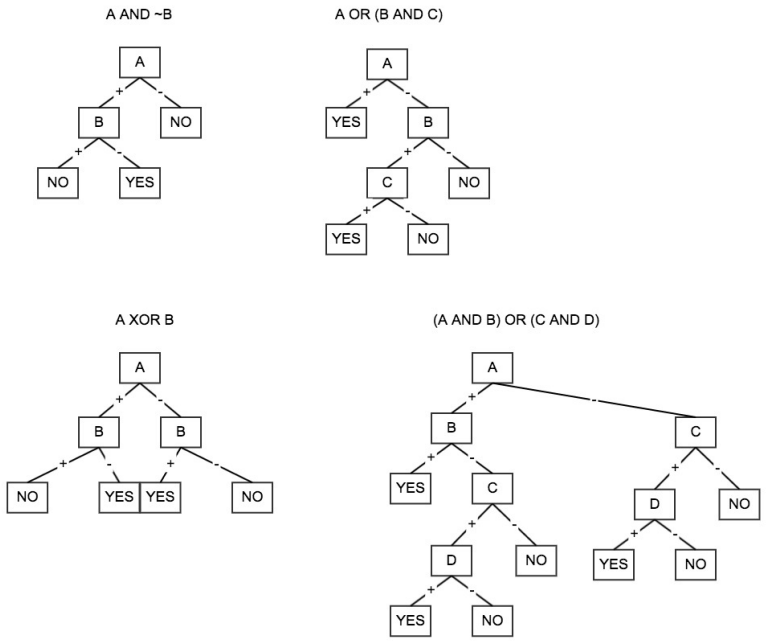

Figure 2. Logical functions Decision Trees

2.2. Decision Trees and Random Forests

(b). Given distribution:

X	[0, 0.45]	[0.45, 0.5]	[0.5, 0.55]	[0.55, 1]
Label	P	N	P	N
Pr of range	0.495	0.005	0.005	0.495
Pdf	Uniform	Uniform	Uniform	Uniform

As we see in the chart, this distribution has four ranges, in each range is uniformly distributed. The probability of each range is significantly different. Because random forests draw a bootstrap sample and use major vote of decision trees in the forest, random forest tend to use 0.5 to do the separation and ignore the outliers. Label the test data with P, when <= 0.5 and label the test data with N, when > 0.5. However, when we use decision tree, as long as n is big enough, we will have a separation between frequent ranges and rare ranges. When n is low, the random forest may not have difference with decision tree because the number in rare ranges will not appear.

Code:

main.m

```
clc
clear
M = csvread('dataset.csv',1,0);
label = M(:, 4);
train = M(:, 1:3);
theta = [0.05 ; -3 ; 2.5];
cons = 0.3;

linear_decision = linear_c(theta, train', cons);
[P_linear, R_linear, F1_linear, Confusion_linear, threshold_linear, threshold_linear_range] = find_t(linear_decision, label);

nonlinear_decision = nonlinear_c(theta, train', cons);
[P_nonlinear, R_nonlinear, F1_nonlinear, Confusion_nonlinear, threshold_nonlinear, threshold_nonlinear_range] = find_t(nonlinear_decision, label);
AUC_nonlinear = plotROC(nonlinear_decision(label == 0), nonlinear_decision(label == 1));
title('f(x) ROC curve');
legend(['ROC curve of f(x) :',num2str(AUC_nonlinear)]);
grid on;
```

linear_c.m

```
function y = linear_c(theta, x, cons)
    y = theta' * x + cons;
    y = y';
end
```

nonlinear_c.m

```
function y = nonlinear_c(theta, x, cons)
	y = 1 ./ (exp(-1 * (theta' * x + cons)) + 1);
	y = y';
end
```

find_t.m

```
function [P, R, F1, Confusion, threshold, threshold_range] = find_t(data, label)
	minError = length(data);
	for i = linspace(min(data), max(data), 2000)
		pe = (length(data(label == 0 & data >= i)) + length(data(label == 1
& data < i)));
		if minError > pe
			threshold = i;
			minError = pe;
		end
	end
	P = length(data(label == 1 & data >= threshold))/(length(data(label == 1
& data >= threshold))+length(data(label == 0 & data >= threshold)));
	R = length(data(label == 1 & data >= threshold))/length(data(label ==
1));
	F1 = 2 * P * R / (P + R);
	Confusion(1,1) = length(data(label == 1 & data >= threshold));
	Confusion(1,2) = length(data(label == 1 & data < threshold));
	Confusion(2,1) = length(data(label == 0 & data >= threshold));
	Confusion(2,2) = length(data(label == 0 & data < threshold));
	t = data(data <= threshold);
	threshold_range(1) = max(t);
	t = data(data >= threshold);
	threshold_range(2) = min(t);
end
```

plotROC.m

```
function AUC=plotROC(H0_1,H1_1)
    index=1;
    min0=min(H0_1);
    min1=min(H1_1);
    max0=max(H0_1);
    max1=max(H1_1);
    mean0=mean(H0_1);
    mean1=mean(H1_1);
    if(min0>=0)
        min0=min0*0.8;
    else
        min0=min0*1.2;
    end

    if(min1>=0)
        min1=min1*0.8;
    else
        min1=min1*1.2;
    end

    if(max0>=0)
        max0=max0*1.2;
    else
        max0=max0*0.8;
    end

    if(max1>=0)
        max1=max1*1.2;
    else
        max1=max1*0.8;
    end
    for beta=min(min0,min1):0.001:max(max0,max1)
        if(mean1>=mean0)
```

```
            Pd_1(index)=sum(H1_1>=beta)/length(H1_1);
            Pf_1(index)=sum(H0_1>=beta)/length(H0_1);
        else
            Pd_1(index)=sum(H1_1<=beta)/length(H1_1);
            Pf_1(index)=sum(H0_1<=beta)/length(H0_1);
        end
        index=index+1;
end
if(mean1>=mean0)
        AUC=-trapz(Pf_1,Pd_1);
else
        AUC=trapz(Pf_1,Pd_1);
end

figure;
p=plot(Pf_1',Pd_1','LineWidth',4);
xlabel('P_f');
ylabel('P_d');
```