Machine Learning Homework 1

Bodong Zhang

u0949206

1 Decision Trees

1 (a) (x1∨x2) ∧x3

```
if x3=0,
then (x1∨x2) ∧x3=0,
else    {
        if x1=1,
        then (x1∨x2) ∧x3=1,
        else    {
                if x2=1,
                then (x1∨x2) ∧x3=1,
                else (x1∨x2) ∧x3=0,
                endif
                }
        endif
        }
endif
```

(b) (x1∧x2)xor(¬x1∨x3)

```
if x1=0,
then    (x1∧x2)xor(¬x1∨x3)=1,
else    {
        if x2=0,
        then    {if x3=1,
                then (x1∧x2)xor(¬x1∨x3)=1,
                else (x1∧x2)xor(¬x1∨x3)=0,
                endif}
        else    {if x3=0,
```

```
                then (x1∧x2)xor(¬x1∨x3)=1,
                else (x1∧x2)xor(¬x1∨x3)=0,
                endif}
        endif
        }
endif
```

(c)2 of 3 function

```
if x1=1,
then    {
        if x2=1
        then function=1,
        else    {if x3=1,
                then function =1,
                else function=0,
                endif
                }
        endif
        }
else    {
        if x2=1,
        then    {
                if x3=1,
                then function=1,
                else function =0,
                endif
                }
        else function=0;
        endif
        }
endif
```

2 (a)2×3×3×4=72

(b)There are 8 yes and 8 no, so the entropy is –(1/2)log(1/2) –(1/2)log(1/2)=1.

(c)Berry: Berry is used in 7 cases and not used in 9 cases.

If Berry is used, there are 6 yes and 1 no, if Berry is not used, there are 2 yes and 7 no.

So information gain is 1- $\frac{7}{16}$×(-$\frac{6}{7}$log$\frac{6}{7}$-$\frac{1}{7}$log$\frac{1}{7}$)- $\frac{9}{16}$×(-$\frac{2}{9}$log$\frac{2}{9}$-$\frac{7}{9}$log$\frac{7}{9}$)=1-$\frac{7}{16}$×0.59167-$\frac{9}{16}$×0.7642=0.311

Ball: Poke has 6 cases, Great has 7 cases, Ultra has 3 cases.

If Poke is used, there are 1 yes and 5 no.

If Great is used, there are 4 yes and 3 no.

If Ultra is used, there are 3 yes.

So information gain is 1-$\frac{6}{16}$×(-$\frac{1}{6}$log$\frac{1}{6}$-$\frac{5}{6}$log$\frac{5}{6}$)- $\frac{7}{16}$×(-$\frac{4}{7}$log$\frac{4}{7}$-$\frac{3}{7}$log$\frac{3}{7}$)- $\frac{3}{16}$×0=1-$\frac{6}{16}$×0.6500-$\frac{7}{16}$×0.9852=0.325

Color: There are 3 green cases, 7 yellow cases, 6 red cases.

If pokemon is green level, there are 2 yes and 1 no.

If pokemon is yellow level, there are 3 yes and 4 no.

If pokemon is red level, there are 3 yes and 3 no.

So information gain is 1-$\frac{3}{16}$×(-$\frac{1}{3}$log$\frac{1}{3}$-$\frac{2}{3}$log$\frac{2}{3}$)- $\frac{7}{16}$×(-$\frac{4}{7}$log$\frac{4}{7}$-$\frac{3}{7}$log$\frac{3}{7}$)- $\frac{6}{16}$×(-$\frac{3}{6}$log$\frac{3}{6}$-$\frac{3}{6}$log$\frac{3}{6}$)=1-$\frac{3}{16}$×0.9183-$\frac{7}{16}$×0.9852-$\frac{6}{16}$=0.0218

Type: There are 6 normal type, 4 water type, 4 flying type, 2 psychic type.

If type is normal, there are 3 yes and 3 no.

If type is water, there are 2 yes and 2 no.

If type is flying, there are 3 yes and 1 no.

If type is psychic, there are 0 yes and 2 no.

So information gain is 1-$\frac{6}{16}$-$\frac{4}{16}$-$\frac{4}{16}$×(-$\frac{1}{4}$log$\frac{1}{4}$-$\frac{3}{4}$log$\frac{3}{4}$)= 1-$\frac{6}{16}$-$\frac{4}{16}$-$\frac{4}{16}$×0.8113=0.172

(d)Information gain with ball has highest value, so the Ball attribute should be root for the decision tree.

(e)if Ball =Poke,

then {

if Berry =yes,

then if Color=Green,

then Caught=yes,

```
                else Caught=no,
                endif
        else    Caught=no,
        endif
        }
else    {
        if Ball=Great,
        then    if Berry=yes,
                then caught=yes,
                else caught=no,
                endif
        else    caught=yes,
        endif
        }
endif
```

(f)According to my decision tree, the label is {yes, yes, yes}, accuracy is 1/3.

(g)I think maybe it is not a good method to achieve high accuracy because the training set is small, also there exists uncertainty and possibility even if all the four features are set.

3(a)Berry: information gain=1- $\frac{7}{16}$×(1-($\frac{6}{7}$)²- ($\frac{1}{7}$)²)- $\frac{9}{16}$×(1-($\frac{2}{9}$)²-$\left(\frac{7}{9}\right)^2$)= 1- $\frac{7}{16}$×$\frac{12}{49}$- $\frac{9}{16}$×$\left(\frac{28}{81}\right)$=0.698

Ball: information gain=1-$\frac{6}{16}$×(1-($\frac{5}{6}$)²- ($\frac{1}{6}$)²)- $\frac{7}{16}$×(1-($\frac{4}{7}$)²- ($\frac{3}{7}$)²)- $\frac{3}{16}$×0=1-$\frac{6}{16}$×$\frac{5}{18}$- $\frac{7}{16}$×$\frac{24}{49}$- $\frac{3}{16}$×0=0.682

Color: information gain=1-$\frac{3}{16}$×(1-($\frac{1}{3}$)²-($\frac{2}{3}$)²)- $\frac{8}{16}$×(1-($\frac{4}{7}$)²-($\frac{3}{7}$)²)- $\frac{6}{16}$×(1-($\frac{3}{6}$)²–($\frac{3}{6}$)²)=1-$\frac{3}{16}$×$\frac{4}{9}$-$\frac{7}{16}$×$\frac{24}{49}$-$\frac{6}{16}$×$\frac{1}{2}$=0.515

Type: information gain=1-$\frac{6}{16}$ × $\frac{1}{2}$-$\frac{4}{16}$×$\frac{1}{2}$-$\frac{4}{16}$×(1-($\frac{1}{4}$)²-($\frac{3}{4}$)²)= 1-$\frac{6}{16}$ × $\frac{1}{2}$-$\frac{4}{16}$ × $\frac{1}{2}$-$\frac{4}{16}$×$\frac{3}{8}$=0.594

(b)Berry should be the root for the decision tree, there two measures lead to different trees.

2 Linear Classifiers

1 $w=(w_1,w_2,w_3,w_4)^T=(0,0,0,2)^T$ b=-1

2

$(0,0,0,2)^T$ *(1,0,1,1)+(-1)=1>0 true

$(0,0,0,2)^T$ *(0,1,0,1)+(-1)=1>0 true

$(0,0,0,2)^T$ *(1,0,1,0)+(-1)=-1<0 false

$(0,0,0,2)^T$ *(1,1,0,0)+(-1)=-1<0 false

$(0,0,0,2)^T$ *(1,1,1,1)+(-1)=1>0 true

$(0,0,0,2)^T$ *(1,1,1,0)+(-1)=-1<0 false

$(0,0,0,2)^T$ *(0,0,1,0)+(-1)=-1<0 true

So the accuracy is 4/7≈0.5714

3

```
public static double[] linear_classifier(double[][] data)
	{
		int row=data.length;
		int col=data[0].length;
		double[] w_p=new double[col];//w_p=(b,w1,w2,w3,w4)T
		double r=0.05;
		int all_correct=0;
		int iter=0;
		while(all_correct==0)
		{
			all_correct=1;
			int i,j;
			for(i=0;i<row;i++)
			{
				double output=w_p[0];
				for(j=1;j<col;j++)
				{
					output=output+w_p[j]*data[i][j];
				}
				if(output*data[i][0]<=0)
				{
					w_p[0]=w_p[0]+r*data[i][0];
					for(j=1;j<col;j++)
					{
						w_p[j]=w_p[j]+r*data[i][0]*data[i][j];
					}
					all_correct=0;
				}
			}
			iter++;
			if(iter>500)
			{
				System.out.println("Reach iteration step limit");
				break;
			}
		}
		System.out.println("Number of iteration steps: "+iter);
		return w_p;
	}
```

So $w=(0.15,0,0,0.15)^T$ $b=-0.1$

3 Experiments

Setting A

1.(a)The ID3 algorithm and tree structure are used. Tree structure saves links to its parent and children and also root attribute we select. In the process, we first save a root node and select an attribute (not been selected before) that best classifies data according to information gain. Then we add tree branches based on value of attribute and disperse data into different branches. If one subset is empty, then we reach leaf node and give value that has majority number to it. If there is only one kind of classification result in one subset, then we can also assume that this is leaf node and give leaf node that value. For other cases, we use the dispersed data to run the algorithm again recursively.

(b)The accuracy on training data is 1, so error rate is 0. (hw1_A1_Bodong.java)

(c)The accuracy on testing data is 1, so error rate is 0. (hw1_A1c_Bodong.java)

(d)The maximum depth is 3. (Assume single node has depth of zero)(hw1_A1d_Bodong.java)

2(a) In cross validation 5 of 6 files are used and the remaining one is used to test accuracy. Since bias is caused because only part of training data is used, this algorithm is implemented 6 times so that every file has a chance to be validation file and every file has same chance to be training file.

The average cross validation and deviation is below. (Generated by hw1_A2a_Bodong.java)

depth	accuracy0	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	deviation
1	1	1	1	1	0.99686	0.86185	0.97645	0.05126
2	1	1	1	1	1	0.88854	0.98142	0.04153
3	1	1	1	1	1	0.88854	0.98142	0.04153
4	1	1	1	1	1	0.88854	0.98142	0.04153
5	1	1	1	1	1	0.88854	0.98142	0.04153
10	1	1	1	1	1	0.88854	0.98142	0.04153
15	1	1	1	1	1	0.88854	0.98142	0.04153
20	1	1	1	1	1	0.88854	0.98142	0.04153

To verify the correctness, the index of category is checked. In training_00 file, we can see that if the 5th category(odor) is pungent, then it is definitely poisonous. If the odor is almond=a, anise=l or none=n, then it is edible for sure. So even if only one category is considered, reaching 100% accuracy in training_00 file is still possible.

(b) From the table, the best accuracy is 0.98142 from depth=2 to depth=20, when there is no limit for depth, the depth it reaches is 3. So if we choose 3 as depth limit, the accuracy is 1. If we choose 2 as depth, the accuracy is 0.99780. (hw1_A2b_Bodong.java)

Setting B

1(a) The accuracy on B_training.data is 0.93982. So error rate is 0.06018. (hw1_B1a_Bodong.java)

(b) The accuracy on B_test.data is 0.93359. So error rate is 0.06641. (hw1_B1b.java)

(c) The accuracy on A's training data is 0.98299. So error rate is 0.01701. (hw1_B1c_Bodong.java)

(d) The accuracy on A's test data is 0.98737. So error rate is 0.01263. (hw1_b1d_Bodong.java)

(e) The depth of the tree is 6. There are more noises in B setting, so we need more categories to classifies them well.

2(a)The cross-validation accuracy and standard deviation are below. When the depth limit reaches 5, the limit would not influence because it is larger than possible depth. To decrease time complexity and avoid overfitting, the depth should not be too big. On the other hand, the average accuracy should be large to have good performance and deviation should be small to maintain stable results. So depth =1 is the best in this case. (hw1_B2a_Bodong.java)

depth	accuracy0	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	deviation
1	0.94505	0.94348	0.94348	0.9529	0.95447	0.83202	0.92857	0.04339
2	0.94819	0.94662	0.94348	0.94348	0.94034	0.69073	0.90214	0.09457
3	0.93563	0.94505	0.95604	0.95447	0.92935	0.82731	0.92464	0.04455
4	0.93406	0.94348	0.94662	0.9529	0.95761	0.83045	0.92752	0.04403
5	0.93406	0.94034	0.94348	0.95447	0.95447	0.83202	0.92647	0.04287
10	0.93406	0.94034	0.94348	0.95447	0.95447	0.83202	0.92647	0.04287
15	0.93406	0.94034	0.94348	0.95447	0.95447	0.83202	0.92647	0.04287
20	0.93406	0.94034	0.94348	0.95447	0.95447	0.83202	0.92647	0.04287

(b) set depth=1, then accuracy on B's test data is 0.93578. (hw1_B2b_Bodong.java)

Setting C

1 To deal with '?' character with method 1 or method 2, the statistics of that feature value should be calculated before setting up tree structure. After getting the majority value, we replace '?' with that value. Then the following process is the same as before. If we want to solve it by method 3, a new value: '?' should be added in the value set of that feature.

2

Method 1: Setting the missing feature as the majority feature value

The majority feature value is 'b'. The accuracy by putting different set as validation group is 1, 1, 1, 1, 1, 0.95038 . So the average accuracy is 0.99173, standard deviation is 0.01849. (hw1_C_method1_Bodong)

Method 2: Setting the missing feature as the majority value of that label

The majority feature value of each label is always 'b'. The accuracy by putting different set as validation group is 1, 1, 1, 1, 1, 0.95038 . So the average accuracy is 0.99173, standard deviation is 0.01849. So it is the same as Method 1. (hw1_C_method2_Bodong)

Method 3: Treating the missing feature as a special feature

Treat '?' as a new feature value, then the accuracy is 1, 1, 1, 1, 1, 0.95038 . So the average accuracy is 0.99173, standard deviation is 0.01849. (hw1_C_method3_Bodong)

3 By using method 3, training on C_training.data and test on C_test.data, the accuracy is 100%.

(hw1_C3_Bodong.java)