CS446: Machine Learning **Fall 2015**

Problem Set Home Work 2

Shivam Upadhyay *Handed In: September 30, 2015*

1. (a) For creating the decision tree based on Id3 algorithm we will first determine the root tree based on the gain from each node:
Initial entropy$(S) = -P_+ * log(p_+) - P_- * log(p_-)$
Given$P_+ = 35/50$ and $P_- = 15/50$
therefore, entropy$(S) = -35/50 * log(35/50) - 15/50 * log(15/50)$
$S = 0.88$
Given the initial entropy in the system we can compute the gain for each parameter which is defined as
Gain(S,a) = Entropy(S) - $\sum_{v \epsilon (Values(a)} \frac{|S_v|}{|S|}$*Entropy(S)

Computing gains for each of the attributes:
Holiday:
Values(Holiday)
YES (+5 , -10) NO (+30, -5)
Gain(S,Holiday)= $Entropy(S) - (35/50 * E(S_{NO}) + 15/50 * E(S_{YES}))$
$E(S_{YES}) = -5/15 * log(5/15) - 10/15 * log(10/15)$
$E(S_{YES}) = 0.92$
$E(S_{NO}) = -30/35 * log(30/35) - 5/35 * log(5/35)$
$E(S_{NO}) = 0.59$
Gain(S,Holiday)= $0.88 - (\frac{35*0.59}{50} + \frac{15*0.92}{50}) = 0.88 - 0.69$
Gain(S,Holiday)= 0.19

Exam Tomorrow:
Values(Exam Tomorrow)
YES (+15 , -1) NO (+20, -14)
Gain(S,Exam Tomorrow)= $Entropy(S) - (34/50 * E(S_{NO}) + 16/50 * E(S_{YES}))$
$E(S_{YES}) = -15/16 * log(15/16) - 1/16 * log(1/16)$
$E(S_{YES}) = 0.34$
$E(S_{NO}) = -20/34 * log(20/34) - 14/34 * log(14/34)$
$E(S_{NO}) = 0.98$
Gain(S,Exam Tomorrow)= $0.88 - (\frac{34*0.98}{50} + \frac{16*0.34}{50}) = 0.88 - 0.77$
Gain(S,Exam Tomorrow)= 0.11

As Gain(S,Holiday) ¿ Gain(S,Exam Tomorrow), Our root node will be **HOLIDAY**

(b) The decision tree will be created on the basis of new heuristic:
MajorityError = min(p; 1-p)
There are four attributes (Color, Size, Act, and Age), thus similar to above problem we will compute the gain for each attribute based on the new splitting heuristic. Following are the gains computed for each value:
Let us first compute the entropy of entire set Entropy(S)= $min(p; 1-p)$
Where p is the fraction of examples with label T= 7/16
Thus 1-p= $1 - 7/16 = 9/16$
Entropy(S)= $min(7/16, 9/16) = 7/16$

Color:
Gain(S,Color) = Entropy(S) - $\sum_{v\epsilon(Yellow,Purple)} \frac{|S_v|}{|S|}$*Entropy(S)

Entropy(S_{yellow}) = $min(p; 1-p)$

Where p is the fraction of examples with label T when Color is yellow= 5/8
Thus 1-p= $1 - 5/8 = 3/8$
Entropy(S) = $min(5/8, 3/8) = 3/8$

Entropy(S_{purple}) = $min(p; 1-p)$
Where p is the fraction of examples with label T when Color is purple= 2/8
Thus 1-p= $1 - 2/8 = 6/8$
Entropy(S) = $min(2/8, 6/8) = 2/8$

Gain(S,Color)= $Entropy(S) - (8/16 * E(S_{yellow}) + 8/16 * E(S_{purple}))$

Gain(S,Color)= $7/16 - (\frac{8*3}{16*8} + \frac{8*2}{16*8}) = 7/16 - 5/16$

Gain(S,Color)= 1/8

Size:

Entropy(S_{small}) = $min(p; 1-p)$

Where p is the fraction of examples with label T when size is small= 5/8
Thus 1-p= $1 - 5/8 = 3/8$
Entropy(S) = $min(5/8, 3/8) = 3/8$

Entropy(S_{long}) = $min(p; 1-p)$
Where p is the fraction of examples with label T when size is large= 2/8
Thus 1-p= $1 - 2/8 = 6/8$
Entropy(S) = $min(2/8, 6/8) = 2/8$

Gain(S,Size)= $Entropy(S) - (8/16 * E(S_{small}) + 8/16 * E(S_{large}))$

Gain(S,Size)= $7/16 - (\frac{8*3}{16*8} + \frac{8*2}{16*8}) = 7/16 - 5/16$

Gain(S,Size)= 1/8

Similarly Gain(S,Act) and Gain(S,Age) can be computed:

Gain(S,Act)= 1/8

Gain(S,Age)= 1/8

Since all the attributes have equal gains any one of them can be selected as the root node(I Chose colour as the root attribute). For the next level we'll repeat the same process again after extending the branches for all the attribute values of Color Yellow, Purple and the Gain will be computed on the basis of:
$Entropy(S_{Yellow} = 3/8)$
$Entropy(S_{Purple} = 2/8 = 1/8)$
Similarly gains can be computed for the remaining attributes with Colour as the root and above mentioned entropies as the starting pint and we'll get the decision tree as shown below:

Decision tree for The Balloons data set:

Decision tree

```
if Color = Yellow:
   if Size = Small:
      class = T
   if Size = Large:
      if Age = Adult:
         if Act = Stretch:
            class = T
         if Act = Dip:
            class = F
      if Age = Child:
         class = F
if Color = Purple:
   if Age = Adult:
      if Act = Stretch:
         class = T
      if Act = Dip:
         class = F
   if Age = Child:
      class = F
```

(c) Id3 works iteratively by searching through the space of possible decision trees from simplest to increasingly complex solutions driven by the information gain heuristic. Since gains are computed per level basis i.e there is no backtracking and the further decisions are made on the basis of attributes selected previously, which makes it susceptible to the risk of hill-climbing search without backtracking. Thus, it converges to the locally optimal solution corresponding to the single search path that it explores based on the test data. In conclusion it might converge to the less desirable tree which is not globally optimal.

2. I used the common features for testing the ID3 and SGD which comprises of minimal features and few additional features which is as following:

   Miniaml Feature Set ϵ "firstName0", "firstName1", "firstName2", "firstName3", "firstName4", "lastName0" ,"lastName1", "lastName2", "lastName3", "lastName4"
   The above feature set correspond to the positions of character in the first name and last name respectively. Now each of these features were scaled to 26 features by appending the feature name with all the alpha bates. Only the lowercase alpah bates were considered and if a name had capitals, it was first converted to small case and then appended in the feature set. Only **"NOMINAL"** values were used for the feature sets with values 0,1. For example if the name in the data file was "Joe" the corresponding feature will have the feature label "1" corresponding to features: "firstname0=j", "firstname0=o", "firstname0=j" will have the feature label as "1" rest all the features will have feature label as "0". Apart from the feature set has the Class label attribute which corresponds to the label of any name in the given data. It takes two nominal values "+","-".
   Additional Feature ϵ "lengthoffirstname", "Lengthof2ndname", "SecondLetterAnVowel(1)", "SecondLetterAnVowel(2)"
   The above features were in addition to the minimal features required for the problem. "lengthoffirstname" and "lengthof2ndname" computes the length of first name and last name respectively. If length was greater than 5 then the features will have labels "1" or else "0". Fetures "SecondLetterAnVowel(1)" and "SecondLetterAnVowel(2)" indicates whether the second alpha bate in first and last name is an vowel or not, if it's an vowel the corresponding label is "1" and "0" otherwise.
   The experiments resulted in classifiers which are arranged in the decreasing order of their corresponding accuracy is as following:

   I Stochastic Gradient Descent over features

   II ID3 for full depth

   III ID3 for max depth 8

   IV ID3 for max depth 4

   V Stochastic Gradient Descent over decision stumps

Algorithm	Percentage Accuracy	Confidence Interval
SGD	78.18	78.18 ± 7.04
ID3(Full)	71.74	71.74 ± 10.65
ID3(Depth 8)	70.76	70.76 ± 8.74
ID3(Depth 4)	67.68	67.68 ± 14.89
SGD over DT	63.60	63.60 ± 17.40

The below table shows the **Statistical Significance** for the paired algorithms (Rank wise).

The initial hypothesis is assumed to be a null hypothesis and distribution is assumed to be two-tailed. The below table lists the value of SE , T-score and p-value determined from the t-score. The hypothesis is rejected or accepted based on the standard p <0.05

Algorithm Pair	Standard Error SE(d)	t-statistics	p -value	Significant
SGD and ID3(Full)	2.91	2.21	0.091626	No
ID3(Full) and ID3(Depth 8)	1.12	0.91	0.414298	No
ID3(Depth 8) and ID3(Depth 4)	2.01	1.502	207509	No
ID3(Depth 4) and SGD over DT	4.67	0.87	00.433395	No

Thus none of the values are statistically significant given the assumption and hypothesis
The below section describes the assumptions, set of experiments performed and best trees for each algorithm
Stochastic Gradient Descent :
SGD was evaluated for the above mentioned feature set. Tuning of SGD for convergence required lots of hit and trials based on value of learning rate, error threshold and number of Max iterations. Below are the values used for learning rate and number of iteration:

Learning rate $\alpha \epsilon$ 0.1, 0.01, 0.001, 0.0001

Error threshold ϵ 1E-4, 1E-5, 1E-6, 1E-8,1E-9

Number of iterations ϵ 10, 100 , 500 ,1000 , 1500

The value of weight vector was initialised to "0" and for various combination of the above mentioned factors experiments were performed to test the convergence of SGD. The values were altered based on the convergence of SGD and corresponding accuracy.

Accuracy value was high for the following combination of parameters:

Learning rate $\alpha \epsilon$ 0.001

Error threshold ϵ 1E-5

Number of iterations ϵ 1000

Decision tree depth 4 Accurecy		
Correctly Classified Instances	44	75.8621 %
Incorrectly Classified Instances	14	24.1379 %
Kappa statistic	0.4637	
Mean absolute error	0.3776	
Root mean squared error	0.4671	
Relative absolute error	78.7751 %	
Root relative squared error	95.4879 %	
Total Number of Instances	58	

=== Confusion Matrix ===

a b ¡– classified as

12 11 — a = +

3 32 — b = -

SGD over DT: New set of features were generated which comprised of outputs of 100 different decision trees repeatedly sampled over 50 percent of the test data. Now each of these tree is used for classifying the training instances. Output of each decision tree is then stored into a 100 dimensional vector which is will form the new set of instances with class attribute same as in the original data. Similar to SGD there were many hit and trials to converge to best performing value.

Learning rate $\alpha \epsilon$ 0.1, 0.01, 0.001, 0.0001

Error threshold ϵ 1E-9, 1E-11, 1E-12, 1E-8

Number of iterations ϵ 10, 100 , 500 ,1000 , 1500

The value of weight vector was initialised to "0" and for various combination of the above mentioned factors experiments were performed to test the convergence of SGD. The values were altered based on the convergence of SGD and corresponding accuracy.

Accuracy value was high for the following combination of parameters:

Learning rate $\alpha \epsilon$ 0.04

Error threshold ϵ 1E-11

ID3 Full tree :

Decision tree Full
firstName4=a = 0
— firstName1=a = 0
— — firstName3=a = 0
— — — firstName2=a = 0
— — — — firstName4=n = 0
— — — — — firstName3=n = 0
— — — — — — firstName2=n = 0
— — — — — — — firstName2=d = 0
— — — — — — — — firstName3=o = 0
— — — — — — — — — firstName2=e = 0: -

```
| — — — — — — — — — — firstName2=e = 1
— — — — — — — — — — — firstName0=a = 0
— — — — — — — — — — — — firstName0=o = 0: -
— — — — — — — — — — — — firstName0=o = 1: +
— — — — — — — — — — — firstName0=a = 1: +
— — — — — — — — — — firstName3=o = 1
— — — — — — — — — — lastName1=i = 0
— — — — — — — — — — — firstName0=a = 0: -
— — — — — — — — — — — firstName0=a = 1: +
— — — — — — — — — — lastName1=i = 1: +
— — — — — — — — — firstName2=d = 1
— — — — — — — — — — lastName0=n = 0: +
— — — — — — — — — — lastName0=n = 1: -
— — — — — — — — — firstName2=n = 1
— — — — — — — — — lastName2=n = 0
— — — — — — — — — — firstName1=o = 0
— — — — — — — — — — — firstName0=l = 0: -
— — — — — — — — — — — firstName0=l = 1: +
— — — — — — — — — — firstName1=o = 1: +
— — — — — — — — — lastName2=n = 1: -
— — — — — — — — firstName3=n = 1
— — — — — — — — lastName1=h = 0
— — — — — — — — — firstName0=b = 0
— — — — — — — — — — lastName2=s = 0: -
— — — — — — — — — — lastName2=s = 1
— — — — — — — — — — — lastName0=c = 0: +
— — — — — — — — — — — lastName0=c = 1: -
— — — — — — — — — firstName0=b = 1: +
— — — — — — — — lastName1=h = 1: +
— — — — — — — firstName4=n = 1
— — — — — — — — firstName0=g = 0: +
— — — — — — — — firstName0=g = 1: -
— — — — — — firstName2=a = 1
— — — — — — — lastName1=e = 0
— — — — — — — — Lengthof2ndname = 0: -
— — — — — — — — Lengthof2ndname = 1
— — — — — — — — firstName0=j = 0
— — — — — — — — — firstName3=s = 0: +
— — — — — — — — — firstName3=s = 1: -
— — — — — — — — firstName0=j = 1: -
— — — — — — — lastName1=e = 1: -
— — — — firstName3=a = 1
— — — — — — firstName0=y = 0
— — — — — — — lastName1=a = 0
— — — — — — — — firstName2=b = 0: +
```

```
— — — — — firstName2=b = 1: -
— — — — lastName1=a = 1
— — — — — firstName0=v = 0: -
— — — — — firstName0=v = 1: +
— — — firstName0=y = 1: -
— firstName1=a = 1
— — lastName3=g = 0
— — — firstName3=y = 0
— — — — lastName1=n = 0
— — — — — lastName2=m = 0
— — — — — — lastName4=a = 0
— — — — — — — — lastName4=m = 0
— — — — — — — — — firstName0=s = 0
— — — — — — — — — — lastName0=a = 0: +
— — — — — — — — — — lastName0=a = 1
— — — — — — — — — — — firstName2=r = 0
— — — — — — — — — — — — firstName0=d = 0: -
— — — — — — — — — — — — firstName0=d = 1: +
— — — — — — — — — — — firstName2=r = 1: +
— — — — — — — — — firstName0=s = 1: -
— — — — — — — — lastName4=m = 1: -
— — — — — — — lastName4=a = 1
— — — — — — — — firstName0=m = 0
— — — — — — — — — firstName0=t = 0: -
— — — — — — — — — firstName0=t = 1: +
— — — — — — — — firstName0=m = 1: +
— — — — — lastName2=m = 1: -
— — — — lastName1=n = 1: -
— — — firstName3=y = 1: -
— — lastName3=g = 1: -
firstName4=a = 1
— firstName3=m = 0
— — lastName0=m = 0: +
— — lastName0=m = 1
— — — lastName3=s = 0: -
— — — lastName3=s = 1: +
— firstName3=m = 1
— — lastName4=t = 0: -
— — lastName4=t = 1: +
```

The tree that gave the most accurate is as shown in table above the confusion matrix (Number of correct and incorrect instances) and accuracies are as following:

Decision tree Accurecy
Correctly Classified Instances 45 76.2712 %
Incorrectly Classified Instances 14 23.7288 %
Kappa statistic 0.5186
Mean absolute error 0.2373
Root mean squared error 0.4871 Relative absolute error 48.1127 %
Root relative squared error 98.1167 %
Total Number of Instances 59
=== Confusion Matrix ===
a b ¡– classified as
26 7 — a = +
7 19 — b = -

ID3 Max depth 8:

Decision tree depth 8
firstName1=a = 0
— firstName4=a = 0
— — firstName4=n = 0
— — — firstName3=d = 0
— — — — lastName3=r = 0
— — — — — firstName3=a = 0
— — — — — — firstName2=a = 0
— — — — — — — firstName2=d = 0
— — — — — — — — firstName3=n = 0: -
— — — — — — — — firstName3=n = 1: +
— — — — — — — firstName2=d = 1
— — — — — — — — lastName1=c = 0: +
— — — — — — — — lastName1=c = 1: -
— — — — — — firstName2=a = 1
— — — — — — — lastName1=e = 0
— — — — — — — — lengthoffirstname = 0: +
— — — — — — — — lengthoffirstname = 1: -
— — — — — — — lastName1=e = 1: -
— — — — — firstName3=a = 1
— — — — — — firstName2=n = 0
— — — — — — — lastName0=s = 0
— — — — — — — — firstName0=r = 0: +
— — — — — — — — firstName0=r = 1: -
— — — — — — — lastName0=s = 1: -
— — — — — — firstName2=n = 1: +
— — — — lastName3=r = 1: +
— — — firstName3=d = 1: +

```
— — firstName4=n = 1
— — — lastName1=i = 0
— — — — lastName0=d = 0: +
— — — — lastName0=d = 1: -
— — — lastName1=i = 1: -
— firstName4=a = 1
— — lastName4=e = 0
— — — lastName4=a = 0
— — — — lastName2=z = 0: +
— — — — lastName2=z = 1: -
— — — lastName4=a = 1: -
— — lastName4=e = 1
— — — firstName1=t = 0: -
— — — firstName1=t = 1: +
firstName1=a = 1
— lastName1=o = 0
— — lastName4=a = 0
— — — lastName0=l = 0
— — — — lastName1=n = 0
— — — — — lastName4=m = 0
— — — — — — lastName1=r = 0
— — — — — — — firstName4=i = 0
— — — — — — — — firstName0=m = 0: +
— — — — — — — — firstName0=m = 1: +
— — — — — — — firstName4=i = 1: -
— — — — — — lastName1=r = 1: +
— — — — — lastName4=m = 1: -
— — — — lastName1=n = 1: -
— — — lastName0=l = 1: -
— — lastName4=a = 1
— — — firstName0=t = 0: -
— — — firstName0=t = 1: +
— lastName1=o = 1: +
```

The tree that gave the most accurate is as shown in table above the confusion matrix (Number of correct and incorrect instances) and accuracies are as following:

Decision tree depth 8 Accurecy

Correctly Classified Instances 44 75.8621 %
Incorrectly Classified Instances 14 24.1379 %
Kappa statistic 0.4957
Mean absolute error 0.2923
Root mean squared error 0.4585
Relative absolute error 60.9915 %
Root relative squared error 93.7353 %
Total Number of Instances 58
=== Confusion Matrix ===
a b ¡– classified as
16 7 — a = +
7 28 — b = -

Id3 Max depth 4:

Decision tree depth 4

firstName1=a = 0
— firstName4=a = 0
— — firstName4=n = 0
— — — firstName3=d = 0
— — — — lastName3=r = 0: -
— — — — lastName3=r = 1: +
— — — firstName3=d = 1: +
— — firstName4=n = 1
— — — lastName1=i = 0
— — — — lastName0=d = 0: +
— — — — lastName0=d = 1: -
— — — lastName1=i = 1: -
— firstName4=a = 1
— — lastName4=e = 0
— — — lastName4=a = 0
— — — — lastName2=z = 0: +
— — — — lastName2=z = 1: -
— — — lastName4=a = 1: -
— — lastName4=e = 1
— — — firstName1=t = 0: -
— — — firstName1=t = 1: +
firstName1=a = 1
— lastName1=o = 0
— — lastName4=a = 0
— — — lastName0=l = 0
— — — — lastName1=n = 0: +
— — — — lastName1=n = 1: -

```
— — — lastName0=l = 1: -
— — lastName4=a = 1
— — — firstName0=t = 0: -
— — — firstName0=t = 1: +
— lastName1=o = 1: +
```

The tree that gave the most accurate is as shown in table above the confusion matrix (Number of correct and incorrect instances) and accuracies are as following:

```
Decision tree depth 4 Accurecy
Correctly Classified Instances      44      75.8621 %
Incorrectly Classified Instances      14      24.1379 %
Kappa statistic      0.4637
Mean absolute error      0.3776
Root mean squared error      0.4671
Relative absolute error      78.7751 %
Root relative squared error      95.4879 %
Total Number of Instances      58
=== Confusion Matrix ===
a b ¡– classified as
12 11 — a = +
3 32 — b = -
```

To conclude the SGD performed the best for the given data under the assumption that only average accuracy is considered for the deciding the rank. ID3 doesn't explore the decision space i.e. as similar to solution to problem 1 it converges to local optimum sometimes over fitting the training data at the same time SGD tries and minimise the weight vector which best separates the decision space at each step. Apart from that using the error threshold value to stop the SGD makes sure that it converges to some local minimum value. Although, global optimal solution can't be guaranteed in the case of SGD but it explores the decision space more that ID3.