CSE 5522 Homework 2

Manjari Akella

10/18/2013

1. I'm trying to figure out what board/card game to play. I have four games that I can play: Settlers of Catan, Fluxx, Apples to Apples, or Scrabble. There are a number of attributes that can be predictors of which game I choose:

- **dayOfWeek: Is today a Weekday, Saturday, or Sunday?**
- **timeOfDay: Is it morning, afternoon, or evening?**
- **timeToPlay: Do I have <30, 30-60, or >60 minutes to play?**
- **mood: Am I in a silly, happy, or tired mood?**
- **friendsVisiting: Do I have friends over who might play? (no/yes)**
- **kidsPlaying: Are there going to be kids in the game? (no/yes)**
- **atHome: Am I at home or out? (no/yes)**
- **snacks: Do I have snacks? (no/yes)**

Your job is to train a decision tree to predict which game I choose. I have provided you training and test materials in two formats: Text-attribute format, with the last column the name of the game (train, test, attributes), and a coded format, where the text attributes are replaced with integers (starting from 0, in the order above) (train, test).

The contents of the training files are exactly the same, just a different encoding; same with the test files.

Write decision tree code (do not use off the shelf software) to train a system for predicting what game I play. Report your results from the test set.

ALTERNATE, NON-CODING VERSION (45 points max). From the training data set above show the calculation for the first level of the decision tree.

- To calculate the first level, we need to determine a node on the basis of which level 1 will be split. For this we need to see which node split maximizes the gain. Let's first calculate the initial entropy in the system. Entropy of a system is given by –

$$\sum_{i=1}^{n} -P(v_i)\log_2 P(v_i)$$

We have
No. of instances of SettlersofCatan(SOC) = 57
No. of instances of Fluxx(F) = 82
No. of instances of Applesto Apples(ATA) = 23
No. of instances of scrabble(Sc) = 38

So, the initial entropy is –

$= -(57/200)*\log_2(57/200)-(82/200)*\log_2(82/200)-(23/200)*\log_2(23/200)-(38/200)*\log_2(38/200)$

= 1.8576

Let us now calculate gain after split at each node –

dayOfWeek:

a).Weekend
No. of instances of SettlersofCatan(SOC) = 21
No. of instances of Fluxx(F) = 23
No. of instances of Applesto Apples(ATA) = 8
No. of instances of scrabble(Sc) = 14

Entropy is –

$= -(21/66)*\log_2(21/66)-(23/66)*\log_2(23/66)-(8/66)*\log_2(8/66)-(14/66)*\log_2(14/66)$

= 1.8992

b). Saturday
No. of instances of SettlersofCatan(SOC) = 16
No. of instances of Fluxx(F) = 27
No. of instances of Applesto Apples(ATA) = 10
No. of instances of scrabble(Sc) = 11

Entropy is –

$= -(16/64)*\log_2(16/64)-(27/64)*\log_2(27/64)-(10/64)*\log_2(10/64)-(11/64)*\log_2(11/64)$

= 1.8804

c). Sunday
No. of instances of SettlersofCatan(SOC) = 20
No. of instances of Fluxx(F) = 32
No. of instances of Applesto Apples(ATA) = 5
No. of instances of scrabble(Sc) = 13

Entropy is –

$= -(20/70)*\log_2(20/70)-(32/70)*\log_2(32/70)-(5/70)*\log_2(5/70)-(13/70)*\log_2(13/70)$

= 1.7557

Gain if split at this node –
G = 1.8576 – [((66/200)* 1.8992)+((64/200)* 1.8804)+((70/200)* 1.7557)]

= 0.0146

$\underline{timeofDay:}$

a). morning
No. of instances of SettlersofCatan(SOC) = 15
No. of instances of Fluxx(F) = 33
No. of instances of Applesto Apples(ATA) = 6
No. of instances of scrabble(Sc) = 12

Entropy is –

= -(15/66)*$\log_2$(15/66)-(33/66)*$\log_2$(33/66)-(6/66)*$\log_2$(6/66)-(12/66)*$\log_2$(12/66)

= 1.7475

b). afternoon
No. of instances of SettlersofCatan(SOC) = 19
No. of instances of Fluxx(F) = 23
No. of instances of Applesto Apples(ATA) = 9
No. of instances of scrabble(Sc) = 14

Entropy is –

= -(19/65)*$\log_2$(19/65)-(23/65)*$\log_2$(23/65)-(9/65)*$\log_2$(9/65)-(14/65)*$\log_2$(14/65)

=1.9211

c). evening
No. of instances of SettlersofCatan(SOC) = 23
No. of instances of Fluxx(F) = 26
No. of instances of Applesto Apples(ATA) = 8
No. of instances of scrabble(Sc) = 12

Entropy is –

= -(23/69)*$\log_2$(23/69)-(26/69)*$\log_2$(26/69)-(8/69)*$\log_2$(8/69)-(12/69)*$\log_2$(12/69)

= 1.8582

Gain if split at this node –
G = 1.8576 – [((66/200)*1.7475)+((65/200)*1.9211)+((69/200)*1.8582)]

= 0.0155

timetoPlay:

a). <30
No. of instances of SettlersofCatan(SOC) = 11
No. of instances of Fluxx(F) = 21
No. of instances of Applesto Apples(ATA) = 0
No. of instances of scrabble(Sc) = 28

Entropy is –

$= -(11/60)*\log_2(11/60)-(21/60)*\log_2(21/60)-(0/60)*\log_2(0/60)-(28/60)*\log_2(28/60)$

= 1.4919

b). 30-60
No. of instances of SettlersofCatan(SOC) = 14
No. of instances of Fluxx(F) = 26
No. of instances of Applesto Apples(ATA) = 17
No. of instances of scrabble(Sc) = 4

Entropy is –

$= -(14/61)*\log_2(14/61)-(26/61)*\log_2(26/61)-(17/61)*\log_2(17/61)-(4/61)*\log_2(4/61)$

= 1.7832

c). >60
No. of instances of SettlersofCatan(SOC) = 32
No. of instances of Fluxx(F) = 35
No. of instances of Applesto Apples(ATA) = 6
No. of instances of scrabble(Sc) = 6

Entropy is –

$= -(32/79)*\log_2(32/79)-(35/79)*\log_2(35/79)-(6/79)*\log_2(6/79)-(6/79)*\log_2(6/79)$

= 1.6133

Gain if split at this node –
G = 1.8576 – [((60/200)*1.4919)+((61/200)*1.7832)+((79/200)*1.6133)]

= 0.2289

mood:

a). silly
No. of instances of SettlersofCatan(SOC) = 43

No. of instances of Fluxx(F) = 12
No. of instances of Applesto Apples(ATA) = 0
No. of instances of scrabble(Sc) = 18

Entropy is –

$= -(43/73)*\log_2(43/73)-(12/73)*\log_2(12/73)-(0/73)*\log_2(0/73)-(18/73)*\log_2(18/73)$

= 1.3760

b). happy
No. of instances of SettlersofCatan(SOC) = 0
No. of instances of Fluxx(F) = 61
No. of instances of Applesto Apples(ATA) = 0
No. of instances of scrabble(Sc) = 0

Entropy is –

$= -(0/61)*\log_2(0/61)-(61/61)*\log_2(61/61)-(0/61)*\log_2(0/61)-(0/61)*\log_2(0/61)$

= 0

c). tired
No. of instances of SettlersofCatan(SOC) = 14
No. of instances of Fluxx(F) = 9
No. of instances of Applesto Apples(ATA) = 23
No. of instances of scrabble(Sc) = 20

Entropy is –

$= -(14/66)*\log_2(14/66)-(9/66)*\log_2(9/66)-(23/66)*\log_2(23/66)-(20/66)*\log_2(20/66)$

= 1.9184

Gain if split at this node –
G = 1.8576 – [((73/200)*1.3760)+((61/200)*0)+((66/200)*1.9184)]

= 0.7223

friendsVisiting

a). yes
No. of instances of SettlersofCatan(SOC) = 22
No. of instances of Fluxx(F) = 44
No. of instances of Applesto Apples(ATA) = 15
No. of instances of scrabble(Sc) = 20

Entropy is –

$= -(22/101)*\log_2(22/101)-(44/101)*\log_2(44/101)-(15/101)*\log_2(15/101)-(20/101)*\log_2(20/101)$

= 1.8724

b). no
No. of instances of SettlersofCatan(SOC) = 35
No. of instances of Fluxx(F) = 38
No. of instances of Applesto Apples(ATA) = 8
No. of instances of scrabble(Sc) = 18

Entropy is –

$= -(35/99)*\log_2(35/99)-(38/99)*\log_2(38/99)-(8/99)*\log_2(8/99)-(18/99)*\log_2(18/99)$

= 1.8010
Gain if split at this node –
G = 1.8576 – [((101/200)*1.8724)+((99/200)*1.8010)]

= 0.0205

kidsPlaying

a). yes
No. of instances of SettlersofCatan(SOC) = 33
No. of instances of Fluxx(F) = 33
No. of instances of Applesto Apples(ATA) = 6
No. of instances of scrabble(Sc) = 18

Entropy is –

$= -(33/90)*\log_2(33/90)-(33/90)*\log_2(33/90)-(6/90)*\log_2(6/90)-(18/90)*\log_2(18/90)$

= 1.7863

b). no
No. of instances of SettlersofCatan(SOC) = 24
No. of instances of Fluxx(F) = 49
No. of instances of Applesto Apples(ATA) = 17
No. of instances of scrabble(Sc) = 20
Entropy is –

$= -(24/110)*\log_2(24/110)-(49/110)*\log_2(49/110)-(17/110)*\log_2(17/110)-(20/110)*\log_2(20/110)$

= 1.8642
Gain if split at this node –
G = 1.8576 – [((90/200)*1.7863)+((110/200)*1.8642)]

= 0.0285

atHome

a). yes

No. of instances of SettlersofCatan(SOC) = 29
No. of instances of Fluxx(F) = 52
No. of instances of Applesto Apples(ATA) = 10
No. of instances of scrabble(Sc) = 12

Entropy is –

$= -(29/103)*\log_2(29/103)-(52/103)*\log_2(52/103)-(10/103)*\log_2(10/103)-(12/103)*\log_2(12/103)$

= 1.7006

b). no

No. of instances of SettlersofCatan(SOC) = 28
No. of instances of Fluxx(F) = 30
No. of instances of Applesto Apples(ATA) = 13
No. of instances of scrabble(Sc) = 26

Entropy is –

$= -(28/97)*\log_2(28/97)-(30/97)*\log_2(30/97)-(13/97)*\log_2(13/97)-(26/97)*\log_2(26/97)$

= 1.9388

Gain if split at this node –
G = 1.8576 – [((103/200)*1.7006)+((97/200)*1.9388)]

= 0.0415

snacks

a). yes

No. of instances of SettlersofCatan(SOC) = 13
No. of instances of Fluxx(F) = 47
No. of instances of Applesto Apples(ATA) = 11
No. of instances of scrabble(Sc) = 28

Entropy is –

$= -(13/99)*\log_2(13/99)-(47/99)*\log_2(47/99)-(11/99)*\log_2(11/99)-(28/99)*\log_2(28/99)$

= 1.7642

b). no

No. of instances of SettlersofCatan(SOC) = 44

No. of instances of Fluxx(F) = 35
No. of instances of Applesto Apples(ATA) = 12
No. of instances of scrabble(Sc) = 10

Entropy is –

$= -(44/101)*\log_2(44/101)-(35/101)*\log_2(35/101)-(12/101)*\log_2(12/101)-(10/101)*\log_2(10/101)$

= 1.7475
Gain if split at this node –
G = 1.8576 – [((99/200)*1.7642)+((101/200)*1.7475)]

= 0.1018

The following table shows the gain when split at different nodes -

	dayofWeek	timeofDay	timetoPlay	**mood**	friendsVisiting	kidsPlaying	atHome	snacks
Gain	0.0146	0.0155	0.2289	**0.7223**	0.0205	0.0285	0.0415	0.1018

From the table we can clearly see that a split at node ‘mood’ maximizes the gain and hence, this is the first level (split) of the decision tree. Even looking at the number of examples classified for each game on each split and the entropy values, we can see that on the question ‘mood’, we get a definite game when the answer is ‘happy’. Intuitively, we can say that this might be the first level of the tree.

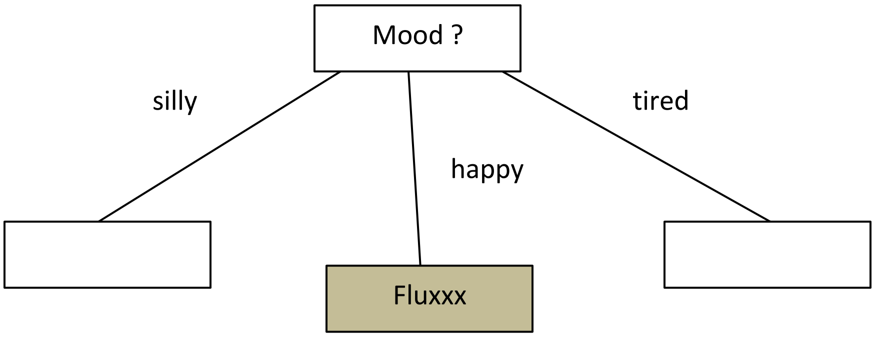
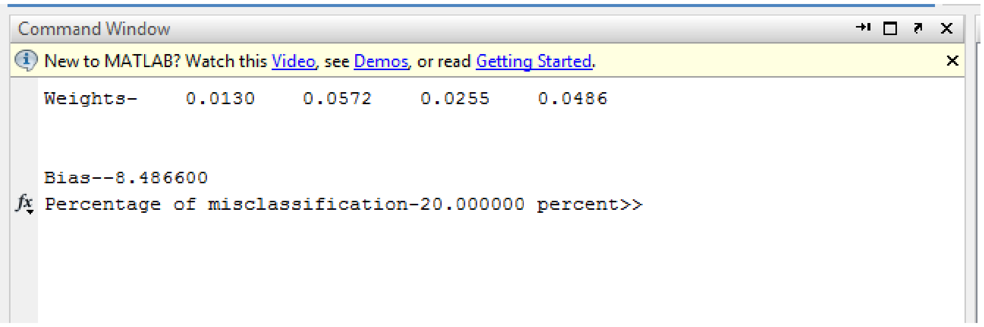


Figure 1: First Level of the decision tree

2. Perceptron learning. Do both parts:

a. An evil professor gives four exams and a minor homework for his class, but doesn't tell the students (a) what the weighting of the exams are, (b) what the grade on the homework is, (c) or what grade is passing for the class. You only have the four exam scores and whether the student passed the class. Write code to train a perceptron on the following data to predict pass (1) or fail (0). This data contains four predictor variables (x1,x2,x3,x4) and a label (0 or 1). Off-the-shelf solutions are not acceptable, you must write your own implementation. Turn in your code and report on the parameters of the perceptron that you have learned. We will test your code on a different dataset. Note: you will need to include a bias term! (train, test)

- Started with bias = -1 and all weights = 0.
- For alpha= 0.0001 and epsilon = 0.001,the following parameters were learnt in 31524 iterations :



New to MATLAB? Watch this Video, see Demos, or read Getting Started.

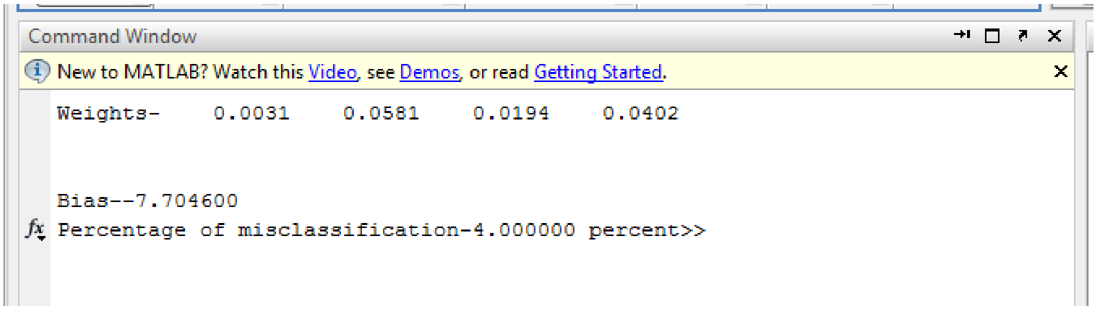
```
Weights-    0.0130    0.0572    0.0255    0.0486


Bias--8.486600
Percentage of misclassification-20.000000 percent>>
```

Figure 2: Weights, bias, accuracy

- The percentage of misclassification was 20%.
- However if I didn't use the error but used number of iterations as the stopping criteria, the following misclassification results were observed –

Iterations	10000	15000	20000	**25000**	30000
Percentage	10%	6%	12%	**4%**	6%



New to MATLAB? Watch this Video, see Demos, or read Getting Started.

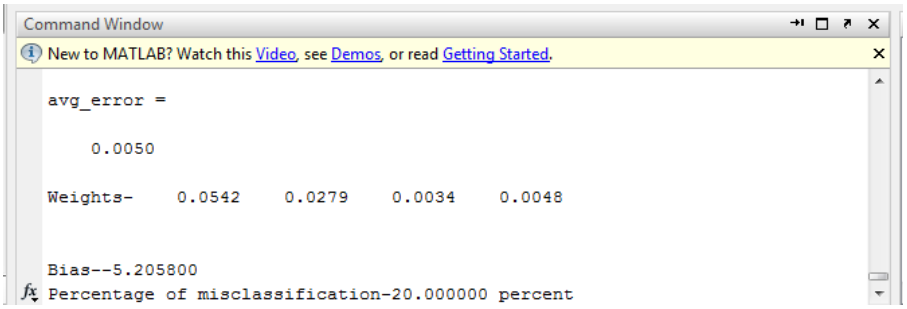
```
Weights-    0.0031    0.0581    0.0194    0.0402


Bias--7.704600
Percentage of misclassification-4.000000 percent>>
```

Figure 3: Results when iterations = 25000

b. Run your code on this dataset which is not linearly separable (train, test). Comment on the behavior of your code.

- Perceptrons cannot classify non-linearly separable problems.
- The average error kept oscillating between 0.0250, 0.0150, 0.0100, 0.0050. It didn't converge even after 25000 iterations.
- The following were the results after 25000 iterations –


Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
avg_error =

    0.0050

Weights-    0.0542    0.0279    0.0034    0.0048


Bias--5.205800
Percentage of misclassification-20.000000 percent
```

Figure 4: Result at 25000 iterations

3. Extra Credit

a. Run the perceptron learner on the game data from problem #1. Note that you will need to find an appropriate encoding for the data (for example, Weekday=0, Saturday=1, Sunday=2 is not a good representation of the day of week because Sunday is not twice as much as Saturday in a linear ordering). Note also that this is no longer a 2-class problem so you'll need to think about that.

- For nodes which are not binary, I did a 1:N encoding. For eg. For node dayOfWeek, instead of a single node I used 3 nodes – dWeekday,dSaturday,dSunday such that the value is {1,0,0} for weekday,{0,1,0} for Saturday and {0,0,1} for Sunday.
- Similarly for the output layer, I have 4 nodes (classes); one node goes to 1 for each of the games. So this is now a 4-class problem with 16 inputs.
- For alpha = 0.0001, the following were the accuracy results at different iterations –

Iterations	1000	5000	**7000**
Percentage	6%	10%	**4%**

b. Build a multi-layer perceptron trainer and run it on the same data. Again, off-the-shelf solutions are not acceptable.

- I assumed the number of hidden layer nodes to be 8.
- I used the log sigmoid activation function and backpropogation algorithm to train the network.
- Started with random weights for both hidden layer and output layer weights.

- For the bias terms (hidden and output), I started with -1 for all.
- For alpha = 0.001, the following were the percentage of misclassification for different iteration values –

Iterations	7000
Percentage	32%