CSE 5523: Homework 2

Soham Mukherjee
mukherjee.126@osu.edu

September 11, 2018

Abstract

Implementation of Decision Tree algorithm popularly knowns as ID3.

1 How to run the program

In the terminal type

```
> python id3.py train test 0.0
```

The last argument 0.0 is the MIN_GAIN. 0 grows full tree. You can vary this parameter from 0 to 1.

2 Brief Description

`InformationGain(data,f)`

Computes the information gain if we split the data based on feature f.It checks if the data element contains feature f or not. If it does it sends the element to left node otherwise to the right and computes the information gain obtained from the split.

`ID3(data,features,min_gain)`

Splits the data according to the highest gain of information and then calls itself recursively for left node and right node.

3 Output

The first line is invoking of the program. The nodes of the tree are printed as follows. The first element is feature used to split the data. Second element is the number of positive instances and the negative. The last float indicates the information gain obtained from

splitting the data based on that feature.
Three sets are shown. First with MIN_GAIN=0.0 , with MIN_GAIN=0.1 and then with MIN_GAIN=1.0. The accuracy obtained on test data is 1.0,0.994454713494 and 0.513863216266 respectively.

```
python id3.py train test 0
>FeatureVal(feature=5, value='n') (3122, 2878) 0.5332284965562999
->FeatureVal(feature=20, value='r') (2523, 79) 0.10332767641736795
-->FeatureVal(feature=22, value='u') (0, 48) 0.0
-->FeatureVal(feature=13, value='y') (2523, 31) 0.06121584721117114
--->FeatureVal(feature=8, value='b') (11, 26) 0.8779620013943912
---->FeatureVal(feature=22, value='u') (11, 0) 0.0
---->FeatureVal(feature=22, value='u') (0, 26) 0.0
--->FeatureVal(feature=2, value='g') (2512, 5) 0.011369742223829376
---->FeatureVal(feature=22, value='u') (0, 3) 0.0
---->FeatureVal(feature=1, value='c') (2512, 2) 0.004273364212711315
----->FeatureVal(feature=22, value='u') (0, 1) 0.0
----->FeatureVal(feature=8, value='n') (2512, 1) 0.001658257809865571
------>FeatureVal(feature=21, value='c') (134, 1) 0.06306780800385679
------->FeatureVal(feature=22, value='u') (0, 1) 0.0
------->FeatureVal(feature=22, value='u') (134, 0) 0.0
------>FeatureVal(feature=22, value='u') (2378, 0) 0.0
->FeatureVal(feature=4, value='t') (599, 2799) 0.38545330036398556
-->FeatureVal(feature=11, value='c') (599, 402) 0.39872187093249534
--->FeatureVal(feature=22, value='u') (385, 0) 0.0
--->FeatureVal(feature=11, value='r') (214, 402) 0.4817492590090897
---->FeatureVal(feature=22, value='u') (147, 0) 0.0
---->FeatureVal(feature=22, value='d') (67, 402) 0.5916727785823275
----->FeatureVal(feature=22, value='u') (67, 0) 0.0
----->FeatureVal(feature=22, value='u') (0, 402) 0.0
-->FeatureVal(feature=22, value='u') (0, 2397) 0.0
Accuracy on test data: 1.0
```

With MIN_GAIN=0.1:

```
python id3.py train test 0.1
>FeatureVal(feature=5, value='n') (3122, 2878) 0.533228496556
->FeatureVal(feature=20, value='r') (2523, 79) 0.103327676417
-->FeatureVal(feature=22, value='u') (0, 48) 0.0
-->FeatureVal(feature=22, value='u') (2523, 31) 0.0612158472112
->FeatureVal(feature=4, value='t') (599, 2799) 0.385453300364
-->FeatureVal(feature=11, value='c') (599, 402) 0.398721870932
--->FeatureVal(feature=22, value='u') (385, 0) 0.0
--->FeatureVal(feature=11, value='r') (214, 402) 0.481749259009
---->FeatureVal(feature=22, value='u') (147, 0) 0.0
---->FeatureVal(feature=22, value='d') (67, 402) 0.591672778582
```

```
----->FeatureVal(feature=22, value='u') (67, 0) 0.0
----->FeatureVal(feature=22, value='u') (0, 402) 0.0
-->FeatureVal(feature=22, value='u') (0, 2397) 0.0
Accuracy on test data: 0.994454713494
```

with MIN_GAIN=1.0:

```
python id3.py train test 1.0
>FeatureVal(feature=22, value='u') (3122, 2878) 0.533228496556
Accuracy on test data: 0.513863216266
```