

Hacker Dojo Machine Learning
Homework 2
Mike Bowles, PhD. and Patricia Hoffman, PhD.

The Entropy of a node is defined by equation(1), where $p(i|\text{node})$ denotes the fraction of records belonging to class i at the given node. The Gini index and Classification Error are defined in the equations (2) and (3) respectively.

$$\text{Entropy}(\text{node}) = - \sum_{i=0}^{c-1} p(i|\text{node}) \log_2 p(i|\text{node}) \quad (1)$$

$$\text{Gini}(\text{node}) = 1 - \sum_{i=0}^{c-1} [p(i|\text{node})]^2 \quad (2)$$

$$\text{Classification Error}(\text{node}) = 1 - \max_i [p(i|\text{node})] \quad (3)$$

If $I(\text{node})$ is any one of the three impurity measure given in the first equation, the Gain in Purity is defined by the following equation:

$$\text{Gain in Purity} = I(\text{parent}) - \sum_{j=1}^k \frac{N(\nu_j)}{N} I(\nu_j) \quad (4)$$

where N is the total number of records at the parent node, k is the number of attribute values, and $N(\nu_j)$ is the number of records associated with each child node, ν_j .

$I(\text{node})$ can be any of the three impurity measures defined by the first three equations. The best split for a node is the one that maximizes the Gain in Purity. When $I(\text{node})$ is Entropy the Gain in Purity is called Information Gain

Homework on Trees

1) For binary classification, consider the training examples in the table below (shown in Table 4.8 from the book on page 198). For this problem ignore columns a_1 and a_2 . For column a_3 , which is a continuous attribute, compute the Information Gain (using entropy as the purity measure) for every possible split. Where would be the best place to split the attribute a_3 ? Next compute the classification error for every possible split of that same variable. Using classification error as the impurity measure, where would be the best place to split the attribute a_3 ? Is there a difference in where to make the split? (You can code this in r or you can do this by hand.)

Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	-
4	F	F	4.0	+
5	F	T	7.0	-
6	F	T	3.0	-
7	F	F	8.0	-
8	T	F	7.0	+
9	F	T	5.0	-

2) The following tree was created using rpart for the table given in this homework problem number one.

node), split, n, loss, yval, (yprob)
 * denotes terminal node

```

1) root 9 4 0 (0.5555556 0.4444444)
  2) a1< 0.5 5 1 0 (0.8000000 0.2000000)
    4) a2>=0.5 3 0 0 (1.0000000 0.0000000) *
    5) a2< 0.5 2 1 0 (0.5000000 0.5000000)
      10) a3>=6 1 0 0 (1.0000000 0.0000000) *
      11) a3< 6 1 0 1 (0.0000000 1.0000000) *
  3) a1>=0.5 4 1 1 (0.2500000 0.7500000)
    6) a2< 0.5 2 1 0 (0.5000000 0.5000000)
      12) a3< 6 1 0 0 (1.0000000 0.0000000) *
      13) a3>=6 1 0 1 (0.0000000 1.0000000) *
    7) a2>=0.5 2 0 1 (0.0000000 1.0000000) *
```

Use this tree to predict the class labels (either a + or -) for the following test observations:

Observation	a_1	a_2	a_3
1	T	T	2.5
2	T	F	5.5
3	F	T	2.5
4	F	F	8.5

3) Consider the table given in the text on page 200 in the book exercise number five (copied below). It is a binary class problem. Determine the best places to split the data for a decision tree. The first time use the Gini Index as the impurity measure. The second time use the classification error as the impurity measure. In both cases use the Gain in Purity to determine the best split. Note any changes relative to the tree built from the two methods.

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	-
T	T	+
F	F	-
F	F	-
F	F	-
T	T	-
T	F	-

4) The UC Irvine web site has many interesting data sets. Sonar data is described at the web site: <http://archive.ics.uci.edu/ml/machine-learning-databases/undocumented/connectionist-bench/sonar/sonar.names> The sonar data set can be found at <http://archive.ics.uci.edu/ml/machine-learning-databases/undocumented/connectionist-bench/sonar/> Divide the sonar data set into a training set ([sonar_train.csv](#)) and a test set ([sonar_test.csv](#)). Use R to compute the classification error on the test set when training on the training set for a tree of depth 5 using all the default values except `control=rpart.control(minsplit=0,minbucket=0,cp=-1, maxcompete=0, maxsurrogate=0, usesurrogate=0, xval=0,maxdepth=5)`. Remember that the 61st column is the response and the other 60 columns are the predictors. Documentation for the rpart package can be found at <http://cran.r-project.org/web/packages/rpart/rpart.pdf>

5) Check out the web page which describes a wine quality data set:

<http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality.names>

Use the Red Wine data set: [winequality-red.csv](#) This data set contains 1599 observations of 11 attributes. The median score of the wine tasters is given in the last column. Note also that the delimiter used in this file is a semi colon and not a comma. Use rpart on this data to create trees for a range of different tree depths. Use cross validation to generate training error and test error. Plot these errors as a function of tree depth. Which tree depth results in the best Test Error? What is that Test Error? Hint: look at the cross validation example given in the lecture.