Using Information Gain for Node Splitting in Decision Trees

This week you will learn an additional method for selecting features to use when building decision trees. Specifically, you will learn about *information gain*, which is a measure of the “reduction in *entropy* (i.e., *disorder*)” of a candidate split *s* at node *t* of a decision tree[[1]](#footnote-1). In engineering applications, *information* is analogous to *signal*,and *entropy* is analogous to *noise*. The optimal split is whichever split maximizes the gain in information (or, equivalently, maximizes the reduction in entropy) over all possible splits at node *t*.

As an alternative to using either “TP-FP” or F(*s*,*t*) to select a feature F to split a node of a decision tree, you will select features that maximize the information gain (i.e., that maximize the entropy reduction).

The *entropy* at node *t* of a decision tree is defined as

H(*t*) = -[*pC,t* log2(*pC,t*) + *pNC,t* log2(*pNC,t*)]

Recall the following definitions from last week’s activity:

* + - * 1. *pC,t* = *n*(*t,* C) / *n*(*t*)

(probability of selecting a class ‘C’ sample at node *t*)

* + - * 1. *pNC,t* = *n*(*t,* NC) / *n*(*t*)

(probability of selecting a class ‘NC’ sample at node *t*)

Sometimes your program may need to calculate the value of log2(0).

Thus, by convention, we define log2(0) = 0.

The entropy of *s*,a candidate split at node *t* of a decision tree, is computed as

H(*s*,*t*) = *PLH(tL) + PRH(tR)*

Recall the following definitions from last week’s activity:

* + - * 1. *PL* = *n*(*tL*) / *n*(*t*)
        2. *PR* = *n*(*tR*) / *n*(*t*)

We may then define *information gain* of split *s* to be

gain(*s*) = H(*t*) - H(*s*,*t*)

Thus, information gain is the increase in information (equivalently, the reduction in entropy) produced by partitioning the samples according to candidate split *s*. At each decision node, the decision tree construction algorithm chooses the optimal split (based on a particular feature F) to be the split that has the greatest information gain, gain(*s*).

Complete the following activities no later than Thursday October 21, 11:59 pm.

PART 1: *using* gain(*s*) *to* *split the root node*

In this activity you will select the best feature (genetic mutation) to split the root node of your decision tree, by identifying the feature F that maximizes the value of gain(*s*). Complete the following activities.

1. In this activity you should compute the values of the following for the root node (denoted as ‘*t*’):
   * + - *n*(*t*), *n*(*t,* C), and *n*(*t,* NC)
       - *pC,t* = *n*(*t,* C) / *n*(*t*) (probability of selecting a class ‘C’ sample at node *t*)
       - *pNC,t* = *n*(*t,* NC) / *n*(*t*) (probability of selecting a class ‘NC’ sample at node *t*)
       - H(*t*)
2. Additionally, you should produce a table that lists the top 10 features in descending order by their gain(*s*) values. For each of the top 10 features, the table should contain the following (as illustrated in Table 1, below):
   * + - the identifier of the specific genetic mutation

(e.g., TEX36\_GRCh37\_10:127371546-127371546\_Nonsense-Mutation\_SNP\_G-G-A)

* + - * *n*(*tL*) - number of samples at *tL*
      * *n*(*tR*) - number of samples at *tR*
      * *n*(*tL*, C) - number of class ‘C’ samples at *tL*
      * *n*(*tL*, NC) - number of class ‘NC’ samples at *tL*
      * *n*(*tR*, C) - number of class ‘C’ samples at *tR*
      * *n*(*tR*, NC) - number of class ‘NC’ samples at *tR*
      * *PL* = *n*(*tL*) / *n*(*t*)
      * *PR* = *n*(*tR*) / *n*(*t*)
      * H(*s*,*t*)
      * H(*t*)
      * gain(*s*)

Table 1. Feature table template for the top features for splitting the root node, based on **F**(s,t) values.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Genetic Mutation** | *n*(*tL*) | *n*(*tR*) | *n*(*tL*, C) | *n*(*tL*, NC) | *n*(*tR*, C) | *n*(*tR*, NC) | *PL* | *PR* | *H*(*s*,*t*) | *H*(*t*) | gain(*s*) |
| GOT1\_GRCh37\_10:101163586-101163586\_Missense-Mutation\_SNP\_C-C-T |  |  |  |  |  |  |  |  |  |  |  |
| TEX36\_GRCh37\_10:127371546-127371546\_Nonsense-Mutation\_SNP\_G-G-A |  |  |  |  |  |  |  |  |  |  |  |
| KIAA1217\_GRCh37\_10:24810824-24810824\_Missense-Mutation\_SNP\_C-C-T |  |  |  |  |  |  |  |  |  |  |  |

PART 2: *completing and evaluating your decision tree*

1. Use gain(*s*) to find the best feature (genetic mutation) for splitting the left child of the root node of your decision tree.
2. Use gain(*s*) to find the best feature (genetic mutation) for splitting the right child of the root node of your decision tree.
3. Manually draw the resulting decision tree.
4. Define the specific classification rules represented in your decision tree. Note that the classification rules for decision trees constructed using gain(*s*) are different from the classification rules that you used previously (when using “TP-FP”), as described below.

The class represented by a leaf node is the class of the *majority of samples at the leaf node*. For example, if a leaf node **L** contains ***X***cancer samples and ***Y*** non-cancer samples, then upon reaching leaf node **L** a sample ***S*** would be classified as follows:

if **X > Y**

then classify **S** as **C**

else classify **S** as **NC**

Specifically, a decision tree can be used to classify a sample **S** by using the following generic classification rules:

if **S** has mutation **F** then

if **S** has mutation **A** then

if leaf node A1 has more cancer samples than non-cancer samples

then classify **S** as **C**

else classify **S** as **NC**

else

if leaf node A2 has more cancer samples than non-cancer samples

then classify **S** as **C**

else classify **S** as **NC**

else

if **S** has mutation **B** then

if leaf node B1 has more cancer samples than non-cancer samples

then classify **S** as **C**

else classify **S** as **NC**

else

if leaf node B2 has more cancer samples than non-cancer samples

then classify **S** as **C**

else classify **S** as **NC**

You should show the SPECIFIC classification rules that show EXACTLY how your decision tree would classify a sample **S**. For example, assume that the majority classes in the leaf nodes of your tree are as follows:

* Leaf node A1: contains *more cancer (C) samples* than non-cancer samples
* Leaf node A2: contains *more cancer (C) samples* than non-cancer samples
* Leaf node B1: contains *more cancer (C) samples* than non-cancer samples
* Leaf node B2: contains *more non-cancer (NC) samples* than cancer samples

In this case, the specific classification rules for the decision tree would be as

follows:

if **S** has mutation **F** then if **S** has mutation **A**

then classify **S** as **C**

else classify **S** as **C**

else if **S** has mutation **B**

then classify **S** as **C**

else classify **S** as **NC**

1. Use 3-fold cross-validation to evaluate the decision tree that resulted from using gain(*s*) to select features for node splitting. Report the resulting evaluation measures.
2. Compare the performance of your decision tree constructed using gain(*s*) to the performance of your decision trees constructed using “TP-FP” and F(*s*,*t*).

**Concepts learned:**

* The concepts of entropy, information, and information gain in the context of the classification problem.
* The objective of the information gain method for producing decision trees.
* The purpose of the gain(*s*), H(*t*), and H(*s*,*t*) functions.
* What kinds of splits the gain(*s*) function prefers.
* Calculation of the gain(*s*) H(*t*), and H(*s*,*t*) functions.
* Decision rules (classification rules) for use with decision trees constructed using the gain(*s*) function.

Complete all activities no later than Thursday October 21, 11:59 pm.

Submit an email to [welch@ohio.edu](mailto:welch@ohio.edu) that contains a brief report, including the following:

* For each fold of cross-validation show
  1. A table that lists the top 10 features in descending order by their gain(*s*) values. For each of the top 10 features, the table should contain the items listed in Table 1 (above).
  2. Drawings of your decision tree generated using information gain.
  3. Your specific classification rules generated using information gain.
* A discussion and interpretation of the cross-validation and comparison activities.
* Your report should include the following sections:
  1. Goals and objectives (2-3 bullet points)
  2. Summary of results (2-3 bullet points)
  3. Summary of methods (2-4 bullet points)
  4. Key results (tables, figures, stats, lists, etc.)
  5. Discussion (2-4 bullet points)

Additionally, attach the computer program that you developed for this activity and the output of your program (either a screenshot(s) or a file).

**NOTE***: you must develop your own computer program to accomplish this assignment. You ARE NOT permitted to use pre-existing programs for building decision trees or any other component of this project.*

**NOTE**: I may respond to your email submissions with questions about your methods, results, and/or interpretation. Please respond promptly to my questions.

1. Explanations and formulas in this document are based on section 8.4 of *Discovering Knowledge in Data*. D.R. Larose and C.D. Larose. Wiley. 2014, which is available at <https://alice.library.ohio.edu/record=b5187242?>. [↑](#footnote-ref-1)