The F Function for Node Splitting in Decision Trees

Perspective:

At this point in the Data Mining course, we have produced a software pipeline for classifying samples. We are laying the foundation for building a rigorous methodology that will be able to find genetic mutations that characterize cancer samples, which can lead to

(1) improved diagnostics, (2) insights into the pathology of the disease, and (3) treatment strategies.

Of course, we will not be able to completely solve such ambitious medical and scientific goals in one semester. However, over the next few weeks, we will focus on enhancing the rigor of our classification model. An obvious enhancement would be to increase the depth of the decision tree; the process for creating a deeper decision tree should be clear at this point in the course. Thus, we will focus on learning advanced techniques for node splitting and for classifier construction. In addition to learning general techniques for solving classification problems, we will also learn how to interpret our models in the context of the larger goals of cancer research, yielding results that will serve as a strong starting point for unraveling the biological story of the cancer phenotype.

There will be no Canvas quiz this week. Instead, you will learn new concepts via an in-class two-part data mining project. Results for part 1 are due at the start of class on Thursday, and results for part 2 are due at the start of class on the following Tuesday. Before coming to class on Tuesday, watch Prof. Welch’s video introduction for this week.

Professor Welch’s Introduction to the F Function:

* <https://www.youtube.com/watch?v=3U4InmGOxkA>

Data Mining Activity:

This week, you will learn a new method for selecting features to use when building decision trees. Specifically, you will learn about the F function, which is a measure of the “goodness” of a candidate split *s* at node *t* of a decision tree[[1]](#footnote-1). The optimal split is the one that maximizes the measure F(*s*,*t*) over all possible splits at node *t*.

As an alternative to using the quantity “**TP**-**FP**” to select a feature F to split a node of a decision tree, you will select features that maximize the value of the F(*s*,*t*) measure, which prefers splits that

1. are homogeneous (have samples from only one class) and
2. have roughly equal numbers of records.

The F measure is defined as F(*s*,*t*) = 2*PLPR* \* *Q*(*s*|*t*).

The first component of the F function, 2*PLPR*, is maximized when the proportions of samples in the left and right child nodes are equal. Therefore, F(*s*,*t*) will tend to favor balanced splits that partition the data into child nodes containing equal numbers of records.

The second component of the F function, *Q*(*s*|*t*), is maximized when the proportions of samples in the child nodes for each class (i.e., C and NC) are as different as possible. The maximum value, therefore, would occur when, for each class, the child nodes are completely uniform (pure).

PART 1: *using* F(*s*,*t*) *to* *split the root node* (in-class activity for Tuesday)

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 F(*s*,*t*) = 2*PLPR* \* *Q*(*s*|*t*). The formulas for computing 2*PLPR* and *Q*(*s*|*t*) are explained below. Complete the following activities before the next class.

The following symbology is used in the formulas for the components of F(*s*,*t*):

* + *t –* a node of the decision tree that needs to be split
  + properties of *t*:
  + *n*(*t*) - number of samples at node *t*
  + *n*(*t,* C) - number of class ‘C’ samples at node *t*
  + *n*(*t,* NC) - number of class ‘NC’ samples at node *t*
  + *s –* a candidate split(based on feature F) at node *t* of a decision tree
  + properties of *s*:
    - *tL* – left child of node *t*
    - *tR* – right child of node *t*
    - *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*
    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*)
    2. Additionally, you should produce a table that lists the top 10 features in descending order by their F(*s*,*t*) 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*
      * *PL* = *n*(*tL*) / *n*(*t*)
      * *PR* = *n*(*tR*) / *n*(*t*)
      * *P*(C| *tL*) = *n*(*tL*, C) / *n*(*tL*)
      * *P*(NC| *tL*) = *n*(*tL*, NC) / *n*(*tL*)
      * *P*(C| *tR*) = *n*(*tR*, C) / *n*(*tR*)
      * *P*(NC| *tR*) = *n*(*tR*, NC) / *n*(*tR*)
      * *Q*(*s*|*t*)= |*P*(C| *tL*) - *P*(C| *tR*)| + |*P*(NC| *tL*) - *P*(NC| *tR*)|
      * F(*s*,*t*) = 2*PLPR* \* *Q*(*s*|*t*)

Table . 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* | *P*(C| *tL*) | *P*(NC| *tL*) | *P*(C| *tR*) | *P*(NC| *tR*) | 2*PLPR* | *Q* | F(*s*,*t*) |
| 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 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Be prepared to present your results and methods for Part 1 in class on Thursday.

PART 2: *completing and evaluating your decision tree* (in-class activity for Thursday)

Be prepared to demonstrate your approach in the next class session.

1. Use F(*s*,*t*) to find the best feature (genetic mutation) for splitting the left child of the root node of your decision tree.
2. Use F(*s*,*t*) 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 F(s,t) are different from the classification rules that you used previously***, 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 F(*s*,*t*) to select features for node splitting. Report the resulting evaluation measures.
2. Compare the performance of your decision tree constructed using F(*s*,*t*) to the performance of your decision tree constructed using “TP-FP.”

**Concepts learned:**

* The objective of the CART method for producing binary decision trees.
* The purpose of the Q component of the f function.
* The purpose of the 2PLPR component of the f function.
* What kinds of splits the f function prefers.
* Calculation of the f function of the CART method for producing binary decision trees.
* Decision rules (classification rules) for use with binary decision trees constructed using the f function.

1. Explanations and formulas in this document are based on section 8.3 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)