Decision Tree: nagdmc_predict_gini_tree

Purpose

nagdmc_predict_gini_tree uses a decision tree computed by **nagdmc_gini_tree** to predict values of data records.

Declaration

```
#include <nagdmc.h>
void nagdmc_predict_gini_tree(long rec1, long nvar, long nrec, long dblk,
                              double data[], long yvar, long bcat[], long iproot,
                              int optrand, long iseed, double res[], double acc[],
                              int *info);
```

Parameters

1: **rec1** – `long` *Input*

On entry: the index in the data of the first data record used in the analysis.

Constraint: **rec1** ≥ 0.

2: **nvar** – `long` *Input*

On entry: the number of variables in the data.

Constraint: **nvar** > 1.

3: **nrec** – `long` *Input*

On entry: the number of consecutive records, beginning at **rec1**, used in the analysis.

Constraint: **nrec** > 1.

4: **dblk** – `long` *Input*

On entry: the total number of records in the data block.

Constraint: **dblk** $\geq$ **rec1** + **nrec**.

5: **data**[**dblk** $*$ **nvar**] – `double` *Input*

On entry: the data values for the jth variable (for $j = 0, 1, \ldots,$ **nvar**-1) are stored in **data**[$i*$**nvar**$+j$], for $i = 0, 1, \ldots,$ **dblk** $- 1$.

6: **yvar** – `long` *Input*

On entry: the index of the dependent variable in the analysis. Indexes of variables used to predict values with a given tree lattice must be the same as those indexes used to compute the lattice.

Constraints: $0 \leq$ **yvar** $<$ **nvar**.

7: **bcat**[**nvar**] – `long` *Input*

On entry: **bcat**[i] contains the base level value for the **ncat**[i] categories on the ith variable. If **ncat**[i] > 0, for $i = 0, 1, \ldots,$ **nvar** $- 1$, the categorical values on the ith variable are given by **bcat**[i] $+ j$, for $j = 0, 1, \ldots,$ **ncat**[i] $- 1$; otherwise **bcat**[i] is not referenced. If the base level for each categorical variable is zero, **bcat** can be 0.

8: **iproot** – `long` *Input*

On entry: the integer value of the root node of a decision tree as returned by **nagdmc_gini_tree**.

9: **optrand** – `int` *Input*

On entry: if the value of **optrand** is set equal to 1, a random number will be used to resolve dichotomies in the decision tree; otherwise **optrand** must be set equal to 0 and some data records may be unclassified, i.e., will be classified as -1.

Constraint: **optrand** $\in \{0, 1\}$.

10: **iseed** – `long` *Input*

On entry: if **optrand** $= 1$, the initial values used to set the seed of the random number generator used to resolve any dichotomies in the tree; otherwise **optrand** is not referenced.

11: **res[nrec]** – `double` *Output*

On exit: **res**[i] contains the decision tree prediction for the (**rec1** + i)th data record, for $i = 0, 1, \ldots,$ **nrec** $- 1$.

12: **acc[nrec]** – `double` *Output*

On exit: **acc**[i] contains the probability of class membership, based on the training data, at the leaf node giving the ith prediction, for $i = 0, 1, \ldots,$ **nrec** $- 1$.

13: **info** – `int *` *Output*

On exit: **info** gives information on the success of the function call:

0: the function successfully completed its task.

-32: a path down the decision tree could not be found for at least one data record, consequently not all data records have been classified; this warning can be avoided by setting **optrand** equal to one.

i; $i = 1, 2, 3, 4, 6, 9$: the specification of the ith formal parameter was incorrect.

99: the function failed to allocate enough memory.

> 100: an error occurred in a function specified by the user.

Notation

nrec	the number of data records used to predict values, n.
data	data records x_i, for $i = 1, 2, \ldots, n$.
res	decision tree classifications y_i, for $i = 1, 2, \ldots, n$.
acc	accuracy of classifications a_i, for $i = 1, 2, \ldots, n$.

Description

Let x_i, for $i = 1, 2, \ldots, n$ be a set of n data records not used to fit a decision tree, T. The ith predcition for the dependent variable in the data is found by using the outcome of a series of tests at the root node and internal nodes in T to associate x_i with leaf node l_i, for $i = 1, 2, \ldots, n$. The value of the dependent variable stored at l_i is then used as the predicted value y_i, for $i = 1, 2, \ldots, n$. In a decision tree calculated by using a Gini index criterion each leaf node stores the modal class of the dependent variable over a subset of the data records.

The outcome of each test depends on the type of variable used to partition data records at the node. Let a test at a node k be on variable j in the data and x_{ij} be the value of the ith data record on variable j.

If j is continuous, x_i is sent to the left child node of node k if $x_{ij} \leq t$, where t is the value of the continuous test as stored in node k; otherwise x_i is sent to the right child node of node k.

If j is categorical, x_i is sent to the node associated with the category value x_{ij}. However, when the decision was fitted there may not have been a category value x_{ij} at node k and, therefore, either the ith data record can be assigned an unclassified value or a child node can be chosen at random from those available to node k.

This process of evaluating tests continues until x_i reaches a leaf node, say l_i, in T.

A measure of the accuracy of the ith prediction can be obtained by considering the class distrbution of data records at leaf node l_i, for $i = 1, 2, \ldots, n$. Suppose that r_i of m_i data records associated with l_i (and used to fit T) belong to the modal class, then a measure of the accuracy a_i of the classification is given by,

$$a_i = \frac{r_i}{m_i}, \quad i = 1, 2, \ldots, n.$$

References and Further Reading

None.

See Also

gini_tree_ex.c the example calling program.
