DOT extension — Handling DOT files[+]

<div align="center">Description   Remarks and examples   Also see</div>

Description

This entry provides a brief introduction to the DOT language and DOT files. These DOT files, which can be created by h2omltree, can be converted into images of decision trees.

The open source software Graphviz can be used to convert DOT files to images.

stata.com

Remarks and examples

Remarks are presented under the following headings:

> *Install Graphviz*
> *How to use Graphviz and DOT language*
> *Modifying the DOT file*

Install Graphviz

Graphviz is available for most operating systems. For the steps to download and install Graphviz, see https://graphviz.org/download/. If prompted during installation, you can allow Graphviz to be installed on the system path so that Graphviz commands can be issued from the terminal and issued from the Command window of Stata using the shell command. For the rest of this entry, we assume that Graphviz is installed.

How to use Graphviz and DOT language

Instead of providing extensive details of DOT language, we will explain by example and focus on options that are relevant to our goal.

First, we open the 1978 automobile data (auto.dta) in Stata and then put the data into an H2O frame. Recall that h2o init initiates an H2O cluster, _h2oframe put loads the current Stata dataset in an H2O frame, and _h2oframe change makes the specified frame the current H2O frame.

```
. use https://www.stata-press.com/data/r18/auto
(1978 automobile data)
. h2o init
  (output omitted )
. _h2oframe _put, into(auto)
. _h2oframe _change auto
```

Next, we perform gradient boosting regression and specify h2orseed(19) for reproducibility.

```
. h2oml gbregress price make mpg, h2orseed(19)
  (output omitted )
```

Finally, we use the h2omltree command to save the second tree in a file called example.dot.

```
. h2omltree, id(2) dotsaving(example.dot, replace)
```

1

The code below is the content of the `example.dot` file. You can look through the content of DOT files using your preferred text editor.

```
digraph G {
rankdir = TB
/* Level 0 */
{
"Node_0" [shape=box, fontsize=20, label="mpg"]
}
/* Level 1 */
{
"Node_9" [fontsize=20, label="286.207"]
"Node_2" [shape=box, fontsize=20, label="mpg"]
}
/* Level 2 */
{
"Node_3" [shape=box, fontsize=20, label="mpg"]
"Node_10" [fontsize=20, label="-172.209"]
}
/* Level 3 */
{
"Node_11" [fontsize=20, label="-125.564"]
"Node_6" [shape=box, fontsize=20, label="mpg"]
}
/* Level 4 */
{
"Node_12" [fontsize=20, label="15.111"]
"Node_13" [fontsize=20, label="-78.548"]
}
/* Edges */
"Node_0" -> "Node_9" [fontsize=20, label="< 17.5
"]
"Node_0" -> "Node_2" [fontsize=20, label="[NA]
>= 17.5
"]
"Node_2" -> "Node_3" [fontsize=20, label="[NA]
< 27.0
"]
"Node_2" -> "Node_10" [fontsize=20, label=">= 27.0
"]
"Node_3" -> "Node_11" [fontsize=20, label="< 20.5
"]
"Node_3" -> "Node_6" [fontsize=20, label="[NA]
>= 20.5
"]
"Node_6" -> "Node_12" [fontsize=20, label="[NA]
< 23.5
"]
"Node_6" -> "Node_13" [fontsize=20, label=">= 23.5
"]
fontsize=40
labelloc="t"
label = "Tree 2"
}
```

The file provides information about nodes of each level in the tree. For example, `Node_2` and `Node_9` belong to level 1. By default, the file provides information about the shape of the node, font size, and label. Those entries can be modified and other options can be added to describe the node. The `Edges` section in the file provides information about the structure of the tree, that is, which nodes are connected and how.
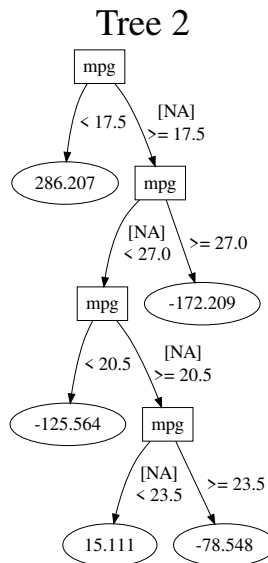
To create a PDF file with a diagram of this tree with Graphviz, we type in Stata

```
. shell dot -Tpdf example.dot -o example.pdf
```

and to create the diagram as a PNG image, we type

```
. shell dot -Tpng example.dot -o example.png
```

The `shell` command of Stata allows you to send commands to the operating system. For details, see [D] **shell**. The resulting tree is shown below.
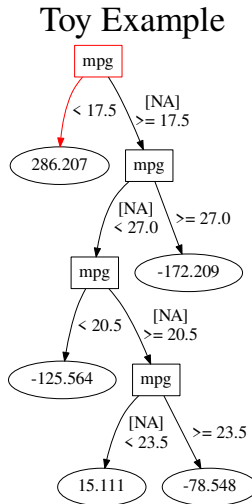

Tree 2

Modifying the DOT file

Having a DOT file gives us the flexibility to modify the tree based on our preference. For example, in the code below, we change the title to "Toy Example", the contour of the Node_0 to red, and the color of the left edge emanating from the Node_0 also to red. Note that the title also can be changed using the `title()` option in h2omltree. Changes are highlighted in bold.

```
digraph G {
rankdir = TB
/* Level 0 */
{
"Node_0" [shape=box, fontsize=20, label="mpg", color = "red"]
}

/* Level 1 */
{
"Node_9" [fontsize=20, label="286.207"]
"Node_2" [shape=box, fontsize=20, label="mpg"]
}

/* Level 2 */
{
"Node_3" [shape=box, fontsize=20, label="mpg"]
"Node_10" [fontsize=20, label="-172.209"]
}

/* Level 3 */
{
"Node_11" [fontsize=20, label="-125.564"]
"Node_6" [shape=box, fontsize=20, label="mpg"]
}

/* Level 4 */
{
"Node_12" [fontsize=20, label="15.111"]
"Node_13" [fontsize=20, label="-78.548"]
}

/* Edges */
"Node_0" -> "Node_9" [fontsize=20, label="< 17.5
", color = "red"]
"Node_0" -> "Node_2" [fontsize=20, label="[NA]
>= 17.5
"]
"Node_2" -> "Node_3" [fontsize=20, label="[NA]
< 27.0
"]
"Node_2" -> "Node_10" [fontsize=20, label=">= 27.0
"]
"Node_3" -> "Node_11" [fontsize=20, label="< 20.5
"]
"Node_3" -> "Node_6" [fontsize=20, label="[NA]
>= 20.5
"]
"Node_6" -> "Node_12" [fontsize=20, label="[NA]
< 23.5
"]
"Node_6" -> "Node_13" [fontsize=20, label=">= 23.5
"]
fontsize=40
labelloc="t"
label = "Toy Example"
}
```

The following plot depicts the changes.

Toy Example

```
                    mpg
              < 17.5 /    \ [NA]
                    /      \ >= 17.5
            286.207        mpg
                     [NA] /   \ >= 27.0
                   < 27.0/     \
                      mpg      -172.209
               < 20.5 /   \ [NA]
                     /     \ >= 20.5
            -125.564       mpg
                     [NA] /   \ >= 23.5
                   < 23.5/     \
                  15.111       -78.548
```

Also see

[H2OML] **h2oml** — Introduction to commands for Stata integration with H2O machine learning[+]