h2omlgraph varimp — Produce variable importance plot

Description

`h2omlgraph varimp` plots the variable importance after `h2oml` *gbm* and `h2oml` *rf*. Variable importance for ensemble decision tree methods, such as random forest and gradient boosting machine, measures the relative influence of a predictor to the predictive performance of the model.

Quick start

Plot the variable importance

```
h2omlgraph varimp
```

Same as above, but plot the top 5 important predictors

```
h2omlgraph varimp, top(5)
```

Plot scaled importance of predictors

```
h2omlgraph varimp, scaled
```

Plot variable importance as a dot graph

```
h2omlgraph varimp, dot
```

Same as above, but save the graph data

```
h2omlgraph varimp, dot savedata(varimp)
```

Menu

Statistics > H2O machine learning

Syntax

```
h2omlgraph varimp [ , options ]
```

options	Description
Main	
top(#)	plot the top # important predictors; default is top(10)
proportion	plot the proportional contribution of the importance of each predictor; the default
relative	plot relative influence of each predictor
scaled	plot scaled importance of each predictor
table	display results as a table
savedata(*filename*[, replace])	save plot data to *filename*
Plot options	
bar	plot variable importance as a bar plot; the default
baropts(*bar_opts*)	affect rendition of the bar plot
dot	plot variable importance as a dot plot
dotopts(*dot_opts*)	affect rendition of the dot plot
valuelabel	display variable importance values
valuelabelopts(*label_opts*)	affect the labeling of important values
twoway_options	any options other than by() documented in [G-3] ***twoway_options***

Options

Main

top(#) plots the top # important predictors. The default is top(10).

proportion, relative, and scaled specify the type of the variable importance contribution to be plotted.

proportion plots the proportional contribution of the importance of each predictor. It is calculated by dividing the importance of each predictor by the total sum of the importance of all predictors. proportion is the default.

relative plots the importance, which is the relative influence of each predictor.

scaled plots the scaled importance. It is calculated by dividing the importance of each predictor by the largest importance score of the predictors.

Only one of proportion, relative, or scaled is allowed.

table displays results as a table. The table is suppressed by default.

savedata(*filename*[, replace]) saves the plot data to a Stata data file (.dta file). replace specifies that *filename* be overwritten if it exists.

Plot options

bar plots the variable importance as a bar plot. This is the default. bar is not allowed with dot.

`baropts(`*bar_opts*`)` affects rendition of the bar plot. *bar_opts* are any of the options documented in [G-2] **graph twoway bar**, excluding `horizontal` and `vertical`.

`dot` plots the variable importance as a dot plot. `dot` is not allowed with `bar`.

`dotopts(`*dot_opts*`)` affects the rendition of the dot plot. *dot_opts* are any of the options documented in [G-2] **graph twoway dot**, excluding `horizontal` and `vertical`.

`valuelabel` displays the values of the variable importance on the graph.

`valuelabelopts(`*label_opts*`)` affects the labeling of variable importance values. *label_opts* includes any of the options documented in [G-3] ***marker_label_options***, excluding `mlabel()`.

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`, `horizontal`, and `vertical`. These include options for titling the graph (see [G-3] ***title_options***) and options for saving the graph to disk (see [G-3] ***saving_option***).

Remarks and examples

We assume you have read the *Interpretation and explanation* in [H2OML] **Intro**.

In a typical machine learning problem, the predictors influence on the outcome differs. Some of the predictors are more relevant than others. In decision trees, the variable importance of a predictor quantifies this relevance by accumulating the improvement of an impurity measure, such as cross-entropy or mean squared error (MSE), from the splitting of this predictor. For a single tree T, Breiman et al. (1984) propose to measure a relative importance of a predictor $\mathbf{X}_i$ by summing the square of relative improvements $\imath_j^2$ associated to all $J-1$ node splits,

$$I_i^2(T) = \sum_{j=1}^{J-1} \imath_j^2 I(v(j) = i)$$

where the split relative improvement $\imath_j$ is defined in (1) of [H2OML] **Intro** and is computed using entropy for classification and MSE for regression. $I(v(j) = i)$ is an indicator function, which takes 1 when the internal node is the predictor X_i. This measure easily extends to ensemble decision trees by taking an average over the number of trees. For example, if the ensemble decision tree method contains 100 trees ($t = 1, 2, \ldots, 100$), then

$$I_i^2 = \frac{1}{100} \sum_{t=1}^{100} I_i^2(T_t)$$

To find the importance for the variable X_i, we take the square root of the measure above.

For multiclass classification with K classes ($k = 1, 2, \ldots K$), there are K different models induced, where each model is an ensemble of classification trees. Then for the class k with 100 trees, the importance of the predictor X_i is computed by

$$I_{ik}^2 = \frac{1}{100} \sum_{t=1}^{100} I_i^2(T_{tk})$$

where T_{tk} is the tth tree for the class k.

It is common to plot the proportional contributions of importance values so that the total importance of all predictors sums to 1. This approach makes it easier to compare predictors. In the `h2omlgraph varimp` command, this is the default behavior. To plot the relative influences, you can specify the `relative` option.

One of the main limitations of variable importance based on impurity measures is their bias toward predictors with more levels. Additionally, they are not reliable when predictors are correlated.

▷ Example 1: Plotting variable importance

In this example, we plot variable importance after performing random forest binary classification.

We consider the `churn` dataset described in example 1 of [H2OML] **h2oml** and where the goal is to build a predictive model that will predict the best behavior of a customer who is more likely to churn or retain the company's services.

We start by opening the `churn` dataset in Stata and then putting the data into an H2O frame. Recall that `h2o init` initiates an H2O cluster, `_h2oframe put` loads the current Stata dataset into an H2O frame, and `_h2oframe change` makes the specified frame the current H2O frame. For details, see *Prepare your data for H2O machine learning in Stata* in [H2OML] **h2oml** and [H2OML] **H2O setup**.

```
. use https://www.stata-press.com/data/r19/churn
(Telco customer churn data)
. h2o init
  (output omitted)
. _h2oframe put, into(churn)
Progress (%): 0 100
. _h2oframe change churn
```

For convenience, we save the name of the predictors in the global macro `predictors` in Stata.

```
. global predictors latitude longitude tenuremonths monthlycharges
> totalcharges gender seniorcitizen partner dependents phoneservice
> multiplelines internetserv onlinesecurity onlinebackup deviceprotect
> techsupport streamtv streammovie contract paperlessbill paymethod
```

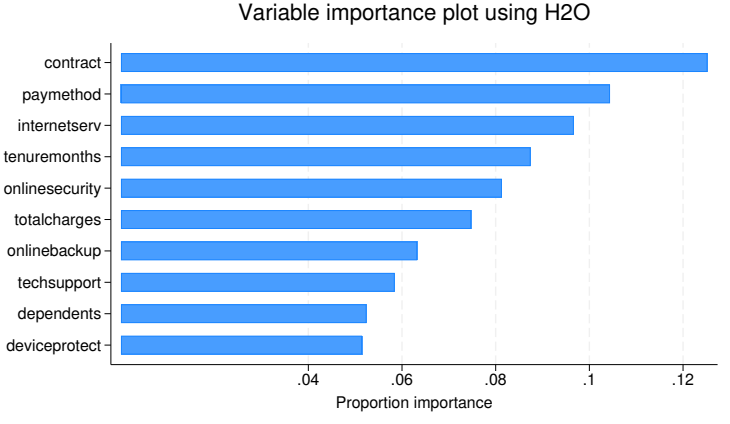
We use `h2oml rfbinclass` to perform random forest binary classification with 200 trees, a maximum tree depth of 3, an observation sampling rate of 0.9, and a predictor sampling value of 1. Then we use `h2omlgraph varimp` to plot the variable importance.

```
. h2oml rfbinclass churn $predictors, h2orseed(19) ntrees(200)
> maxdepth(3) samprate(0.9) predsampvalue(1)
Progress (%): 0 10.0 33.5 63.9 93.5 100
Random forest binary classification using H2O
Response: churn
Frame:                                  Number of observations:
  Training: churn                                 Training =  7,043
Model parameters
Number of trees      = 200
              actual = 200
Tree depth:                             Pred. sampling value =      1
           Input max =   3              Sampling rate        =     .9
                 min =   3              No. of bins cat.     =  1,024
                 avg = 3.0              No. of bins root     =  1,024
                 max =   3              No. of bins cont.    =     20
Min. obs. leaf split =   1              Min. split thresh.   = .00001
```

Metric summary

Metric	Training
Log loss	.480982
Mean class error	.2400372
AUC	.8284618
AUCPR	.6263171
Gini coefficient	.6569236
MSE	.1572825
RMSE	.3965886

```
. h2omlgraph varimp
```



The proportion of importance for the top 10 predictors is plotted. Based on this model, `contract`, `paymethod`, and `internetserv` are the three most important predictors of churn.

◁

▷ Example 2: Assessing stability of variable importance

Recent literature shows an increased attention on assessing stability of variable importance (Wang et al. 2016). In this example, we study the stability of variable importance by showing dependence of variable rankings from the predictor sampling number. That is, our goal is to vary the predictor sampling value `predsampvalue()` in random forest and explore the change in rankings of predictors based on the importance. Wang et al. (2016) implement a more extensive study and use rank-based tests to quantify stability. Our example is limited only to graphical comparison.

In the previous example, we specified a predictor sampling value of 1. Here we will compare this with the results using three other values. For convenience, we save a list of possible `predsampvalues` in the local macro `sratelist` in Stata.

```
. local sratelist 1 -1 10 -2
```

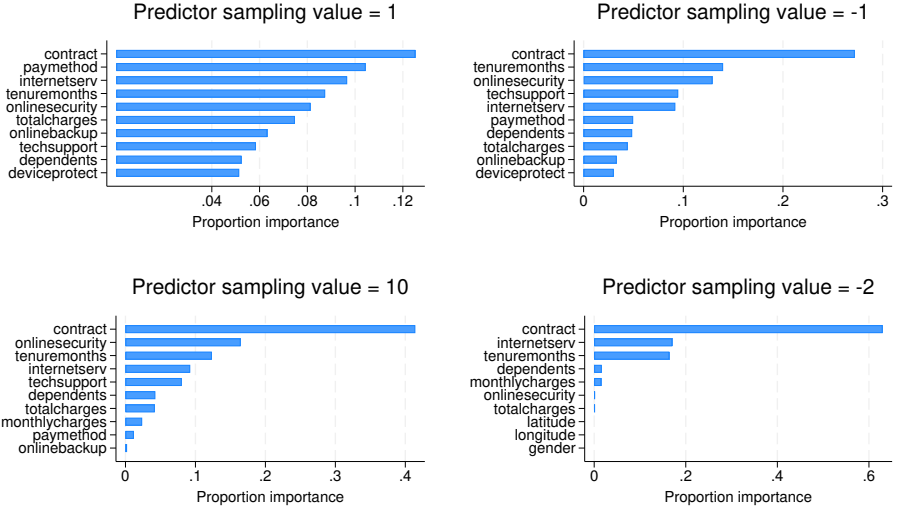
Next we use a loop to perform random forest binary classification with the predictor sampling values of $\{1, -1, 10, -2\}$, iteratively specifying each of these values in the `predsampvalue()` option of `h2oml rfbinclass`. We plot the variable importance after each estimation by using the `h2omlgraph varimp` command. Note that `predsampvalue(-2)` corresponds to selecting all predic-

tors, and `predsampvalue(-1)` corresponds to selecting the square root of the number of predictors. In `h2omlgraph varimp`, we also specify the option `saving()` to save the graphs and the option `title()` to provide a title for each graph.

```
. local i = 1
. foreach rate in ‘sratelist’{
  2.         quietly h2oml rfbinclass churn $predictors, h2orseed(19)
>         ntrees(200) maxdepth(3) samprate(0.9) predsampvalue(‘rate’)
  3.         h2omlgraph varimp, saving(imp‘i’, replace)
>         title("Predictor sampling value = ‘rate’")
  4.         local i = ‘i’ + 1
  5. }
file imp1.gph saved
file imp2.gph saved
file imp3.gph saved
file imp4.gph saved
```

Finally, we display the saved graphs by using the `graph combine` command in Stata.

```
. graph combine imp1.gph imp2.gph imp3.gph imp4.gph
```



As the predictor sampling value changes, except for the `contract` predictor, the ranking of the importance of predictors changes substantially, indicating instability in the variable importance measure. In practice, this instability can be explained as follows: For smaller numbers of sampled predictors, predictors with smaller effects are assigned greater importance. Conversely, for larger numbers of sampled predictors, such as when all predictors are sampled with `predsampvalue(-2)`, the random forest focuses on highly influential predictors, resulting in only a few predictors considered important.

◁

References

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*. Boca Raton, FL: Chapman and Hall/CRC.

Wang, L., C. S. McMahan, M. G. Hudgens, and Z. P. Qureshi. 2016. A flexible, computationally efficient method for fitting the proportional hazards model to interval-censored data. *Biometrics* 72: 222–231. https://doi.org/10.1111/biom.12389.

Also see

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