h2omlgraph scorehistory — Produce score history plot+

+This command includes features that are part of StataNow.

Description

`h2omlgraph scorehistory` plots the evolution of a performance metric (a score) as the number of trees grows in a machine learning model fit using either `h2oml` *gbm* or `h2oml` *rf*. The performance metric is based on the training set. If validation was specified during estimation, the performance metric on the validation set is also plotted. If cross-validation was specified during estimation, the performance metric based on the cross-validation results and based on the training on cross-validation results is also plotted.

Quick start

Plot the score history

```
h2omlgraph scorehistory
```

As above, but show the best score reference line

```
h2omlgraph scorehistory, bsline
```

Menu

Statistics > H2O machine learning

Syntax

`h2omlgraph scorehistory` [, *options*]

options	Description
Main	
metric(*metric*)	specify the metric (score) to be plotted
table	display results as a table
savedata(*filename*[, replace])	save plot data to *filename*
Plot options	
bsline	plot the best score reference line
bslineopts(*line_options*)	affect rendition of the best score reference line
lineopts(*line_options*)	affect rendition of all training, validation, and cross-validation curves
trainlineopts(*line_options*)	affect rendition of training curve
validlineopts(*line_options*)	affect rendition of validation curve
cvtrainlineopts(*line_options*)	affect rendition of the training on cross-validation curve
cvlineopts(*line_options*)	affect rendition of cross-validation curve
nocvtrainsd	do not plot the standard deviation band for the training on cross-validation curve
cvtrainsdopts(*area_options*)	affect rendition of the standard deviation band for the training on cross-validation curve
nocvsd	do not plot the standard deviation band for the cross-validation curve
cvsdopts(*area_options*)	affect rendition of the standard deviation band for the cross-validation curve
twoway_options	any options other than by() documented in [G-3] ***twoway_options***
trainopts(*line_options*)	synonym for trainlineopts()
validopts(*line_options*)	synonym for validlineopts()
cvtrainopts(*line_options*)	synonym for cvtrainlineopts()
cvopts(*line_options*)	synonym for cvlineopts()

Options

Main

metric(*metric*) specifies the metric to be plotted. The allowed options are the following:

After regression: deviance, rmse, and mae.

After binary classification: logloss, misclassification, auc, aucpr, and rmse.

After multiclass classification: logloss, misclassification, and rmse.

deviance is the default metric for regression. logloss is the default metric for binary and multiclass classification.

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

`bsline` plots the best score reference line for the training, validation, or cross-validation curve. The best score corresponds to the optimal training score (the optimal metric) if neither validation nor cross-validation is performed during estimation. When validation or cross-validation is performed, the best score corresponds to the optimal validation or cross-validation score, respectively.

`bslineopts(`*line_options*`)` affects rendition of the best score reference line. For options, see [G-3] ***line_options***.

`lineopts(`*line_options*`)` affects the rendition of both training and validation curves when `validframe()` is specified during estimation or the rendition of training, training on cross-validation, and cross-validation curves when `cv()` is specified during estimation. If neither `validframe()` nor `cv()` is specified, only training curve is affected. See [G-3] ***line_options***.

`trainlineopts(`*line_options*`)` affects the rendition of the training curve. See [G-3] ***line_options***.

`validlineopts(`*line_options*`)` affects the rendition of the validation curve when `validframe()` is specified during estimation. See [G-3] ***line_options***.

`cvtrainlineopts(`*line_options*`)` affects the rendition of the training on cross-validation curve when `cv()` is specified during estimation. During k-fold cross-validation, the training data are separated into k folds, from which $k - 1$ are used for training and 1 for prediction. The training on cross-validation curve plots the average across the k cross-validation iterations of the metrics computed on the training data (from $k - 1$ folds). See [G-3] ***line_options***.

`cvlineopts(`*line_options*`)` affects the rendition of the cross-validation curve when `cv()` is specified during estimation. See [G-3] ***line_options***.

`nocvtrainsd` suppresses plotting the standard deviation band for the mean training on cross-validation curve. The standard deviation band is included by default.

`cvtrainsdopts(`*area_options*`)` affects rendition of the standard deviation band for mean training on cross-validation metrics. See [G-3] ***area_options***.

`nocvsd` suppresses plotting the standard deviation band for the mean cross-validation curve.

`cvsdopts(`*area_options*`)` affects rendition of the standard deviation band for the mean cross-validation curve. See [G-3] ***area_options***.

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`. 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***).

`trainopts(`*line_options*`)` is a synonym for `trainlineopts()`.

`validopts(`*line_options*`)` is a synonym for `validlineopts()`.

`cvtrainopts(`*line_options*`)` is a synonym for `cvtrainlineopts()`.

`cvopts(`*line_options*`)` is a synonym for `cvlineopts()`.

stata.com

Remarks and examples

We assume you have read [H2OML] **Intro**.

Overfitting occurs when a machine learning model fits the training data too well. This harms the ability of the model to generalize to new data, increasing the generalization error. Underfitting occurs when performance can be improved by increasing complexity of the model by modifying the hyperparameters. The score history curve, also known as the learning curve, is a useful graphical tool for examining the overfitting or underfitting of a model. It plots a performance metric (a score) as a function of the number of trees and allows you to evaluate the optimal number of trees.

▷ Example 1: Over- and underfitting with score history

Consider `churn.dta`, 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. We use the `_h2oframe split` command to randomly split the `churn` frame into a training frame (80% of observations) and a validation frame (20% of observations), which we name `train` and `valid`, respectively. We also change the current frame to `train`. 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/r18/churn
(Telco customer churn data)
. h2o init
  (output omitted)
. _h2oframe put, into(churn)
Progress (%): 0 100
. _h2oframe split churn, into(train valid) split(0.8 0.2) rseed(19)
. _h2oframe change train
```

Next we define a global macro, `predictors`, to store predictors, and perform gradient boosting binary classification with 200 trees.

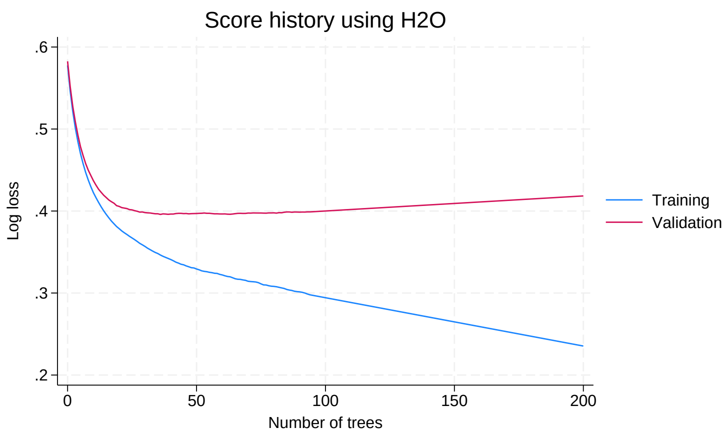
```
. global predictors latitude longitude tenuremonths monthlycharges
> totalcharges gender seniorcitizen partner dependents phoneservice
> multiplelines internetserv onlinesecurity onlinebackup deviceprotect
> techsupport streamtv streammovie contract paperlessbill paymethod
. h2oml gbbinclass churn $predictors, validframe(valid) ntrees(200) h2orseed(19)
Progress (%): 0 2.9 12.5 28.9 46.5 100
Gradient boosting binary classification using H2O
Response: churn
Loss:     Bernoulli
Frame:                                  Number of observations:
  Training:   train                                Training =  5,643
  Validation: valid                              Validation =  1,400
Model parameters
Number of trees      = 200              Learning rate       =     .1
              actual = 200              Learning rate decay =      1
Tree depth:                             Pred. sampling rate =      1
           Input max =   5              Sampling rate       =      1
                 min =   5              No. of bins cat.    =  1,024
                 avg = 5.0              No. of bins root    =  1,024
                 max =   5              No. of bins cont.   =     20
Min. obs. leaf split =  10              Min. split thresh.  = .00001
```

```
Metric summary
```

Metric	Training	Validation
Log loss	.2353826	.4184287
Mean class error	.0982787	.2314265
AUC	.9692747	.8515924
AUCPR	.9264498	.6724044
Gini coefficient	.9385495	.7031848
MSE	.0679986	.1370254
RMSE	.2607655	.3701694

Next we plot the score history curve by using the `h2omlgraph scorehistory` command.

```
. h2omlgraph scorehistory
Training frame:   train
Validation frame: valid
```



We can see that when the number of trees is fewer than 10, learning and generalization behave similarly. In other words, the log loss is similar for the training and validation data. For these small numbers of trees, the log-loss metric is large; the model is underfitting the training data, and performance can be improved. However, when the number of trees exceeds 40, the log-loss metric for the validation data starts to increase. Generalization stops improving, even though the training metrics continue to improve. This indicates that the model learns patterns specific to training data that cannot be extended to new data points. At this stage, the model is overfitting.

◁

▷ Example 2: Score history with cross-validation

In example 1, we used a validation frame during estimation. When cross-validation is used, the `h2omlgraph scorehistory` command provides not only the score history curves for cross-validation but also standard deviation bands for quantifying uncertainty.

We open auto.dta in Stata and then put it into an H2O frame. Because we are focused on evaluating cross-validation, we do not split the data into training and testing sets as we typically would in practice.

```
. use https://www.stata-press.com/data/r18/auto
(1978 automobile data)

. h2o init
  (output omitted)

. _h2oframe put, into(auto)
Progress (%): 0 100

. _h2oframe change auto
```

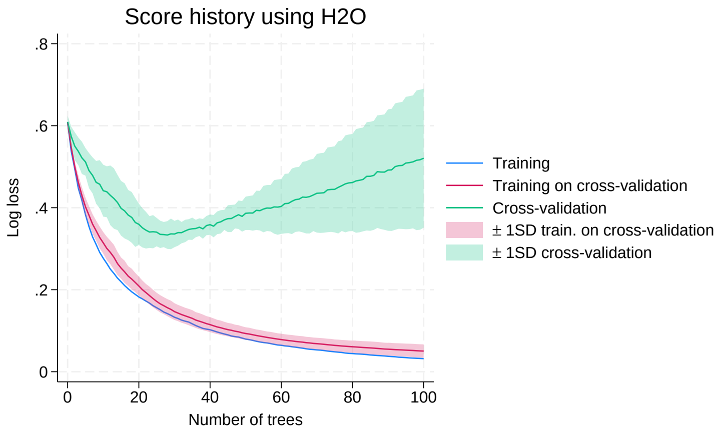
We perform gradient boosting binary classification with 3-fold cross-validation and use 100 trees.

```
. h2oml gbbinclass foreign price length weight trunk mpg, h2orseed(19)
> cv(3, modulo) ntrees(100)
Progress (%): 0 15.7 58.9 100

Gradient boosting binary classification using H2O

Response: foreign
Loss:     Bernoulli
Frame:                                   Number of observations:
  Training: auto                                    Training =     74
                                            Cross-validation =     74
Cross-validation: Modulo                 Number of folds     =      3

Model parameters

Number of trees      = 100               Learning rate       =     .1
              actual = 100               Learning rate decay =      1
Tree depth:                              Pred. sampling rate =      1
           Input max =   5               Sampling rate       =      1
                 min =   2               No. of bins cat.    =  1,024
                 avg = 4.3               No. of bins root    =  1,024
                 max =   5               No. of bins cont.   =     20
Min. obs. leaf split =  10               Min. split thresh.  = .00001

Metric summary
```

Metric	Training	Cross-validation
Log loss	.0319483	.5174966
Mean class error	0	.1153846
AUC	1	.9143357
AUCPR	1	.802104
Gini coefficient	1	.8286713
MSE	.0050191	.1460853
RMSE	.0708458	.3822111

Next we plot the score history using the `h2omlgraph scorehistory` command.

```
. h2omlgraph scorehistory
Training frame: auto
```



The band representing the cross-validation standard deviation, displayed in green, has an hourglass-like shape. The uncertainty is greater at the beginning, where the model is underfitting. It then narrows in the regions where the model's performance is likely to generalize well before widening again at the end, where the model overfits the training data.

◁

Also see

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

For suggested citations, see the FAQ on citing Stata documentation.