h2omlexplore — Explore models after grid search

Description | Quick start | Menu | Syntax
Remarks and examples | Stored results | Reference | Also see

Description

h2omlexplore allows you to compare models with different hyperparameter configurations after h2omlestat gridsummary. In the process of tuning hyperparameters with h2oml *gbm* and h2oml *rf*, you can use h2omlestat gridsummary to report the specified metric for different hyperparameter configurations. h2omlexplore allows you to further explore a few selected models by reporting several performance metrics.

Quick start

After performing multiclass classification and obtaining the grid-search summary, view the performance metrics of the models with IDs 2, 4, and 8

```
h2oml rfmulticlass y1 x1-x20, ntrees(10(5)100) maxdepth(3(1)10)
h2omlestat gridsummary
h2omlexplore id = 2 4 8
```

Menu

Statistics > H2O machine learning

Syntax

h2omlexplore id = # | *numlist*

where # is a grid ID from h2omlestat gridsummary corresponding to a model with the desired hyperparameter configuration, and *numlist* is a list of grid IDs.

Remarks and examples

Building a machine learning model that generalizes well to new data involves choosing an appropriate method and selecting a model by tuning hyperparameters. We can perform a grid search using gradient boosting and random forest methods and then use h2omlestat gridsummary to report the hyperparameter configurations that achieve the top performance based on the specified metric. In some cases, you may decide to choose the best-performing model reported in h2omlestat gridsummary; in other cases, you may want to explore other well-performing models further, which you can do using h2omlexplore. With h2omlexplore, you can report several performance metrics for models with different hyperparameter configurations.

▷ Example 1: Exploring different models

In example 1 of [H2OML] **h2omlselect**, we used the social pressure dataset (Gerber, Green, and Larimer 2008) to implement a hyperparameter tuning, and we used the `h2omlselect` command to select the second-best model, which was comparably less complex than the best model. In that example, our decision was based on the area under the precision–recall curve (AUCPR) metric. Suppose now we want to compare those two models based on different performance metrics to make sure that the same pattern holds.

We start by opening the social pressure 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 in an H2O frame, and `_h2oframe change` makes the specified frame the current H2O frame. We use the `_h2oframe split` command to randomly split the `social` frame into a training frame (80% 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 see [H2OML] **H2O setup**.

```
. use https://www.stata-press.com/data/r19/socialpressure
(Social pressure data)

. h2o init
  (output omitted)

. _h2oframe _put, into(social)
Progress (%): 0 100

. _h2oframe _split social, into(train valid) split(0.8 0.2) rseed(19)

. _h2oframe _change train
```

We define a global macro, `predictors`, to store the names of our predictors. We perform random forest binary classification, and we specify the `maxdepth()` and `predsampvalue()` options to tune the maximum tree depth and predictor sampling rate hyperparameters. For illustration, we use the AUCPR metric for tuning.

```
. global predictors gender g2000 g2002 p2000 p2002 p2004 treatment age

. h2oml rfbinclass voted $predictors, validframe(valid) h2orseed(19)
> ntrees(200) maxdepth(3(3)12) predsampvalue(-1, 1(2)8) tune(metric(aucpr))

Progress (%): 0 100

Random forest binary classification using H2O

Response: voted
Frame:                                     Number of observations:
  Training:   train                                   Training = 183,607
  Validation: valid                                 Validation =  45,854

Tuning information for hyperparameters

Method: Cartesian
Metric: AUCPR
```

Hyperparameters	Grid values Minimum	Maximum	Selected
Max. tree depth	3	12	6
Pred. sampling value	-1	7	7

```
Model parameters

Number of trees      = 200
              actual = 200
Tree depth:                                Pred. sampling value =      7
           Input max =   6                 Sampling rate        =   .632
                 min =   6                 No. of bins cat.     =  1,024
                 avg = 6.0                 No. of bins root     =  1,024
                 max =   6                 No. of bins cont.    =     20
Min. obs. leaf split =   1                 Min. split thresh.   = .00001

Metric summary
```

Metric	Training	Validation
Log loss	.5724664	.5705699
Mean class error	.3935492	.3943867
AUC	.6705554	.6734867
AUCPR	.4658395	.4725543
Gini coefficient	.3411109	.3469735
MSE	.1946923	.1935647
RMSE	.4412395	.4399599

Next we obtain the grid-search summary by using the `h2omlestat gridsummary` command. This command lists the configuration of the hyperparameters we are tuning ranked by AUCPR.

```
. h2omlestat gridsummary
Grid summary using H2O
```

ID	Max. tree depth	Pred. sampling value	AUCPR
1	6	7	.4725543
2	6	5	.4723736
3	6	3	.4714554
4	9	3	.4712076
5	6	-1	.4708614
6	12	-1	.4706606
7	9	-1	.4705794
8	9	5	.4689799
9	9	7	.4682457
10	9	1	.4674565

To compare the first two models based on other metrics, we use the `h2omlexplore` command.

```
. h2omlexplore id = 1 2
Performance metric summary using H2O
Training frame  : train
Validation frame: valid
```

	Model index 1	2
Training		
No. of observations	183,607	183,607
Log loss	.5724664	.57237
Mean class error	.3935492	.3979593
AUC	.6705554	.671146
AUCPR	.4658395	.4670326
Gini coefficient	.3411109	.342292
MSE	.1946923	.1946602
RMSE	.4412395	.4412031
Validation		
No. of observations	45,854	45,854
Log loss	.5705699	.5704978
Mean class error	.3943867	.3945857
AUC	.6734867	.6737527
AUCPR	.4725543	.4723736
Gini coefficient	.3469735	.3475054
MSE	.1935647	.1935627
RMSE	.4399599	.4399576

The first section of the output corresponds to the training metrics, while the second presents the validated metrics of the specified models. For each of the metrics, we see that the difference between the best and second-best models is not substantial. Therefore, the decision to switch to the less complex model may be justified.

◁

Stored results

`h2omlexplore` stores the following in `r()`:

Macros

`r(id)`	model IDs

Matrices

`r(table)`	performance metrics for selected models

Reference

Gerber, A. S., D. P. Green, and C. W. Larimer. 2008. Social pressure and voter turnout: Evidence from a large-scale field experiment. *American Political Science Review* 102: 33–48. https://doi.org/10.1017/S000305540808009X.

Also see

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

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