

Stacking with Random Grids in H2O



Mountain View, CA August 2016



A decorative graphic in the bottom left corner consists of numerous thin, yellowish-gold lines that curve and overlap, creating a sense of depth and motion.

Erin LeDell Ph.D.
Machine Learning Scientist
H2O.ai

Introduction

- Statistician & Machine Learning Scientist at H2O.ai, in Mountain View, California, USA
- Ph.D. in Biostatistics with Designated Emphasis in Computational Science and Engineering from UC Berkeley (focus on Machine Learning)
- Worked as a data scientist at several startups
- Founder of Bay Area WiMLDS meetup (2013)

Agenda



- Ensemble Learning Overview
- Stacking / Super Learner
- Ensembles vs Grid Search
- Random Grid Search
- H2O Stacking Code Demo

Ensemble Learning



In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained by any of the constituent algorithms.

– Wikipedia (2016)

Common Types of Ensemble Methods

Bagging

- Reduces variance and increases accuracy
 - Robust against outliers or noisy data
 - Often used with Decision Trees (i.e. Random Forest)
-

Boosting

- Also reduces variance and increases accuracy
 - Not robust against outliers or noisy data
 - Flexible – can be used with any loss function
-

Stacking / Super Learning

- Used to ensemble a diverse group of strong learners
- Involves training a second-level machine learning algorithm called a “metalearner” to learn the optimal combination of the base learners

The Super Learner Algorithm

$$n \left\{ \begin{bmatrix} & \\ & m \end{bmatrix} \begin{bmatrix} x \\ \end{bmatrix} \right] \begin{bmatrix} & \\ & y \end{bmatrix}$$

“Level-zero”
data

- Start with design matrix, X , and response, y
- Specify L base learners (with model params)
- Specify a metalearner (just another algorithm)
- Perform k -fold CV on each of the L learners

The Super Learner Algorithm

$$n \left\{ \begin{bmatrix} p_1 \\ \vdots \\ p_L \end{bmatrix} \cdots \begin{bmatrix} p_1 \\ \vdots \\ p_L \end{bmatrix} \begin{bmatrix} y \end{bmatrix} \right\} \rightarrow n \left\{ \underbrace{\begin{bmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \\ z & & \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{bmatrix}}_L \begin{bmatrix} y \end{bmatrix} \right\}$$

"Level-one"
data

- Collect the predicted values from k-fold CV that was performed on each of the L base learners
- Column-bind these prediction vectors together to form a new design matrix, Z
- Train the metalearner using Z, y

Super Learning vs. Parameter Tuning/Search

- A common task in machine learning is to perform model selection by specifying a number of models with different parameters.
- An example of this is Grid Search or Random Search.
- The first phase of the Super Learner algorithm is computationally equivalent to performing model selection via cross-validation.
- The latter phase of the Super Learner algorithm (the metalearning step) is just training another single model (no CV).
- With Super Learner, your computation does not go to waste!

H2O Ensemble R Package

Branch: master ▾

[h2o-3](#) / [h2o-r](#) / [ensemble](#) / +



ledell Update h2oEnsemble README

Latest commit 4824ede a minute ago

..

	demos	Added save/load functions to h2oEnsemble	8 days ago
	h2oEnsemble-package	Optimized predict.h2o.ensemble function	2 days ago
	README.md	Update h2oEnsemble README	a minute ago
	SuperLearner_wrappers.R	Added h2o-3 version of h2oEnsemble package	4 months ago
	create_h2o_wrappers.R	Added example to h2o-r/ensemble/create_h2o_wrappers.R	4 months ago
	example_twoClass_higgs.R	Updated higgs example in h2oEnsemble	5 days ago

README.md

H2O Ensemble

The `h2oEnsemble` R package provides functionality to create ensembles from the base learning algorithms that are accessible via the `h2o` R package (H2O version 3.0 and above). This type of ensemble learning is called "super learning", "stacked regression" or "stacking." The Super Learner algorithm learns the optimal combination of the base learner fits. In a 2007 article titled, "[Super Learner](#)," it was shown that the super learner ensemble represents an asymptotically optimal system for learning.

H2O Cartesian Grid Search

Example

```
hidden_opt <- list(c(200,200), c(100,300,100), c(500,500))
l1_opt <- c(1e-5,1e-7)
hyper_params <- list(hidden = hidden_opt, l1 = l1_opt)

grid <- h2o.grid(algorithm = "deeplearning",
                  hyper_params = hyper_params,
                  x = x, y = y,
                  training_frame = train,
                  validation_frame = valid)
```

H2O Random Grid Search

Example

```
search_criteria <- list(strategy = "RandomDiscrete",
                         max_runtime_secs = 600)

grid <- h2o.grid(algorithm = "deeplearning",
                  hyper_params = hyper_params,
                  search_criteria = search_criteria,
                  x = x, y = y,
                  training_frame = train,
                  validation_frame = valid)
```

Stacking with Random Grids

Use `h2o.stack()` to stack lists of models, for example, random grids.

<http://tinyurl.com/h2o-randomgrid-stack-demo>

Stacking in H2O

Example

```
# Create a list of all the base models
models <- c(gbm_models, rf_models, dl_models, glm_models)

# Let's stack!
stack <- h2o.stack(models = models,
                    response_frame = train[,y],
                    metalearner = metalearner)
```

H2O Ensemble Resources

H2O Ensemble training guide:

<http://tinyurl.com/learn-h2o-ensemble>

H2O Ensemble homepage on Github:

<http://tinyurl.com/github-h2o-ensemble>

H2O Ensemble R Demos:

<http://tinyurl.com/h2o-ensemble-demos>

H2O R & Python Tutorial

<http://tinyurl.com/h2o-tutorial-rpy>



Tutorial: Intro to H2O Algorithms

The “Intro to H2O” tutorial introduces five popular supervised machine learning algorithms in the context of a binary classification problem.

The training module demonstrates how to train models and evaluating model performance on a test set.

- Generalized Linear Model (GLM)
- Random Forest (RF)
- Gradient Boosting Machine (GBM)
- Deep Learning (DL)
- Naive Bayes (NB)

Tutorial: Grid Search for Model Selection

```
> print(gbm_gridperf)
H2O Grid Details
=====
Grid ID: gbm_grid2
Used hyper parameters:
- sample_rate
- max_depth
- learn_rate
- col_sample_rate
Number of models: 72
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by decreasing auc
  sample_rate max_depth learn_rate col_sample_rate      model_ids          auc
1           1         3       0.19  1 gbm_grid2_model_38 0.685166598389755
2           0.9       3       0.15  1 gbm_grid2_model_53 0.684956999713052
3           0.8       5       0.06  1 gbm_grid2_model_22 0.684843506375254
4           0.6       4       0.07  1 gbm_grid2_model_4   0.684327718715252
5           0.95      4       0.13  1 gbm_grid2_model_48 0.684042497773235
```

The second training module demonstrates how to find the best set of model parameters for each model using Grid Search.

H2O Resources

- H2O Online Training: <http://learn.h2o.ai>
- H2O Tutorials: <https://github.com/h2oai/h2o-tutorials>
- H2O Slidedecks: <http://www.slideshare.net/0xdata>
- H2O Video Presentations: <https://www.youtube.com/user/0xdata>
- H2O Community Events & Meetups: <http://h2o.ai/events>



Thank you!

@ledell on Github, Twitter
erin@h2o.ai

<http://www.stat.berkeley.edu/~ledell>