h2oEnsemble: Scalable Ensemble Learning in R

Erin LeDell

H2O.ai
Mountain View, California
USA

July 2, 2015
Overview

- Ensemble Learning
- Model Stacking (aka. Super Learning)
- H2O Machine Learning via h2o R package
- h2oEnsemble R package
Ensemble Learning

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


- Ensemble of weak learners (e.g. Random Forest)
- Generalized Model Stacking (combine the predictions from multiple models)
The **Super Learner algorithm** is a loss-based supervised learning method that finds the optimal combination of a collection of prediction algorithms.

Super Learner performs asymptotically as well as best possible weighted combination of the base learners.
K-fold Cross-validation

Example: 5-fold cross validation
Super Learner: The setup

1. Define a base learner library of $L$ learners, $\Psi^1, \ldots, \Psi^L$.
2. Specify a metalearning method, $\Phi$.
3. Partition the training observations into $V$ folds.
Super Learner: The algorithm

1. Generate a matrix $Z$, of dimension $n \times L$, of cross-validated predictions as follows: During cross-validation, we obtain fits, $\hat{\Psi}_v^l$, defined as fitting $\Psi^l$ on the observations that are not in fold $v$. Predictions are then generated for the observations in the $v^{th}$ fold.

2. Find the optimal combination of subset-specific fits according to a user-specified metalearner algorithm, $\hat{\Phi}$, with a new design matrix, $Z$.

3. Fit $L$ models (one for each base learner) on the original training set, $X$, and save the $L$ individual model fit objects along with $\hat{\Phi}$. This ensemble model can be used to generate predictions on new data.
Practical solutions to this problem:

1. Develop alternative formulations of Super Learner that learn on subsets of data to overcome memory limitations.
2. Use candidate learners that can learn iteratively and thus do not require loading the entire training set into memory at once. (i.e., online learning)
3. Make use of distributed algorithms.
4. Rather than native R or Python, use a more “scalable” language (C++, Java, Scala, Fortran, Julia).
H2O Machine Learning platform

H2O is an open source, distributed, Java machine learning library.

APIs available in:
R, Python, Java, Scala and REST/JSON
H2O Machine Learning platform

Distributed Supervised ML Algorithms available in H2O

- Generalized Linear Model with Elastic Net regularization
- Gradient Boosting Machines (w/ trees)
- Random Forest
- Deep Learning: Multi-Layer Feed-Forward Neural Networks
library(h2o)  # First install from CRAN
localH2O <- h2o.init()  # Initialize the H2O cluster

# Data directly into H2O cluster (avoids R)
train <- h2o.importFile(path = "train.csv")

# Data into H2O from R data.frame
train <- as.h2o(my_df)
h2o: How to train & test

Example

```r
y <- "Class"
x <- setdiff(names(train), y)

fit <- h2o.gbm(x = x, y = y, training_frame = train)
pred <- h2o.predict(fit = fit, validation_frame = test)
```
h2oEnsemble: Set up the ensemble

Example

learner <- c("h2o.randomForest.1",
              "h2o.deeplearning.1",
              "h2o.deeplearning.2")

metalearner <- "h2o.glm.wrapper"

family <- "binomial"
**h2oEnsemble** R package

**Example**

```r
fit <- h2o.ensemble(x = x, y = y, training_frame = train,
                      family = family,
                      learner = learner,
                      metalearner = metalearner)

pred <- h2o.predict(fit = fit, validation_frame = test)
```
H2O Ensemble: Performance

Runtime Performance of H2O Ensemble

- Workstation (32 vCPU)
- Cluster (96 vCPU)
- Cluster (320 vCPU)

R color palette: https://github.com/karthik/wesanderson
Thank you!

@ledell on Twitter, GitHub
http://www.stat.berkeley.edu/~ledell

More info at http://h2o.ai
Email: erin@h2o.ai