Package 'h2oEnsemble'

November 6, 2015

Type Package

Title H2O Ensemble Learning

Version 0.1.3

Date 2015-11-06

Author Erin LeDell

Maintainer Erin LeDell <erin@h2o.ai>

Description H2O Ensemble implements the Super Learner ensemble (stacking) algorithm using the H2O R interface to provide base learning algorithms.

License Apache License (== 2.0)

Depends R (>= 2.13.0), h2o (>= 3.2.0.9)

NeedsCompilation no

SystemRequirements Java (>= 1.6)

Suggests SuperLearner, cvAUC, testthat

URL https://github.com/h2oai/h2o-3/tree/master/h2o-r/ensemble/h2oEnsemble-package

R topics documented:

h2oEnsemble-package *H2O Ensemble Package*

Description

H2O Ensemble implements the Super Learner ensemble (stacking) algorithm using the H2O R interface to provide base learning algorithms.

Details

Package:	h2oEnsemble
Type:	Package
Version:	0.1.3
Date:	2015-11-06
License:	Apache License (== 2.0)

Author(s)

Erin LeDell

Maintainer: Erin LeDell <erin@h2o.ai>

References

LeDell, E. (2015) Scalable Ensemble Learning and Computationally Efficient Variance Estimation (Doctoral Dissertation). University of California, Berkeley, USA.
http://www.stat.berkeley.edu/~ledell/papers/ledell-phd-thesis.pdf

van der Laan, M. J., Polley, E. C. and Hubbard, A. E. (2007) Super Learner, Statistical Applications of Genetics and Molecular Biology, 6, article 25.
http://dx.doi.org/10.2202/1544-6115.1309
http://biostats.bepress.com/ucbbiostat/paper222

Breiman, L. (1996) Stacked Regressions, Machine Learning, 24:49-64.
http://dx.doi.org/10.1007/BF00117832
http://statistics.berkeley.edu/sites/default/files/tech-reports/367.pdf

See Also

SuperLearner, subsemble

h2o.ensemble *H2O Ensemble*

Description

This function creates a "Super Learner" (stacking) ensemble using the H2O base learning algorithms specified by the user.

Usage

```
h2o.ensemble(x, y, training_frame,
  model_id = "", validation_frame = NULL,
  family = c("AUTO", "binomial", "gaussian"),
  learner = c("h2o.glm.wrapper", "h2o.randomForest.wrapper", "h2o.gbm.wrapper", "h2o.deeplearning
  metalearner = "h2o.glm.wrapper",
  cvControl = list(V = 5, shuffle = TRUE),
  seed = 1, parallel = "seq",
  keep_levelone_data = TRUE)
```

Arguments

x	A vector containing the names of the predictors in the model.
y	The name of the response variable in the model.
training_frame	An H2O Frame object containing the variables in the model.
family	A description of the error distribution and link function to be used in the model. This must be a character string. Currently supports "binomial" and "gaussian".
model_id	(Optional) The unique id assigned to the resulting model. If none is given, an id will automatically be generated.
validation_frame	(Optional) An H2O Frame object indicating the validation dataset used to construct the confusion matrix. If left blank, this defaults to the training data when nfolds = 0. Currently not used.
learner	A string or character vector naming the prediction algorithm(s) used to train the base models for the ensemble. The functions must have the same format as the h2o wrapper functions.
metalearner	A string specifying the prediction algorithm used to learn the optimal combination of the base learners. Supports both h2o and SuperLearner wrapper functions.
cvControl	A list of parameters to control the cross-validation process. The V parameter is an integer representing the number of cross-validation folds and defaults to 10. Other parmeters are stratifyCV and shuffle, which are not yet enabled.
seed	A random seed to be set (integer); defaults to 1. If NULL, then a random seed will not be set. The seed is set prior to creating the CV folds and prior to model training for base learning and metalearning.
parallel	A character string specifying optional parallelization. Use "seq" for sequential computation (the default) of the cross-validation and base learning steps. Use "multicore" to perform the V-fold (internal) cross-validation step as well as the final base learning step in parallel over all available cores. Or parallel can

be a snow cluster object. Both parallel options use the built-in functionality of the R core "parallel" package. Currently, only "seq" is compatible with the parallelized H2O algorithms, so this argument may be removed or modified in the future.

keep_levelone_data
: Logical, defaults to TRUE. Keep the levelone H2O Frame of cross-validated predicted values (Z matrix) and original response vector, y (cbind to Z).

Value

x
: A vector containing the names of the predictors in the model.

y
: The name of the response variable in the model.

family
: Returns the family argument from above.

cvControl
: Returns the cvControl argument from above.

folds
: A vector of fold ids for each observation, ordered by row index. The number of unique fold ids is specified in cvControl$V.

ylim
: Returns range of y.

seed
: An integer. Returns seed argument from above.

parallel
: An character vector. Returns character argument from above.

basefits
: A list of H2O models, each of which are trained using the data object. The length of this list is equal to the number of base learners in the learner argument.

metafit
: The predictive model which is learned by regressing y on Z (see description of Z below). The type of model is specified using the metalearner argument.

levelone
: An H2O Frame object. The levelone H2O Frame includes the "Z matrix" (the cross-validated predicted values for each base learner) and original response vector, y. In the stacking ensemble literature, the Z matrix is the design matrix used to train the metalearner.

runtime
: A list of runtimes for various steps of the algorithm. The list contains cv, metalearning, baselearning and total elements. The cv element is the time it takes to create the Z matrix (see above). The metalearning element is the training time for the metalearning step. The baselearning element is a list of training times for each of the models in the ensemble. The time to run the entire h2o.ensemble function is given in total.

h2o_version
: The version of the h2o R package.

h2oEnsemble_version
: The version of the h2oEnsemble R package.

Author(s)

Erin LeDell <erin@h2o.ai>

References

LeDell, E. (2015) Scalable Ensemble Learning and Computationally Efficient Variance Estimation (Doctoral Dissertation). University of California, Berkeley, USA.
http://www.stat.berkeley.edu/~ledell/papers/ledell-phd-thesis.pdf

van der Laan, M. J., Polley, E. C. and Hubbard, A. E. (2007) Super Learner, Statistical Applications of Genetics and Molecular Biology, 6, article 25.
http://dx.doi.org/10.2202/1544-6115.1309
http://biostats.bepress.com/ucbbiostat/paper222

Breiman, L. (1996) Stacked Regressions, Machine Learning, 24:49-64.
http://dx.doi.org/10.1007/BF00117832
http://statistics.berkeley.edu/sites/default/files/tech-reports/367.pdf

See Also

SuperLearner, subsemble

Examples

```
## Not run:

# An example of binary classification on a local machine using h2o.ensemble

library(h2oEnsemble)  # Requires version >=0.0.4 of h2oEnsemble
library(cvAUC)  # Used to calculate test set AUC (requires version >=1.0.1 of cvAUC)
localH2O <-  h2o.init(nthreads = -1)  # Start an H2O cluster with nthreads = num cores on your machine


# Import a sample binary outcome train/test set into R
train <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_10k.csv")
test <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_test_5k.csv")
y <- "C1"
x <- setdiff(names(train), y)
family <- "binomial"

#For binary classification, response should be a factor
train[,y] <- as.factor(train[,y])
test[,y] <- as.factor(test[,y])


# Specify the base learner library & the metalearner
learner <- c("h2o.glm.wrapper", "h2o.randomForest.wrapper",
               "h2o.gbm.wrapper", "h2o.deeplearning.wrapper")
metalearner <- "h2o.deeplearning.wrapper"


# Train the ensemble using 5-fold CV to generate level-one data
# More CV folds will take longer to train, but should increase performance
fit <- h2o.ensemble(x = x, y = y,
                    training_frame = train,
                    family = family,
                    learner = learner,
                    metalearner = metalearner,
                    cvControl = list(V = 5, shuffle = TRUE))


# Generate predictions on the test set
pp <- predict(fit, test)
predictions <- as.data.frame(pp$pred)[,3]  #third column, p1 is P(Y==1)
labels <- as.data.frame(test[,y])[,1]
```

```
# Ensemble test AUC
cvAUC::AUC(predictions = predictions, labels = labels)
# 0.7888723


# Base learner test AUC (for comparison)
L <- length(learner)
auc <- sapply(seq(L), function(l) cvAUC::AUC(predictions = as.data.frame(pp$basepred)[,l], labels = labels))
data.frame(learner, auc)
#                   learner       auc
#1          h2o.glm.wrapper 0.6871288
#2 h2o.randomForest.wrapper 0.7711654
#3          h2o.gbm.wrapper 0.7817075
#4 h2o.deeplearning.wrapper 0.7425813

# Note that the ensemble results above are not reproducible since
# h2o.deeplearning is not reproducible when using multiple cores,
# and we did not set a seed for h2o.randomForest.wrapper or h2o.gbm.wrapper.

# Additional note: In a future version, performance metrics such as AUC
# will be computed automatically, as in the other H2O algos.

# Here is an example of how to generate a base learner library using custom base learners:
h2o.randomForest.1 <- function(..., ntrees = 1000, nbins = 100, seed = 1) {
h2o.randomForest.wrapper(..., ntrees = ntrees, nbins = nbins, seed = seed)
}
h2o.deeplearning.1 <- function(..., hidden = c(500,500), activation = "Rectifier", seed = 1) {
  h2o.deeplearning.wrapper(..., hidden = hidden, activation = activation, seed = seed)
}
h2o.deeplearning.2 <- function(..., hidden = c(200,200,200), activation = "Tanh", seed = 1) {
h2o.deeplearning.wrapper(..., hidden = hidden, activation = activation, seed = seed)
}
learner <- c("h2o.randomForest.1", "h2o.deeplearning.1", "h2o.deeplearning.2")


## End(Not run)
```

`h2o.example.wrapper` *Wrapper functions for h2o algorithms*

Description

This is an example h2o algorithm wrapper function.

Usage

```
h2o.example.wrapper(x, y, training_frame, model_id = "",
  family = c("gaussian", "binomial"), ...)
```

Arguments

x	A vector containing the names of the predictors in the model.

y	The name of the response variable in the model.
training_frame	An H2O Frame object containing the variables in the model.
model_id	(Optional) The unique id assigned to the resulting model. If none is given, an id will automatically be generated.
family	A description of the error distribution and link function to be used in the model. This must be a character string. Currently supports "binomial" and "gaussian".
...	Additional arguments to be passed to or from methods.

Value

An H2O model.

Author(s)

Erin LeDell <erin@h2o.ai>

See Also

h2o.ensemble

---

h2o.load_ensemble *Load an H2O Ensemble model*

Description

Load an H2O Ensemble model from disk. This includes a set of cross-validated H2OModels serialized using the h2o.saveModel, a serialized RData object (the ensemble object) and optionally, the level-one matrix of cross-validated predicted values as a CSV file.

Usage

```
h2o.load_ensemble(path = "", import_levelone = FALSE)
```

Arguments

path	String indicating the directory the model will be loaded from.
import_levelone	Logical, defaults to FALSE. Will load the level-one matrix of cross-validated predicted values from a CSV file if the model was saved with h2o.save_ensemble using the export_levelone = TRUE. The level-one matrix is required for re-training the metalearner using the h2o.metalearn function.

Author(s)

Erin LeDell <erin@h2o.ai>

See Also

h2o.save_ensemble for saving an H2O Ensemble to disk.

Examples

```
## Not run:
# An example of binary classification on a local machine using h2o.ensemble

library(h2oEnsemble)  # Requires version >=0.0.4 of h2oEnsemble
library(cvAUC)  # Used to calculate test set AUC (requires version >=1.0.1 of cvAUC)
localH2O <-  h2o.init(nthreads = -1)  # Start an H2O cluster with nthreads = num cores on your machine


# Import a sample binary outcome train/test set into R
train <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_10k.csv")
test <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_test_5k.csv")
y <- "C1"
x <- setdiff(names(train), y)
family <- "binomial"

#For binary classification, response should be a factor
train[,y] <- as.factor(train[,y])
test[,y] <- as.factor(test[,y])


# Specify the base learner library & the metalearner
learner <- c("h2o.glm.wrapper", "h2o.randomForest.wrapper",
               "h2o.gbm.wrapper", "h2o.deeplearning.wrapper")
metalearner <- "h2o.deeplearning.wrapper"


# Train the ensemble using 5-fold CV to generate level-one data
# More CV folds will take longer to train, but should increase performance
fit <- h2o.ensemble(x = x, y = y,
                    training_frame = train,
                    family = family,
                    learner = learner,
                    metalearner = metalearner,
                    cvControl = list(V = 5, shuffle = TRUE))

# Save ensemble model (a collection of H2OModels and an RData object) to disk
h2o.save_ensemble(fit, path = "./h2o-ensemble-model-loadtest", force = FALSE, export_levelone = FALSE)
rm(fit)

# Load model from disk
fit <- h2o.load_ensemble(path = "./h2o-ensemble-model-loadtest")

# Generate predictions from ensemble model on a test set
pp <- predict(fit, test)
predictions <- as.data.frame(pp$pred)[,3]  #third column, p1 is P(Y==1)
labels <- as.data.frame(test[,y])[,1]

# Ensemble test AUC
cvAUC::AUC(predictions = predictions, labels = labels)
# 0.7888723

## End(Not run)
```

h2o.metalearn *H2O Metalearn*

Description

Re-trains an existing H2O Ensemble fit using a new metalearning function.

Usage

```
h2o.metalearn(object,
  metalearner = "h2o.glm.wrapper",
  seed = 1,
  keep_levelone_data = TRUE)
```

Arguments

object	An object of class, "h2o.ensemble".
metalearner	A string specifying the prediction algorithm used to learn the optimal combination of the base learners. Supports both h2o and SuperLearner wrapper functions.
seed	A random seed to be set (integer); defaults to 1. If NULL, then a random seed will not be set. The seed is set prior to creating the CV folds and prior to model training for base learning and metalearning.
keep_levelone_data	Logical, defaults to TRUE. Keep the levelone H2O Frame of cross-validated predicted values (Z matrix) and original response vector, y (cbind to Z).

Value

x	A vector containing the names of the predictors in the model.
y	The name of the response variable in the model.
family	Returns the family argument from above.
cvControl	Returns the cvControl argument from above.
folds	A vector of fold ids for each observation, ordered by row index. The number of unique fold ids is specified in cvControl$V.
ylim	Returns range of y.
seed	An integer. Returns seed argument from above.
parallel	An character vector. Returns character argument from above.
basefits	A list of H2O models, each of which are trained using the data object. The length of this list is equal to the number of base learners in the learner argument.
metafit	The predictive model which is learned by regressing y on Z (see description of Z below). The type of model is specified using the metalearner argument.
levelone	An H2O Frame object. The levelone Frame includes the Z matrix (the cross-validated predicted values for each base learner), fold id column and original response vector, y. In the stacking ensemble literature, the Z matrix is the design matrix used to train the metalearner.

`runtime` A list of runtimes for various steps of the algorithm. The list contains `cv`, `metalearning`, `baselearning` and `total` elements. The `cv` element is the time it takes to create the `Z` matrix (see above). The `metalearning` element is the training time for the metalearning step. The `baselearning` element is a list of training times for each of the models in the ensemble. The time to run the entire `h2o.ensemble` function is given in `total`.

`h2o_version` The version of the h2o R package.

`h2oEnsemble_version` The version of the h2oEnsemble R package.

Author(s)

Erin LeDell <erin@h2o.ai>

References

LeDell, E. (2015) Scalable Ensemble Learning and Computationally Efficient Variance Estimation (Doctoral Dissertation). University of California, Berkeley, USA.
http://www.stat.berkeley.edu/~ledell/papers/ledell-phd-thesis.pdf

van der Laan, M. J., Polley, E. C. and Hubbard, A. E. (2007) Super Learner, Statistical Applications of Genetics and Molecular Biology, 6, article 25.
http://dx.doi.org/10.2202/1544-6115.1309
http://biostats.bepress.com/ucbbiostat/paper222

Breiman, L. (1996) Stacked Regressions, Machine Learning, 24:49-64.
http://dx.doi.org/10.1007/BF00117832
http://statistics.berkeley.edu/sites/default/files/tech-reports/367.pdf

See Also

h2o.ensemble

Examples

```
## Not run:

# An example of binary classification on a local machine using h2o.ensemble

library(h2oEnsemble)  # Requires version >=0.0.4 of h2oEnsemble
library(cvAUC)  # Used to calculate test set AUC (requires version >=1.0.1 of cvAUC)
localH2O <-  h2o.init(nthreads = -1)  # Start an H2O cluster with nthreads = num cores on your machine


# Import a sample binary outcome train/test set into R
train <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_10k.csv")
test <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_test_5k.csv")
y <- "C1"
x <- setdiff(names(train), y)
family <- "binomial"

#For binary classification, response should be a factor
train[,y] <- as.factor(train[,y])
test[,y] <- as.factor(test[,y])
```

```
# Specify the base learner library & the metalearner
# Let's use a reproducible library (set seed on RF and GBM):
h2o.randomForest.1 <- function(..., ntrees = 100, seed = 1) h2o.randomForest.wrapper(..., ntrees = ntrees, se
h2o.gbm.1 <- function(..., ntrees = 100, seed = 1) h2o.gbm.wrapper(..., ntrees = ntrees, seed = seed)
learner <- c("h2o.glm.wrapper", "h2o.randomForest.1", "h2o.gbm.1")
metalearner <- "h2o.glm.wrapper"


# Train the ensemble using 10-fold CV to generate level-one data
# More CV folds will take longer to train, but should increase performance
fit <- h2o.ensemble(x = x, y = y,
                    training_frame = train,
                    family = family,
                    learner = learner,
                    metalearner = metalearner,
                    cvControl = list(V = 10, shuffle = TRUE))


# Generate predictions on the test set
pp <- predict(fit, test)
predictions <- as.data.frame(pp$pred)[,3]  #third column, p1 is P(Y==1)
labels <- as.data.frame(test[,y])[,1]

# Ensemble test AUC
cvAUC::AUC(predictions = predictions, labels = labels)
# 0.787226


# Now let's re-train the metalearner fit to see if we get better performance.
# Previously, we used a GLM metalearner, and now we will try a GBM.
newfit <- h2o.metalearn(fit, metalearner = "h2o.gbm.1")

# Generate predictions on the test set
pp <- predict(newfit, test)
predictions <- as.data.frame(pp$pred)[,3]  #third column, p1 is P(Y==1)
labels <- as.data.frame(test[,y])[,1]

# Ensemble test AUC
cvAUC::AUC(predictions = predictions, labels = labels)
# 0.7777479

# We see that on this dataset & base learner combination,
# that an ensemble with a GLM metalearner performs better,
# in terms of test set AUC, than an emseble wtih a GBM metalearner.


## End(Not run)
```

`h2o.save_ensemble` *Save an H2O Ensemble model*

Description

Save an H2O Ensemble model to disk. This includes a set of cross-validated H2OModels serialized using the `h2o.saveModel`, a serialized RData object (the ensemble object) and optionally, the level-

one matrix of cross-validated predicted values as a CSV file.

Usage

```
h2o.save_ensemble(object, path = "", force = FALSE, export_levelone = FALSE)
```

Arguments

object	An object of class, "h2o.ensemble".
path	String indicating the directory the model will be written to.
force	Logical, defaults to FALSE. Indicates how to deal with files that already exist.
export_levelone	Logical, defaults to FALSE. Will write the level-one matrix of cross-validated predicted values to a CSV file.

Details

In the case of existing files `force = TRUE` will overwrite the file. Otherwise, the operation will fail.

Author(s)

Erin LeDell <erin@h2o.ai>

See Also

h2o.load_ensemble for loading an H2O Ensemble from disk.

Examples

```
## Not run:
# An example of binary classification on a local machine using h2o.ensemble

library(h2oEnsemble)  # Requires version >=0.0.4 of h2oEnsemble
library(cvAUC)  # Used to calculate test set AUC (requires version >=1.0.1 of cvAUC)
localH2O <-  h2o.init(nthreads = -1)  # Start an H2O cluster with nthreads = num cores on your machine


# Import a sample binary outcome train/test set into R
train <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_10k.csv")
test <- h2o.importFile("http://www.stat.berkeley.edu/~ledell/data/higgs_test_5k.csv")
y <- "C1"
x <- setdiff(names(train), y)
family <- "binomial"

#For binary classification, response should be a factor
train[,y] <- as.factor(train[,y])
test[,y] <- as.factor(test[,y])


# Specify the base learner library & the metalearner
learner <- c("h2o.glm.wrapper", "h2o.randomForest.wrapper",
               "h2o.gbm.wrapper", "h2o.deeplearning.wrapper")
metalearner <- "h2o.deeplearning.wrapper"
```

```
# Train the ensemble using 5-fold CV to generate level-one data
# More CV folds will take longer to train, but should increase performance
fit <- h2o.ensemble(x = x, y = y,
                    training_frame = train,
                    family = family,
                    learner = learner,
                    metalearner = metalearner,
                    cvControl = list(V = 5, shuffle = TRUE))

# Save ensemble model (a collection of H2OModels and an RData object) to disk
h2o.save_ensemble(fit, path = "./h2o-ensemble-model-savetest", force = FALSE, export_levelone = FALSE)
rm(fit)

# Load model from disk
fit <- h2o.load_ensemble(path = "./h2o-ensemble-model-savetest")

## End(Not run)
```

`predict.h2o.ensemble` *Predict method for an 'h2o.ensemble' object*

Description

Obtains predictions on a new data set from a `h2o.ensemble` fit.

Usage

```
## S3 method for class 'h2o.ensemble'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class 'h2o.ensemble', which is returned from the `h2o.ensemble` function.
`newdata`	An H2O Frame object in which to look for variables with which to predict.
`...`	Additional arguments passed on to the function.

Value

`pred`	A vector of predicted values from ensemble fit.
`basepred`	An H2O Frame object with the predicted values from each base learner algorithm for the rows in `newdata`.

Author(s)

Erin LeDell `<erin@h2o.ai>`

See Also

`h2o.ensemble`

Examples

```
# See h2o.ensemble documentation for an example.
```

`print.h2o.ensemble` *Print method for an 'h2o.ensemble' object*

Description

Print metadata for an `h2o.ensemble` fit.

Usage

```
  ## S3 method for class 'h2o.ensemble'
print(x, ...)
```

Arguments

`x`	An object of class 'h2o.ensemble', which is returned from the `h2o.ensemble` function.
`...`	Additional arguments passed on to the function.

Author(s)

Erin LeDell <erin@h2o.ai>

See Also

`h2o.ensemble`

Index