Package 'wsrf'

July 2, 2014

Type Package

Title Weighted Subspace Random Forest

Version 1.4.0

Date 2014-05-30

Maintainer He Zhao <Simon.Yansen.Zhao@gmail.com>

Description The wsrf package is a parallel implementation of the
Weighted Subspace Random Forest algorithm proposed (wsrf). A
novel variable weighting method is used for variable subspace
selection in place of the traditional approach of random
variable sampling. This new approach is particularly useful
in building models for high dimensional data---often
consisting of thousands of variables. Parallel computation is
used to take advantage of multi-core machines and clusters of
machines to build random forest models from high dimensional data with reduced elapsed times.

License GPL (>= 2)

Depends R (>= 3.0.0), Rcpp (>= 0.10.2), parallel

LinkingTo Rcpp

Suggests rattle (>= 2.6.26), randomForest (>= 4.6.7), party (>= 1.0.7), stringr (>= 0.6.2), knitr (>= 1.5)

VignetteBuilder knitr

Author Qinghan Meng [aut],He Zhao [aut, cre],Graham Williams [aut],Junchao Lv [ctb]

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-05-30 09:07:45

R topics documented:

`correlation.wsrf` *Correlation*

Description

Give the measure for the diversity of the trees in the forest model built from `wsrf`.

Usage

```
## S3 method for class 'wsrf'
correlation(object, ...)
```

Arguments

`object`	object of class `wsrf`.
`...`	Optional additional arguments. At present no additional arguments are used.

Details

The measure was introduced in Breiman (2001).

Value

A numeric value.

Author(s)

He Zhao and Graham Williams (SIAT)

References

Breiman L (2001). "Random forests." Machine learning, 45(1), 5-32.

See Also

`wsrf`

oobErrorRate.wsrf *Out-of-Bag Error Rate*

Description

Return out-of-bag error rate for "wsrf" model.

Usage

```
## S3 method for class 'wsrf'
oobErrorRate(object, tree)
```

Arguments

object	object of class wsrf.
tree	logical or an integer vector for the index of a specific tree in the forest model. If provided as an integer vector, oobErrorRate.wsrf will give the corresponding out-of-bag error rates of the exact trees specified by tree. If TRUE, all error rates will be presented. If FALSE or missing, the gross error rate for the forest will be given.

Value

return a vector of error rates.

Author(s)

He Zhao and Graham Williams (SIAT)

See Also

wsrf

predict.wsrf *Predict Method for* wsrf *Model*

Description

Give the predictions for the new data by the forest model built from wsrf.

Usage

```
 ## S3 method for class 'wsrf'
predict(object, newdata, type=c("response",
  "class", "prob", "vote", "aprob", "waprob"), ...)
```

Arguments

object	object of class wsrf.
newdata	the data that needs to be predicted. Its format should be the same as that for wsrf.
type	the type of prediction required, indicating the type of output, and can be one of the values below: **vote** matrix of vote counts **response** predicted values. **class** the same as response. **prob** matrix of class probabilities. The probability is the proportion of trees in the forest voting for the particular outcome (prob = votes / ntrees) **aprob** the average score from the decision trees for each class rather than the proportion of decision trees for each class (aprob = scores / ntrees) **waprob** the weighted average, weighted by the accuracy of the tree (waprob = scores * accuracy / sum(accuracy))
...	Optional additional arguments. At present no additional arguments are used.

Value

a vector of length nrow(newdata) if given type of response or class, otherwise, a matrix of nrow(newdata) * (number of class label).

Author(s)

He Zhao and Graham Williams (SIAT)

See Also

wsrf

---

print.wsrf *Print Method for* wsrf *model*

Description

Print all trees or one specific tree in the forest model built from wsrf.

Usage

```
## S3 method for class 'wsrf'
print(x, tree, ...)
```

Arguments

x	object of class wsrf.
tree	the index of a specific tree. If missing, print will print the whole forest, which will take a long time for a big forest.
...	Optional additional arguments. At present no additional arguments are used.

Note

It will take quite amount of time printing the whole forest if there are a large number of trees and a large number of nodes for each tree.

Author(s)

He Zhao and Graham Williams (SIAT)

See Also

wsrf

---

strength.wsrf *Strength*

Description

Give the measure for the collective performance of individual trees in the forest model built from wsrf.

Usage

```
## S3 method for class 'wsrf'
strength(object, ...)
```

Arguments

object	object of class wsrf.
...	Optional additional arguments. At present no additional arguments are used.

Details

The measure was introduced in Breiman (2001).

Value

A numeric value.

Author(s)

He Zhao and Graham Williams (SIAT)

References

Breiman L (2001). "Random forests." Machine learning, 45(1), 5-32.

See Also

wsrf

---

summary.wsrf *Summarize a* wsrf *Model*

Description

Summary method for class "wsrf".

Usage

```
## S3 method for class 'wsrf'
summary(object, tree, ...)
```

Arguments

object	object of class wsrf.
tree	the index of a specific tree in the forest model. If provided, summary.wsrf will give the summary of the exact tree in the forest.
...	Optional additional arguments. At present no additional arguments are used.

Author(s)

He Zhao and Graham Williams (SIAT)

See Also

wsrf

varCounts.wsrf *Number of Times of Variables Selected as Split Condition*

Description

Return the times of each variable being selected as split condition. For evaluating the bias of wsrf towards attribute types (categorical and numerical) and the number of values each attribute has.

Usage

```
## S3 method for class 'wsrf'
varCounts(object)
```

Arguments

object	object of class wsrf.

Value

A vector of integer. The length is the same as the training data for building that wsrf model.

Author(s)

He Zhao and Graham Williams (SIAT)

See Also

wsrf

wsrf *Build a Forest of Weighted Subspace Decision Trees*

Description

Build weighted subspace decision trees to construct a forest.

Usage

```
wsrf(formula, data, nvars, mtry, ntrees=500, weights=TRUE,
                parallel=TRUE, na.action=na.fail)
```

Arguments

formula	a formula, with a response but no interaction terms.
data	a data frame in which to interpret the variables named in the formula.
ntrees	number of trees to build on each server; By default, 500
nvars, mtry	number of variables to choose, with Breiman's default for random forests being the integer less than or equal to $log_2(ninputs) + 1$. For compatibility with other R packages like randomForest, both nvars and mtry are supported, however, only one of them should be specified.
weights	logical. TRUE for weighted subspace selection, which is the default; FALSE for random selection.
na.action	indicate the behaviour when encountering NA values in data.
parallel	whether to run multiple cores (TRUE), nodes, or sequentially (FALSE).

Details

See Xu, Huang, Williams, Wang, and Ye (2012) for details

Value

An object of class wsrf.

Author(s)

He Zhao and Graham Williams (SIAT)

References

Xu B, Huang JZ, Williams G, Wang Q, Ye Y (2012). "Classifying very high-dimensional data with random forests built from small subspaces." International Journal of Data Warehousing and Mining (IJDWM), 8(2), 44-63.

Examples

```
# prepare parameters
ds <- get("weather")
dim(ds)
names(ds)
target <- "RainTomorrow"
id     <- c("Date", "Location")
risk   <- "RISK_MM"
ignore <- c(id, if (exists("risk")) risk)
vars   <- setdiff(names(ds), ignore)
if (sum(is.na(ds[vars]))) ds[vars] <- na.roughfix(ds[vars])
ds[target] <- as.factor(ds[[target]])
(tt <- table(ds[target]))
form <- as.formula(paste(target, "~ ."))
```

```
set.seed(42)
train <- sample(nrow(ds), 0.7*nrow(ds))
test <- setdiff(seq_len(nrow(ds)), train)

# build model
model.wsrf.1 <- wsrf(form, data=ds[train, vars])

# view model
print(model.wsrf.1, tree=1)
summary(model.wsrf.1)
summary(model.wsrf.1, tree=c(1,500))

# evaluate
strength(model.wsrf.1)
correlation(model.wsrf.1)
cl <- predict(model.wsrf.1, newdata=ds[test, vars], type="response")
actual <- ds[test, target]
(accuracy.wsrf <- sum(cl==actual)/length(actual))
```

`wsrfParallelInfo` *Query about underlying parallel implementation information*

Description

Give the information about underlying parallel implementation.

Usage

```
wsrfParallelInfo(...)
```

Arguments

...	Optional additional arguments. At present no additional arguments are used.

Details

The parallel implementation cannot be changed after installation of the package. So this function is used to query which techniques are actually adopted as the underlying parallel implementation, among which are C++11, Boost, or no parallelism at all.

Value

A diagnostic message about the parallel implementation depends on the actual situation. Currently possible messages are:

1. With parallel computing disabled
2. Use C++ standard thread library for parallel computing
3. Use C++ Boost thread library for parallel computing

Author(s)

He Zhao (SIAT)

See Also

wsrf

Index