Package ‘rknn’

June 9, 2015

Type Package

Title Random KNN Classification and Regression

Version 1.2-1

Date 2015-06-07

Author Shengqiao Li <lishengqiao@yahoo.com>

Maintainer Shengqiao Li <lishengqiao@yahoo.com>

Depends R (>= 2.14), gmp (>= 0.5-5)

Suggests Hmisc, Biobase, genefilter, golubEsets, chemometrics

Description Random knn classification and regression are implemented. Random knn based feature selection methods are also included. The approaches are mainly developed for high-dimensional data with small sample size.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-06-09 00:14:51

R topics documented:

rknn-package *Random KNN Classification and Regression*

Description

Random KNN Classification and Regression

Details

Package:	rknn
Type:	Package
Version:	1.1
Date:	2013-08-05
Depends:	R (>= 2.15.0), gmp
Suggests:	Hmisc, Biobase, genefilter, golubEsets, chemometrics
Imports:	class, FNN
License:	GPL (>=2)
LazyLoad:	yes
Packaged:	2013-08-5

Index:

```
PRESS                   Predicted Residual Sum of Squares
begKNN                  Backward Elimination Feature Selection with
                        Random KNN
bestset                 Extract the best subset of feature from
                        selection process
confusion               Classification Confusion Matrix and Accuracy
cv.coef                 Coefficient of Variation
eta                     Coverage Probability
fitted.randomKNN        Extract Model Fitted Values
knn.reg                 KNN Regression
knn.reg.cv              KNN Regression Cross-Validation
lambda                  Compute Number of Silent Features
```

normalize.decscale	Data Normalization
plot.begKNN	Plot Function for Recursive Backward Elimination Feature Selection
plot.supportRKNN	Plot Function for Support Criterion
predicted	Prediced Value From a Linear Model
print.KNNregcv	Print Method for KNN Regression Cross-validation
print.beKNN	Print Method for Recursive Backward Elimination Feature Selection
print.randomKNN	Print method for Random KNN regression cross-validation
print.supportRKNN	Print Method for Random KNN Support Criterion
r	Choose number of KNNs
randomKNN	Random KNN Classification and Regression
rknn-package	Random KNN Classification and Regression
rsqp	Predicted R-square
supportRKNN	Support Criterion
varUsed	Features Used or Not Used in Random KNN

Author(s)

Shengqiao Li

Maintainer: Shengqiao Li <lishengqiao@yahoo.com>

References

Shengqiao Li, E James Harner and Donald A Adjeroh. *Random KNN feature selection - a fast and stable alternative to Random Forests.* BMC Bioinformatics 2011, 12:450. http://www.biomedcentral.com/1471-2105/12/450

bestset *Extract the Best Subset of Feature from Selection Process*

Description

Extract the best subset of feature from selection process.

Usage

```
bestset(x, criterion=c("mean_accuracy", "mean_support"))
prebestset(x,  criterion=c("mean_accuracy", "mean_support"))
```

Arguments

x	An object returned by rknnBeg or rknnBel.
criterion	either uses mean_accuracy or mean_support for best.

Value

A character vector of feature names.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

See Also

rknnBeg, rknnBel

Examples

```
## Not run: bestset{x}
```

confusion *Classification Confusion Matrix and Accuracy*

Description

Compute classification confusion matrix and accuracy

Usage

```
confusion(obs, pred)
confusion2acc(ct)
```

Arguments

obs	A vector of observed classes.
pred	A vector of predicted classes.
ct	A table retruned from confusion

Value

confusion return a matrix of cross classification counts. confusion return a overall classification accuracy.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

Examples

```
obs<- rep(0:1, each =5);
pre<- c(obs[3:10], obs[1:2])
confusion(obs, pre)
confusion2acc(  confusion(obs, pre))
```

`cv.coef` *Coefficient of Variation*

Description

Compute coefficient of variation.

Usage

```
cv.coef(x)
```

Arguments

`x`	A numeric vector.

Value

A number within [0, 1].

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

`eta` *Coverage Probability*

Description

Calculate the coverage probability

Usage

```
eta(p, m, r, method = c("binomial", "poisson", "exact"))
```

Arguments

`p`	Total number of available features.
`m`	Number of features to be drawn by each KNN.
`r`	Number of KNN to be generated.
`method`	Either binomial approximation, poisson approximaton or exact method.

Details

"exact" method needs gmp pakage and is slow for large `p`.

Value

The coverage probability is returned.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

See Also

`r`

Examples

```
eta(1000, 32, 100)
```

`fitted` *Extract Model Fitted Values*

Description

Extract Random KNN fitted values.

Usage

```
## S3 method for class 'rknn'
fitted(object, ...)
```

Arguments

`object`	A rknnobject.
`...`	Additional arguments.

Value

A vector of fitted values.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

Examples

```
## Not run:
  golub.train<- t(scale(golubTrain));
  golub.test<- t(scale(golubTest));
  golub.rnn<- randomKNN(data=golub.train, newdata=golub.test, y=golubTrain.cl,
     r=821, mtry=55);
  fitted(golub.rnn)

## End(Not run)
```

`lambda` *Compute Number of Silent Features*

Description

Compute number of silent features for Random KNN

Usage

```
lambda(p, m, r)
```

Arguments

`p`	Total number of available features.
`m`	Number of features to be drawn for each KNN.
`r`	Number of KNN to be generated.

Value

A scalar for the mean silent features.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

`normalize` *Data Normalization*

Description

Data matrix normalization procedures.

Usage

```
normalize.decscale(data)
normalize.sigmoidal(data)
normalize.softmax(data)
normalize.unit(data)
```

Arguments

`data`	A data matrix to be normalized.

Value

A normalized data matrix.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

---

plot backward elimination *Plot Function for Recursive Backward Elimination Feature Selection*

Description

Plot the recursive backward elimination feature selection process.

Usage

```
## S3 method for class 'rknnBeg'
plot(x, col = "springgreen4", xlab = "no. of features", ylab = "mean accuracy", ...)
## S3 method for class 'rknnBel'
plot(x, col = "springgreen4", xlab = "no. of features", ylab = "mean accuracy", ...)
```

Arguments

`x`	An object returned from `rknnBel` or `rknnBeg`.
`col`	Plot line color.
`xlab`	Plot x label.
`ylab`	Plot x label.
`...`	Addional plot parameters.

Author(s)

Shengqiao Li

Maintainer: Shengqiao Li<lishengqiao@yahoo.com>

---

plot rknn support *Plot Function for Support Criterion*

Description

Plot `support` of the important features.

Usage

```
## S3 method for class 'rknnSupport'
plot(x, n.var = min(30, length(x$support)),
    main = deparse(substitute(x)), bg = "gold", lcolor = "blue", ...)
```

Arguments

`x`	a list with `support` returned from `rknnSupport`.
`n.var`	number of variables to be displayed.
`main`	text for main title.
`bg`	background color.
`lcolor`	line color.
`...`	addtional `plot` arguments.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

`predicted` *Prediced Value From a Linear Model*

Description

Extraceted predicted values from a linear model.

Usage

```
predicted(obj)
```

Arguments

`obj`	A linear model.

Value

A vector of ppredicted values.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

PRESS *Predicted Residual Sum of Squares*

Description

Predicted Residual Sum of Squares

Usage

```
PRESS(obj)
```

Arguments

`obj`	A linear model or knn regression

Value

returns predicted residual sum of squares

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

print.rknn *Print method for Random KNN*

Description

Print method for Random KNN

Usage

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

Arguments

`x`	A `rknn` object.
`...`	Additonal `print` arguments.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

print.rknnBE *Print Method for Recursive Backward Elimination Feature Selection*

Description

Print summary of recursive backward elimination feature selection.

Usage

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

Arguments

x	A beKNN object returned by rknnBeg or rknnBel.
...	Addtional arguments to print method.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

print.rknnSupport *Print Method for Random KNN Support Criterion*

Description

Print support summary of the features.

Usage

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

Arguments

x	A list returned from rknnSupport or rknnRegSupport.
...	Additonal print arguments.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

r *Choose number of KNNs*

Description

Choose number of KNNs

Usage

```
r(p, m = floor(sqrt(p)), eta = 0.99, nu = 20, rmax = p, nsim = 1000,
    lambda = 0.01, method = c("binomial", "poisson", "nu",
    "geo.independent", "geo.sim", "geo.dependent", "lambda"))
```

Arguments

`p`	Total number of available features.
`m`	Number of features to be drawn by each KNN.
`eta`	Coverage Probability.
`nu`	mean mutiplicity of a feature
`rmax`	number of series terms for independent geometric approximation
`nsim`	number of simulations for geometric simulation.
`lambda`	mean number of silient features.
`method`	one of binomial, poisson, nu, geo.independent, geo.sim, geo.dependent, lambda

Details

Method `binomial` and `poisson` are approximation method for a given `eta` value.

Method `nu` computes `r` for a given `nu`.

Method `geo.independent`, `geo.sim` and `geo.dependent` compute `r` using geomerical random variables `z` until eachh is at least drawn once. The difference is that `geo.independent` ignores dependency,`geo.sim` is a simulation method and `geo.independent` is an exact method using `gmp` package.

Method `lambda` computes `r` for a given `lambda`.

Value

An integer.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

Examples

```
r(100, 10);
```

rknn *Random KNN Classification and Regression*

Description

Random KNN Classification and Regression

Usage

```
rknn(data, newdata, y, k = 1, r = 500, mtry = trunc(sqrt(ncol(data))),
        cluster = NULL, seed = NULL)
rknn.cv(data, y, k = 1, r = 500, mtry=trunc(sqrt(ncol(data))),
        cluster=NULL, seed = NULL)
rknnReg(data, newdata, y, k=1, r=500,  mtry=trunc(sqrt(ncol(data))),
        cluster=NULL, seed=NULL)
```

Arguments

`data`	A training dataset.
`newdata`	A testing dataset.
`y`	A vector of responses.
`k`	Number of nearest neighbors.
`r`	Number of KNNs.
`mtry`	Number of features to be drawn for each KNN.
`cluster`	An object of class ‘c("SOCKcluster", "cluster")’
`seed`	An integer seed.

Value

Return a `RandomKNN` object.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

rknnBeg *Backward Elimination Feature Selection with Random KNN*

Description

Recursive Backward Elimination Feature Selection with Random KNN

Usage

```
rknnBeg(data, y, k = 1, r = 500, mtry = trunc(sqrt(ncol(data))),
  fixed.partition = FALSE, pk = 0.5, stopat = 4, cluster=NULL, seed = NULL)
rknnBel(data, y, k = 1, r = 500, mtry = trunc(sqrt(ncol(data))),
  fixed.partition = FALSE, d = 1, stopat = 4,  cluster=NULL, seed = NULL)
```

Arguments

data	An n x p numeric design matrix.
y	A vector of responses. For a numeric vector, Random Knn regression is performed. For a factor, Random classification is performed.
k	An integer for the number of nearest neighbors.
r	An integer for the number of base KNN models.
mtry	Number of features to be drawn for each KNN.
fixed.partition	Logical. Use fixed partition of dynamic partition of the data into training and testing subsets for each KNN.
pk	A real number between 0 and to indicate the proportion of the feature set to be kept in each step.
d	A integer to indicate the number of features to be dropped in each step.
stopat	an integer for the minimum number of variables.
cluster	An object of class ‘c("SOCKcluster", "cluster")’
seed	An integer seed.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

rknnSupport *Support Criterion*

Description

Compute support criterion using Random KNN classification or regression

Usage

```
rknnSupport(data, y, k = 1, r = 500, mtry = trunc(sqrt(ncol(data))),
    fixed.partition = FALSE, cluster=NULL, seed = NULL)
rknnRegSupport(data, y, k = k, r = 500, mtry = trunc(sqrt(ncol(data))),
    fixed.partition = FALSE, cluster=NULL, seed = NULL)
```

Arguments

data	The input dataset.
y	A vector of responses.
k	Number of nearest neighbors.
r	Number of KNNs.
mtry	Number of features to be drawn for each KNN.
fixed.partition	Logical. Use fixed partition of dynamic partition of the data into training and testing subsets for each KNN.
cluster	An object of class ‘c("SOCKcluster", "cluster")’
seed	An integer seed.

Value

A supportKNN object.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

`rsqp` *Predicted R-square*

Description

Computed predicted R-square

Usage

```
rsqp(obj)
```

Arguments

`obj`	A regression model.

Value

returns predicted R-square

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

`varUsed` *Features Used or Not Used in Random KNN*

Description

List the features used or not used in Random KNN modeling.

Usage

```
varUsed(x, by.KNN = FALSE, count = TRUE)
varNotUsed(x)
```

Arguments

`x`	A `randomKNN` object.
`by.KNN`	Logical. Should list features used in each KNN?
`count`	Logical. Should list number of times each used.

Value

`varNotUsed` returns a list features not used. `varUsed` returns a list of feartures used.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

Index