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 ...................................................................... 2
bestset ............................................................................... 3
confusion .............................................................................. 4
cv.coef ............................................................................... 5
eta ....................................................................................... 5
fitted ................................................................................... 6
lambda ............................................................................... 7
normalize .......................................................................... 7
plot backward elimination ....................................................... 8
plot rknn support ................................................................. 8
predicted ........................................................................... 9
PRESS ............................................................................... 10
rknn-package

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
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.

Author(s)

Shengqiao Li
Maintainer: Shengqiao Li <lishengqiao@yahoo.com>

References

Value
   A character vector of feature names.

Author(s)
   Shengqiao Li<lishengqiao@yahoo.com>

See Also
   rknnBeg, rknnBel

Examples
   ## Not run: bestset(x)

---

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 returned 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 approximation or exact method.

**Details**
“exact” method needs gmp package and is slow for large `p`. 
Value

The coverage probability is returned.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

See Also

r

Examples

es(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**  
\[ \text{lambda}(p, m, r) \]

**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**  
- \( \text{data} \): 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
```r
## S3 method for class 'rknnBel'
plot(x, col = "springgreen4", xlab = "no. of features", ylab = "mean accuracy", ...)
## S3 method for class 'rknnBeg'
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.
- `...`: Additional 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
```r
## S3 method for class 'rknnSupport'
plot(x, n.var = min(30, length(x$support)),
     main = deparse(substitute(x)), bg = "gold", lcolor = "blue", ...)
```
predicted

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.
- **...**: additional plot arguments.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>

---

**Predicted Value From a Linear Model**

Description

Extraceted predicted values from a linear model.

Usage

```r
predicted(obj)
```

Arguments

- **obj**: A linear model.

Value

A vector of predicted values.

Author(s)

Shengqiao Li<lishengqiao@yahoo.com>
**PRESS**

*Predicted Residual Sum of Squares*

**Description**

Predicted Residual Sum of Squares

**Usage**

```r
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**

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

**Arguments**

- `x` A rknn object.
- `...` Additional print arguments.

**Author(s)**

Shengqiao Li<lishengqiao@yahoo.com>
**Print Method for Recursive Backward Elimination Feature Selection**

**Description**

Print summary of recursive backward elimination feature selection.

**Usage**

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

**Arguments**

- `x` A `bkNN` object returned by `rknnBeg` or `rknnBel`.
- `...` Additional arguments to `print` method.

**Author(s)**

Shengqiao Li\<lishengqiao@yahoo.com>

---

**Print Method for Random KNN Support Criterion**

**Description**

Print support summary of the features.

**Usage**

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

**Arguments**

- `x` A list returned from `rknnSupport` or `rknnRegSupport`.
- `...` Additional print arguments.

**Author(s)**

Shengqiao Li\<lishengqiao@yahoo.com>
Choose number of KNNs

Description
Choose number of KNNs

Usage
\[ r(p, m = \text{floor}(\sqrt{p})), \text{eta} = 0.99, \text{nu} = 20, \text{rmax} = p, \text{nsim} = 1000, \text{lambda} = 0.01, \text{method} = \text{c} ("\text{binomial}", "\text{poisson}", "\text{nu}", "\text{geo.independent}", "\text{geo.sim}", "\text{geo.dependent}", "\text{lambda}") \]

Arguments
- **p**: Total number of available features.
- **m**: Number of features to be drawn by each KNN.
- **eta**: Coverage Probability.
- **nu**: Mean multiplicity of a feature
- **rmax**: Number of series terms for independent geometric approximation
- **nsim**: Number of simulations for geometric simulation.
- **lambda**: Mean number of silent 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 geometrical random variables z until each 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

\[
\text{rknn}(\text{data}, \text{newdata}, \text{y}, k = 1, r = 500, \text{mtry} = \text{trunc}(\sqrt{\text{ncol}(\text{data}))}, \\
\text{cluster} = \text{NULL}, \text{seed} = \text{NULL}) \\
\text{rknn.ccv}(\text{data}, \text{y}, k = 1, r = 500, \text{mtry} = \text{trunc}(\sqrt{\text{ncol}(\text{data}))}, \\
\text{cluster} = \text{NULL}, \text{seed} = \text{NULL}) \\
\text{rknnReg}(\text{data}, \text{newdata}, \text{y}, k = 1, r = 500, \text{mtry} = \text{trunc}(\sqrt{\text{ncol}(\text{data}))}, \\
\text{cluster} = \text{NULL}, \text{seed} = \text{NULL})
\]

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>
**Description**

Recursive Backward Elimination Feature Selection with Random KNN

**Usage**

```r
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

<table>
<thead>
<tr>
<th>rknnSupport</th>
<th>Support Criterion</th>
</tr>
</thead>
</table>

**Description**

Compute support criterion using Random KNN classification or regression

**Usage**

```r
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 features used.

**Author(s)**

Shengqiao Li<lishengqiao@yahoo.com>
Index

*Topic **classif**
  confusion, 4
  rknnBeg, 14

*Topic **distribution**
  eta, 5
  lambda, 7
  r, 12

*Topic **hplot**
  plot backward elimination, 8
  plot rknn support, 8

*Topic **manip**
  normalize, 7

*Topic **models**
  fitted, 6
  predicted, 9
  rknn, 13

*Topic **multivariate**
  bestset, 3
  rknn, 13
  rknnBeg, 14
  rknnSupport, 15
  varUsed, 16

*Topic **package**
  rknn-package, 2

*Topic **print**
  print rknn, 10
  print rknnBE, 11
  print rknnSupport, 11

*Topic **regression**
  PRESS, 10
  rknnBeg, 14
  rsqp, 16

*Topic **univar**
  cv.coef, 5

bestset, 3

confusion, 4
confusion2acc (confusion), 4
cv.coef, 5

eta, 5
fitted, 6
lambda, 7
normalize, 7
plot backward elimination, 8
plot rknn support, 8
plot rknnBeg (plot backward elimination), 8
plot rknnBel (plot backward elimination), 8
plot rknnSupport (plot rknn support), 8
prebestset (bestset), 3
predicted, 9
PRESS, 10
print rknn, 10
print rknnBE, 11
print rknnSupport, 11
r, 6, 12
rknn, 13
rknn-package, 2
rknnBeg, 4, 14
rknnBel, 4
rknnBel (rknnBeg), 14
rknnReg (rknn), 13
rknnRegSupport (rknnSupport), 15
rknnSupport, 15
rsqp, 16
varNotUsed (varUsed), 16
varUsed, 16