

# Package ‘rknn’

July 2, 2014

**Type** Package

**Title** Random KNN Classification and Regression

**Version** 1.2

**Date** 2013-09-09

**Author** Shengqiao Li <shli@stat.wvu.edu>

**Maintainer** Shengqiao Li <shli@stat.wvu.edu>

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

**Suggests** Hmisc, Biobase, genefilter, golubEsets, chemometrics

**Description** Random KNN Classification and Regression

**License** GPL (>= 2)

**LazyLoad** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2013-09-10 15:50:35

## 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
print.rknn	10
print.rknnBE	11
print.rknnSupport	11
r	12
rknn	13
rknnBeg	14
rknnSupport	15
rsqp	16
varUsed	16

## Index 17

---

rknn-package	<i>Random KNN Classification and Regression</i>
--------------	---

---

### 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 &lt;shli@stat.wvu.edu&gt;

**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<shli@stat.wvu.edu>

**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<shli@stat.wvu.edu>

**Examples**

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

---

cv.coef	<i>Coefficient of Variation</i>
---------	---------------------------------

---

**Description**

Compute coefficient of variation.

**Usage**

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

**Arguments**

x                    A numeric vector.

**Value**

A number within [0, 1].

**Author(s)**

Shengqiao Li<shli@stat.wvu.edu>

---

eta	<i>Coverage Probability</i>
-----	-----------------------------

---

**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<shli@stat.wvu.edu>

**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<shli@stat.wvu.edu>

**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	<i>Compute Number of Silent Features</i>
--------	--

---

**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<shli@stat.wvu.edu>

---

normalize	<i>Data Normalization</i>
-----------	---------------------------

---

**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<shli@stat.wvu.edu>

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.
...	Additional plot parameters.

**Author(s)**

Shengqiao Li

Maintainer: Shengqiao Li<shli@stat.wvu.edu>

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.
- ... additional plot arguments.

**Author(s)**

Shengqiao Li<shli@stat.wvu.edu>

---

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<shli@stat.wvu.edu>

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<shli@stat.wvu.edu>

---

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<shli@stat.wvu.edu>

---

print.rknnBE	<i>Print Method for Recursive Backward Elimination Feature Selection</i>
--------------	--

---

**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.
...	Additional arguments to print method.

**Author(s)**

Shengqiao Li<shli@stat.wvu.edu>

---

print.rknnSupport	<i>Print Method for Random KNN Support Criterion</i>
-------------------	--

---

**Description**

Print support summary of the features.

**Usage**

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

**Arguments**

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

**Author(s)**

Shengqiao Li<shli@stat.wvu.edu>

---

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 geometrical 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

Maintainer: Shengqiao Li<shli@stat.wvu.edu>

**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<shli@stat.wvu.edu>

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 1 to indicate the proportion of the feature set to be kept in each step.
d	An 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<shli@stat.wvu.edu>

---

rknnSupport	<i>Support Criterion</i>
-------------	--------------------------

---

### 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<shli@stat.wvu.edu>

---

rsqp	<i>Predicted R-square</i>
------	---------------------------

---

**Description**

Computed predicted R-square

**Usage**

```
rsqp(obj)
```

**Arguments**

obj	A regression model.
-----	---------------------

**Value**

returns predicted R-square

**Author(s)**

Shengqiao Li<shli@stat.wvu.edu>

---

varUsed	<i>Features Used or Not Used in Random KNN</i>
---------	--

---

**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<shli@stat.wvu.edu>



# 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
- 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
- bestset, 3
- confusion, 4
- confusion2acc (confusion), 4
- cv.coef, 5