Package 'wNNSel'

November 9, 2017

Type Package

Title Weighted Nearest Neighbor Imputation of Missing Values using Selected Variables

Version 0.1

Author Shahla Faisal

Maintainer Shahla Faisal <shahla_ramzan@yahoo.com>

Description New tools for the imputation of missing values in high-dimensional data are introduced using the non-parametric nearest neighbor methods. It includes weighted nearest neighbor imputation methods that use specific distances for selected variables. It includes an automatic procedure of cross validation and does not require prespecified values of the tuning parameters. It can be used to impute missing values in high-dimensional data when the sample size is smaller than the number of predictors. For more information see Faisal and Tutz (2017) <doi:10.1515/sagmb-2015-0098>.

Imports stats

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-11-09 11:32:18 UTC

R topics documented:

`wNNSel-package` *Weighted Nearest Neighbor Imputation of Missing Values using Selected Variables*

Description

This package introduces new non-parametric tools for the imputation of missing values in high-dimensional data. It includes weighted nearest neighbor imputation methods that use distances for selected covariates. The careful selection of distances that carry information about the missing values yields an imputation tool. It does not require pre-specified k, unlike other kNN methods. It can be used to impute missing values in high-dimensional data when $n < p$.

Details

Package:	wNNSel
Version:	0.1
Date:	2017-11-08
Depends:	R (>= 2.10)
License:	GPL (>= 2)

The main function of the package is `wNNSel` for implementing the nonparameteric procedure of nearest neighbors imputaiton. See `wNNSel` for more details.

Note

*Author's Last name changed to *Faisal* from *Ramzan* in 2016.

Author(s)

Shahla Faisal <shahla_ramzan@yahoo.com>

References

Tutz, G. and Ramzan,S*. (2015). Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics and Data Analysis*, Vol. 90, pp. 84-99.

Faisal, S.* and Tutz, G. (2017). Missing value imputation for gene expression data by tailored nearest neighbors. *Statistical Application in Genetics and Molecular Biology*. Vol. 16(2), pp. 95-106.

artifNA *Introduce MCAR Missing Values in a matrix*

Description

This function artificially introduces missing values in a data matrix under missing completely at random (MCAR) mechanism.

Usage

```
artifNA(x, miss.prop = 0.1)
```

Arguments

x	a matrix, in which missing values are to be created.
miss.prop	proportion of missing values

Value

a matrix with missing values

Examples

```
set.seed(3)
x = matrix(rnorm(100),10,10)
## create 10% missing values in x
artifNA(x, 0.10)
```

artifNA.cv *Introduce MCAR Missing Values in a matrix for cross validation*

Description

This function introduces additional missing values in a missing data matrix artificially. The missing values are introduced under missing completely at random (MCAR) mechanism.

Usage

```
artifNA.cv(x, testNA.prop = 0.1)
```

Arguments

x	a matrix, in which missing values are to be created.
testNA.prop	proportion of missing values

Value

a list contatining a matrix with artifical missing values, removed indices and the provided x matrix

See Also

cv.wNNSel

Examples

```
set.seed(3)
x = matrix(rnorm(100),10,10)
## create 10% missing values in x
x.miss<- artifNA(x, 0.10)
## create another 10% missing values in x
x.miss.cv<- artifNA.cv(x, 0.10)
summary(x.miss)
summary(x.miss.cv)
```

---

computeMAIE *Mean Absolute Imputation Error*

Description

This function computes the mean absolute imputation error for a given complete/true data matrix, imputed data matrix and the data matrix with missing values.

Usage

```
computeMAIE(x.miss, x.impute, x.true)
```

Arguments

x.miss	a matrix, having missing values
x.impute	an imputed data matrix. Note that it should not contain any missing values.
x.true	complete/true data matrix. Note that it should not contain any missing values.

Value

value of MSIE

Examples

```
set.seed(3)
x.true = matrix(rnorm(100),10,10)
## create 10% missing values in x
x.miss = artifNA(x.true, 0.10)
## impute using wNNSel method
x.impute = wNNSel.impute(x.miss)
computeMAIE(x.miss, x.impute, x.true)
```

computeMSIE *Mean Squared Imputation Error*

Description

This function computes the mean squared imputation error for a given complete/true data matrix, imputed data matrix and the data matrix with missing values.

Usage

```
computeMSIE(x.miss, x.impute, x.true)
```

Arguments

x.miss	a matrix, having missing values
x.impute	an imputed data matrix. Note that it should not contain any missing values.
x.true	complete/true data matrix. Note that it should not contain any missing values.

Value

value of MSIE

Examples

```
set.seed(3)
x.true = matrix(rnorm(100),10,10)
## create 10% missing values in x
x.miss = artifNA(x.true, 0.10)
## impute using wNNSel method
x.impute = wNNSel.impute(x.miss)
computeMSIE(x.miss, x.impute, x.true)
```

computeNRMSE *Normalized Root Mean Squared Imputatoin Error*

Description

This function computes the nrmalized root mean squared imputation error for a given complete/true data matrix, imputed data matrix and the data matrix with missing values.

Usage

```
computeNRMSE(x.miss, x.impute, x.true)
```

Arguments

x.miss	a matrix, having missing values
x.impute	an imputed data matrix. Note that it should not contain any missing values.
x.true	complete/true data matrix. Note that it should not contain any missing values.

Value

value of MSIE

Examples

```
set.seed(3)
x.true = matrix(rnorm(100),10,10)
## create 10% missing values in x
x.miss = artifNA(x.true, 0.10)
## impute using wNNSel method
x.impute = wNNSel.impute(x.miss)
computeNRMSE(x.miss, x.impute, x.true)
```

---

cv.wNNSel *Cross Validation for wNNSel Imputation*

Description

This function aims to search for optimal values of the tuning parameters for the wNNSel imputation.

Usage

```
cv.wNNSel(x, kernel = "gaussian", x.dist = "euclidean", method = "2",
  m.values = seq(2, 8, by = 2), c.values = seq(0.1, 0.5, by = 0.1),
  lambda.values = seq(0, 0.6, by = 0.01)[-1], times.max = 5,
  testNA.prop = 0.05)
```

Arguments

x	a matrix containing missing values
kernel	kernel function to be used in nearest neighbors imputation. Default kernel function is "gaussian".
x.dist	distance to compute, The default is x.dist="euclidean" to compute Euclidean distance. Set x.dist to NULL to use Manhattan distance.
method	convex function, performs selection of variables. If method="1", linear function is used and when if method="c", power function is used.
m.values	a vector of integer values, required when mehtod="2".
c.values	a vector between 0 and less than 1. It is required when mehtod="1".
lambda.values	a vector, for the tuning parameter λ

times.max	maximum number of repititions for the cross validation procedure.
testNA.prop	proportion of values to be deleted artificially for cross validation in the missing matrix x. Default method uses 5 percent.

Details

Some values are artificially deleted and wNNSel is run multiple times, varying λ and m. For each pair of λ and m, compute MSIE on the subset of the data matrix x for which the the values were deleted artificially. (See References for more detail).

Value

a list containing

lambda.opt	optimal parameter selected by cross validation
m.opt	optimal parameter selected by cross validation
MSIE.cv	cross validation error

Author(s)

Shahla Faisal <shahla_ramzan@yahoo.com>

References

Tutz, G. and Ramzan,S. (2015). Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics and Data Analysis*, Vol. 90, pp. 84-99.

Faisal, S. and Tutz, G. (2017). Missing value imputation for gene expression data by tailored nearest neighbors. *Statistical Application in Genetics and Molecular Biology*. Vol. 16(2), pp. 95-106.

See Also

artifNA.cv, wNNSel

Examples

```
set.seed(3)
x.true = matrix(rnorm(100),10,10)
## create 10% missing values in x
x.miss = artifNA(x.true, 0.10)
## use cross validation to find optimal values
result = cv.wNNSel(x.miss)
## optimal values are
result$lambda.opt
result$m.opt
## Now use these values to get final imputation
x.impute = wNNSel.impute(x.miss, lambda=result$lambda.opt, m=result$m.opt)
## and final MSIE
computeMSIE(x.miss, x.impute, x.true)
```

wNNSel *Imputatin using wNNSel method.*

Description

'wNNSel' is used to impute the missing values particularly in high dimensional data. It uses a cross validation procedure for selecting the best values of the tuning parameters. It also works when the samples are smaller than the covariates.

Usage

```
wNNSel(x, x.initial = NULL, x.true = NULL, k, useAll = TRUE,
  x.dist = "euclidean", kernel = "gaussian", method = "2", impute.fn,
  convex = TRUE, m.values = seq(2, 8, by = 2), c.values = seq(0.1, 0.5, by
  = 0.1), lambda.values = seq(0, 0.6, by = 0.01)[-1], times.max = 5,
  testNA.prop = 0.05, withinFolds = FALSE, folds, verbose = TRUE)
```

Arguments

x	a numeric data matrix containing missing values
x.initial	an optional. A complete data matrix e.g. using mean imputation of x. If provided, it will be used for the computation of correlations.
x.true	a matrix of true or complete data. If provided, MSIE will be returned in the results list.
k	an optional, the number of nearest neighbors to use for imputation.
useAll	logical. If TRUE, all *available* neighbors are used for the imputation.
x.dist	distance to compute. The default is x.dist="euclidean", that uses the Euclidean distance. Set x.dist to NULL for Manhattan distance.
kernel	kernel function to be used in nearest neighbors imputation. Default kernel function is "gaussian".
method	convex function, performs selection of variables. If method="1", linear function is used and the power function is used when method="2".
impute.fn	the imputation function to run on the length k vector of values for a missing feature. Defaults to a weighted mean of the neighboring values, weighted by the specified kernel. If not specified then wNN imputation will be used by default.
convex	logical. If TRUE, selected variables are used for the computation of distance. The default is TRUE.
m.values	a vector of integer values, required when mehtod="2".
c.values	a vector between 0 and less than 1. It is required when mehtod="1".
lambda.values	a vector, for the tuning parameter λ
times.max	maximum number of repititions for the cross validation procedure.
testNA.prop	proportion of values to be deleted artificially for cross validation in the missing matrix x. Default method uses 5 percent.

withinFolds	logical. Use only if the neighbors/rows belong to particular folds/groups. Default is set to FALSE.
folds	a list of vectors specifying folds/groups for neighbors. lenght of list is equal to the number of folds/groups. Each element/vector of the list indicates row indices belonging to that particular group/fold.
verbose	logical. If TRUE, prints status updates

Details

For each sample, identify missinng features. For each missing feature find the nearest neighbors which have that feature. Impute the missing value using the imputation function on the *selected* vector of values found from the neighbors. By default the wNNSel method automatically searches for optimal values for a given data matrix.

The default method uses x.dist="euclidean" including selected covariates. The specific distancs are computed using important covariates only. If mehtod="1", the linear function in absolute value of r is used, defined by

$$\frac{|r|}{1-c} - \frac{c}{1-c},$$

for $|r| > c$, and, 0 , otherwise. By default, the power function $|r|^m$ is used when mehtod="2". For more detailed discussion, see references.

Value

a list containing imputed data matrix, and cross validation results

x.impute	imputed data matrix
MSIE	True error. Note it is only available when x.true is provided.
lambda.opt	optimal parameter selected by cross validation
m.opt	optimal parameter selected by cross validation
MSIE.cv	cross validation error

References

Tutz, G. and Ramzan,S. (2015). Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics and Data Analysis*, Vol. 90, pp. 84-99.

Faisal, S. and Tutz, G. (2017). Missing value imputation for gene expression data by tailored nearest neighbors. *Statistical Application in Genetics and Molecular Biology*. Vol. 16(2), pp. 95-106.

See Also

cv.wNNSel, wNNSel.impute

Examples

```
set.seed(3)
x.true = matrix(rnorm(100),10,10)
## create 10% missing values in x
x.miss = artifNA(x.true, 0.10)
## imputed matrix
result <- wNNSel(x.miss)
result$x.impute
## cross validation result can be accessed using
result$cross.val
```

`wNNSel.impute` *Weighted Nearest Neighbor Imputation of Missing Values using Selected Variables*

Description

This function imputes the missing values using user-spefied values of the tuning parameters. It also works when the samples are smaller than the covariates.

Usage

```
wNNSel.impute(x, k, useAll = TRUE, x.initial = NULL, x.dist = "euclidean",
  kernel = "gaussian", lambda = 0.3, impute.fn, convex = TRUE,
  method = "2", m = 2, c = 0.3, withinFolds = FALSE, folds,
  verbose = TRUE, verbose2 = FALSE)
```

Arguments

`x`	a `matrix` containing missing values
`k`	an optional, the number of nearest neighbors to use for imputation.
`useAll`	`logical`. The default is `useALL=TRUE`, that is, all *available* neighbors are used for the imputation.
`x.initial`	an optional. A complete data matrix e.g. using mean imputation of `x`. If provided, it will be used for the computation of correlations.
`x.dist`	distance to compute. The default is `x.dist="euclidean"`, that uses the Euclidean distance. Set `x.dist` to `NULL` for Manhattan distance.
`kernel`	kernel function to be used in nearest neighbors imputation. Default kernel function is "gaussian".
`lambda`	`scaler`, a tuning parameter
`impute.fn`	the imputation function to run on the length k vector of values for a missing feature. Defaults to a weighted mean of the neighboring values, weighted by the specified `kernel`. If not specified then wNN imputation will be used by default.
`convex`	logical. If `TRUE`, selected variables are used for the computation of distance. The default is `TRUE`.

method	convex function, performs selection of variables. If method="1", linear function is used and the power function is used when method="2".
m	scaler, a tuning parameter required by the power function.
c	scaler, a tuning parameter required by the linear function.
withinFolds	logical. Use only if the neighbors/rows belong to particular folds/groups. Default is set to FALSE.
folds	a list of vectors specifying folds/groups for neighbors. lenght of list is equal to the number of folds/groups. Each element/vector of the list indicates row indices belonging to that particular group/fold.
verbose	logical. If TRUE, prints status updates
verbose2	logical. If TRUE, prints status updates with more detail

Details

For each sample, identify missinng features. For each missing feature find the nearest neighbors which have that feature. Impute the missing value using the imputation function on the *selected* vector of values found from the neighbors.

Value

imputed data matrix

See Also

cv.wNNSel, wNNSel

Examples

```
set.seed(3)
x = matrix(rnorm(100),10,10)
x.miss = x > 1
x[x.miss] = NA
wNNSel.impute(x)
wNNSel.impute(x, lambda=0.5, m=2)
```

Index