Package 'FORD'

May 30, 2025

Type Package

Title Feature Ordering by Integrated R Square Dependence

Version 0.1.2

Description Feature Ordering by Integrated R square Dependence (FORD) is a variable selection algorithm based on the new measure of dependence: Integrated R2 Dependence Coefficient (IRDC). For more information, see the paper: Azadkia and Roudaki (2025),``A New Measure Of Dependence: Integrated R2'' <doi:10.48550/arXiv.2505.18146>.

License GPL-3

Encoding UTF-8

URL https://github.com/PouyaRoudaki/FORD

BugReports https://github.com/PouyaRoudaki/FORD/issues

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, markdown, xfun, testthat, minerva, devtools, FOCI, XICOR, KPC, dplyr, ggplot2

VignetteBuilder knitr

Depends R (>= 3.6.0), data.table

Imports RANN, parallel

NeedsCompilation no

Author Pouya Roudaki [aut, cre],
Mona Azadkia [aut, ctb]

Maintainer Pouya Roudaki <roudaki.pouya@gmail.com>

Repository CRAN

Date/Publication 2025-05-30 09:40:12 UTC

Contents

---

ford   *Variable selection by the FORD algorithm*

Description

FORD is a variable selection algorithm based on Integrated R square dependence coefficient [irdc](#).

Usage

```
ford(
  Y,
  X,
  dist.type.X = "continuous",
  num_features = NULL,
  stop = TRUE,
  na.rm = TRUE,
  standardize = "scale",
  numCores = parallel::detectCores(),
  parPlat = "none",
  printIntermed = TRUE
)
```

Arguments

Y	Vector of responses (length n)
X	Matrix of predictors (n by p)
dist.type.X	A string specifying the distribution type of X: either "continuous" or "discrete". Default is "continuous".
num_features	Number of variables to be selected, cannot be larger than p. The default value is NULL and in that case it will be set equal to p. If stop == TRUE (see below), then num_features is irrelevant.
stop	Stops at the first instance of negative irdc, if TRUE.
na.rm	Removes NAs if TRUE.
standardize	Standardize covariates if set equal to "scale" or "bounded". Otherwise will use the raw inputs. The default value is "scale" and normalizes each column of X to have mean zero and variance 1. If set equal to "bounded" map the values of each column of X to [0, 1].
numCores	Number of cores that are going to be used for parallelizing the variable selection process.
parPlat	Specifies the parallel platform to chunk data by rows. It can take three values: 1- The default value is set to 'none', in which case no row chunking is done; 2- the `parallel` cluster to be used for row chunking; 3- "locThreads", specifying that row chunking will be done via threads on the host machine.
printIntermed	The default value is TRUE, in which case print intermediate results from the cluster nodes before final processing.

Details

ford is a forward stepwise algorithm that uses the conditional dependence coefficient ([irdc](#)) at each step, instead of the multiple correlation coefficient as in ordinary forward stepwise. If stop == TRUE, the process is stopped at the first instance of nonpositive irdc, thereby selecting a subset of variables. Otherwise, a set of covariates of size num_features, ordered according to predictive power (as measured by irdc) is produced.

Parallel computation:

The computation can be lengthy, so the package offers two kinds of parallel computation.

The first, controlled by the argument numCores, specifies the number of cores to be used on the host machine. If at a given step there are k candidate variables under consideration for inclusion, these k tasks are assigned to the various cores.

The second approach, controlled by the argument parPlat ("parallel platform"), involves the user first setting up a cluster via the **parallel** package. The data are divided into chunks by rows, with each cluster node applying ford to its data chunk. The union of the results is then formed, and fed through ford one more time to adjust the discrepancies. The idea is that that last step will not be too lengthy, as the number of candidate variables has already been reduced. A cluster size of r may actually produce a speedup factor of more than r (Matloff 2016).

Potentially the best speedup is achieved by using the two approaches together.

The first approach cannot be used on Windows platforms, as parallel::mcapply has no effect. Windows users should thus use the second approach only.

In addition to speed, the second approach is useful for diagnostics, as the results from the different chunks gives the user an idea of the degree of sampling variability in the ford results.

In the second approach, a random permutation is applied to the rows of the dataset, as many datasets are sorted by one or more columns.

Note that if a certain value of a feature is rare in the full dataset, it may be absent entirely in some chunk.

Value

An object of class "ford", with attributes selectedVar, showing the selected variables in decreasing order of predictive power, and step_nu, listing the 'irdc' values. Typically the latter will begin to level off at some point, with additional marginal improvements being small.

Author(s)

Mona Azadkia, Pouya Roudaki

References

Azadkia, M. and Roudaki, P. (2025). A New Measure Of Dependence: Integrated R2 http://arxiv.org/abs/2505.18146.

Matloff, N. (2016). Software Alchemy: Turning Complex Statistical Computations into Embarrassingly-Parallel Ones. *J. of Stat. Software.*

See Also

irdc, foci, KFOCI

Examples

```
# Example 1
n = 500
p = 10
x <- matrix(rnorm(n * p), nrow = n)
colnames(x) = paste0(rep("x", p), seq(1, p))
y <- x[, 1] * x[, 8] + x[, 10]^2
# with num_features equal to 3 and stop equal to FALSE, ford will give a list of
# three selected features
result1 = ford(y, x, num_features = 3, stop = FALSE, numCores = 1)
result1
# Example 2
# same example, but stop according to the stopping rule
result2 = ford(y, x, numCores = 1)
result2
```

---

irdc   *Estimate the Integrated R-squared Dependence Coefficient (irdc)*

Description

The Integrated R-squared Dependence Coefficient (irdc) is a measure of dependence between a random variable Y and a random vector X, based on an i.i.d. sample of (Y, X). The estimated coefficient is asymptotically guaranteed to lie between 0 and 1. The measure is asymmetrical; that is, irdc(X, Y) != irdc(Y, X). The measure equals 0 if and only if X is independent of Y, and it equals 1 if and only if Y is a measurable function of X. This coefficient has several applications; for example, it can be used for variable selection, as demonstrated in the ford function.

Usage

```
irdc(Y, X, dist.type.X = "continuous", na.rm = TRUE)
```

Arguments

Y	A vector of length n.
X	A vector or matrix of length n (or with n rows).
dist.type.X	A string specifying the distribution type of X: either "continuous" or "discrete". Default is "continuous".
na.rm	Logical; if TRUE, missing values (NAs) will be removed. Default is TRUE.

Details

The value returned by 'irdc' can be positive or negative for finite samples, but asymptotically, it is guaranteed to be between 0 and 1. A small value indicates low dependence between Y and X, while a high value indicates strong dependence. The 'irdc' function is used by the [ford](#) function for variable selection.

Value

The Integrated R-squared Dependence Coefficient (irdc) between Y and X.

Author(s)

Mona Azadkia, Pouya Roudaki

References

Azadkia, M. and Roudaki, P. (2025). A New Measure Of Dependence: Integrated R2 http://arxiv.org/abs/2505.18146.

See Also

[ford](#), [irdc_simple](#), [codec](#), [xicor](#), [KPCgraph](#), [KPCRKHS](#)

Examples

```
n = 1000
x <- matrix(runif(n * 3), nrow = n)
y <- (x[, 1] + x[, 2])
irdc(y, x[, 1])
irdc(y, x[, 2])
irdc(y, x[, 3])
```

---

irdc_simple	*Simple Estimator of the Integrated R-squared Dependence Coefficient (irdc_simple) for 1 Dimensional continuous X and Y*

Description

The Simple Integrated R-squared Dependence Coefficient (irdc_simple) is a measure of dependence between a continuous random variables Y and X, based on an i.i.d. sample of (Y, X). The estimated coefficient is asymptotically guaranteed to lie between 0 and 1. The measure is asymmetrical; that is, irdc_simple(X, Y) != irdc_simple(Y, X). The measure equals 0 if and only if X is independent of Y, and it equals 1 if and only if Y is a measurable function of X. This coefficient has several applications; for example, it can be used for independence test. This coefficient only implemented for the 1-Dimensional continuous random variable X and Y.

Usage

```
irdc_simple(Y, X, na.rm = TRUE)
```

Arguments

Y	A vector of length n.
X	A vector of length n.
na.rm	Logical; if TRUE, missing values (NAs) will be removed. Default is TRUE.

Details

The value returned by 'irdc_simple' can be positive or negative for finite samples, but asymptotically, it is guaranteed to be between 0 and 1. A small value indicates low dependence between Y and X, while a high value indicates strong dependence. The 'irdc_simple' function is used for testing the independence of variables.

Value

The Simple Integrated R-squared Dependence Coefficient (irdc_simple) between Y and X.

Author(s)

Mona Azadkia, Pouya Roudaki

References

Azadkia, M. and Roudaki, P. (2025). A New Measure Of Dependence: Integrated R2 http://arxiv.org/abs/2505.18146.

See Also

[irdc](), [codec](), [xicor](), [KPCgraph](), [KPCRKHS]()

Examples

```
n = 1000
x <- matrix(runif(n * 3), nrow = n)
y <- (x[, 1] + x[, 2])
irdc_simple(y, x[, 1])
irdc_simple(y, x[, 2])
irdc_simple(y, x[, 3])
```

Index