Package ‘GreedyExperimentalDesign’

July 15, 2018

Type Package

Title Greedy Experimental Design Construction

Version 1.2

Date 2018-07-15

Author Adam Kapelner, David Azriel and Abba M. Krieger

Maintainer Adam Kapelner <kapelner@qc.cuny.edu>

Description Computes experimental designs for a
two-arm experiment with covariates by greedily optimizing a
balance objective function. This optimization provides lower
variance for the treatment effect estimator (and higher power)
while preserving a design that is close to complete randomization.
We return all iterations of the designs for use in a permutation test.
Additional functionality includes using branch and bound optimization
(via Gurobi) and exhaustive enumeration.

License GPL-3

Depends R (>= 3.2.0), rJava (>= 0.9-6), GreedyExperimentalDesignJARs (>= 1.0)

SystemRequirements Java (>= 7.0)

Imports graphics, grDevices, stats

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-15 21:40:10 UTC

R topics documented:

`automobile` *Data concerning automobile prices.*

Description

The `automobile` data frame has 201 rows and 25 columns and concerns automobiles in the 1985 Auto Imports Database. The response variable, `price`, is the log selling price of the automobile. There are 7 categorical predictors and 17 continuous / integer predictors which are features of the automobiles. 41 automobiles have missing data in one or more of the feature entries. This dataset is true to the original except with a few of the predictors dropped.

Usage

```
data(automobile)
```

Source

K Bache and M Lichman. UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml/datasets/Automobile

`complete_randomization` *Implements complete randomization*

Description

Implements complete randomization

Usage

```
complete_randomization(n, r)
```

Arguments

`n`	number of observations
`r`	number of randomized designs you would like

Value

a matrix where each column is one of the `r` designs

Author(s)

Adam Kapelner

`complete_randomization_with_balanced_ns` *Implements complete balanced randomization*

Description

Implements complete balanced randomization

Usage

```
complete_randomization_with_balanced_ns(n, r)
```

Arguments

`n`	number of observations
`r`	number of randomized designs you would like

Value

a matrix where each column is one of the `r` designs

Author(s)

Adam Kapelner

compute_objective_val

Returns the objective value given a design vector as well an an objective function. This is code duplication since this is implemented within Java. This is only to be run if...

Description

Returns the objective value given a design vector as well an an objective function. This is code duplication since this is implemented within Java. This is only to be run if...

Usage

```
compute_objective_val(X, indic_T, objective = "abs_sum_diff",
  inv_cov_X = NULL)
```

Arguments

`X`	The n x p design matrix
`indic_T`	The n-length binary allocation vector
`objective`	The objective function to use. Default is `abs_sum_diff`.
`inv_cov_X`	Optional: the inverse sample variance covariance matrix. Use this argument if you will be doing many calculations since passing this in will cache this data.

Author(s)

Adam Kapelner

compute_randomization_metrics

Computes Randomization Metrics (explained in paper) about a design algorithm

Description

Computes Randomization Metrics (explained in paper) about a design algorithm

Usage

```
compute_randomization_metrics(designs)
```

Arguments

`designs`	A matrix where each column is one design.

Value

A list of resulting data: the probability estimates for each pair in the design of randomness where estmates close to ~0.5 represent random assignment, then the entropy metric and the distance metric.

Author(s)

Adam Kapelner

`generate_stdzied_design_matrix` *Generates a design matrix with standardized predictors. Useful for debugging.*

Description

Generates a design matrix with standardized predictors. Useful for debugging.

Usage

```
generate_stdzied_design_matrix(n = 50, p = 1, covariate_gen = rnorm, ...)
```

Arguments

`n`	Number of rows in the design matrix
`p`	Number of columns in the design matrix
`covariate_gen`	The function to use to draw the covariate realizations (assumed to be iid). This defaults to `rnorm` for $N(0,1)$ draws.
`...`	Optional arguments to be passed to the `covariate_dist` function.

Value

THe design matrix

Author(s)

Adam Kapelner

`GreedyExperimentalDesign` *Greedy Experimental Design Search*

Description

A tool to find a priori experimental designs with good balance greedily

Author(s)

Adam Kapelner <`kapelner@qc.cuny.edu`>

References

Kapelner, A

`gurobi_multiple_designs` *Find multiple designs*

Description

This method searches through $1_T$ space using Gurobi's optimization many times. It finds many different solutions by permuting the rows of the design matrix and rerunning the optimization.

Usage

```
gurobi_multiple_designs(X, r, ...)
```

Arguments

`X` The design matrix with $n$ rows (one for each subject) and $p$ columns (one for each measurement on the subject). This is the design matrix you wish to search for a more optimal design.

`r` The number of vectors that should be returned

`...` Additional arguments to be passed to `initGurobiNumericalOptimizationExperimentalDesignObjec`

Value

A matrix of allocation vectors of dimension `r x n`.

Author(s)

Kapelner

initGreedyExperimentalDesignObject

This method creates an object of type greedy_experimental_design and will immediately initiate a search through $1_T$ space.

Description

This method creates an object of type greedy_experimental_design and will immediately initiate a search through $1_T$ space.

Usage

```
initGreedyExperimentalDesignObject(X, max_designs = 10000,
  objective = "abs_sum_diff", wait = FALSE, start = TRUE,
  max_iters = Inf, semigreedy = FALSE, diagnostics = FALSE,
  num_cores = 1)
```

Arguments

`X`	The design matrix with $n$ rows (one for each subject) and $p$ columns (one for each measurement on the subject). This is the design matrix you wish to search for a more optimal design.
`max_designs`	The maximum number of designs to be returned. Default is 10,000. Make this large so you can search however long you wish as the search can be stopped at any time by using the `stopSearch` method
`objective`	The objective function to use when greedily searching design space. This is a string "`abs_sum_diff`" (default) or "`mahal_dist.`"
`wait`	Should the `R` terminal hang until all `max_designs` vectors are found? The deafult is `FALSE`.
`start`	Should we start searching immediately (default is `TRUE`).
`max_iters`	Should we impose a maximum number of greedy switches? The default is `Inf` which a flag for “no limit.”
`semigreedy`	Should we use a fully greedy approach or the quicker semi-greedy approach? The default is `FALSE` corresponding to the fully greedy approach.
`diagnostics`	Returns diagnostic information about the iterations including (a) the initial starting vectors, the switches at every iteration and information about the objective function at every iteration (default is `FALSE` due to speed concerns).
`num_cores`	The number of CPU cores you wish to use during the search. The default is `1`.

Value

An object of type `greedy_experimental_design_search` which can be further operated upon

Author(s)

Adam Kapelner

Examples

```
 ## Not run:
library(MASS)
data(Boston)
 #pretend the Boston data was an experiment setting
#first pull out the covariates
 X = Boston[, 1 : 13]
 #begin the greedy design search
ged = initGreedyExperimentalDesignObject(X,
max_designs = 1000, num_cores = 3, objective = "abs_sum_diff")
#wait
ged

## End(Not run)
```

---

initGurobiNumericalOptimizationExperimentalDesignObject

This method creates an object of type optimal_experimental_design and will immediately initiate a search through $1_T$ space.

Description

This method creates an object of type optimal_experimental_design and will immediately initiate a search through $1_T$ space.

Usage

```
initGurobiNumericalOptimizationExperimentalDesignObject(X, num_cores = 1,
  time_limit_min = 5, node_limit = NULL, max_solutions = 10,
  verbose = TRUE, log_file = "")
```

Arguments

`X` The design matrix with $n$ rows (one for each subject) and $p$ columns (one for each measurement on the subject). This is the design matrix you wish to search for a more optimal design.

`num_cores` The number of CPU cores you wish to use during the search. The default is `1`.

`time_limit_min` The maximum amount of time the optimizer can run for in minutes. The default is `5`.

`node_limit` The maximum number of nodes Gurobi should explore. Default is `NULL` for no limit.

`max_solutions` The maximum number of solutions Gurobi should retain (if possible given the time limit and constraint of the node limit). The default is `NULL` for Gurobi's default of 10.

`verbose` Should Gurobi print its log to screen? Default is `TRUE`.

`log_file` Log filename for Gurobi e.g. `my_log.txt`. Default is `""` for no file log.

Value

An object of type `optimal_experimental_design_search` which can be further operated upon

Author(s)

Adam Kapelner and Bracha Blau

`initKarpExperimentalDesignObject` *This method creates an object of type karp_experimental_design and will immediately initiate a search through $1_T$ space.*

Description

This method creates an object of type karp_experimental_design and will immediately initiate a search through $1_T$ space.

Usage

```
initKarpExperimentalDesignObject(X, objective = "abs_sum_diff",
  wait = FALSE, balanced = TRUE, start = TRUE)
```

Arguments

`X`	The design matrix with $n$ rows (one for each subject) and $p$ columns (one for each measurement on the subject). This is the design matrix you wish to search for a more karp design.
`objective`	The objective function to use when greedily searching design space. This is a string "`abs_sum_diff`" (default) or "`mahal_dist`."
`wait`	Should the `R` terminal hang until all `max_designs` vectors are found? The deafult is `FALSE`.
`balanced`	Should the final vector be balanced? Default and recommended is `TRUE`.
`start`	Should we start searching immediately (default is `TRUE`).

Value

An object of type `karp_experimental_design_search` which can be further operated upon

Author(s)

Adam Kapelner

initOptimalExperimentalDesignObject

This method creates an object of type optimal_experimental_design and will immediately initiate a search through $1_T$ space.

Description

This method creates an object of type optimal_experimental_design and will immediately initiate a search through $1_T$ space.

Usage

```
initOptimalExperimentalDesignObject(X, objective = "abs_sum_diff",
  wait = FALSE, start = TRUE, num_cores = 1)
```

Arguments

`X`	The design matrix with $n$ rows (one for each subject) and $p$ columns (one for each measurement on the subject). This is the design matrix you wish to search for a more optimal design.
`objective`	The objective function to use when greedily searching design space. This is a string "`abs_sum_diff`" (default) or "`mahal_dist`."
`wait`	Should the `R` terminal hang until all `max_designs` vectors are found? The deafult is `FALSE`.
`start`	Should we start searching immediately (default is `TRUE`).
`num_cores`	The number of CPU cores you wish to use during the search. The default is `1`.

Value

An object of type `optimal_experimental_design_search` which can be further operated upon

Author(s)

Adam Kapelner

initRerandomizationExperimentalDesignObject

This method creates an object of type rerandomization_experimental_design and will immediately initiate a search through $1_T$ space.

Description

This method creates an object of type rerandomization_experimental_design and will immediately initiate a search through $1_T$ space.

Usage

```
initRerandomizationExperimentalDesignObject(X, max_designs = 1000,
  obj_val_cutoff_to_include = NULL, objective = "mahal_dist",
  wait = FALSE, start = TRUE, num_cores = 1)
```

Arguments

`X`	The design matrix with $n$ rows (one for each subject) and $p$ columns (one for each measurement on the subject). This is the design matrix you wish to search for a more optimal design.
`max_designs`	The maximum number of designs to be returned. Default is 10,000. Make this large so you can search however long you wish as the search can be stopped at any time by using the `stopSearch` method
`obj_val_cutoff_to_include`	Only allocation vectors with objective values lower than this threshold will be returned. The default is `NULL` which means all vectors are returned.
`objective`	The objective function to use when searching the design space. This is a string `"abs_sum_diff"` (default) or `"mahal_dist."`
`wait`	Should the `R` terminal hang until all `max_designs` vectors are found? The default is `FALSE`.
`start`	Should we start searching immediately (default is `TRUE`).
`num_cores`	The number of CPU cores you wish to use during the search. The default is `1`.

Value

An object of type `rerandomization_experimental_design_search` which can be further operated upon.

Author(s)

Adam Kapelner

plot.greedy_experimental_design_search

Plots a summary of a `greedy_experimental_design_search` *object*

Description

Plots a summary of a `greedy_experimental_design_search` object

Usage

```
## S3 method for class 'greedy_experimental_design_search'
plot(x, ...)
```

Arguments

`x`	The `greedy_experimental_design_search` object to be summarized in the plot
`...`	Other parameters to pass to the default plot function

Value

An array of order statistics from plot_obj_val_order_statistic as a list element

Author(s)

Adam Kapelner

`plot_obj_val_by_iter` *Plots the objective value by iteration*

Description

Plots the objective value by iteration

Usage

```
plot_obj_val_by_iter(res, runs = NULL)
```

Arguments

`res`	Results from a greedy search object
`runs`	A vector of run indices you would like to see plotted (default is to plot the first up to 9)

Author(s)

Adam Kapelner

plot_obj_val_order_statistic

Plots an order statistic of the object value as a function of number of searches

Description

Plots an order statistic of the object value as a function of number of searches

Usage

```
plot_obj_val_order_statistic(obj, order_stat = 1, skip_every = 5,
  type = "o", ...)
```

Arguments

obj	The greedy_experimental_design_search object whose search history is to be visualized
order_stat	The order statistic that you wish to plot. The default is 1 for the minimum.
skip_every	Plot every nth point. This makes the plot generate much more quickly. The default is 5.
type	The type parameter for plot.
...	Other arguments to be passed to the plot function.

Value

An array of order statistics as a list element

Author(s)

Adam Kapelner

print.greedy_experimental_design_search

Prints a summary of a greedy_experimental_design_search *object*

Description

Prints a summary of a greedy_experimental_design_search object

Usage

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

Arguments

x	The greedy_experimental_design_search object to be summarized in the console
...	Other parameters to pass to the default print function

Author(s)

Adam Kapelner

---

print.karp_experimental_design_search
Prints a summary of a karp_experimental_design_search *object*

Description

Prints a summary of a karp_experimental_design_search object

Usage

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

Arguments

x	The karp_experimental_design_search object to be summarized in the console
...	Other parameters to pass to the default print function

Author(s)

Adam Kapelner

---

print.optimal_experimental_design_search
Prints a summary of a optimal_experimental_design_search *object*

Description

Prints a summary of a optimal_experimental_design_search object

Usage

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

Arguments

x	The `optimal_experimental_design_search` object to be summarized in the console
...	Other parameters to pass to the default print function

Author(s)

Adam Kapelner

`print.rerandomization_experimental_design_search` *Prints a summary of a* `rerandomization_experimental_design_search` *object*

Description

Prints a summary of a `rerandomization_experimental_design_search` object

Usage

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

Arguments

x	The `rerandomization_experimental_design_search` object to be summarized in the console
...	Other parameters to pass to the default print function

Author(s)

Adam Kapelner

`resultsGreedySearch` *Returns the results (thus far) of the greedy design search*

Description

Returns the results (thus far) of the greedy design search

Usage

```
resultsGreedySearch(obj, max_vectors = 9)
```

Arguments

`obj` The `greedy_experimental_design` object that is currently running the search

`max_vectors` The number of design vectors you wish to return. `NULL` returns all of them. This is not recommended as returning over 1,000 vectors is time-intensive. The default is 9.

Author(s)

Adam Kapelner

Examples

```
 ## Not run:
library(MASS)
data(Boston)
 #pretend the Boston data was an experiment setting
#first pull out the covariates
 X = Boston[, 1 : 13]
 #begin the greedy design search
ged = initGreedyExperimentalDesignObject(X,
max_designs = 1000, num_cores = 2, objective = "abs_sum_diff")
#wait
res = resultsGreedySearch(ged, max_vectors = 2)
design = res$ending_indicTs[, 1] #ordered already by best-->worst
 design
 #what is the balance on this vector?
res$obj_vals[1]
#compute balance explicitly in R to double check
compute_objective_val(X, design) #same as above
#how far have we come?
ged
#we can cut it here
stopSearch(ged)

## End(Not run)
```

---

resultsGurobiNumericalOptimizeExperimentalDesign

Returns the results (thus far) of the Gurobi numerical optimization design search

---

Description

Returns the results (thus far) of the Gurobi numerical optimization design search

Usage

```
resultsGurobiNumericalOptimizeExperimentalDesign(obj)
```

Arguments

`obj` The `gurobi_numerical_optimization_experimental_design_search` object that is currently running the search

Author(s)

Adam Kapelner

`resultsKarpSearch` *Returns the results (thus far) of the karp design search*

Description

Returns the results (thus far) of the karp design search

Usage

```
resultsKarpSearch(obj)
```

Arguments

`obj` The `karp_experimental_design` object that is currently running the search

Author(s)

Adam Kapelner

`resultsOptimalSearch` *Returns the results (thus far) of the optimal design search*

Description

Returns the results (thus far) of the optimal design search

Usage

```
resultsOptimalSearch(obj)
```

Arguments

`obj` The `optimal_experimental_design` object that is currently running the search

Author(s)

Adam Kapelner

resultsRerandomizationSearch *Returns the results (thus far) of the rerandomization design search*

Description

Returns the results (thus far) of the rerandomization design search

Usage

```
resultsRerandomizationSearch(obj, include_assignments = FALSE)
```

Arguments

obj	The `rerandomization_experimental_design` object that is currently running the search
include_assignments	Do we include the assignments (takes time) and default is `FALSE`.

Author(s)

Adam Kapelner

searchTimeElapsed *Returns the number of vectors found by the greedy design search*

Description

Returns the number of vectors found by the greedy design search

Usage

```
searchTimeElapsed(obj)
```

Arguments

obj	The `experimental_design` object that is currently running the search

Author(s)

Adam Kapelner

`startSearch` *Starts the parallelized greedy design search. Once begun, this function cannot be run again.*

Description

Starts the parallelized greedy design search. Once begun, this function cannot be run again.

Usage

```
startSearch(obj)
```

Arguments

`obj`	The `experimental_design` object that will be running the search

Author(s)

Adam Kapelner

`stopSearch` *Stops the parallelized greedy design search. Once stopped, it cannot be restarted.*

Description

Stops the parallelized greedy design search. Once stopped, it cannot be restarted.

Usage

```
stopSearch(obj)
```

Arguments

`obj`	The `experimental_design` object that is currently running the search

Author(s)

Adam Kapelner

summary.greedy_experimental_design_search

Prints a summary of a `greedy_experimental_design_search` *object*

Description

Prints a summary of a `greedy_experimental_design_search` object

Usage

```
## S3 method for class 'greedy_experimental_design_search'
summary(object, ...)
```

Arguments

`object`	The `greedy_experimental_design_search` object to be summarized in the console
`...`	Other parameters to pass to the default summary function

Author(s)

Adam Kapelner

summary.karp_experimental_design_search

Prints a summary of a `karp_experimental_design_search` *object*

Description

Prints a summary of a `karp_experimental_design_search` object

Usage

```
## S3 method for class 'karp_experimental_design_search'
summary(object, ...)
```

Arguments

`object`	The `karp_experimental_design_search` object to be summarized in the console
`...`	Other parameters to pass to the default summary function

Author(s)

Adam Kapelner

summary.optimal_experimental_design_search

Prints a summary of a `optimal_experimental_design_search` *object*

Description

Prints a summary of a `optimal_experimental_design_search` object

Usage

```
## S3 method for class 'optimal_experimental_design_search'
summary(object, ...)
```

Arguments

`object`	The `optimal_experimental_design_search` object to be summarized in the console
`...`	Other parameters to pass to the default summary function

Author(s)

Adam Kapelner

summary.rerandomization_experimental_design_search

Prints a summary of a `rerandomization_experimental_design_search` *object*

Description

Prints a summary of a `rerandomization_experimental_design_search` object

Usage

```
## S3 method for class 'rerandomization_experimental_design_search'
summary(object, ...)
```

Arguments

`object`	The `rerandomization_experimental_design_search` object to be summarized in the console
`...`	Other parameters to pass to the default summary function

Author(s)

Adam Kapelner

Index