Package 'NHEMOtree'

January 27, 2015

Type Package

Title Non-hierarchical evolutionary multi-objective tree learner to perform cost-sensitive classification

Depends partykit, emoa, sets, rpart

Version 1.0

Date 2013-05-05

Author Swaantje Casjens

Maintainer Swaantje Casjens <swaantje.casjens@tu-dortmund.de>

Description NHEMOtree performs cost-sensitive classification by solving the two-objective optimization problem of minimizing misclassification rate and minimizing total costs for classification. The three methods comprised in NHEMOtree are based on EMOAs with either tree representation or bitstring representation with an enclosed classification tree algorithm.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2013-05-09 15:46:02

R topics documented:

1

NHEMO   *Non-hierarchical evolutionary multi-objective tree learner to perform*
                               cost-sensitive classification

Description

NHEMO performs cost-sensitive classification by solving the non-hierarchical evolutionary two-objective optimization problem of minimizing misclassification rate and minimizing total costs for classification. NHEMO is based on an EMOA with tree representation and cutoff optimization conducted by the EMOA.

Usage

```
NHEMO(data, CostMatrix,
      gens = 50, popsize = 50, max_nodes = 10,
      ngens = 14, bound = 10^-10,
      init_prob = 0.8,
      ps = c("tournament", "roulette", "winkler"), tournament_size = 4,
      crossover = c("standard", "brood", "poli"), brood_size=4,
      crossover_prob=0.5,  mutation_prob=0.5,
      CV=5, vim=0, ...)
```

Arguments

data	A data frame containing in the first column the class for each observation and in the other columns from which variables specified in formula are preferentially to be taken.
CostMatrix	A data frame containing the names (first column) and the costs (second column) for each explanatory variable in formula or x. NHEMOtree does not work with missing data in CostMatrix.
gens	Maximal number of generations of the EMOA (default: gens=50).
popsize	Population size in each generation of the EMOA (default: popsize=50).
max_nodes	Maximal number of nodes within each tree (default: max_nodes=10).
ngens	Preceeding generations for the Online Convergence Detection (OCD, default: ngens=14) (see below for details).
bound	Variance limit for the Online convergence detection (default: bound=10^10).
init_prob	Degree of initilization $\in [0, 1]$, i.e. the probability of a node having a subnode.
ps	Type of parent selection, "tournament" for tournament selection (default), "roulette" for roulette-wheel selection, and "winkler" for random selection of the first parent and the second parent by roulette-wheel selection.
tournament_size	Size of tournament for ps="tournament" (default: tournament_size=4).

crossover	Crossover operator, "standard" for one point crossover swapping two randomly chosen subtrees of the parents, "brood" for brood crossover (for details see Tackett (1994)), "poli" for a size-dependent crossover with the crossover point from the so called common region of both parents (for details see Poli and Langdon (1998)).
brood_size	Number of offspring created by brood crossover (default: brood_size=4).
crossover_prob	Probability to perform crossover $\in [0, 1]$ (default: crossover_prob=0.50).
mutation_prob	Probability to perform mutation $\in [0, 1]$ (default: mutation_prob=0.50).
CV	Cross validation steps as natural number bigger than 1 (default: CV=5).
vim	Variable importance measure to be used to improve standard crossover. vim=0 for no variable improtance measure (default), vim=1 for 'simple absolute frequency', vim=2 for 'simple relative frequency', vim=3 for 'relative frequency', vim=4 for 'linear weigthed relative frequency', vim=5 for 'exponential weigthed relative frequency', and vim=6 for 'permutation accuracy importance'.
...	Arguments passed to or from other methods.

Details

With NHEMO a cost-sensitive classification can be performed, i.e. building a tree learner by solving a two-objective optimization problem with regard to minimizing misclassification rate and minimizing total costs for classification (sum of costs for all used variables in the classifier). The non-hierarchical evolutionary multi-objective tree learner performs this multi-objective optimization based on an EMOA with tree representation. It optimizes both objectives simultaneously without any hierarchy and generates Pareto-optimal classifiers being trees to solve the problem. Cutoffs of the tree learners are optimized with the EMOA.

Termination criteria are the maximal amount of generations and the Online Convergence Detection (OCD) proposed by Wagner and Trautmann (2010). Here, OCD uses the dominated hypervolume as quality criterion. If its variance over the last g generations is significantly below a given threshold L according to the one-sided χ^2-variance test OCD stops the run. We followed the suggestion of Wagner and Trautmann (2010) and considered their parameter settings as default values.

Missing data in the grouping variable or the explanatory variables are excluded from the analysis automatically.

NHEMO does not work with missing data in CostMatrix. Costs of all explanatory variables set to 1 results in optimizing the amount of explanatory variables in the tree learner as second objective.

Author(s)

Swaantje Casjens

References

R. Poli and W.B. Langdon. Schema theory for genetic programming with one-point crossover and point mutation. Evolutionary Computation, 6(3):231-252, 1998a.

W.A. Tackett. Recombination, selection und the genetic construction of computer programs. PhD thesis, University of Southern California, 1994.

T. Wagner and H. Trautmann. Online convergence detection for evolutionary multiobjective algorithms revisited. In: IEEE Congress on Evolutionary Computation, 1-8, 2010.

See Also

[NHEMOtree](#)

Examples

```
# Simulation of data and costs
  d         <- Sim_Data(Obs=200)
  CostMatrix<- Sim_Costs()

# NHEMO calculations with function NHEMOtree and type="NHEMO"
  res<- NHEMOtree(method="NHEMO", formula=Y2~., data=d, CostMatrix=CostMatrix,
                  gens=5, popsize=10,
                  max_nodes=5, ngens=5, bound=10^-10, init_prob=0.8,
                  ps="tournament", tournament_size=4, crossover="standard",
                  crossover_prob=0.1, mutation_prob=0.1,
                  CV=5, vim=1)
  res
```

NHEMOtree	*Non-hierarchical evolutionary multi-objective tree learner to perform cost-sensitive classification*

Description

NHEMOtree performs cost-sensitive classification by solving the two-objective optimization problem of minimizing misclassification rate and minimizing total costs for classification. The three methods comprised in NHEMOtree ("NHEMO", "NHEMO_Cutoff", "Wrapper") are based on EMOAs with either tree representation or bitstring representation with a classification tree as enclosed classifier.

Usage

```
NHEMOtree(method = c("NHEMO", "NHEMO_Cutoff", "Wrapper"),
          formula, data, x, grouping,
          CostMatrix, ...)
```

Arguments

method	Optimization method, "NHEMO" for standard non-hierarchical evolutionary multi-objective optimization with cutoff optimization by the EMOA, "NHEMO_Cutoff" for NHEMO with local cutoff optimization, "Wrapper" for the wrapper approach (see below for details).
formula	A formula of the form groups ~ x1 + x2 + ... The response is the grouping factor and the right hand side specifies the explanatory variables.
data	Data frame from which variables specified in formula are preferentially to be taken.

x   (Required if no formula is given as the principal argument.) A matrix or data
                      frame containing the explanatory variables.

grouping   (Required if no formula is given as the principal argument.) A factor specifying
                      the class for each observation.

CostMatrix   A data frame containing the names (first column) and the costs (second column)
                      for each explanatory variable in formula or x. NHEMOtree does not work with
                      missing data in CostMatrix.

...   Arguments passed to or from other methods.

Details

With NHEMOtree three types of cost-sensitive classification can be performed, i.e. building a tree
learner by solving a two-objective optimization problem with regard to minimizing misclassification
rate and minimizing total costs for classification (summarized costs for all used variables in the
classifier).

First, the non-hierarchical evolutionary multi-objective tree learner (NHEMOtree) performs this
multi-objective optimization based on an EMOA with tree representation (method="NHEMO").
It optimizes both objectives simultaneously without any hierarchy and generates Pareto-optimal
classifiers being binary trees to solve the problem. Cutoffs of the tree learners are optimized with
the EMOA.

Second, NHEMOtree with local cutoff optimization works like NHEMOtree but the cutoffs of the
tree learner are optimized analogous to a classification tree with recursive partitioning either based
on the Gini index or the misclassification rate (method="NHEMO_Cutoff").

Third, a wrapper approach based on NSGA-II with enclosed classification tree algorithm can be
chosen to solve the multi-objective optimization problem (method="Wrapper"). The classification
trees are built with rpart rpart. However, wrapper approaches suffer from a hierarchy in the objec-
tives at which misclassification is minimized at first followed by optimizing costs.

Termination criteria of the EMOAs are the maximal amount of generations and the Online Conver-
gence Detection (OCD) proposed by Wagner and Trautmann (2010). Here, OCD uses the dominated
hypervolume as quality criterion. If its variance over the last g generations is significantly below
a given threshold L according to the one-sided χ^2-variance test OCD stops the run. We followed
the suggestion of Wagner and Trautmann (2010) and considered their parameter settings as default
values.

Missing data in the grouping variable or the explanatory variables are excluded from the analysis
automatically.

NHEMOtree does not work with missing data in "CostMatrix". Costs of all explanatory variables
set to 1 results in optimizing the amount of explanatory variables in the tree learner as second
objective.

Value

Values returned by NHEMOtree are

S_Metric   S metric of the final population

Misclassification_total
                      Range of the misclassification rate of the final population

Costs_total	Range of the costs of the final population
VIMS	Variable improtance measures of all explanatory variables for method=c("NHEMO", "NHEMO_Cutoff")
Trees	Individuals of the final population
Best_Tree	Individual of the final population with the lowest misclassification rate, its misclassification rate and costs for method=c("Wrapper")
S_Metrik_temp	S metric values of each generation
method	Method run with NHEMOtree

Author(s)

Swaantje Casjens

References

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees. Wadsworth, 1984.

K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation, 6(2):182-197, 2002.

R. Poli and W.B. Langdon. Schema theory for genetic programming with one-point crossover and point mutation. Evolutionary Computation, 6(3):231-252, 1998a.

W.A. Tackett. Recombination, selection und the genetic construction of computer programs. PhD thesis, University of Southern California, 1994.

T. Wagner and H. Trautmann. Online convergence detection for evolutionary multiobjective algorithms revisited. In: IEEE Congress on Evolutionary Computation, 1-8, 2010.

See Also

plot.NHEMOtree, NHEMO, NHEMO_Cutoff, Wrapper

Examples

```
# Simulation of data and costs
  d         <- Sim_Data(Obs=200)
  CostMatrix<- Sim_Costs()

# NHEMOtree calculations
  res<- NHEMOtree(method="NHEMO", formula=Y2~., data=d, CostMatrix=CostMatrix,
                  gens=5, popsize=5, crossover_prob=0.1, mutation_prob=0.1)
  res
```

NHEMO_Cutoff	*Non-hierarchical evolutionary multi-objective optimization with local cutoff optimization*

Description

NHEMO_Cutoff performs cost-sensitive classification by solving the non-hierarchical evolutionary two-objective optimization problem of minimizing misclassification rate and minimizing total costs for classification. NHEMO_Cutoff is based on an EMOA with tree representation and local cutoff optimization. Cutoffs of the tree learner are optimized analogous to a classification tree with recursive partitioning either based on the Gini index or the misclassification rate.

Usage

```
NHEMO_Cutoff(data, CostMatrix,
             gens = 50, popsize = 50, max_nodes = 10,
             ngens = 14, bound = 10^-10,
             init_prob = 0.8,
             ps = c("tournament", "roulette", "winkler"), tournament_size = 4,
             crossover = c("standard", "brood", "poli"), brood_size = 4,
             crossover_prob = 0.5, mutation_prob = 0.5,
             CV = 5, vim = 0,
             ncutoffs = 10, opt = c("gini", "mcr"))
```

Arguments

data	A data frame containing in the first column the class for each observation and in the other columns from which variables specified in formula are preferentially to be taken.
CostMatrix	A data frame containing the names (first column) and the costs (second column) for each explanatory variable in formula or x. NHEMOtree does not work with missing data in CostMatrix.
gens	Maximal number of generations of the EMOA (default: gens=50).
popsize	Population size in each generation of the EMOA (default: popsize=50).
max_nodes	Maximal number of nodes within each tree (default: max_nodes=10).
ngens	Preceeding generations for the Online Convergence Detection (OCD, default: ngens=14) (see below for details).
bound	Variance limit for the Online convergence detection (default: bound=10^10).
init_prob	Degree of initilization $\in [0,1]$, i.e. the probability of a node having a subnode (default: init_prob=0.80).
ps	Type of parent selection, "tournament" for tournament selection (default), "roulette" for roulette-wheel selection, and "winkler" for random selection of the first parent and the second parent by roulette-wheel selection.
tournament_size	
	Size of tournament for ps="tournament" (default: tournament_size=4).

crossover	Crossover operator, "standard" for one point crossover swapping two randomly chosen subtrees of the parents, "brood" for brood crossover (for details see Tackett (1994)), "poli" for a size-dependent crossover with the crossover point from the so called common region of both parents (for details see Poli and Langdon (1998)).
brood_size	Number of offspring created by brood crossover (default: brood_size=4).
crossover_prob	Probability to perform crossover $\in [0, 1]$ (default: crossover_prob=0.50).
mutation_prob	Probability to perform mutation $\in [0, 1]$ (default: mutation_prob=0.50).
CV	Cross validation steps as natural number bigger than 1 (default: CV=5).
vim	Variable importance measure to be used to improve standard crossover. vim=0 for no variable improtance measure (default), vim=1 for 'simple absolute frequency', vim=2 for 'simple relative frequency', vim=3 for 'relative frequency', vim=4 for 'linear weigthed relative frequency', vim=5 for 'exponential weigthed relative frequency', and vim=6 for 'permutation accuracy importance'.
ncutoffs	Number of cutoffs per explanatory variable to be tested for optimality (default: ncutoffs=10).
opt	Type of local cutoff optimization, "gini" for local cutoff optimization by Gini index (default), "mcr" for local cutoff optimization by misclassification rate.

Details

The non-hierarchical evolutionary multi-objective tree learner (NHEMOtree) with local cutoff optimization solves a two-objective optimization problem with regard to minimizing misclassification rate and minimizing total costs for classification (summarized costs for all used variables in the classifier). It is based on an EMOA with tree representation. It optimizes both objectives simultaneously without any hierarchy and generates Pareto-optimal classifiers being binary trees to solve the problem. Cutoffs of the tree learner are optimized analogous to a classification tree with recursive partitioning either based on the Gini index or the misclassification rate.

Termination criteria of NHEMO_Cutoff are the maximal amount of generations and the Online Convergence Detection (OCD) proposed by Wagner and Trautmann (2010). Here, OCD uses the dominated hypervolume as quality criterion. If its variance over the last g generations is significantly below a given threshold L according to the one-sided χ^2-variance test OCD stops the run. We followed the suggestion of Wagner and Trautmann (2010) and considered their parameter settings as default values.

Missing data in the grouping variable or the explanatory variables are excluded from the analysis automatically. NHEMO_Cutoff does not work with missing data in "CostMatrix". Costs of all explanatory variables set to 1 results in optimizing the amount of explanatory variables in the tree learner as second objective.

Author(s)

Swaantje Casjens

References

R. Poli and W.B. Langdon. Schema theory for genetic programming with one-point crossover and point mutation. Evolutionary Computation, 6(3):231-252, 1998a.

W.A. Tackett. Recombination, selection und the genetic construction of computer programs. PhD thesis, University of Southern California, 1994.

T. Wagner and H. Trautmann. Online convergence detection for evolutionary multiobjective algorithms revisited. In: IEEE Congress on Evolutionary Computation, 1-8, 2010.

See Also

NHEMOtree

Examples

```
# Simulation of data and costs
  d         <- Sim_Data(Obs=200)
  CostMatrix<- Sim_Costs()

# NHEMO_Cutoff calculations with function NHEMOtree and type="NHEMO_Cutoff"
  res<- NHEMOtree(method="NHEMO_Cutoff", formula=Y2~., data=d, CostMatrix=CostMatrix,
                  gens=5, popsize=5,
                  max_nodes=5, ngens=5, bound=10^-10, init_prob=0.8,
                  ps="tournament", tournament_size=4, crossover="standard",
                  crossover_prob=0.1, mutation_prob=0.1,
                  CV=5, vim=1,
                  ncutoffs=5, opt="mcr")
  res
```

---

plot.NHEMOtree	*Plot Method for Class NHEMOtree*

Description

Generates different plots for class NHEMOtree

Usage

```
## S3 method for class 'NHEMOtree'
plot(x, data, which = c("Pareto", "S_Metric", "VIMS", "Tree"), vim = 0, ...)
```

Arguments

x	An object of class NHEMOtree.
data	Data which was used in NHEMOtree.
which	Type of plot, "Pareto" for the final Pareto front, "S_Metric" for increase of the dominated hypervolume by generation, "VIMS" for variable improtance for each explanatory variable, "Tree" for the individual with the lowest misclassification rate.

vim   Variable improtance measure to be plottet for which="VIMS", vim=1 for 'simple absolute frequency', vim=2 for 'simple relative frequency', vim=3 for 'relative frequency', vim=4 for 'linear weigthed relative frequency', vim=5 for 'exponential weigthed relative frequency', vim=6 for 'permutation accuracy importance' (default: plots for all VIMs).

...   Arguments passed to or from other methods.

Author(s)

Swaantje Casjens

See Also

[NHEMOtree](#)

Examples

```
# Simulation of data and costs
  d          <- Sim_Data(Obs=200)
  CostMatrix<- Sim_Costs()

  res<- NHEMOtree(type="NHEMO", formula=Y2~., data=d, CostMatrix=CostMatrix,
                  gens=5, popsize=5, crossover_prob=0.1, mutation_prob=0.1)
  plot(data=d, x=res, which="P")
  plot(data=d, x=res, which="S", col=3, type="l")
  plot(data=d, x=res, which="V", vim=5)
  plot(data=d, x=res, which="T")
```

Sim_Costs   *Simulation of a cost matrix for use in NHEMOtree*

Description

Simulation of a cost matrix for 10 variables in combination with 'Sim_Data' and to be analyszed with "NHEMOtree".

Usage

```
Sim_Costs()
```

Author(s)

Swaantje Casjens

See Also

[Sim_Data](#), [NHEMOtree](#)

Examples

```
Sim_Costs()
```

Sim_Data   *Simulation of data for use in NHEMOtree*

Description

Simulation of data with one grouping variable containing four classes and 20 explanatory variables. Variables X1 to X3 are informative for seperating the four classes. Variable X1 separates class 1, X2 separates class 1 and class 2, and X3 separates class 3 from class 4. Variables X4, X5, and X6 are created on basis of X3 and can also be used to separate class 3 from class 4 but with decreased prediction accuracy.

Usage

```
Sim_Data(Obs, VG=1, VP1=0.05, VP2=0.1, VP3=0.3)
```

Arguments

Obs	Amount of observations.
VG	Overall accuracy for data separation $\in [0, 1]$ with VG=1 (default) for perfect seperation.
VP1	Decrease of prediction accuracy for variable X4 in comparison with X3 to separate class 3 from class 4 (default: VP1=0.05).
VP2	Decrease of prediction accuracy for variable X5 in comparison with X3 to separate class 3 from class 4 (default: VP2=0.1).
VP3	Decrease of prediction accuracy for variable X6 in comparison with X3 to separate class 3 from class 4 (default: VP3=0.3).

Details

With this function data with one grouping variable containing four classes and 20 explanatory variables X1 to X10 is simulated.

Variable X1 separates class 1, X2 separates class 1 and class 2, and X3 separates class 3 from class 4. For all samples belonging to the according classes the explanatory variables X1 to X3 are drawn from a normal distribution with $\mu = 80$ and $\sigma^2 = 25$. Samples which are not allocated to the corresponding class are drawn from a uniform distribution with minimum 0 and an adjustable maximum value. The maximum values of the uniform distributions are the smallest drawn random values of each variable.

Variables X4, X5, and X6 are created on basis of X3 and separate class 3 from class 4, too. However, the prediction accuracy of these variables decreases gradually. The decrease is assigned by 'VP1', 'VP2', and 'VP3'. Thus, the according amount of the discriminating samples of former class 3 are disturbed by assigning a value drawn from a uniform distribution. Accordingly, X4, X5 and X6

discriminate class 3 worse than X3. X7 to X10 are noisy variables drawn from a normal distribution that contain no information.

Noise is added to the class assignment by a binomial distribution. Each potential class is only with probability "VG" the equivalent class and with probability 1-"VG" one of the other classes.

Variable costs correlate with their prediction accuracy so that variables containing more information are more expensive than variables with less or none information. The costs of the variables are generated with function "Sim_Costs".

Author(s)

Swaantje Casjens

See Also

[Sim_Costs](), [NHEMOtree]()

Examples

```
d<- Sim_Data(Obs=200)
head(d)
```

Wrapper   *NSGA-II wrapper with enclosed classification tree algorithm*

Description

This wrapper approach is based on the Nondominated Sorting Genetic Algorithm II (NSGA-II) with an enclosed classification tree algorithm. It performs cost-sensitive classification by solving the two-objective optimization problem of minimizing misclassification rate and minimizing total costs for classification.

Usage

```
Wrapper(data, CostMatrix,
        gens = 50, popsize = 50,
        ngens = 14, bound = 10^-10,
        init_prob = 0.8,
        crossover_prob = 0.5, mutation_prob = 0.5,
        CV = 5, ...)
```

Arguments

data	A data frame containing in the first column the class for each observation and in the other columns from which variables specified in formula are preferentially to be taken.
CostMatrix	A data frame containing the names (first column) and the costs (second column) for each explanatory variable in formula or x. NHEMOtree does not work with missing data in CostMatrix.

gens	Maximal number of generations of the EMOA (default: gens=50).
popsize	Population size in each generation of the EMOA (default: popsize=50).
ngens	Preceeding generations for the Online Convergence Detection (OCD, default: ngens=14) (see below for details).
bound	Variance limit for the Online convergence detection (default: bound=10^10).
init_prob	Degree of initilization $\in [0, 1]$, i.e. the amount of activated explanatory variables in the individuals of the initial population (default: init_prob=0.80).
crossover_prob	Probability to perform crossover $\in [0, 1]$ (default: crossover_prob=0.50).
mutation_prob	Probability to perform mutation $\in [0, 1]$ (default: mutation_prob=0.50).
CV	Cross validation steps as natural number bigger than 1 (default: CV=5).
...	Arguments passed to or from other methods.

Details

With Wrapper a wrapper approach based on the Nondominated Sorting Genetic Algorithm II (NSGA-II) with an enclosed classification tree algorithm is used to solve the two-objective optimization problem of minimizing misclassification rate and minimizing total costs for classification (summarized costs for all used variables in the classifier). The classification trees are constructed with rpart rpart. However, wrapper approaches suffer from a hierarchy in the objectives at which misclassification is minimized at first followed by optimizing costs. Parent selection is always performed by a binary tournament and there is only one-point crossover.

Alternatives are the proposed non-hierarchical evolutionary multi-objective tree learners which do not suffer from a hierarchy in the objectives (see NHEMO NHEMOtree and NHEMO_Cutoff NHEMOtree).

Termination criteria of the NSGA-II are the maximal amount of generations and the Online Convergence Detection (OCD) proposed by Wagner and Trautmann (2010). Here, OCD uses the dominated hypervolume as quality criterion. If its variance over the last g generations is significantly below a given threshold L according to the one-sided χ^2-variance test OCD stops the run. We followed the suggestion of Wagner and Trautmann (2010) and considered their parameter settings as default values.

Missing data in the grouping variable or the explanatory variables are excluded from the analysis automatically. Wrapper does not work with missing data in "CostMatrix". Costs of all explanatory variables set to 1 results in optimizing the amount of explanatory variables in the tree learner as second objective.

Author(s)

Swaantje Casjens

References

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees. Wadsworth, 1984.

K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation, 6(2):182-197, 2002.

R. Poli and W.B. Langdon. Schema theory for genetic programming with one-point crossover and point mutation. Evolutionary Computation, 6(3):231-252, 1998a.

W.A. Tackett. Recombination, selection und the genetic construction of computer programs. PhD thesis, University of Southern California, 1994.

T. Wagner and H. Trautmann. Online convergence detection for evolutionary multiobjective algorithms revisited. In: IEEE Congress on Evolutionary Computation, 1-8, 2010.

See Also

[NHEMOtree](#)

Examples

```
# Simulation of data and costs
  d         <- Sim_Data(Obs=200)
  CostMatrix<- Sim_Costs()

# Wrapper calculations with function NHEMOtree and type="Wrapper"
  res<- NHEMOtree(method="Wrapper", formula=Y2~., data=d, CostMatrix=CostMatrix,
                  gens=5, popsize=5,
                  init_prob=0.8,
                  ngens=14, bound=10^-10,
                  crossover_prob=0.1, mutation_prob=0.1,
                  CV=5)
  res
```

Index