Package 'emma'

February 19, 2015

Type Package

Title Evolutionary model-based multiresponse approach

Version 0.1-0

Date 2011-10-22

Author Laura Villanova <laura.villanova@monash.edu>, Kate Smith-Miles <kate.smith-miles@monash.edu> and Rob J Hyndman <rob.hyndman@monash.edu>.

Maintainer Laura Villanova <laura.villanova@monash.edu>

Depends R (>= 2.9.2), earth, clusterSim

Imports methods

Description The evolutionary model-based multiresponse approach (EMMA) is a novel methodology to process optimisation and product improvement. The approach is suitable to contexts in which the experimental cost and/or time limit the number of implementable trials.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2011-10-26 17:59:36

NeedsCompilation no

R topics documented:

emma-package 2
ackley 3
distance 4
emma 5
emmacheck 9
emmat0 12
emmatn 14
peaks 19
plot.emma 20

Index 21

emma-package *Designing experiments for process optimization*

Description

The evolutionary model-based multiresponse approach (EMMA) is a procedure for process optimization and product improvement. It is particularly suited to processes featuring irregular experimental region due to constraints on the input variables (factors), multiple responses not accomodated by polynomial models, and expensive or time-consuming experiments. EMMA iterativelly selects new experimental points that increasingly concentrate on the most promising regions of the experimental space. The selection of the new experimental points is performed on the basis of the results achieved during previous trials. A multivariate adaptive regression splines (MARS) model and a particle swarm optimization (PSO) algorithm are used to drive the search of the optimum.

Details

Package:	emma
Type:	Package
Version:	1.0
Date:	2011-02-22
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

Maintainer: Laura Villanova <laura.villanova@monash.edu>

References

Villanova L., Falcaro P., Carta D., Poli I., Hyndman R., Smith-Miles K. (2010) 'Functionalization of Microarray Devices: Process Optimization Using a Multiobjective PSO and Multiresponse MARS Modelling', IEEE CEC 2010, DOI: 10.1109/CEC.2010.5586165

Carta D., Villanova L., Costacurta S., Patelli A., Poli I., Vezzu' S., Scopece P., Lisi F., Smith-Miles K., Hyndman R. J., Hill A. J., Falcaro P. (2011) 'Method for Optimizing Coating Properties Based on an Evolutionary Algorithm Approach', Analytical Chemistry 83 (16), 6373-6380.

Examples

```
in.name <- c("x1","x2")
nlev <- c(20, 20)
```

```
lower <- c(-3, -3)
upper <- c(3, 3)
out.name <- "y"
weight <- 1
C <- 20
pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

emma.peaks<-emma(in.name, nlev, lower, upper, out.name, opt = "mx", nd = 10,
na = 5, weight, C , w1 = 0.7, w2 = 0.4, c1i = 2.5, c1f = 0.5,
c2i = 0.5, c2f = 2.5, b = 5, pr.mut, graph = "no", fn1 = peaks)

plot(emma.peaks, fn = peaks, n = 50, C = 20)
```

---

ackley *Ackley test problem*

Description

Generates the Ackley benchmark function. The Ackley function is a commonly used test problem for global optimization procedures.

Usage

```
ackley(x)
```

Arguments

x	A matrix containing the values of the input variables.

Value

Vector of the same length as x giving the values of the Ackley function.

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

References

http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page295.htm

Examples

```
x1 <- seq(-2, 2, length = 50)
x2 <- x1

x <- expand.grid(x1, x2)

z <- matrix(ackley(x),nrow = length(x1))

nrz <- nrow(z)
ncz <- ncol(z)
jet.colors <- colorRampPalette( c("red", "yellow") )
nbcol <- 100
color <- jet.colors(nbcol)
zfacet <- z[-1, -1] + z[-1, -ncz] + z[-nrz, -1] + z[-nrz, -ncz]
facetcol <- cut(zfacet, nbcol)

persp(x1, x2, z, col = color[facetcol], theta = 0, phi = 10,
expand = 1, xlab = "x1", ylab = "x2", zlab = "f(x1, x2)",
ticktype = "detailed")
```

---

distance *Distance measure from the target*

Description

Computes a scalar distance between the target (a set of desirable values for the responses) and the responses values that have been either observed or estimated for each point in the experimental space. Such a distance is used to identify additional experimental points to be investigated.

Usage

```
distance(xpop, xspace, yspace, weight, opt)
```

Arguments

xpop	A data frame containing the factor values for the experimental points investigated; the row names uniquely identify each experimental point (ID).
xspace	A data frame containing the factor values for the experimental points defining the entire experimental region; the row names uniquely identify each experimental point (ID).
yspace	A data frame containing the response values (either observed or estimated) for the points in the experimental region.
weight	A numerical vector, of the same length as the number of responses, containing the weights assigned to the each response; the sum of the weights must be equal to 1.
opt	A character vector, of the same length as the number of responses, defining if each response needs to be minimized or maximized. The allowed values are 'mn' (minimize) and 'mx' (maximize).

Details

The function normalizes the response values with respect to the estimated limits of the response space, so that the response values lie between 0 and 1. Subsequently, the function identifies the target and computes a scalar distance between the target and the response values.

Value

`fit`	The scalar distances between the target and the response(s) values for the experimental points in `xpop`.
`obj.nn`	Scalar distance from the target for the best experimental point identified by EMMA.

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

References

Friedman J. H. (1991) 'Multivariate adaptive regression splines' (with discussion), The Annals of Statistics 19, 1:141.

Villanova L., Falcaro P., Carta D., Poli I., Hyndman R., Smith-Miles K. (2010) 'Functionalization of Microarray Devices: Process Optimization Using a Multiobjective PSO and Multiresponse MARS Modelling', IEEE CEC 2010, DOI: 10.1109/CEC.2010.5586165

`emma` *Evolutionary Model-based Multiresponse Approach*

Description

EMMA designs the experiments using a procedure based on the Particle Swarm Optimization (PSO) algorithm. Firstly, EMMA selects a set of initial experimental points (see `emmat0`) that define the initial position of the particles; subsequently, for a given number of iterations, the particles are moved and new experimental points are selected (see `emmatn`).

Usage

```
emma(in.name, nlev, lower, upper, out.name, opt="mn", nd=10, na=5,
weight, C = 20, w1 = 0.7, w2 = 0.4, c1i = 2.5, c1f = 0.5,
c2i = 0.5, c2f = 2.5, b = 5, pr.mut, graph, fn1 = NULL,
fn2 = NULL, fn3 = NULL, fn4 = NULL, nresp)
```

Arguments

in.name	A vector containing the names of the input variables (factors).
nlev	A numeric vector of the same length as in.name, containing the number of factor levels.
lower	A numeric vector of the same length as in.name, containing the lower values of the factors.
upper	A numeric vector of the same length as in.name, containing the upper values of the factors.
out.name	A vector containing the name(s) of the output/response variable(s).
opt	A character vector of the same length as the number of responses, indicating for each response function, if the response must be minimized ('mn') or maximized ('mx').
nd	Number of experimental points to be selected when t = 0.
na	A numeric value indicating the number of experimental points to be selected when t > 0.
weight	A numerical vector of the same length as the number of responses, reflecting the relevance of each response. Use weight = 1 if only one response is investigated; if multiple responses are investigated, the sum of the values in weight must be 1.
C	A numeric value indicating the maximum number of iterations.
w1	The first numeric value used to calculate the inertia weight parameter of the time variant PSO algorithm; the default is w1 = 0.7.
w2	The second numeric value used to calculate the inertia weight parameter of the time variant PSO algorithm; The default is w2 = 0.4.
c1i	The first numeric value used to calculate the acceleration coefficient c1 of the time variant PSO algorithm; the default is c1i = 2.5.
c1f	The second numeric value used to calculate the acceleration coefficient c1 of the time variant PSO algorithm; the default is c1f = 0.5.
c2i	The first numeric value used to calculate the acceleration coefficient c2 of the time variant PSO algorithm; the default is c2i = 0.5.
c2f	The second numeric value used to calculate the acceleration coefficient c2 of the time variant PSO algorithm; the default is c2f = 2.5.
b	A numeric value, used in the mutation operator, that determines the degree of dependence of the mutation on the iteration number; the default is b = 5.
pr.mut	A numeric vector of the same length as the number of iterations C containing the probability of mutation for each time instant.
graph	Logical; if 'yes', a plot of the MARS model is produced. A plot is produced only if the model contains more than one explanatory variable.
fn1	The first function to be optimised. Use fn1 = NULL if the function is unknown (e.g. when designing experiments in applied problems).
fn2	The second function to be optimised. Use fn2 = NULL if the function is unknown (e.g. when designing experiments in applied problems).

fn3	The third function to be optimised. Use fn3 = NULL if the function is unknown (e.g. when designing experiments in applied problems).
fn4	The forth function to be optimised. Use fn4 = NULL if the function is unknown (e.g. when designing experiments in applied problems).
nresp	The response to be plotted. Use nresp = 1 to plot the first response...

Details

To select the new experimental points to be investigated, the following steps are iterated. A MARS model is fitted to the collected data so that an approximated function is obtained for each response; these approximated functions are used to predict the response values at the non-investigated experimental points. Each point in the experimental region E (xspace) is now associated with a vector of response values that has been either measured or estimated. The best (measured or estimated) value of each response is selected and used to identify the target. Subsequently, for each experimental point in E, the scalar distance between the response values and the target is computed and the solution that is closest to the target is selected. If such solution has not been tested yet (see emmacheck), the experiment needs to be performed and its response values are measured. The target is then updated, as well as the scalar distances of all the experimental points from the target. The scalar distances are used to identify the good performing experimental points. The experimental point whose response values are closest to the target is referred to as the global best. Similarly, a personal best is identified for each particle by considering the experimental points visited by that particle and selecting that point featuring the response values that are closest to the target. Finally, the particles velocity and position are updated and a new set of experimental points is identified.

The parameters w1 and w2 are used to calculate the inertia weight w of the PSO algorithm, namely the parameter that controls the influence of the previous particle velocity on the present velocity. High values of w favour a global search, whereas lower values of w encourage a local search. In EMMA the inertia weight is allowed to decrease linearly with iteration from w1 to w2 thus favouring the exploration initially and the exploitation subsequently. The parameters c1i and c1f are used to calculate the cognitive acceleration coefficient c1 of the PSO algorithm, whereas the parameters c2i and c2f are used to calculate the social acceleration coefficient c2 of the PSO algorithm.Higher values of c1 ensure larger deviation of the particle in the search space (exploration), while higher values of c2 signify the convergence to the current global best (exploitation). In EMMA c1 is allowed to decrease from c1i to c1f and c2 is allowed to increase from c2i to c2f. See Tripathi et al. (2007) for more details.

Value

An object of class emma with the components listed below:

xpop	Experimental points investigated.
ypop	Response values observed at the experimental points investigated.
xspace	Experimental region. It is given by all the possible combinations of the factors' levels and contains xpop. The rownames uniquely identify the experimental points and are reported also in xpop.
yspace	Response values that have been either observed or predicted. Observed response values are stored also in ypop. Predicted response values are obtained using a MARS model fitted to the available data.

opt	Indicates if each single function is either minimized ('mn') or maximized ('mx').
nd	Number of experimental points selected initially (t=0).
na	Number of experimental points selected in subsequent iterations (t>0).
tested	IDs of the tested experimental points.
time	Current time instant of the EMMA procedure.
weight	Relative importance of each response. If only one response is investigated, then weight = 1; if multiple responses are investigated, the sum of the values in weight must be 1.
Gb	ID of the best experimental point investigated (global best). Use xspace[Gb,] to visualise the global best and use yspace[Gb,] to visualise its measured response value(s). Gb identifies the experimental point whose response values are closest to the target; the target is a set of desirable response values which are automatically selected on the basis of the measured and predicted response values.
Pb	ID of the best experimental point investigated by each particle (personal best). Use xspace[Pb,] to identify the personal bests and use yspace[Pb,] to visualise their measured response values. Among the experimental points associated to one particle, the Pb identifies the experimental point that is whose response values are closest to the target.
Gb.arch	Archive of the global bests identified. Because the global best changes as new experimental points are investigated, an archive is maintained.
Pb.arch	Archive of the personal bests identified. Because the personal bests change as new experimental points are investigated, an archive is maintained.
v	Velocities used to update the particles position. The position of a particle is uniquely determined by the predictors' values; it also defines the experiment to be performed. At each step of EMMA, the position of a particle is updated by adding a numerical value (velocity) to the current value of each single predictor.
sam.x	IDs of the experiments that have been selected in the current iteration of the procedure. Use xspace[sam.x,] to visualise the experiments to be performed.
add	Logical. If '0' indicates that an additional experimental point needs to be investigated; if '1' indicates that an additional experimental point is not required.

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

References

Villanova L., Falcaro P., Carta D., Poli I., Hyndman R., Smith-Miles K. (2010) 'Functionalization of Microarray Devices: Process Optimization Using a Multiobjective PSO and Multiresponse MARS Modelling', IEEE CEC 2010, DOI: 10.1109/CEC.2010.5586165

Carta D., Villanova L., Costacurta S., Patelli A., Poli I., Vezzu' S., Scopece P., Lisi F., Smith-Miles K., Hyndman R. J., Hill A. J., Falcaro P. (2011) 'Method for Optimizing Coating Properties Based on an Evolutionary Algorithm Approach', Analytical Chemistry 83 (16), 6373-6380.

Friedman J. H. (1991) 'Multivariate adaptive regression splines' (with discussion), The Annals of Statistics 19, 1:141.

Tripathi P. K., Bandyopadhyay S., Pal S. K. (2007) 'Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients' Information Sciences, 177, 5033:5049.

Examples

```
#########################
## 1 response variable ##
#########################
in.name <- c("x1","x2")
nlev <- c(20, 20)
lower <- c(-2.048, -2.048)
upper <- c(2.048, 2.048)
out.name <- "y"
weight <- 1
C <- 10
pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

emma(in.name, nlev, lower, upper, out.name, opt = "mn", nd = 10, na = 5,
weight, C , w1 = 0.7, w2 = 0.4, c1i = 2.5, c1f = 0.5, c2i = 0.5,
c2f = 2.5, b = 5, pr.mut, graph = "yes", fn1 = ackley)

#########################
## 2 response variables ##
#########################
in.name <- c("x1", "x2")
nlev <- c(20, 20)
lower <- c(-3, -3)
upper <- c(3, 3)
out.name <- c("y1", "y2")
weight <- c(0.2, 0.8)
C <- 10
pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

emma(in.name, nlev, lower, upper, out.name, opt = c("mn", "mx"), nd = 10,
na = 5, weight, C , w1 = 0.7, w2 = 0.4, c1i = 2.5, c1f = 0.5,
c2i = 0.5, c2f = 2.5, b = 5, pr.mut, graph = "yes", fn1 = ackley,
fn2 = peaks, nresp = 2)
```

---

emmacheck *Checking the need for additional experiments*

Description

The function evaluates if one additional experimental point is required. If this is the case, the function provides with details about the additional experiment to be performed.

Usage

```
emmacheck(x, graph, fn1 = NULL, fn2 = NULL, fn3 = NULL, fn4 = NULL, nresp)
```

Arguments

x	An object of class emmatn.
graph	Logical; if "yes", a plot of the MARS model is produced. Note that a plot is produced only if the model contains more than one explanatory variable.
fn1	The first function to be optimised. Use fn1 = NULL if the function is unknown (e.g. when designing experiments in applied problems).
fn2	The second function to be optimised. Use fn2 = NULL if the function is unknown (e.g. when designing experiments in applied problems).
fn3	The third function to be optimised. Use fn3 = NULL if the function is unknown (e.g. when designing experiments in applied problems).
fn4	The fourth function to be optimised. Use fn4 = NULL if the function is unknown (e.g. when designing experiments in applied problems).
nresp	The response to be plotted. Use nresp = 1 to plot the first response...

Details

Once the experiments identified by emma are implemented, the observed response values, the predicted response values, the target and the scalar distances from the target are updated. The solution with the response values closest to the target is thus identified. If such a solution has not been tested yet, emmacheck selects it as an additional experimental point that needs to be investigated.

Value

An object of class emmatn with the components listed below:

xpop	Experimental points investigated.
ypop	Response values observed at the experimental points investigated.
xspace	Experimental region. It is given by all the possible combinations of the factors' levels and contains xpop. The rownames uniquely identify the experimental points and are reported also in xpop.
yspace	Response values that have been either observed or predicted. Observed response values are stored also in ypop. Predicted response values are obtained using a MARS model fitted to the available data.
opt	Indicates if each single function is either minimized ('mn') or maximized ('mx').
nd	Number of experimental points selected initially (t=0).
na	Number of experimental points selected in subsequent iterations (t>0).
Gb	ID of the best experimental point investigated. Use xspace[Gb,] to visualise the best experimental point and use yspace[Gb,] to visualise the measured response value(s). Gb identifies the experimental point whose response values are closest to the target. The target is a set of desirable response values which are automatically selected on the basis of the measured and predicted response values.
add	Logical. If '0' indicates that an additional experimental point needs to be investigated; if '1' indicates that an additional experimental point is not required.

test	IDs of the tested experimental points.
time	Current time instant of the EMMA procedure.
weight	Importance of each response. If only one response is investigated, then weight = 1; if multiple responses are investigated, the sum of the values in weight must be 1.

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

References

Villanova L., Falcaro P., Carta D., Poli I., Hyndman R., Smith-Miles K. (2010) 'Functionalization of Microarray Devices: Process Optimization Using a Multiobjective PSO and Multiresponse MARS Modelling', IEEE CEC 2010, DOI: 10.1109/CEC.2010.5586165

Carta D., Villanova L., Costacurta S., Patelli A., Poli I., Vezzu' S., Scopece P., Lisi F., Smith-Miles K., Hyndman R. J., Hill A. J., Falcaro P. (2011) 'Method for Optimizing Coating Properties Based on an Evolutionary Algorithm Approach', Analytical Chemistry 83 (16), 6373-6380.

Friedman J. H. (1991) 'Multivariate adaptive regression splines' (with discussion), The Annals of Statistics 19, 1:141.

Examples

```
## define the problem variables
in.name <- c("x1", "x2")
nlev <- c(20, 20)
lower <- c(-2.048, -2.048)
upper <- c(2.048, 2.048)
out.name <- "y"
weight <- 1
C <- 10
pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

## Not run:
#########################################################
## simulated problem (with known objective function) ##
#########################################################
##   identify the initial set of experimental runs (initialization)
tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10, fn1 = ackley)

##   identify the experimental runs during subsequent steps of the
## EMMA procedure
for(t in 1:(C - 1))
{
  tn <- emmatn(t, tn, na = 5, opt = "mn", weight, pr.mut = pr.mut,
graph = "yes", fn1 = ackley)
  tn <- emmacheck(tn, graph = "no", fn1 = ackley)
}

## End(Not run)
```

```
############################################################
## applicative problem (with unknown objective function) ##
############################################################
##   identify the initial set of experimental runs (initialization)
tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10)

## perform the experiments in \code{tn$xpop} and measure the response
## values, then load the measured response values in \code{tn$ypop}
tn$ypop <- ackley(tn$xpop)

##   identify the experimental runs during subsequent steps of the
## EMMA procedure
for(t in 1:(C-1))
{
  tn <- emmatn(t, tn, na = 5, opt = "mn", weight, pr.mut = pr.mut,
graph = "no")
  tn$ypop <- ackley(tn$xpop)
  tn <- emmacheck(tn, graph = "no")
  if(tn$add==1) tn$ypop <- ackley(tn$xpop)
}
```

`emmat0` *Defining the initial design*

Description

The function initializes the EMMA procedure. It generates the experimental space and selects the initial set of experimental points, namely the initial set of experiments to be performed. Random sampling is used for that purpose.

Usage

```
emmat0(in.name, nlev, lower, upper, out.name, nd, fn1 = NULL,
fn2 = NULL, fn3 = NULL, fn4 = NULL)
```

Arguments

`in.name`	A vector containing the names of the input variables (factors).
`nlev`	A numeric vector of the same length as `in.name`, containing the number of factor levels.
`lower`	A numeric vector of the same length as `in.name`, containing the lower values of the factors.
`upper`	A numeric vector of the same length as `in.name`, containing the upper values of the factors.
`out.name`	A vector containing the name(s) of the output/response variable(s).
`nd`	Number of experimental points to be selected when `t = 0`.

fn1	The first function to be optimised; use fn1 = NULL if the objective function is unknown, like in applied problems.
fn2	The first function to be optimised; use fn2 = NULL if the objective function is unknown, like in applied problems.
fn3	The third function to be optimised; use fn3 = NULL if the objective function is unknown, like in applied problems.
fn4	The fourth function to be optimised; use fn4 = NULL if the objective function is unknown, like in applied problems.

Details

At the moment the function does not implement the use of constraints for the factors. Unfeasible experiments are easily excluded by manipulating the matrix xspace in an object of class emmat0.

Value

An object of class emmat0 with the components listed below:

xpop	Experimental points investigated.
ypop	Response values observed at the experimental points investigated.
xspace	Experimental region.
yspace	Response values that have been either observed or predicted. Observed response values are stored also in ypop. Predicted response values are obtained using a MARS model fitted to the available data.
opt	Indicates if each single function is either minimized ('mn') or maximized ('mx').
nd	Number of experimental points selected initially (t=0).
na	Number of experimental points selected in subsequent iterations (t>0).
tested	ID of the tested experimental points.
time	Current time instant of the EMMA procedure.
opt	Indicates if each single objective function is either minimized ('mn') or maximized ('mx').

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

References

Villanova L., Falcaro P., Carta D., Poli I., Hyndman R., Smith-Miles K. (2010) 'Functionalization of Microarray Devices: Process Optimization Using a Multiobjective PSO and Multiresponse MARS Modelling', IEEE CEC 2010, DOI: 10.1109/CEC.2010.5586165

Carta D., Villanova L., Costacurta S., Patelli A., Poli I., Vezzu' S., Scopece P., Lisi F., Smith-Miles K., Hyndman R. J., Hill A. J., Falcaro P. (2011) 'Method for Optimizing Coating Properties Based on an Evolutionary Algorithm Approach', Analytical Chemistry 83 (16), 6373-6380.

Examples

```
#########################
## 1 response variable ##
#########################
## define the problem variables
in.name <- c("x1","x2")
nlev <- c(20, 20)
lower <- c(-2.048, -2.048)
upper <- c(2.048, 2.048)
out.name <- "y"

## identify the initial set of experimental runs (initialization)
## simulated problem (with known objective function)
tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10, fn1 = ackley)

## applicative problem (with unknown objective function)
tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10)
## perform the experiments in \code{tn$xpop} and measure the
## response values, then load in \code{tn$ypop} the measured
## response values
# tn$ypop<-...


#########################
## 2 response variables ##
#########################
in.name <- c("x1", "x2")
nlev <- c(20, 20)
lower <- c(-3, -3)
upper <- c(3, 3)
out.name <- c("y1", "y2")
weight <- c(0.2, 0.8)
C <- 10
pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10, fn1 = ackley,
fn2 = peaks)
```

`emmatn` *Defining the additional experimental points*

Description

Given the set of experimental points investigated in previous steps of the EMMA procedure and their measured response values, `emmatn` returns a new set of experimental points to be investigated (and thus new experiments to be performed).

Usage

```
emmatn(t, x, na, opt, weight, C, w1, w2, c1i, c1f, c2i, c2f,
b, pr.mut, graph, fn1 = NULL, fn2 = NULL, fn3 = NULL, fn4 = NULL, nresp)
```

Arguments

t	A numeric value indicating the current time instant of the EMMA procedure.
x	An object of class emmat0 or emmatn. Use emmat0 if t = 1; use emmatn if t > 1.
na	A numeric value indicating the number of experimental points to be selected when t > 0.
opt	A character vector of the same length as the number of responses, indicating for each response function, if the response must be minimized ('mn') or maximized ('mx').
weight	A numerical vector of the same length as the number of responses, reflecting the relevance of each response. Use weight = 1 if only one response is investigated; if multiple responses are investigated, the sum of the values in weight must be 1.
C	A numeric value indicating the maximum number of iterations. The default is C = 20.
w1	The first numeric value used to calculate the inertia weight parameter of the time variant PSO algorithm. The default is w1 = 0.7.
w2	The second numeric value used to calculate the inertia weight parameter of the time variant PSO algorithm. The default is w2 = 0.4.
c1i	The first numeric value used to calculate the acceleration coefficient c1 of the time variant PSO algorithm. The default is c1i = 2.5.
c1f	The second numeric value used to calculate the acceleration coefficient c1 of the time variant PSO algorithm. The default is c1f = 0.5.
c2i	The first numeric value used to calculate the acceleration coefficient c2 of the time variant PSO algorithm. The default is c2i = 0.5.
c2f	The second numeric value used to calculate the acceleration coefficient c2 of the time variant PSO algorithm. The default is c2f = 2.5.
b	A numeric value, used in the mutation operator, that determines the degree of dependence of the mutation on the iteration number. The default is b = 5.
pr.mut	A numeric vector of the same length as the number of iterations C containing the probability of mutation for each time instant.
graph	Logical; if 'yes', a plot of the MARS model is produced. A plot is produced only if the model contains more than one explanatory variable.
fn1	The first function to be optimised; use fn1 = NULL if the objective function is unknown, like in applied problems.
fn2	The first function to be optimised; use fn2 = NULL if the objective function is unknown, like in applied problems.

fn3	The third function to be optimised; use fn3 = NULL if the objective function is unknown, like in applied problems.
fn4	The fourth function to be optimised; use fn4 = NULL if the objective function is unknown, like in applied problems.
nresp	The response to be plotted. Use nresp = 1 to plot the first response...

Details

The parameters w1 and w2 are used to calculate the inertia weight w of the PSO algorithm, namely the parameter that controls the influence of the previous particle velocity on the present velocity. High values of w favour a global search, whereas lower values of w encourage a local search. In EMMA the inertia weight is allowed to decrease linearly with iteration from w1 to w2 thus favouring the exploration initially and the exploitation subsequently. The parameters c1i and c1f are used to calculate the cognitive acceleration coefficient c1 of the PSO algorithm, whereas the parameters c2i and c2f are used to calculate the social acceleration coefficient c2 of the PSO algorithm.Higher values of c1 ensure larger deviation of the particle in the search space (exploration), while higher values of c2 signify the convergence to the current global best (exploitation). In EMMA c1 is allowed to decrease from c1i to c1f and c2 is allowed to increase from c2i to c2f. See Tripathi et al. (2007) for more details.

Value

An object of class emma with the components listed below:

xpop	Experimental points investigated.
ypop	Response values observed at the experimental points investigated.
xspace	Experimental region. It is given by all the possible combinations of the factors' levels and contains xpop. The rownames uniquely identify the experimental points and are reported also in xpop.
yspace	Response values that have been either observed or predicted. Observed response values are stored also in ypop. Predicted response values are obtained using a MARS model fitted to the available data (xpop, ypop).
opt	Indicates if each single function is either minimized ('mn') or maximized ('mx').
nd	Number of experimental points selected initially (t=0).
na	Number of experimental points selected in subsequent iterations (t>0).
tested	IDs of the tested experimental points.
time	Current time instant of the EMMA procedure.
weight	Relative importance of each response. If only one response is investigated, then weight = 1; if multiple responses are investigated, the sum of the values in weight must be 1.
Gb	ID of the best experimental point investigated (global best). Gb identifies the experimental point whose response values are closest to the desirable response values (target); the target is automatically selected on the basis of the measured and predicted response values. Use xspace[Gb,] to visualise the global best and use yspace[Gb,] to visualise its measured response value(s).

Pb	ID of the best experimental point investigated by each particle (personal best). Use xspace[Pb,] to identify the personal bests and use yspace[Pb,] to visualise their measured response values. Among the experimental points associated to one particle, the Pb identifies the experimental point that is whose response values are closest to the target.
Gb.arch	Archive of the global bests identified. Because the global best changes as new experimental points are investigated, an archive is maintained.
Pb.arch	Archive of the personal bests identified. Because the personal bests change as new experimental points are investigated, an archive is maintained.
v	Velocities used to update the particles position. The position of a particle is uniquely determined by the predictors' values; it also defines the experiment to be performed. At each step of EMMA, the position of a particle is updated by adding a numerical value (velocity) to the current value of each single predictor.
sam.x	IDs of the experiments that have been selected in the current iteration of the procedure. Use xspace[sam.x,] to visualise the experiments to be performed.
add	Logical. If '0' indicates that an additional experimental point needs to be investigated; if '1' indicates that an additional experimental point is not required.

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

References

Villanova L., Falcaro P., Carta D., Poli I., Hyndman R., Smith-Miles K. (2010) 'Functionalization of Microarray Devices: Process Optimization Using a Multiobjective PSO and Multiresponse MARS Modelling', IEEE CEC 2010, DOI: 10.1109/CEC.2010.5586165

Carta D., Villanova L., Costacurta S., Patelli A., Poli I., Vezzu' S., Scopece P., Lisi F., Smith-Miles K., Hyndman R. J., Hill A. J., Falcaro P. (2011) 'Method for Optimizing Coating Properties Based on an Evolutionary Algorithm Approach', Analytical Chemistry 83 (16), 6373-6380.

Friedman J. H. (1991) 'Multivariate adaptive regression splines' (with discussion), The Annals of Statistics 19, 1:141.

Tripathi P. K., Bandyopadhyay S., Pal S. K. (2007) 'Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients' Information Sciences, 177, 5033:5049.

Examples

```
## Not run:
#########################
## 1 response variable ##
#########################

in.name <- c("x1","x2")
nlev <- c(20, 20)
lower <- c(-2.048, -2.048)
upper <- c(2.048, 2.048)
out.name <- "y"
weight <- 1
```

```
C <- 10
pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

## simulated problem (with known objective function)
tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10, fn1 = ackley)

for(t in 1:(C-1))
{
  tn <- emmatn(t, tn, na = 5, opt = "mn", weight, pr.mut = pr.mut,
graph = "yes", fn1 = ackley)
  tn <- emmacheck(tn, graph = "no", fn1 = ackley)
}


## applicative problem (with unknown objective function)
tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10)
## use the measured response values to manually fill 'tn$ypop'
tn$ypop<-ackley(tn$xpop)

  for(t in 1:(C-1))
  {
    tn <- emmatn(t, tn, na = 5, opt = "mn", weight, pr.mut = pr.mut,
graph = "yes")
    tn$ypop <- ackley(tn$xpop)
    tn <- emmacheck(tn, graph = "no")
    if(tn$add == 1) tn$ypop<-ackley(tn$xpop)
  }

## End(Not run)

##########################
## 2 response variables ##
##########################
in.name <- c("x1", "x2")
nlev <- c(20, 20)
lower <- c(-3, -3)
upper <- c(3, 3)
out.name <- c("y1", "y2")
weight <- c(0.2, 0.8)
C <- 10
pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

tn <- emmat0(in.name, nlev, lower, upper, out.name, nd = 10, fn1 = ackley,
fn2 = peaks)

for(t in 1:(C-1))
  {
    tn <- emmatn(t, tn, na = 5, opt = c("mn", "mx"), weight,
pr.mut = pr.mut, graph = "yes", fn1 = ackley, fn2 = peaks)
    tn <- emmacheck(tn, graph = "no", fn1 = ackley, fn2 = peaks)
  }
```

peaks *Test problem with multiple peaks*

Description

Generates a benchmark function with multiple peaks.

Usage

```
peaks(x)
```

Arguments

x	A matrix containing the values of the input variables.

Value

Vector of the same length as x giving the values of the benchmark function.

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

Examples

```
x1 <- seq(-3, 3, length = 50)
x2 <- x1

x<-expand.grid(x1, x2)

z <- matrix(peaks(x), nrow = length(x1))

nrz <- nrow(z)
ncz <- ncol(z)
jet.colors <- colorRampPalette( c("red", "yellow") )
nbcol <- 100
color <- jet.colors(nbcol)
zfacet <- z[-1, -1] + z[-1, -ncz] + z[-nrz, -1] + z[-nrz, -ncz]
facetcol <- cut(zfacet, nbcol)

persp(x1, x2, z, col = color[facetcol], theta = 0, phi = 10,
expand = 1, xlab = "x1", ylab = "x2", zlab = "f(x1, x2)",
ticktype = "detailed")
```

`plot.emma` *3D simulation plot*

Description

For a problem with 1 response and 2 input variables (factors) plots a 3D graph and shows how the simulation evolves.

Usage

```
## S3 method for class 'emma'
plot(x, n = 50, fn, C = 10,...)
```

Arguments

`x`	An object of class 'emma'.
`n`	The number of factors' levels to be plotted.
`fn`	The optimization function.
`C`	The number of time instants used in the EMMA procedure.
`...`	Other arguments not used.

Value

...

Author(s)

Laura Villanova, Kate Smith-Miles and Rob J Hyndman

Examples

```
  in.name <- c("x1","x2")
  nlev <- c(20, 20)
  lower <- c(-3, -3)
  upper <- c(3, 3)
  out.name <- "y"
  weight <- 1
  C <- 20
  pr.mut <- c(0.1, 0.07, 0.04, rep(0.01, C-3))

  emma.peaks<-emma(in.name, nlev, lower, upper, out.name, opt = "mx",
nd = 10, na = 5, weight, C , w1 = 0.7, w2 = 0.4, c1i = 2.5,
c1f = 0.5, c2i = 0.5, c2f = 2.5, b = 5, pr.mut, graph = "no",
fn1 = peaks)

  plot(emma.peaks, fn = peaks, n = 50, C = 20)
```

Index

∗Topic **\textasciitildeDesigned Experiments**
distance, 4
emma, 5
emma-package, 2
emmacheck, 9
emmat0, 12
emmatn, 14
∗Topic **\textasciitildeOptimization**
ackley, 3
distance, 4
emma, 5
emmacheck, 9
emmat0, 12
emmatn, 14
peaks, 19
plot.emma, 20

ackley, 3

distance, 4

EMMA *(*emma-package*)*, 2
emma, 5, *7*, *10*, *16*
emma-package, 2
emmacheck, *7*, 9
emmat0, *5*, 12, *13*, *15*
emmatn, *5*, *10*, 14, *15*

peaks, 19
plot.emma, 20