Package: OSFD (via r-universe)

October 6, 2026

Type Package

Title Output Space-Filling Design

Version 3.1

Date 2025-03-01

Description Methods to generate a design in the input space that sequentially fills the output space of a black-box function. The output space-filling designs are helpful in inverse design or feature-based modeling problems. See Wang, Shangkun, Adam P. Generale, Surya R. Kalidindi, and V. Roshan Joseph. (2024), Sequential designs for filling output spaces, Technometrics, 66, 65–76. for details. This work is supported by U.S. National Foundation grant CMMI-1921646.

License GPL (>= 2)

Depends Rcpp (>= 1.0.8), lhs, twinning, dplyr

LinkingTo Rcpp, RcppArmadillo, twinning

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Maintainer Shangkun Wang <sk_wang@gatech.edu>

Author Shangkun Wang [aut, cre], Roshan Joseph [aut]

Repository https://shangkunwang01.r-universe.dev

Date/Publication 2025-03-03 22:20:13 UTC

RemoteUrl https://github.com/cran/OSFD

RemoteRef HEAD

RemoteSha 68ead203f4f49d72263a10d554b6a8dbd3c2844f

Contents

OSFD-package 2
ball_unif 2

IOSFD 3
mMdist 5
OSFD 6
spanfill 8

Index **9**

`OSFD-package` *Sequential algorithms to generate designs that fill the output space.*

Description

Sequential algorithms to generate a design that produces points filling the output space. The underlying mapping `f` from input space to output space is assumed to be a black-box function that can be evaluated in the forward direction. Please see Wang et al. (2024) and Wang & Joseph (2025) for details.

Author(s)

Shangkun Wang, V. Roshan Joseph

Maintainer: Shangkun Wang <sk_wang@gatech.edu>

References

Wang, Shangkun, Adam P. Generale, Surya R. Kalidindi, and V. Roshan Joseph. (2024), "Sequential designs for filling output spaces", Technometrics, 66, 65–76.

Wang, Shangkun, and V. Roshan Joseph. (2025), "Comment: A Model-free Method for Input-Output Space-Filling Design." Technometrics, to appear.

`ball_unif` *(Quasi) uniform points in a p-dimensional ball*

Description

`ball_unif` generates random or quasi-random uniform points in a p-dimensional ball.

Usage

```
ball_unif(cen, rad, n, rand = TRUE)
```

Arguments

`cen`	a vector specifying the center of the ball.
`rad`	radius of the ball.
`n`	number of points.
`rand`	whether to generate random or quasi random points. The default value is TRUE.

Details

`ball_unif` generates random uniform points or quasi uniform points by twinning algorithm in a p-dimensional ball.

Value

a matrix of the generated points.

References

Vakayil, Akhil, and V. Roshan Joseph. (2022). "Data twinning". Statistical Analysis and Data Mining: The ASA Data Science Journal, 15(5), 598-610.

Wang, Shangkun, Adam P. Generale, Surya R. Kalidindi, and V. Roshan Joseph. (2024), "Sequential designs for filling output spaces", Technometrics, 66, 65–76.

Examples

```
x = ball_unif(c(0,0), 1, 10, rand=FALSE)
plot(x, type='p')
```

IOSFD *Input-Output Space-Filling Design*

Description

This function is for producing designs that explicitly balance the input and output points.

Usage

```
IOSFD(
  D = NULL,
  f,
  p,
  q,
  lambda = 0.5,
  n_ini = NA,
  n,
  scale = TRUE,
  CAND = NULL,
  rand_out = FALSE,
  rand_in = FALSE
)
```

Arguments

D	a matrix of the initial design. If not specified, a random Latin hypercube design of size n_ini and dimension p will be generated as initial design.
f	black-box function.
p	input dimension.
q	output dimension.
lambda	the weight for the input space. Its value should be within [0, 1]. The default value is 0.5. When lambda=0, please directly use OSFD.
n_ini	the size of initial design. This initial size must be specified if D is not provided.
n	the size of the final design.
scale	whether to scale the output points to 0 to 1 for each dimension.
CAND	the candidate points in the input space. If Null, it will be automatically generated.
rand_out	whether to use random uniform points or quasi random points by twinning algorithm for generating points in spheres for output space approximation. The default value is FALSE.
rand_in	whether to use random uniform points or quasi random points by twinning algorithm for generating points in spheres for input space candidate sets. The default value is FALSE.

Details

IOSFD produces a design that balances the input and output points by Wang et al. (2025).

Value

D	the final design points in the input space
Y	the output points

References

Wang, Shangkun, and V. Roshan Joseph. (2025), "Comment: A Model-free Method for Input-Output Space-Filling Design." Technometrics, to appear.

Examples

```
# test function: inverse-radius function (Wang et.al 2023)
inverse_r = function(x){
epsilon = 0.1
y1 = 1 / (x[1]^2 + x[2]^2 + epsilon^2) ^ (1/2)
if (x[2]==0){
 y2 = 0
}else if (x[1]==0) {
  y2 = pi / 2}else{
    y2 = atan(x[2] / x[1])
  }
```

```
return (c(y1=y1, y2=y2))
}

set.seed(2022)
p = 2
q = 2
f = inverse_r
n_ini = 10
n = 50
iosfd = IOSFD(f=f, p=p, q=q, n_ini=n_ini, n=n)
D = iosfd$D
Y = iosfd$Y
```

`mMdist` *Minimax distance*

Description

`mMdist` computes the minimax distance of a deisng in a specified region. A large uniform sample from the specified region is needed to compute the minimax distance.

Usage

```
mMdist(X, X_space)
```

Arguments

`X`	a matrix specifying the design.
`X_space`	a large sample of uniform points in the space of interest.

Details

`mMdist` approximates the minimax distance of a set of points `X` by the large sample `X_space` in the space of interest.

Value

the minimax distance.

References

Johnson, Mark E., Leslie M. Moore, and Donald Ylvisaker. (1990), "Minimax and Maximin Distance Designs", Journal of Statistical Planning and Inference, 26, 131–148.

Wang, Shangkun, Adam P. Generale, Surya R. Kalidindi, and V. Roshan Joseph. (2024), "Sequential designs for filling output spaces", Technometrics, 66, 65–76.

Examples

```
# the minimax distance of a random Latin hypercube design
D = randomLHS(5, 2)
mMdist(D, replicate(2, runif(1e5)))
```

OSFD *Output Space-Filling Design*

Description

This function is for producing designs that fill the output space.

Usage

```
OSFD(
  D = NULL,
  f,
  p,
  q,
  n_ini = NA,
  n,
  scale = TRUE,
  method = "EI",
  CAND = NULL,
  rand_out = FALSE,
  rand_in = FALSE
)
```

Arguments

D	a matrix of the initial design. If not specified, a random Latin hypercube design of size n_ini and dimension p will be generated as initial design.
f	black-box function.
p	input dimension.
q	output dimension.
n_ini	the size of initial design. This initial size must be specified if D is not provided.
n	the size of the final design.
scale	whether to scale the output points to 0 to 1 for each dimension.
method	two choices: 'EI' or 'Greedy'; the default is 'EI'.
CAND	the candidate points in the input space. If Null, it will be automatically generated.

rand_out	whether to use random uniform points or quasi random points by twinning algorithm for generating points in spheres for output space approximation. The default value is FALSE.
rand_in	whether to use random uniform points or quasi random points by twinning algorithm for generating points in spheres for input space candidate sets. The default value is FALSE.

Details

OSFD produces a design that fills the output space using the sequential algorithm by Wang et al. (2024).

Value

D	the final design points in the input space
Y	the output points

References

Wang, Shangkun, Adam P. Generale, Surya R. Kalidindi, and V. Roshan Joseph. (2024), "Sequential designs for filling output spaces", Technometrics, 66, 65–76.

Examples

```
# test function: inverse-radius function (Wang et.al 2024)
inverse_r = function(x){
epsilon = 0.1
y1 = 1 / (x[1]^2 + x[2]^2 + epsilon^2) ^ (1/2)
if (x[2]==0){
 y2 = 0
}else if (x[1]==0) {
  y2 = pi / 2}else{
    y2 = atan(x[2] / x[1])
  }
return (c(y1=y1, y2=y2))
}

set.seed(2022)
p = 2
q = 2
f = inverse_r
n_ini = 10
n = 50
osfd = OSFD(f=f, p=p, q=q, n_ini=n_ini, n=n)
D = osfd$D
Y = osfd$Y
```

spanfill *Generate points to approximate the space spanned by the existing points*

Description

spanfill generates points to approximate a space based on existing points. These approximate points can be used to find local fill distance in the space or be used as candidate points in active learning.

Usage

```
spanfill(X, bound = FALSE)
```

Arguments

X	a matrix specifying the existing points
bound	a binary variable indicating whether to bound the generated points to 0 to 1 in each dimension. If bound=TRUE, all the generated points will be projected to the unit hypercube. The default value is FALSE.

Details

spanfill generates points to approximate the space spanned by the existing points. Details can be found in Wang et al. (2024).

Value

a matrix of the generated points to approximate the space.

References

Wang, Shangkun, Adam P. Generale, Surya R. Kalidindi, and V. Roshan Joseph. (2024). "Sequential designs for filling output spaces", Technometrics, 66, 65–76.

Examples

```
X = matrix(runif(20), ncol=2)
spanfill_points = spanfill(X)
plot(spanfill_points, type='p')
```

Index

ball_unif, 2

IOSFD, 3

mMdist, 5

OSFD, 6
OSFD-package, 2

spanfill, 8