

# Package ‘SLSEdesign’

May 30, 2024

**Title** Optimal Regression Design under the Second-Order Least Squares Estimator

**Version** 0.0.3

**Description** With given inputs that include number of points, discrete design space, a measure of skewness, models and parameter value, this package calculates the objective value, optimal designs and plot the equivalence theory under A- and D-optimal criteria under the second-order Least squares estimator. This package is based on the paper “Properties of optimal regression designs under the second-order least squares estimator” by Chi-Kuang Yeh and Julie Zhou (2021) <[doi:10.1007/s00362-018-01076-6](https://doi.org/10.1007/s00362-018-01076-6)>.

**URL** <https://github.com/chikuang/SLSEdesign>

**BugReports** <https://github.com/chikuang/SLSEdesign/issues>

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Imports** CVXR

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Chi-Kuang Yeh [aut, cre] (<<https://orcid.org/0000-0001-7057-2096>>),  
Julie Zhou [aut, ctb],  
Jason Hou-Liu [ctb]

**Maintainer** Chi-Kuang Yeh <[chi-kuang.yeh@uwaterloo.ca](mailto:chi-kuang.yeh@uwaterloo.ca)>

**Repository** CRAN

**Date/Publication** 2024-05-29 23:30:08 UTC

## R topics documented:

Aopt . . . . .	2
calc_phiA . . . . .	3
calc_phiD . . . . .	4

Dopt . . . . .	5
plot_direction_Aopt . . . . .	6
plot_direction_Dopt . . . . .	7
plot_weight . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

Aopt	<i>Calculate the A-optimal design under the second-order Least squares estimator</i>
------	--

---

## Description

Calculate the A-optimal design under the second-order Least squares estimator

## Usage

```
Aopt(N, u, tt, FUN, theta, num_iter = 1000)
```

## Arguments

N	The number of sample points in the design space.
u	The discretized design space.
tt	The level of skewness between 0 to 1 (inclusive). When tt=0, it is equivalent to compute the A-optimal design under the ordinary least squares estimator.
FUN	The function to calculate the derivative of the given model.
theta	The parameter value of the model.
num_iter	Maximum number of iteration.

## Details

This function calculates the A-optimal design and the loss function under the A-optimality. The loss function under A-optimality is defined as the trace of the inverse of the Fisher information matrix

## Value

A list that contains 1. Value of the objective function at solution. 2. Status. 3. Optimal design

## Examples

```
poly3 <- function(xi, theta){
  matrix(c(1, xi, xi^2, xi^3), ncol = 1)
}
Npt <- 101
my_design <- Aopt(N = Npt, u = seq(-1, +1, length.out = Npt),
  tt = 0, FUN = poly3, theta = rep(0,4), num_iter = 2000)
round(my_design$design, 3)
my_design$val
```

---

`calc_phiA`*Calculate the loss function of the A-optimal design*

---

**Description**

Calculate the loss function of the A-optimal design

**Usage**

```
calc_phiA(design, theta, FUN, tt, A)
```

**Arguments**

<code>design</code>	The resulted design that contains the design points and the associated weights
<code>theta</code>	The parameter value of the model
<code>FUN</code>	The function to calculate the derivative of the given model.
<code>tt</code>	The level of skewness
<code>A</code>	The calculated covariance matrix

**Details**

This function calculates the loss function of the design problem under the A-optimality. The loss function under A-optimality is defined as the trace of the inverse of the Fisher information matrix

**Value**

The loss of the model at each design points

**Examples**

```
my_design <- data.frame(location = c(0, 180), weight = c(1/2, 1/2))
theta <- c(0.05, 0.5)
peleg <- function(xi, theta){
  deno <- (theta[1] + xi * theta[2])^2
  rbind(-xi/deno, -xi^2/deno)
}
A <- matrix(c(1, 0, 0, 0, 0.2116, 1.3116, 0, 1.3116, 15.462521), byrow = TRUE, ncol = 3)
res <- calc_phiA(my_design, theta, peleg, 0, A)
res
```

---

 calc\_phiD

*Calculate the loss function of the D-optimal design*


---

### Description

Calculate the loss function of the D-optimal design

### Usage

```
calc_phiD(design, theta, FUN, tt, A)
```

### Arguments

design	The resulted design that contains the design points and the associated weights
theta	The parameter value of the model
FUN	The function to calculate the derivative of the given model.
tt	The level of skewness
A	The calculated covariance matrix

### Details

This function calculates the loss function of the design problem under the D-optimality. The loss function under D-optimality is defined as the log determinant of the inverse of the Fisher information matrix

### Value

The loss of the model at each design points

### Examples

```
my_design <- data.frame(location = c(0, 180), weight = c(1/2, 1/2))
theta <- c(0.05, 0.5)
peleg <- function(xi, theta){
  deno <- (theta[1] + xi * theta[2])^2
  rbind(-xi/deno, -xi^2/deno)
}
A <- matrix(c(1, 0, 0, 0, 0.2116, 1.3116, 0, 1.3116, 15.462521), byrow = TRUE, ncol = 3)
res <- calc_phiA(my_design, theta, peleg, 0, A)
res
```

---

Dopt *Calculate the D-optimal design under the SLSE*

---

### Description

Calculate the D-optimal design under the SLSE

### Usage

```
Dopt(N, u, tt, FUN, theta, num_iter = 1000)
```

### Arguments

N	The number of sample points in the design space.
u	The discretized design space.
tt	The level of skewness. When tt=0, it is equivalent to compute the D-optimal design under the ordinary least squares estimator.
FUN	The function to calculate the derivative of the given model.
theta	The parameter value of the model.
num_iter	Maximum number of iteration.

### Details

This function calculates the D-optimal design and the loss function under the D-optimality. The loss function under D-optimality is defined as the log determinant of the inverse of the Fisher information matrix.

### Value

A list that contains 1. Value of the objective function at solution. 2. Status. 3. Optimal design

### Examples

```
poly3 <- function(xi, theta){  
  matrix(c(1, xi, xi^2, xi^3), ncol = 1)  
}  
Npt <- 101  
my_design <- Dopt(N = Npt, u = seq(-1, +1, length.out = Npt),  
  tt = 0, FUN = poly3, theta = rep(0,4), num_iter = 2000)  
round(my_design$design, 3)  
my_design$val
```

---

plot\_direction\_Aopt    *Verify the optimality condition for the A-optimal design*

---

### Description

Verify the optimality condition for the A-optimal design

### Usage

```
plot_direction_Aopt(u, design, tt, FUN, theta)
```

### Arguments

u	The discretized design points.
design	The A-optimal design that contains the design points and the associated weights
tt	The level of skewness.
FUN	The function to calculate the derivative of the given model.
theta	The parameter value of the model.

### Details

This function produces the figure for the directional derivative of the given A-optimal design of the compact supports. According to the general equivalence theorem, for an optimal design, all the negative value of the directional derivative should be below zero line.

### Value

The plot of the negative value of the directional derivative of an A-optimal design

### Examples

```
poly3 <- function(xi, theta){  
  matrix(c(1, xi, xi^2, xi^3), ncol = 1)  
}  
design = data.frame(location = c(-1, -0.464, 0.464, 1),  
                   weight = c(0.151, 0.349, 0.349, 0.151))  
u = seq(-1, 1, length.out = 201)  
plot_direction_Aopt(u=u, design=design, tt=0, FUN = poly3, theta = rep(0,4))
```

---

plot\_direction\_Dopt    *Verify the optimality condition for the D-optimal design*

---

### Description

Verify the optimality condition for the D-optimal design

### Usage

```
plot_direction_Dopt(u, design, tt, FUN, theta)
```

### Arguments

u	The discretized design points.
design	The D-optimal design that contains the design points and the associated weights.
tt	The level of skewness.
FUN	The function to calculate the derivative of the given model.
theta	The parameter value of the model.

### Details

This function produces the figure for the negative value of the directional derivative of the given D-optimal design of the compact supports. According to the general equivalence theorem, for an optimal design, all the directional derivative should be below zero line.

### Value

The plot of the negative value of the directional derivative of a D-optimal design

### Examples

```
poly3 <- function(xi, theta){  
  matrix(c(1, xi, xi^2, xi^3), ncol = 1)  
}  
design = data.frame(location = c(-1, -0.447, 0.447, 1),  
  weight = rep(0.25, 4))  
u = seq(-1, 1, length.out = 201)  
plot_direction_Dopt(u, design, tt=0, FUN = poly3,  
  theta = rep(0, 4))
```

---

plot_weight	<i>Plot the weight distribution of the optimal design for univariate regression model</i>
-------------	---

---

**Description**

Plot the weight distribution of the optimal design for univariate regression model

**Usage**

```
plot_weight(design)
```

**Arguments**

design            The resulted design that contains the design points and the associated weights

**Details**

This functions produce a figure that contains the location and their associated weights of the resulted optimal design measures.

**Value**

The plot that shows the given optimal design

**Examples**

```
Des = list(location = c(-1, +1), weight = c(0.5, 0.5))  
plot_weight(Des)
```



# Index

Aopt, [2](#)

calc\_phiA, [3](#)

calc\_phiD, [4](#)

Dopt, [5](#)

plot\_direction\_Aopt, [6](#)

plot\_direction\_Dopt, [7](#)

plot\_weight, [8](#)