# Package 'GDSARM'

July 13, 2022

**Title** Gauss - Dantzig Selector: Aggregation over Random Models

**Version** 0.1.0

**Description** The method aims to identify important factors in screening experiments by aggregation over random models as studied in Singh and Stufken (2022) <doi:10.48550/arXiv.2205.13497>. This package provides functions to run the Gauss-Dantzig selector on screening experiments when interactions may be affecting the response. Currently, all functions require each factor to be at two levels coded as +1 and -1.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.0

**Depends** R (>= 2.10)

**LazyData** true

**Imports** lpSolve

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** yes

**URL** https://github.com/agrakhi/GDSARM

**BugReports** https://github.com/agrakhi/GDSARM/issues

**Author** Rakhi Singh [cre, aut] (<https://orcid.org/0000-0003-3469-295X>), John Stufken [aut] (<https://orcid.org/0000-0001-9026-3610>)

**Maintainer** Rakhi Singh <agrakhi@gmail.com>

## R topics documented:

dantzig.delta *Dantzig selector with an option to make profile plot*

## Description

The Dantzig selector (DS) finds a solution for the model parameters of a linear model, beta using linear programming. For a given delta, DS minimizes the L_1-norm (sum of absolute values) of beta subject to the constraint that max(|t(X)(y-X * beta)|) <= delta.

## Usage

```
dantzig.delta(X, y, delta, plot = FALSE)
```

## Arguments

| | |
|---|---|
| X | a design matrix. |
| y | a vector of responses. |
| delta | a vector with the values of delta for which the DS optimization needs to be solved. |
| plot | a boolean value of either TRUE or FALSE with TRUE indicating that the profile plot should be drawn. |

## Value

A matrix of the estimated values of beta with each row corresponding to a particular value of delta.

## Source

Cand\‘es, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. Annals of Statistics 35 (6), 2313–2351.

Phoa, F. K., Pan, Y. H. and Xu, H. (2009). Analysis of supersaturated designs via the Dantzig selector. Journal of Statistical Planning and Inference 139 (7), 2362–2372.

## See Also

GDS_givencols, GDSARM

## Examples

```
data(dataHamadaWu)
X = dataHamadaWu[,-8]
Y = dataHamadaWu[,8]
#scale and center X and y
scaleX = base::scale(X, center= TRUE, scale = TRUE)
scaleY = base::scale(Y, center= TRUE, scale = FALSE)
maxDelta = max(abs(t(scaleX)%*%matrix(scaleY, ncol=1)))
# Dantzig Selector on 4 equally spaced delta values between 0 and maxDelta
dantzig.delta(scaleX, scaleY, delta = seq(0,maxDelta,length.out=4))
```

dataCompoundExt *Compound Extraction experiment of Dopico-Garc\' ia et al. (2007)*

## Description

An analytical experiment conducted by Dopico-Garc\' ia et al. (2007) to characterize the chemical composition of white grapes simultaneously determining the most important phenolic compounds and organic acids for the grapes. This example has been further studied in Phoa et al. (2009b) for one phenolic compound, kaempferol-3-Orutinoside + isorhamnetin-3-O glucoside, which is also what we studied. It is accepted for these data that fitting a main-effects model suggests that V3 (Factor C), V4 (Factor D), and interaction V1:V4 (A:D) are active effects.

## Usage

```
data(dataCompoundExt)
```

## Format

A data frame with 12 rows and 9 columns:

**V1** Factor A

**V2** Factor B

**V3** Factor C

**V4** Factor D

**V5** Factor E

**V6** Factor F

**V7** Factor G

**V8** Factor H

**V9** Response

## Source

Dopico-Garc\' ia, M.S., Valentao, P., Guerra, L., Andrade, P. B., and Seabra, R. M. (2007). Experimental design for extraction and quantification of phenolic compounds and organic acids in white "Vinho Verde" grapes Analytica Chimica Acta, 583(1): 15–22.

Phoa, F. K., Wong, W. K., and Xu, H (2009b). The need of considering the interactions in the analysis of screening designs. Journal of Chemometrics: A Journal of the Chemometrics Society, 23(10): 545–553.

## Examples

```
data(dataCompoundExt)
X = dataCompoundExt[,-9]
Y= dataCompoundExt[,9]
```

dataHamadaWu *Cast fatigue experiment of Hunter et al. (1982)*

## Description

A cast fatigue experiment with 12 runs and 7 factors was originally studied by Hunter et al. (1982), and was later revisited by Hamada and Wu (1992) and Phoa et al. (2009), among others. It is widely accepted for these data that V6 (F) and interaction V6:V7 (F:G) are active effects, with interaction of V1:V5 (A:E) possibly being active as well.

## Usage

```
data(dataHamadaWu)
```

## Format

A data frame with 12 rows and 8 columns:

**V1** Factor A

**V2** Factor B

**V3** Factor C

**V4** Factor D

**V5** Factor E

**V6** Factor F

**V7** Factor G

**V8** Response

## Source

Hamada, M. and C. F. J. Wu (1992). Analysis of designed experiments with complex aliasing. Journal of Quality Technology 24 (3), 130–137.

Hunter, G. B., F. S. Hodi, and T. W. Eagar (1982). High cycle fatigue of weld repaired cast Ti-6AI-4V. Metallurgical Transactions A 13 (9), 1589–1594.

Phoa, F. K., Y. H. Pan, and H. Xu (2009). Analysis of supersaturated designs via the Dantzig selector. Journal of Statistical Planning and Inference 139 (7), 2362–2372.

## Examples

```
data(dataHamadaWu)
X = dataHamadaWu[,-8]
Y= dataHamadaWu[,8]
```

GDSARM *Gauss-Dantzig Selector - Aggregation over Random Models (GDS-ARM)*

## Description

The GDS-ARM procedure consists of three steps. First, it runs the Gauss Dantzig Selector (GDS) `nrep` times, each time with a different set of `nint` randomly selected two-factor interactions. All `m` main effects are included in each GDS run. Second, the best `ntop` models are identified with the smallest BIC. Effects that appear in at least `pkeep x ntop` of the `ntop` models are then passed on to the third stage. In the third stage, stepwise regression is used. With `n` being the number of runs, the stepwise regression starts with at most `n-3` selected effects from the previous step. The remaining effects from the previous step as well as all main effects are given a chance to enter into the model using the forward-backward stepwise regression. The function also has the option of using the modified GDS-ARM. The modified version incorporates effect heredity in two steps, first, for each model found by GDS, we ignore active interactions when at least one of the main effects is not active (for weak heredity) or when both main effects are not active (for strong heredity); and second, we do the same for the model found after the stepwise stage of GDS-ARM.

## Usage

```
GDSARM(
  delta.n = 10,
  nint,
  nrep,
  ntop,
  pkeep,
  design,
  Y,
  cri.penter = 0.01,
  cri.premove = 0.05,
  opt.heredity = c("none"),
  seedvalue = 1234
)
```

## Arguments

| | |
|---|---|
| `delta.n` | a positive integer suggesting the number of delta values to be tried. `delta.n` equally spaced values of `delta` will be used strictly between 0 and `max(\|t(X)y\|)`. The default value is set to 10. |
| `nint` | a positive integer representing the number of randomly chosen interactions. The suggested value to use is the ceiling of 20% of the total number of interactions, that is, for `m` factors, we have `ceiling(0.2(m choose 2))`. |
| `nrep` | a positive integer representing the number of times GDS should be run. The suggested value is `(m choose 2)`. |
| `ntop` | a positive integer representing the number of top models to be selected among the `nrep` models. The suggested value is `max(20,(nrep x nint)/(m(m-1))`. The value of `ntop` should not exceed `nrep`. |
| `pkeep` | a number between 0 and 1 representing the proportion of `ntop` models in which an effect needs to appear in order to be selected for the stepwise regression stage. |

| | |
|---|---|
| design | a $n \times m$ matrix of m two-level factors. The levels should be coded as +1 and -1. |
| Y | a vector of n responses. |
| cri.penter | the p-value cutoff for the most significant effect to enter into the stepwise regression model. The suggested value is 0.01. |
| cri.premove | the p-value cutoff for the least significant effect to exit from the stepwise regression model. The suggested value is 0.05. |
| opt.heredity | a string with either none, or weak, or strong. Denotes whether the effect-heredity (weak or strong) should be embedded in GDS-ARM. The default value is none as suggested in Singh and Stufken (2022). |
| seedvalue | a seed value that will fix the set of interactions being selected. The default value is seed to 1234. |

## Value

A list returning the selected effects as well as the corresponding important factors.

## Source

Cand\‘es, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. Annals of Statistics 35 (6), 2313–2351.

Dopico-Garc\’ ia, M.S., Valentao, P., Guerra, L., Andrade, P. B., and Seabra, R. M. (2007). Experimental design for extraction and quantification of phenolic compounds and organic acids in white "Vinho Verde" grapes Analytica Chimica Acta, 583(1): 15–22.

Hamada, M. and Wu, C. F. J. (1992). Analysis of designed experiments with complex aliasing. Journal of Quality Technology 24 (3), 130–137.

Hunter, G. B., Hodi, F. S. and Eagar, T. W. (1982). High cycle fatigue of weld repaired cast Ti-6AI-4V. Metallurgical Transactions A 13 (9), 1589–1594.

Phoa, F. K., Pan, Y. H. and Xu, H. (2009). Analysis of supersaturated designs via the Dantzig selector. Journal of Statistical Planning and Inference 139 (7), 2362–2372.

Singh, R. and Stufken, J. (2022). Factor selection in screening experiments by aggregation over random models, 1–31. doi: 10.48550/arXiv.2205.13497

## See Also

GDS_givencols, dantzig.delta

## Examples

```
data(dataHamadaWu)
X = dataHamadaWu[,-8]
Y = dataHamadaWu[,8]
delta.n = 10
n = dim(X)[1]
m = dim(X)[2]
nint = ceiling(0.2*choose(m,2))
nrep = choose(m,2)
ntop = max(20, nint*nrep/(2*choose(m,2)))
pkeep = 0.25
cri.penter = 0.01
cri.premove = 0.05
design = X
```

```
# GDS-ARM with default values
GDSARM(delta.n, nint, nrep, ntop, pkeep, X, Y, cri.penter, cri.premove)

# GDS-ARM with default values but with weak heredity
opt.heredity="weak"
GDSARM(delta.n, nint, nrep, ntop, pkeep, X, Y, cri.penter, cri.premove, opt.heredity)


data(dataCompoundExt)
X = dataCompoundExt[,-9]
Y = dataCompoundExt[,9]
delta.n = 10
n = dim(X)[1]
m = dim(X)[2]
nint = ceiling(0.2*choose(m,2))
nrep = choose(m,2)
ntop = max(20, nint*nrep/(2*choose(m,2)))
pkeep = 0.25
cri.penter = 0.01
cri.premove = 0.05
design = X
# GDS-ARM on compound extraction
GDSARM(delta.n, nint, nrep, ntop, pkeep, X, Y, cri.penter, cri.premove)

# GDS-ARM on compound extraction with strong heredity
opt.heredity = "strong"
GDSARM(delta.n, nint, nrep, ntop, pkeep, X, Y, cri.penter, cri.premove, opt.heredity)
```

---

`GDS_givencols` *Gauss-Dantzig Selector*

---

## Description

This function runs the Gauss-Dantzig selector on the given columns. We have two options: either (a) GDS(m) on the `m` main effects, and (b) GDS(m+2fi) on the `m` main effects and the corresponding two-factor interactions. For a given `delta`, DS minimizes the L_1-norm (sum of absolute values) of `beta` subject to the constraint that `max(|t(X)(y-X * beta)|) <= delta`. The GDS is run for multiple values of `delta`. We use kmeans and BIC to select a best model.

## Usage

```
GDS_givencols(delta.n = 10, design, Y, which.cols = c("main2fi"))
```

## Arguments

| | |
|---|---|
| `delta.n` | a positive integer suggesting the number of delta values to be tried. `delta.n` equally spaced values of `delta` will be used strictly between 0 and `max(|t(X)y|)`. The default value is set to 10. |
| `design` | a $n \times m$ matrix of `m` two-level factors. The levels should be coded as +1 and -1. |
| `Y` | a vector of `n` responses. |

`which.cols` a string with either `main` or `main2fi`. Denotes whether the Gauss-Dantzig Selector should be run on the main effect columns (`main`), or on all main effects plus all 2 factor interaction columns (`main2fi`). The default value is `main2fi`.

## Value

A list returning the selected effects as well as the corresponding important factors.

## Source

Cand\'es, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. Annals of Statistics 35 (6), 2313–2351.

Dopico-Garc\' ia, M.S., Valentao, P., Guerra, L., Andrade, P. B., and Seabra, R. M. (2007). Experimental design for extraction and quantification of phenolic compounds and organic acids in white "Vinho Verde" grapes Analytica Chimica Acta, 583(1): 15–22.

Hamada, M. and Wu, C. F. J. (1992). Analysis of designed experiments with complex aliasing. Journal of Quality Technology 24 (3), 130–137.

Hunter, G. B., Hodi, F. S. and Eagar, T. W. (1982). High cycle fatigue of weld repaired cast Ti-6AI-4V. Metallurgical Transactions A 13 (9), 1589–1594.

Phoa, F. K., Pan, Y. H. and Xu, H. (2009). Analysis of supersaturated designs via the Dantzig selector. Journal of Statistical Planning and Inference 139 (7), 2362–2372.

Singh, R. and Stufken, J. (2022). Factor selection in screening experiments by aggregation over random models, 1–31. doi: 10.48550/arXiv.2205.13497

## See Also

GDSARM, dantzig.delta

## Examples

```
data(dataHamadaWu)
X = dataHamadaWu[,-8]
Y = dataHamadaWu[,8]
delta.n = 10
# GDS on main effects
GDS_givencols(delta.n, design = X, Y=Y, which.cols = "main")

# GDS on main effects and two-factor interactions
GDS_givencols(delta.n, design = X, Y=Y)

data(dataCompoundExt)
X = dataCompoundExt[,-9]
Y = dataCompoundExt[,9]
delta.n = 10
# GDS on main effects
GDS_givencols(delta.n, design = X, Y=Y, which.cols = "main")
# GDS on main effects and two-factor interactions
GDS_givencols(delta.n, design = X, Y=Y, which.cols = "main2fi")
```

# Index