Package 'designGLMM'

February 16, 2016

Type Package

Title Finding Optimal Block Designs for a Generalised Linear Mixed Model

Version 0.1.0

Date 2016-02-16

Author Stephen Bush and Katya Ruggiero

Maintainer Stephen Bush <stephen.bush@uts.edu.au>

Description
Use simulated annealing to find optimal designs for Poisson regression models with blocks.

License GPL-3

LazyData TRUE

RoxygenNote 5.0.1

Imports stats

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-16 13:50:36

R topics documented:

`findOptimalApproxDesign` *Find an Optimal Approximate Design for Poisson Regression*

Description

This function will calculate the design weights for an A-optimal completely randomised design with a Poisson response.

Usage

```
findOptimalApproxDesign(means, silent = FALSE)
```

Arguments

`means`	A list of length v containing conditional means for each treatment e.g. c(1,1,2) for three treatments with means 1, 1, and 2 respectively
`silent`	a logical to indicate whether the design should be supressed out (TRUE) or not (FALSE)

Value

`design`	A list of the same dimension as means that contains the design weights for the corresponding treatments.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

`findOptimalExactDesign`, `findOptimalBlockDesign`

Examples

```
# Calculating design weights for an approximate design with means 1, 2, and 4

findOptimalApproxDesign(c(1,2,4), silent = FALSE)
```

`findOptimalBlockDesign` *Find an efficient block design for a Poisson GLMM*

Description

`findOptimalBlockDesign` finds a D-optimal or A-optimal block design for a Poisson generalised linear mixed model with gamma distributed random effects (Poisson-Gamma mixture) using simulated annealing.

Usage

```
findOptimalBlockDesign(numblock,blksize,means,sigma,sigmaB,probs=c(1),
                       criterion="D",link="Poisson",trace=FALSE,iter=10000,
                       temp=10,tmax=10,silent=FALSE,tol=0.0001,maxtime=60,
                       startdes="random")
```

Arguments

`numblock`	an integer specifying the number of blocks in the block design to be constructed.
`blksize`	an integer specifying the number of experimental units that are in each block. For this function, the block size is constant across all blocks.
`means`	a vector specifying the means for each treatment group.
`sigma`	a numeric indicating the within block standard deviation in excess of the Poisson residual error
`sigmaB`	a numeric indicating the between block standard deviation.
`probs`	a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.
`criterion`	Optimality criterion. For D-optimality use `"D"` and for A-optimality use `"A"`.
`link`	The link function used. Currently only `"Poisson"` is supported.
`trace`	a logical indicating whether the function should return information about the iterations of the simulated annealing algorithm.
`iter`	the number of simulated annealing iterations per round. This variable is passed to the `optim` function.
`temp`	starting temperature for simulated annealing algorithm. This variable is passed to the `optim` function.
`tmax`	the number of function evaluations at each temperature in the simulated annealing algorithm. This variable is passed to the `optim` function.
`silent`	a logical indicating whether any output should be returned.

`tol`	a numeric value indicating the numerical tolerance required to stop the simulated annealing algorithm from restarting
`maxtime`	a numeric value indicating the maximum amount of time the fiunction is allowed to take before stopping optimisation
`startdes`	either `"random"` or `"approximate"`. If random is chosen then a random starting design will be used. If "approximate" is used then the approximate proportions of each treatment in the optimal completely randomised design will appear in each block. If the block size is less than or close to the number of treatments, then "approximate" is not recommended.

Details

This function uses the simulated annealing algorithm provided in the `optim` function to find optimal block designs for a Poisson generalised linear mixed model. The linear predictor for the experimental unit j in block i using this model is:

$\eta_{R(i,j)} = \mu + \tau_{R(i,j)} + B_i + e_{i,j}$

where i = 1,...,b, j=1,...,k and there are b blocks each with k experimental units. R(i,j) is the treatment assigned to experimental unit j in block i.

Value

`design`	a matrix containing the optimal design. Each row corresponds to a single block.
`value`	the value of the D or A optimality criterion value for the optimal design, depending on whether `"D"` or `"A"` was specified under the criterion option.
`iter`	lists the evolution of the criterion value through iterations of the simulated annealing algorithm.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

`findOptimalExactDesign`, `findOptimalApproxDesign`, `updateDesign_BD`, `objfnA_BD`, `objfnD_BD`

Examples

```
## Constructing a D-optimal block design with 4 blocks of size 3 with seven treatments
## with means c(5,5.5,6,5.5,7,10,4) with between block standard deviation 0.3
## and no overdispersion (sigma=0). In each round of simulated annealing, we use 1000
```

```
## iterations

findOptimalBlockDesign(numblock = 4, blksize = 3, means = c(5,5.5,6,5.5,7,10,4),
  sigma = 0, sigmaB = 0.3,iter=1000)

## Constructing an A-optimal design with the same means

findOptimalBlockDesign(numblock = 4, blksize = 3, means = c(5,5.5,6,5.5,7,10,4),
  sigma = 0, sigmaB = 0.3, criterion = "A",iter=1000)
```

`findOptimalExactDesign` *Find optimal Completely Randomised Design using Simulated Annealing*

Description

`findOptimalExactDesign` finds a D-optimal or A-optimal completely randomised design for a Poisson generalised linear model using simulated annealing.

Usage

```
findOptimalExactDesign(numunits, means, sigma, probs = c(1),criterion="D",
                       link="Poisson", trace = FALSE, iter = 10000, temp = 10,
                       tmax = 10, silent = FALSE, tol = 1e-04, maxtime = 60)
```

Arguments

`numunits`	the total number of experimental units in the design
`means`	a vector specifying the means for each treatment group.
`sigma`	a numeric indicating the within block standard deviation in excess of the Poisson residual error
`probs`	a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.
`criterion`	Optimality criterion. For D-optimality use `"D"` and for A-optimality use `"A"`.
`link`	The link function used. Currently only `"Poisson"` is supported.
`trace`	a logical indicating whether the function should return information about the iterations of the simulated annealing algorithm.
`iter`	the number of simulated annealing iterations per round. This variable is passed to the `optim` function.

temp	starting temperature for simulated annealing algorithm. This variable is passed to the optim function.
tmax	the number of function evaluations at each temperature in the simulated annealing algorithm. This variable is passed to the optim function.
silent	a logical indicating whether any output should be returned.
tol	a numeric value indicating the numerical tolerance required to stop the simulated annealing algorithm from restarting.
maxtime	a numeric value indicating the maximum amount of time the fiunction is allowed to take before stopping optimisation.

Details

This function uses the simulated annealing algorithm provided in the optim function to find optimal completely randomised designs for a Poisson generalised linear model. The linear predictor for the experimental unit j in using this model is:

$\eta_{R(j)} = \mu + \tau_{R(j)} + e_j$

where j=1,...,k and R(j) is the treatment assigned to experimental unit j.

Value

design	a matrix containing the optimal design.
value	the value of the D or A optimality criterion value for the optimal design, depending on whether "D" or "A" was specified under the criterion option.
iter	lists the evolution of the criterion value through iterations of the simulated annealing algorithm.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

findOptimalBlockDesign, findOptimalApproxDesign, updateDesign_CRD, objfnA_CRD, objfnD_CRD

Examples

```
# Finding a D-optimal completely randomised design with 10 observations, three treatments
# with means 1,2, and 4 and no overdispersion.In each round of simulated annealing, we use
# 1000 iterations
```

```
findOptimalExactDesign(numunits=10, means=c(1,2,4), sigma=0, criterion="D",iter=1000)

# Finding an A-optimal completely randomised design with 10 observations, three treatments
# with means 1,2, and 4 and no overdispersion.

findOptimalExactDesign(numunits=10, means=c(1,2,4), sigma=0, criterion="A",iter=1000)
```

---

objfnA_BD *Calculate A-optimality criterion for Block Design with Poisson response*

Description

This function calculates the trace of the inverse of the Fisher information matrix for a block design with a Poisson response. This function assumes that there are b blocks, each containing k experimental units and v treatments in total. Block effects are assumed to be normal with mean 0 and standard deviation sigb. Additional residual error can be specified through sige.

Usage

```
objfnA_BD(des, ntmt, blksz, sigb, sige, means, probs=c(1))
```

Arguments

des	The design under consideration, a list of length b x k containing treatment indices (1, 2, ..., v)
ntmt	The number of distinct treatments (v)
blksz	The number of treatments in each block (k)
sigb	Standard deviation of block effect
sige	Standard deviation of excess error
means	A list of length v containing conditional means for each treatment e.g. c(1,1,2) for three treatments with means 1, 1, and 2 respectively
probs	a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.

Details

This function is designed to work with findOptimalBlockDesign, and as such shares the arguments of updateDesign_BD. It can, however, be used on its own. The probs argument is not used in this function, but is in updateDesign_BD.

Value

Returns the negative of the determinant of the Fisher information matrix for the proveded design.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

findOptimalBlockDesign, updateDesign_BD

Examples

```
# Finding the A-optimality objective value for the design ((1,1,2,2),(1,1,2,3),(1,2,2,3))
# where there are three treatments in three blocks of size four when the treatment means
# are 1, 2, and 4, the between block standard deviation is 0.2 (sigb=0.2) and there is
# no overdispersion (sige=0)

objfnA_BD(c(1,1,2,2,1,1,2,3,1,2,2,3), ntmt=3, blksz=4, sigb=0.2, sige=0, means=c(1,2,4))
```

---

objfnA_CRD — *Calculate A-optimality criterion for Completely Randomised Design with Poisson response*

Description

This function calculates the trace of the inverse of the Fisher information matrix for a completely randomised design with a Poisson response.

Usage

```
objfnA_CRD(des,ntmt,sige,means,probs=c(1))
```

Arguments

des	The design under consideration, a list of length b x k containing treatment indices (1, 2, ..., v)
ntmt	The number of distinct treatments (v)
sige	Standard deviation of excess error

`means` A list of length v containing conditional means for each treatment e.g. c(1,1,2) for three treatments with means 1, 1, and 2 respectively

`probs` a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.

Details

This function is designed to work with `findOptimalExactDesign`, and as such shares the arguments of `updateDesign_CRD`. It can, however, be used on its own. The `probs` argument is not used in this function, but is in `updateDesign_CRD`.

Value

Returns the negative of the determinant of the Fisher information matrix for the provided design.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

`findOptimalExactDesign`, `updateDesign_CRD`

Examples

```
# Finding the A-optimality objective value for the design (1,1,1,1,2,2,2,3,3,3)
# where there are three treatments, the treatment means are 1, 2, and 4, and
# there is no overdispersion (sige=0)

objfnA_CRD(c(1,1,1,1,2,2,2,3,3,3),ntmt=3,sige=0,means=c(1,2,4))
```

objfnD_BD *Calculate D-optimality criterion value for a Block Design*

Description

This function calculates the determinant of the Fisher information matrix for a block design with a Poisson response. This function assumes that there are b blocks, each containing k experimental units and v treatments in total. Block effects are assumed to be normal with mean 0 and standard deviation `sigb`. Additional residual error can be specified through `sige`.

Usage

```
objfnD_BD(des, ntmt, blksz, sigb, sige, means, probs)
```

Arguments

`des`	The design under consideration, a list of length b x k containing treatment indices (1, 2, ..., v)
`ntmt`	The number of distinct treatments (v)
`blksz`	The number of treatments in each block (k)
`sigb`	Standard deviation of block effect
`sige`	Standard deviation of excess error
`means`	A list of length v containing conditional means for each treatment e.g. c(1,1,2) for three treatments with means 1, 1, and 2 respectively
`probs`	a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.

Details

This function is designed to work with `findOptimalBlockDesign`, and as such shares the arguments of `updateDesign_BD`. It can, however, be used on its own. The `probs` argument is not used in this function, but is in `updateDesign_BD`.

Value

Returns the negative of the determinant of the Fisher information matrix for the proveded design.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

findOptimalBlockDesign, updateDesign_BD

Examples

```
# Finding the D-optimality objective value for the design ((1,1,2,2),(1,1,2,3),(1,2,2,3))
# where there are three treatments in three blocks of size four when the treatment means
# are 1, 2, and 4, the between block standard deviation is 0.2 (sigb=0.2) and there is
# no overdispersion (sige=0)

objfnD_BD(c(1,1,2,2,1,1,2,3,1,2,2,3), ntmt=3, blksz=4, sigb=0.2, sige=0, means=c(1,2,4))
```

---

objfnD_CRD *Calculate D-optimality criterion for Completely Randomised Design with Poisson response*

Description

This function calculates the determinant of the Fisher information matrix for a completely randomised design with a Poisson response.

Usage

```
objfnD_CRD(des, ntmt, sige, means, probs=c(1))
```

Arguments

des	The design under consideration, a list of length b x k containing treatment indices (1, 2, ..., v)
ntmt	The number of distinct treatments (v)
sige	Standard deviation of excess error
means	A list of length v containing conditional means for each treatment e.g. c(1,1,2) for three treatments with means 1, 1, and 2 respectively
probs	a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.

Details

This function is designed to work with `findOptimalExactDesign`, and as such shares the arguments of `updateDesign_CRD`. It can, however, be used on its own. The `probs` argument is not used in this function, but is in `updateDesign_CRD`.

Value

Returns the negative of the determinant of the Fisher information matrix for the proveded design.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

`findOptimalExactDesign`, `updateDesign_CRD`

Examples

```
# Finding the D-optimality objective value for the design (1,1,1,1,2,2,2,3,3,3)
# where there are three treatments, the treatment means are 1, 2, and 4, and
# there is no overdispersion (sige=0)

objfnD_CRD(c(1,1,1,1,2,2,2,3,3,3),ntmt=3,sige=0,means=c(1,2,4))
```

`updateDesign_BD` *Updating a Block Design*

Description

This function provides an update function for finding block designs. This function works by by making one or more substitutions. The number of substitutions made and the probability of making a particular number of substitution is given in a vector of probabilities.

Usage

```
updateDesign_BD(des,ntmt,blksz,sigb=0,sige=0,means=c(1,1),probs=c(1))
```

Arguments

des	a vector indicating the initial design to which we will make substitutions.
ntmt	a numeric indicating the number of treatments in the design.
blksz	an integer specifying the number of experimental units that are in each block. For this function, the block size is constant across all blocks.
sigb	a numeric indicating the between block standard deviation.
sige	a numeric indicating the within block standard deviation in excess of the Poisson residual error.
means	a vector specifying the means for each treatment group.
probs	a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.

Details

This function is designed to work with `findOptimalBlockDesign`, and as such shares the arguments of `objfnA_BD` and `objfnD_BD`. It can, however, be used on its own. The `sige`, `sigb`, and `means` arguments are not used in this function, but are in `objfnA_BD` and `objfnD_BD`. If using this function on its own, then these terms may be omitted without changing the result of the function.

Value

Returns a vector containing the new design in flattened form.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

findOptimalBlockDesign

Examples

```
# Suppose that we have a block design ((1,1,2,2),(1,1,2,3),(1,2,2,3)) and would like to
# replace exactly one entry with a randomly sampled treatment. Then we set the probability
# that one entry is replaced equal to 1.

updateDesign_BD(des=c(1,1,2,2,1,1,2,3,1,2,2,3),ntmt=3,blksz=4,probs=c(1))

# Now suppose that we would like to replace one entry with probability 0.6 and replace
# two entries with probability 0.4 then probs=c(0.6,0.4).

updateDesign_BD(des=c(1,1,2,2,1,1,2,3,1,2,2,3),ntmt=3,blksz=4,probs=c(0.6,0.4))
```

---

updateDesign_CRD *Update function for Completely Randomised Design search*

Description

This function provides an update function for finding completely randomised designs. This function works by by making one or more substitutions. The number of substitutions made nd the probability of making a particular number of substitution is given in a vector of probabilities.

Usage

```
updateDesign_CRD(des, ntmt, sige=0,means=c(1,1), probs=c(1))
```

Arguments

des	a vector indicating the initial design to which we will make substitutions.
ntmt	a numeric indicating the number of treatments in the design.
sige	a numeric indicating the within block standard deviation in excess of the Poisson residual error
means	a vector specifying the means for each treatment group.
probs	a list of probabilities specifying the probability that each step of the simulated annealing substitutes a certain number of design points. The first entry corresponds to the probability that only one substitution is made in a simulated annealing step, the second is the probability that two substitutions are made and so on. By default this is set to c(1) which means that only one substitution is made in each simulated annealing step.

Details

This function is designed to work with `findOptimalExactDesign`, and as such shares the arguments of `objfnA_CRD` and `objfnD_CRD`. It can, however, be used on its own. The `sige` and `means` arguments are not used in this function, but are in `objfnA_CRD` and `objfnD_CRD`. If using this function on its own, then these terms may be omitted without changing the result of the function.

Value

Returns a vector containing the new design.

Author(s)

Stephen Bush (stephen.bush@uts.edu.au)

Katya Ruggiero (k.ruggiero@auckland.ac.nz)

References

Bush, S., and Ruggiero, K. (2016) Optimal block designs for experiments with responses drawn from a Poisson distribution, *Under Review*, preprint available at http://arxiv.org/abs/1601.00477

See Also

findOptimalExactDesign

Examples

```
# Suppose that we have a completely randomised design (1,1,1,1,2,2,2,3,3,3) and would like
# to replace exactly one entry with a randomly sampled treatment. Then we set the
# probability that one entry is replaced equal to 1.

updateDesign_CRD(des=c(1,1,1,1,2,2,2,3,3,3),ntmt=3,probs=c(1))

# Now suppose that we would like to replace one entry with probability 0.6 and replace
# two entries with probability 0.4 then probs=c(0.6,0.4).

updateDesign_CRD(des=c(1,1,1,1,2,2,2,3,3,3),ntmt=3,probs=c(0.6,0.4))
```

Index