Package ‘linearERRfit’

November 12, 2020

Type Package

Title Fit linear ERR models, including dose information to multiple locations

Version 0.1.2

Author Sander Roberti

Maintainer Sander Roberti <`s.roberti@nki.nl`>

Description Fit linear ERR models, including dose information to multiple locations.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports bbmle

Depends R (>= 3.5.0)

R topics documented:

`linearERR` *Fit linear ERR model and perform jackknife correction*

Description

Fits the linear ERR model on matched case-control data and performs first and second order jackknife correction

Usage

```
linearERR(
  data,
  doses,
  set,
  status,
  loc,
  corrvars = NULL,
  ccmethod = "CCAL",
  repar = FALSE,
  initpars = rep(0, length(doses) + length(corrvars)),
  fitopt = NULL,
  uplimBeta = 5,
  profCI = TRUE,
  doJK1 = FALSE,
  doJK2 = FALSE,
  jkscorethresh = 0.01,
  jkvalrange = c(-Inf, Inf)
)
```

Arguments

`data` data frame containing matched case-control data, with a number of columns for doses to different locations, a column containing matched set numbers, a column containing the case's tumor location (value between 1 and the number of locations, with location x corresponding to the x-th column index in `doses`) and a column serving as a case-control indicator. Other covariates can also be included, in this case a parameter for each covariate column will be estimated. Hence factor variables need to be converted to dummy variables using `model.matrix`. If using `ccmethod='meandose'`, a column for tumor location is still required but in this case the column can be a vector of ones.

`doses` vector containing the indices of columns containing dose information.

`set` column index containing matched set numbers.

`status` column index containing case status.

`loc` column index containing the location of the matched set's case's second tumor.

`corrvars` vector containing the indices of columns containing variables to be corrected for.

`ccmethod` choice of method of analysis: one of meandose, CCML, CCAL or CL. Defaults to CCAL

`repar` reparametrize to $\beta = exp(\xi)$? Defaults to `FALSE`

`initpars` initial values for parameters, default is 0 for all parameters. If supplying a different vector, use a vector with an initial value for β or ξ, a 0 for the reference location, one for all of the other location effects and one for each other covariate (in that order). Note that if `repar=TRUE`, the initial value is used for ξ.

`fitopt` list with options to pass to `control` argument of optimizer (see details)

`uplimBeta` upper limit for $\beta = exp(\xi)$, default value 5. This is used for constraining the MLE estimation in some settings and for the jackknife inclusion criteria, and can be infinite except when Brent optimization is used (see details) `profCI` boolean: compute 95% profile likelihood confidence interval for β/ξ? Default value TRUE.

`doJK1` perform first order jackknife correction? Automatically set to `TRUE` when `doJK2=TRUE`. Caution: this can take a long time to run. Default value FALSE

`doJK2` perform second order jackknife correction? Caution: this can take a very long time to run. Default value FALSE

`jkscorethresh` square L2 norm threshold for leave-one-out and leave-two-out estimates to be included in the computation of the first and second order jackknife corrected estimate, respectively

`jkvalrange` range of leave-one-out and leave-two-out beta/xi estimates to be allowed in the computation of the first and second order jackknife corrected estimate, respectively

Details

This is the main function of the package, used for fitting the linear ERR model in matched case-control data. Use this function to estimate the MLE (including a profile likelihood confidence interval for the dose effect) and to perform first and second order jackknife corrections.

The model being fit is HR=$\sum(1+\beta d_l)exp(\alpha_l + X^T\gamma)$, where the sum is over organ locations. Here β is the dose effect, α are the location effects and γ are other covariate effects. The model can be reparametrized to HR=$\sum(1+exp(\xi)d_l)exp(\alpha_l + X^T\gamma)$ using `repar=TRUE`. In the original parametrization, β is constrained such that HR cannot be negative. There are different choices for the design used to estimate the parameters: mean organ dose, CCML, CL, and CCAL. Mean organ dose (`ccmethod='meandose'`) uses the mean of the supplied location doses and compares that mean dose between case and matched controls. The other choices (CCML, CL and CCAL) use the tumor location for the case and compare either only between patients (CCML), only within patients (CL) or both between and within patients (CCAL). CCML only compares the same location between patients, and hence cannot be used to estimate location effects. Similarly, CL compares within patients and cannot be used to estimate covariate effects other than dose, meaning `corrvars` should not be supplied for CL.

For one-dimensional models (i.e., mean dose or CCML without additional covariates), the Brent algorithm is used with a search interval (-10,log(`uplimBeta`)) when `repar=TRUE` and (L,`uplimBeta`) otherwise, where L is determined by the positivity constraint for HR. For other optimizations, the L-BFGS-B algorithm (with constraint `uplimBeta`) is used when `repar=FALSE`, and the unconstrained Nelder-Mead is used when `repar=TRUE`. For details refer to the function `optim`, also for `fitopt` settings. Note that when supplying `ndeps` to `fitopt`, a value needs to be specified for every free parameter in the model. For more flexibility in optimizion, use `linERRloglik` and optimize directly.

The jackknife procedure allows for filtering of the leave-one-out and leave-two-out estimates, which is important as the model can be unstable and produce extreme estimates. All estimates reaching the maximum number of iterations are excluded, as well as estimates larger than uplimBeta (if applicable). Further, the user can set a threshold for the square L2 norm of the score for an estimate (default .01), as well as an allowed value range for the β/ξ estimate itself. When the jackknife is run, the output object contains an element `details`, allowing the user to inspect the produced leave-one-out and leave-two-out estimates.

Value

Object with components `MLE` and `jackknife`. `MLE` has components:

`coef` estimated model coefficients

`sd` estimated standard deviation for all coefficient estimates `vcov` variance-covariance matrix for all estimates

`score` score in the MLE

`convergence` convergence code produced by the optimizer (for details refer to `optim`)

`message` convergence message produced by the optimizer

`dosepval` p-value for the LRT comparing the produced model with a model without dose effect. Note that the null model this is based on uses the same optimization algorithm used for the MLE, meaning one-dimensional Nelder-Mead is used when `repar=TRUE` and the full model has 2 free parameters (see details)

`profCI` the 95% profile likelihood confidence interval. In some cases one or both of the bounds of the CI cannot be obtained automatically. In that case, it is possible to use the `proflik` function that is an output of `linearERRfit` directly. Note: the same optimization algorithm that was used for the MLE will be used, even if this model only has one parameter (see details)

`jackknife` has components `firstorder` and `secondorder`. Both of these have components:

`coef` the jackknife-corrected coefficient estimates

`details` data frame with information on leave-one-out or leave-two-out estimates, with columns:

- `set` or `set1` and `set2`, the left-out set(s)
- `included`, a 0/1 variable indicating whether this row was used to produce the corrected estimate
- `conv`, convergence code for each model produced by the optimizer
- `coef`, the leave-one-out or leave-two-out coefficient estimates
- `score`, the score in the leave-one-out or leave-two-out estimate

Note that the `details` for the second order jackknife only include leave-two-out estimates. To access leave-one-out estimates, use `details` for the first order jackknife.

Examples

```
data(linearERRdata1)

fitCCML <- linearERR(data=linearERRdata1, set=1, doses=2:6, status=8,
loc=7, corrvars=9, repar=FALSE, ccmethod="CCML", doJK1=TRUE)

fitCCML$MLE
fitCCML$jackknife$firstorder$coef
```

`linearERRdata1` *Matched case-control data*

Description

Simulated matched case-control data set. A data frame with columns for matched set number, dose information, second tumor location for the set's case, and a confounder C. The true value for β was 0.3, while α_2, ... , α_5 were equal to log(.25) and γ was log(2)

Usage

```
data(linearERRdata1)
```

Format

An object of class `data.frame` with 600 rows and 9 columns.

`linearERRfirth` *Derive the Firth-corrected estimate for the linear ERR model*

Description

Finds roots to the Firth-corrected score equations for the linear ERR model using a matched case-control study.

Usage

```
linearERRfirth(
  data,
  doses,
  set,
  status,
  loc,
  corrvars = NULL,
  repar = FALSE,
  ccmethod = "CCAL",
  initpars = NULL,
  lowerlim = NULL,
  upperlim = NULL,
  fitopt = list(maxit = 5000)
)
```

Arguments

`data` data frame containing matched case-control data, with a number of columns for doses to different locations, a column containing matched set numbers, a column containing the case's tumor location (value between 1 and the number of locations, with location x corresponding to the x-th column index in `doses`) and a column serving as a case-control indicator. Other covariates can also be included, in this case a parameter for each covariate column will be estimated. Hence factor variables need to be converted to dummy variables using `model.matrix`. If using `ccmethod='meandose'`, a column for tumor location is still required but in this case the column can be a vector of ones.

`doses` vector containing the indices of columns containing dose information.

`set` column index containing matched set numbers.

`status` column index containing case status.

`loc` column index containing the location of the matched set's case's second tumor.

`corrvars` vector containing the indices of columns containing variables to be corrected for. Not used with `ccmethod='CL'`

`repar` reparametrize to $\beta = exp(\xi)$? It is recommended to reparametrize when using CL or CCAL or when using additional covariates. Defaults to `FALSE`

`ccmethod` choice of method of analysis: one of meandose, CCML, CCAL or CL. Defaults to CCAL `initpars` initial values for parameters, default is 0 for all parameters. If supplying a different vector, use a vector with an initial value for all free parameters (β or ξ, one for each location effect (except the reference) when using CCAL or CCAL, and for each other covariate if applicable, in that order). This is different from `linearERR`, where all location effects have to be supplied regardless of the chosen `ccmethod`. Note that if `repar=TRUE`, the first initial value is used for ξ.

`lowerlim` lower bound for model parameters, in the same order as `initpars`. When not supplied, `-Inf` is used for all coefficients (except when `repar=FALSE`, see details). Note that when `repar=TRUE`, the first entry is the lower limit for ξ

`upperlim` upper bound for model parameters, in the same order as `initpars`. When not supplied, `Inf` is used for all coefficients. Note that when `repar=TRUE`, the first entry is the upper limit for ξ

`fitopt` list with options to pass to `control` argument of optimizer (see details)

Details

This function looks for roots of the Firth-corrected score functions.

The underlying model is HR=$\sum(1+\beta d_l)exp(\alpha_l+X^T\gamma)$, where the sum is over organ locations. Here β is the dose effect, α are the location effects and γ are other covariate effects. The model can be reparametrized to HR=$\sum(1+exp(\xi)d_l)exp(\alpha_l+X^T\gamma)$ using `repar=TRUE`. In the original parametrization, β is constrained such that HR cannot be negative. There are different choices for the design used to estimate the parameters: mean organ dose, CCML, CL, and CCAL. Mean organ dose (`ccmethod='meandose'`) uses the mean of the supplied location doses and compares that mean dose between case and matched controls. The other choices (CCML, CL and CCAL) use the tumor location for the case and compare either only between patients (CCML), only within patients (CL) or both between and within patients (CCAL). CCML only compares the same location between patients, and hence cannot be used to estimate location effects. Similarly, CL compares within patients and cannot be used to estimate covariate effects other than dose, meaning `corrvars` should not be supplied for CL. For this model, the Firth correction (Firth 1993) is used as a method for bias correction, or for obtaining an estimate when there is separation in the data.

To avoid using unstable multidimensional root finders, this function minimizes the square L2 norm of the modified score instead. This is done using the `optim` function. If desired, it is possible to use `linERRscore` and optimize or search for roots directly. For one-dimensional models (i.e., mean dose or CCML without additional covariates), the Brent algorithm is used with the user-supplied search interval (`lowerlim`,`upperlim`). Note that the choice for search interval is crucial as this determines convergence. For this reason, there is no default setting in this case. For other optimizations, the L-BFGS-B algorithm (with constraints `lowerlim` and `upperlim`) is used. For details refer to the function optim, also for `fitopt` settings. When `repar=FALSE`, if the lower bound for β is set too small, it is automatically changed according to the positivity constraint for HR.

It is advisable to interpret the results with caution. It was found that the modified score function sometimes has multiple roots, which makes setting initial values and search intervals crucial. It is recommended to try different settings for these inputs. Further, it seemed that reparametrizing improved the performance for multidimensional models.

Value

`optim` object with fit results.

References

David Firth, Bias reduction of maximum likelihood estimates, Biometrika, Volume 80, Issue 1, March 1993, Pages 27–38, https://doi.org/10.1093/biomet/80.1.27

Examples

```
data(linearERRdata1)

fitMLE <- linearERR(data=linearERRdata1,doses=2:6,set=1,status=8,loc=7,
corrvars=9,repar=TRUE,ccmethod="CCAL",profCI=FALSE)

fitfirth <- linearERRfirth(data=linearERRdata1,doses=2:6,set=1,status=8,loc=7,
corrvars=9,repar=TRUE,ccmethod="CCAL",initpars=fitMLE$MLE$coef)

data.frame(MLE=fitMLE$MLE$coef, Firth=fitfirth$par)
```

linearERRfit *Fit linear ERR model*

Description

Fits the linear ERR model on a dataset

Usage

```
linearERRfit(
  data,
  doses,
  set,
  status,
  loc,
  corrvars = NULL,
  repar = FALSE,
  ccmethod = "CCAL",
  initpars = rep(0, length(doses) + length(corrvars)),
  fitopt = list(maxit = 5000),
  fitNull = TRUE,
  useOld = FALSE,
  uplimBeta = 5
)
```

Arguments

`data` data frame containing matched case-control data, with a number of columns for doses to different locations, a column containing matched set numbers, a column containing the case's tumor location (value between 1 and the number of locations, with location x corresponding to the x-th column index in doses) and a column serving as a case-control indicator. Other covariates can also be included, in this case a parameter for each covariate column will be estimated. Hence factor variables need to be converted to dummy variables using `model.matrix`. If using `ccmethod='meandose'`, a column for tumor location is still required but in this case the column can be a vector of ones.

`doses` vector containing the indices of columns containing dose information.

`set` column index containing matched set numbers.

`status` column index containing case status.

`loc` column index containing the location of the matched set's case's second tumor.

`corrvars` vector containing the indices of columns containing variables to be corrected for.

`repar` reparametrize to $\beta = exp(\xi)$? Defaults to `FALSE`

`ccmethod` choice of method of analysis: one of meandose, CCML, CCAL or CL. Defaults to CCAL

`initpars` initial values for parameters, default is 0 for all parameters. If supplying a different vector, use a vector with an initial value for β or ξ, a 0 for the reference location, one for all of the other location effects and one for each other covariate (in that order). Note that if `repar=TRUE`, the initial value is used for ξ.

`fitopt` list with options to pass to `control` argument of optimizer

`fitNull` boolean: also fit model without dose effect? Defaults to `TRUE`. Note: the same optimization algorithm that was used for the MLE will be used for the null model, even if the null model only has one parameter (see details)

`useOld` if TRUE, a previous (slower) implementation of the log-likelihood function will be used. Defaults to `FALSE`

`uplimBeta` upper limit for $\beta = exp(\xi)$, default value 5. This is used for constraining the MLE estimation in some settings and for the jackknife inclusion criteria, and can be infinite except when Brent optimization is used (see help for `linearERR`)

Details

This is a stripped down version of `linearERR`, and should only be used when that function does not suffice. For more details refer to the help of `linearERR`.

Value

Object with components:

`fit` object produced by `mle2`

`nullfit` fit without dose effect produced by `mle2`

`proflik` profile likelihood: one-dimensional function of β or ξ. Note that the optimization used is the same as for the MLE, leading to one-dimensional Nelder-Mead optimization in certain cases (see details of `linearERR`)

See Also

linearERR

Examples

```
data(linearERRdata1)

fitmeandose <- linearERRfit(data=linearERRdata1, set=1, doses=2:6,
status=8, loc=7, corrvars=9, repar=FALSE, ccmethod="meandose")

fitCCML <- linearERRfit(data=linearERRdata1, set=1, doses=2:6,
status=8, loc=7, corrvars=9, repar=FALSE, ccmethod="CCML")

fitCCAL <- linearERRfit(data=linearERRdata1, set=1, doses=2:6,
status=8, loc=7, corrvars=9, repar=FALSE, ccmethod="CCAL")

fitCL <- linearERRfit(data=linearERRdata1, set=1, doses=2:6,
status=8, loc=7, corrvars=9, repar=FALSE, ccmethod="CL")

bbmle::coef(fitmeandose$fit, exclude.fixed=TRUE)
bbmle::coef(fitCCML$fit, exclude.fixed=TRUE)
bbmle::coef(fitCCAL$fit, exclude.fixed=TRUE)
bbmle::coef(fitCL$fit, exclude.fixed=TRUE)
```

linERRloglik *Negative log-likelihood*

Description

Compute the negative log-likelihood for the linear ERR model in a matched case-control dataset

Usage

```
linERRloglik(
  params,
  data,
  doses,
  set,
  status,
  loc,
  corrvars = NULL,
  ccmethod = "CCAL"
)
```

Arguments

params vector of parameter values (β, α_2, ... , α_L, γ_1, ... , γ_p) to evaluate the log-likelihood at

data data frame containing matched case-control data, with a number of columns for doses to different locations, a column containing matched set numbers, a column containing the case's tumor location (value between 1 and the number of locations, with location x corresponding to the x-th column index in doses) and a column serving as a case-control indicator. Other covariates can also be included, in this case a parameter for each covariate column will be estimated. Hence factor variables need to be converted to dummy variables using model.matrix. If using ccmethod='meandose', a column for tumor location is still required but in this case the column can be a vector of ones.

doses vector containing the indices of columns containing dose information, in the desired order.

set column index containing matched set numbers.

status column index containing case status.

loc column index containing the location of the matched set's case's second tumor.

`corrvars` vector containing the indices of columns containing variables to be corrected for.

`ccmethod` choice of method of analysis: one of meandose, CCML, CCAL or CL. Defaults to CCAL

Value

Minus log likelihood in params

Examples

```
data(linearERRdata1)

#log-likelihood in the truth
-linERRloglik(params=c(.3,rep(-1.386294,4),log(2)),
data=linearERRdata1,set=1, doses=2:6, status=8, loc=7, corrvars=9)

#log-likelihood in 0
-linERRloglik(params=c(0,rep(0,4),0),
data=linearERRdata1,set=1, doses=2:6, status=8, loc=7, corrvars=9)
```

`linERRloglikold` *Negative log-likelihood*

Description

Compute the negative log-likelihood for the linear ERR model in a matched case-control dataset. This is an outdated function, replaced by the substantially faster linERRloglik

Usage

```
linERRloglikold(
  params,
  data,
  doses,
  set,
  status,
  loc,
  corrvars = NULL,
  ccmethod = "CCAL"
)
```

Arguments

`params` vector of parameter values (β, α_2, ... , α_L, γ_1, ... , γ_p) to evaluate the log-likelihood at

`data` data frame containing matched case-control data, with a number of columns for doses to different locations, a column containing matched set numbers, a column containing the case's tumor location (value between 1 and the number of locations, with location x corresponding to the x-th column index in `doses`) and a column serving as a case-control indicator. Other covariates can also be included, in this case a parameter for each covariate column will be estimated. Hence factor variables need to be converted to dummy variables using `model.matrix`. If using `ccmethod='meandose'`, a column for tumor location is still required but in this case the column can be a vector of ones.

`doses` vector containing the indices of columns containing dose information, in the desired order.

`set` column index containing matched set numbers.

`status` column index containing case status.

`loc` column index containing the location of the matched set's case's second tumor.

`corrvars` vector containing the indices of columns containing variables to be corrected for.

`ccmethod` choice of method of analysis: one of meandose, CCML, CCAL or CL. Defaults to CCAL

Value

Minus log likelihood in params

Examples

```
data(linearERRdata1)

#log-likelihood in the truth
-linERRloglikold(params=c(.3,rep(log(.25),4),log(2)),
data=linearERRdata1,set=1, doses=2:6, status=8, loc=7, corrvars=9)

#log-likelihood in 0
-linERRloglikold(params=c(0,rep(0,4),0),
data=linearERRdata1,set=1, doses=2:6, status=8, loc=7, corrvars=9)
```

`linERRscore` *Compute the Firth-corrected score function*

Description

Compute the Firth-corrected score for a matched case-control dataset

Usage

```
linERRscore(
  params,
  data,
  doses,
  set,
  status,
  loc,
  ccmethod,
  corrvars = NULL,
  repar = FALSE
)
```

Arguments

`params` vector of parameters (β/ξ, α_2, ... , α_L, γ_1, ... , γ_p) for which to compute the modified score. Note that when `repar=TRUE`, ξ needs to be supplied

`data` data frame containing matched case-control data, with a number of columns for doses to different locations, a column containing matched set numbers, a column containing the case's tumor location (value between 1 and the number of locations, with location x corresponding to the x-th column index in `doses`) and a column serving as a case-control indicator. Other covariates can also be included, in this case a parameter for each covariate column will be estimated. Hence factor variables need to be converted to dummy variables using `model.matrix`. If using `ccmethod='meandose'`, a column for tumor location is still required but in this case the column can be a vector of ones.

`doses` vector containing the indices of columns containing dose information.

`set` column index containing matched set numbers.

`status` column index containing case status.

`loc` column index containing the location of the matched set's case's second tumor.

`ccmethod` choice of method of analysis: one of meandose, CCML, CCAL or CL. Defaults to CCAL

`corrvars` vector containing the indices of columns containing variables to be corrected for.

`repar` reparametrize to $\beta = exp(\gamma)$? Note that this only affects the Firth modified score, not the original score. Defaults to FALSE.

Value

List object with components:

`U` the original score

`A` the modification to the score

The Firth corrected score is equal to `U+A`.

Examples

```
data(linearERRdata1)

# score in the truth
score1 <- linERRscore(params=c(.3,rep(-1.386294,4),log(2)),
data=linearERRdata1,set=1, doses=2:6, status=8, loc=7, corrvars=9,ccmethod="CCAL")

# score in the truth, reparametrized
score1_repar <- linERRscore(params=c(.3,rep(-1.386294,4),log(2)),
data=linearERRdata1,set=1, doses=2:6, status=8, loc=7, corrvars=9,ccmethod="CCAL", repar=TRUE)

score1$U # Original score
score1$U+score1$A # Firth score

score1_repar$U+score1_repar$A # Firth score under reparametrization

# score in 0
score2 <- linERRscore(params=c(0,rep(0,4),0),
data=linearERRdata1,set=1, doses=2:6, status=8, loc=7, corrvars=9, ccmethod="CCAL")
```

```
score2$U # Original score
score2$U+score2$A # Firth score
```

Index