Package 'coxphSGD'

July 5, 2017

Type Package

Title Stochastic Gradient Descent log-Likelihood Estimation in Cox Proportional Hazards Model

Version 0.2.1

Date 2017-07-05

Maintainer Marcin Kosinski <m.p.kosinski@gmail.com>

Description Estimate coefficients of Cox proportional hazards model using stochastic gradient descent algorithm for batch data.

License GPL-2

Depends R (>= 3.3.0), survival

URL https://github.com/MarcinKosinski/coxphSGD/blob/master/README.md

BugReports https://github.com/MarcinKosinski/coxphSGD/issues

RoxygenNote 6.0.1

NeedsCompilation no

Author Marcin Kosinski [aut, cre], Przemyslaw Biecek [ctb]

Repository CRAN

Date/Publication 2017-07-05 11:43:29 UTC

R topics documented:

coxphSGD *Stochastic Gradient Descent log-likelihood Estimation in Cox Proportional Hazards Model*

Description

coxphSGD estimates coefficients using stochastic gradient descent algorithm in Cox proportional hazards model.

Usage

```
coxphSGD(formula, data, learn.rates = function(x) {     1/x },
  beta.zero = 0, epsilon = 1e-05, max.iter = 500, verbose = FALSE)
```

Arguments

formula	a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the Surv function.
data	a list of batch data.frames in which to interpret the variables named in the formula. See Details.
learn.rates	a function specifing how to define learning rates in steps of the algorithm. By default the f(t)=1/t is used, where t is the number of algorithm's step.
beta.zero	a numeric vector (if of length 1 then will be replicated) of length equal to the number of variables after using formula in the model.matrix function
epsilon	a numeric value with the stop condition of the estimation algorithm.
max.iter	numeric specifing maximal number of iterations.
verbose	whether to cat the number of the iteration

Details

A data argument should be a list of data.frames, where in every batch data.frame there is the same structure and naming convention for explanatory and survival (times, censoring) variables. See Examples.

Note

If one of the conditions is fullfiled (j denotes the step number)

- $||\beta_{j+1} - \beta_j|| <$ epsilon parameter for any j
- $j > max.iter$

the estimation process is stopped.

Author(s)

Marcin Kosinski, <m.p.kosinski@gmail.com>

Examples

```
library(survival)
set.seed(456)
x <- matrix(sample(0:1, size = 20000, replace = TRUE), ncol = 2)
head(x)
dCox <- dataCox(10^4, lambda = 3, rho = 2, x,
                beta = c(2,2), cens.rate = 5)
batch_id <- sample(1:90, size = 10^4, replace = TRUE)
dCox_split <- split(dCox, batch_id)
results <-
  coxphSGD(formula     = Surv(time, status) ~ x.1+x.2,
           data        = dCox_split,
           epsilon     = 1e-5,
           learn.rates = function(x){1/(100*sqrt(x))},
           beta.zero   = c(0,0),
           max.iter    = 10*90)
coeff_by_iteration <-
  as.data.frame(
    do.call(
      rbind,
      results$coefficients
    )
  )
head(coeff_by_iteration)
```

`dataCox` *Cox Proportional Hazards Model Data Generation From Weibull Distribution*

Description

Function `dataCox` generaters random survivaldata from Weibull distribution (with parameters `lambda` and `rho` for given input `x` data, model coefficients `beta` and censoring rate for censoring that comes from exponential distribution with parameter `cens.rate`.

Usage

```
dataCox(n, lambda, rho, x, beta, cens.rate)
```

Arguments

`n`	Number of observations to generate.
`lambda`	lambda parameter for Weibull distribution.
`rho`	rho parameter for Weibull distribution.
`x`	A data.frame with an input data to generate the survival times for.
`beta`	True model coefficients.
`cens.rate`	Parameter for exponential distribution, which is responsible for censoring.

Details

For each observation true survival time is generated and a censroing time. If censoring time is less then survival time, then the survival time is returned and a status of observations is set to `0` which means the observation had censored time. If the survival time is less than censoring time, then for this observation the true survival time is returned and the status of this observation is set to `1` which means that the event has been noticed.

Value

A `data.frame` containing columns:

- `id` an integer.
- `time` survival times.
- `status` observation status (event occured (1) or not (0)).
- `x` a `data.frame` with an input data to generate the survival times for.

References

http://onlinelibrary.wiley.com/doi/10.1002/sim.2059/abstract

`Generating survival times to simulate Cox proportional hazards models`, 2005 by Ralf Bender, Thomas Augustin, Maria Blettner.

Examples

```
## Not run:
x <- matrix(sample(0:1, size = 20000, replace = TRUE), ncol = 2)
dataCox(10^4, lambda = 3, rho = 2, x,
beta = c(1,3), cens.rate = 5) -> dCox

## End(Not run)
```

Index