

coxphSGD

December 9, 2015

coxphSGD	<i>Stochastic Gradient Descent log-likelihood estimation in Cox proportional hazards model</i>
----------	--

Description

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

Usage

```
coxphSGD(formula, data, learningRates = function(x) {  
  1/x  
}, beta_0 = 0, epsilon = 1e-05, max.iter = 500)
```

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.
learningRates	a function specifying 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_0	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.

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 fulfilled (j denotes the step number)

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

the estimation process is stopped.

Author(s)

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

Examples

```
library(survival)
## Not run:
coxphSGD(Surv(time, status) ~ ph.ecog + age, data = split(lung, sample(1:4,
  size = 228, replace = TRUE)))

## End(Not run)
```

dataCox

Cox Proportional Hazards Model Data Generation From Weibull Distribution

Description

Function dataCox generates random survival data from Weibull distribution (with parameters λ and ρ for given input x data, model coefficients β and censoring rate for censoring that comes from exponential distribution with parameter censRate).

Usage

```
dataCox(N, lambda, rho, x, beta, censRate)
```

Arguments

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

Details

For each observation true survival time is generated and a censoring time. If censoring time is less than 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.

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)
dCox <- dataCox(10^4, lambda = 3, rho = 2, x, beta = c(1, 3), censRate = 5)

## End(Not run)
```

simulateCoxSGD	<i>Optimize Partial Log-Likelihood Function For Cox Propotional Hazards Model Using Stochastic Gradient Descent</i>
----------------	---

Description

Function simulateCoxSGD splits input dCox data on 10, 30, 60, 90, 120 and 200 groups and for each split it uses [coxphSGD](#) function to generate estimates for Cox Proportional Hazards Model with stochastic gradient descent optimization.

Usage

```
simulateCoxSGD(dCox = dCox, learningRates = function(x) {
  1/x
}, epsilon = 0.001, beta_0 = c(0, 0), max.iter = 100)
```

Arguments

dCox	Input data.frame containing survival times (columnnd should be named time) and status (columnnd should be named status). So far this function only supports 2 explanatory variable (that should be named x1 and x2).
learningRates	Parameter passed to coxphSGD .
epsilon	Parameter passed to coxphSGD .
beta_0	Parameter passed to coxphSGD .
max.iter	Parameter passed to coxphSGD to multiple the maximal iterations by the rows number.

Examples

```
## Not run:
x <- matrix(sample(0:1, size = 20000, replace = TRUE), ncol = 2)
dCox <- dataCox(10^4, lambda = 3, rho = 2, x, beta = c(1, 3), censRate = 5)
d2ggplot <- simulateCoxSGD(dCox, learningRates = function(x) {
  1/(100 * sqrt(x))
}, max.iter = 10, epsilon = 1e-05)

## End(Not run)
```