Package ‘transmdl’

October 14, 2022

Type Package

Title Semiparametric Transformation Models

Version 0.1.0

Maintainer Fengyu Wen <Wenfy1207@qq.com>

Description To make the semiparametric transformation models easier to apply in real studies,
we introduce this R package, in which the MLE in transformation models via
an EM algorithm proposed by Zeng D, Lin DY(2007) <doi:10.1111/j.1369-7412.2007.00606.x>
and adaptive lasso method in transformation models proposed by Liu XX, Zeng
D(2013) <doi:10.1093/biomet/ast029> are implemented.
C++ functions are used to compute complex loops. The coefficient vector and
cumulative baseline hazard function can be estimated, along with the
corresponding standard errors and P values.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.2

Imports graphics, Rcpp, statmod, stats, survival

LinkingTo Rcpp, RcppEigen

Suggests MASS

NeedsCompilation yes

Author Fengyu Wen [aut, cre],
Baosheng Liang [aut]

Repository CRAN

Date/Publication 2021-10-14 11:50:02 UTC

R topics documented:

EM_est 2
generate_data 3
trans_alasso 4
trans_m 6

Index **9**

EM_est *Estimate parameters and hazard function via EM algorithm.*

Description

Estimate the vector of parameters for baseline covariates β and baseline cumulative hazard function $\Lambda(\cdot)$ using the expectation-maximization algorithm. $\Lambda(t)$ is estimated as a step function with jumps only at the observed failure times. Typically, it would only be used in a call to `trans.m` or `Simu`.

Usage

```
EM_est(Y, X, delta, alpha, Q = 60, EM_itmax = 250)
```

Arguments

`Y`	observed event times
`X`	design matrix
`delta`	censoring indicator. If Y_i is censored, `delta`=0. If not, `delta`=1.
`alpha`	parameter in transformation function
`Q`	number of nodes and weights in Gaussian quadrature. Defaults to 60.
`EM_itmax`	maximum iteration of EM algorithm. Defaults to 250.

Value

a list containing

`beta_new`	estimator of β
`Lamb_Y`	estimator of $\Lambda(Y)$
`lamb_Y`	estimator of $\lambda(Y)$
`lamb_Ydot`	estimator of $\lambda(Y')$
`Y_eq_Yhat`	a matrix used in `trans.m` and `Simu`
`Y_geq_Yhat`	a matrix used in `trans.m` and `Simu`

References

Abramowitz, M., and Stegun, I.A. (1972). Handbook of Mathematical Functions (9th ed.). Dover Publications, New York.

Evans, M. and Swartz, T. (2000). Approximating Integrals via Monte Carlo and Deterministic Methods. Oxford University Press.

Liu, Q. and Pierce, D.A. (1994). A note on Gauss-Hermite quadrature. Biometrika 81: 624-629.

Examples

```
gen_data = generate_data(200, 1, 0.5, c(-0.5, 1))
```

```
delta = gen_data$delta
Y = gen_data$Y
X = gen_data$X
EM_est(Y, X, delta, alpha = 1)$beta_new - c(-0.5, 1)
```

---

generate_data *Generate data for simulation.*

Description

Generate observed event times, covariates and other data used for simulations in the paper.

Usage

```
generate_data(n, alpha, rho, beta_true, now_repeat = 1)
```

Arguments

n	number of subjects
alpha	parameter in transformation function
rho	parameter in baseline cumulative hazard function $\Lambda(t) = \rho log(1 + t)$ assumed in simulation
beta_true	parameter β
now_repeat	number of duplication of simulation

Details

The survival function for t of the ith observation takes the form

$$S_i(t|X_i) = \exp\left\{-H\{\Lambda(t)\exp(\beta^T X_i)\}\right\}.$$

The failure time T_i can be generated by

$$T_i = \left\{ \begin{array}{ll} \exp\{\frac{U^{-\alpha}-1}{\alpha\rho\exp\{\beta^T X_i\}}\} - 1 & \alpha > 0, \\ \exp\{\frac{-log(U)}{\rho\exp\{\beta^T X_i\}}\} - 1, & \alpha = 0. \end{array} \right\}$$

Value

a list containing

X	design matrix
beta_X	$X \cdot \beta^T$
Y	observed event time

References

Abramowitz, M., and Stegun, I.A. (1972). Handbook of Mathematical Functions (9th ed.). Dover Publications, New York. +- Evans, M. and Swartz, T. (2000). Approximating Integrals via Monte Carlo and Deterministic Methods. Oxford University Press.

Liu, Q. and Pierce, D.A. (1994). A note on Gauss-Hermite quadrature. Biometrika 81: 624-629.

Examples

```
generate_data(200,0.5,1,c(0.5,-1))
```

`trans_alasso` *Adaptive LASSO for Semiparametric Transformation Models.*

Description

Select the important variables in semiparametric transformation models for right censored data using adaptive lasso.

Usage

```
trans_alasso(Z, Y, delta_i, r, lamb_vec, solu = TRUE)
```

Arguments

`Z`	the baseline covariates
`Y`	observed event times
`delta_i`	censoring indicator. If Y is censored, `delta_i`=0. If not, `delta_i`=1.
`r`	parameter in transformation function
`lamb_vec`	the grad of the tuning parameter λ
`solu`	determines whether the solution path will be plotted. The default is TRUE.

Details

The initial value of the coefficient β used as the adapting weights is EM estimator, which is computed by the function `EM_est`. The tuning parameter λ is data-dependent and we select it using generalized crossvalidation. There may be some errors for small λ, in which case the λ and the number of adaptive lasso iteration are recorded in the `skip_para`.

Value

a list containing

beta_res	the estimated β with the selected tuning parameter λ
GCV_res	the value of GCV with the selected tuning parameter λ
lamb_res	the selected tuning parameter λ
beta_all	estimated β with all tuning parameters
CSV_all	value of GCV with all tuning parameters
skip_para	a list containing the λ and the number of adaptive lasso iteration when adaptive lasso doesn't work.

References

Xiaoxi, L. , & Donglin, Z. . (2013). Variable selection in semiparametric transformation models for right-censored data. Biometrika(4), 859-876.

Examples

```
 if(!requireNamespace("MASS", quietly = TRUE))
 {stop("package MASS needed for this example. Please install it.")}

 gen_lasdat = function(n,r,rho,beta_true,a,b,seed=66,std = FALSE)
{

  set.seed(seed)
  beta_len = length(beta_true)
  beta_len = beta_len
  sigm = matrix(0, nrow = beta_len, ncol = beta_len)
  for(i in 1:(beta_len-1))
  {
    diag(sigm[1:(beta_len+1-i),i:beta_len]) = rho^(i-1)
  }
  sigm[1,beta_len] = rho^(beta_len-1)
  sigm[lower.tri(sigm)] = t(sigm)[lower.tri(sigm)]

  Z = MASS::mvrnorm(n, mu = rep(0, beta_len), Sigma = sigm)
  beta_Z.true = c(Z %*% beta_true)
  U = runif(n)
  if(r>0)
  {
    t = ((U^(-r)-1)/(a*r*exp(beta_Z.true)))^(1/b)
  }else if(r == 0)
  {
    t = (-log(U)/(a*exp(beta_Z.true)))^(1/b)
    #t = (exp(-log(U)/(0.5 * exp(beta_Z.true))) - 1)
  }
  C = runif(n,0,8)
  Y = pmin(C,t)
  delta_i = ifelse( C >= t, 1, 0)
```

```
  if(std)
  {
    Z = apply(Z,2,normalize)
  }
  return(list(Z = Z, Y = Y, delta_i = delta_i,censor = mean(1-delta_i)))
}

now_rep=1
dat = gen_lasdat(100,1,0.5,c(0.3,0.5,0.7,0,0,0,0,0,0,0),2,5,seed= 6+60*now_rep,std = FALSE)
Z = dat$Z
Y = dat$Y
delta_i = dat$delta_i

tra_ala = trans_alasso(Z,Y,delta_i,lamb_vec = c(5,7),r=1)
tra_ala$GCV_res
tra_ala$beta_res
tra_ala$lamb_res
```

`trans_m` *Regression Analysis of Right-censored Data using Semiparametric Transformation Models.*

Description

This function is used to conduct the regression analysis of right-censored data using semiparametric transformation models. It calculates the estimators, standard errors and p values. A plot of estimated baseline cumulative hazard function and confidence intervals can be produced.

Usage

```
trans_m(
  X,
  delta,
  Y,
  plot.Lamb = TRUE,
  alpha = seq(0.1, 1.1, by = 0.1),
  trsmodel = TRUE,
  EM_itmax = 250,
  show_res = TRUE
)
```

Arguments

`X`	design matrix
`delta`	censoring indicator. If Y_i is censored, `delta`=0. If not, `delta`=1.
`Y`	observed event times

`plot.Lamb`	If TRUE, plot the estimated baseline cumulative hazard function and confidence intervals. The default is TRUE.
`alpha`	parameter in transformation function. Generally, α can not be observed in medical applications. In that situation, `alpha` indicates the scale of choosing α. The default is $(0.1, 0.2, ..., 1.1)$. If α is known, `alpha` indicates the true value of α.
`trsmodel`	logical value indicating whether to implement transformation models. The default is TRUE.
`EM_itmax`	maximum iteration of EM algorithm. Defaults to 250.
`show_res`	show results after `trans_m`.

Details

If α is unknown, we firse set $\alpha =$`alpha`. Then, for each α, we estimate the parameters and record the value of observed log-likelihood function. The α that maximizes the observed log-likelihood function and the corresponding $\hat{\beta}$ and $\hat{\Lambda}(\cdot)$ are chosen as the best estimators. Nonparametric maximum likelihood estimators are developed for the regression parameters and cumulative intensity functions of these models based on censored data.

Value

a list containing

`beta.est`	estimators of β
`SE.beta`	standard errors of the estimated β
`SE.Ydot`	standard errors of the estimated $\Lambda(Y')$
`Ydot`	vector of sorted event times with duplicate values removed
`Lamb.est`	estimated baseline cumulative hazard
`lamb.est`	estimated jump sizes of baseline cumulative hazard function
`choose.alpha`	the chosen α
`Lamb.upper`	upper confidence limits for the estimated baseline cumulative hazard function
`Lamb.lower`	lower confidence limits for the estimated baseline cumulative hazard function
`p.beta`	P values of estimated β
`p.Lamb`	P values of estimated baseline cumulative hazard
p.beta	

References

Cheng, S.C., Wei, L.J., and Ying, Z. (1995). Analysis of transformation models with censored data. Biometrika 82, 835-845.

Zeng, D. and Lin, D.Y. (2007). Maximum likelihood estimation in semiparametric regression models with censored data. J. R. Statist. Soc. B 69, 507-564.

Abramowitz, M., and Stegun, I.A. (1972). Handbook of Mathematical Functions (9th ed.). Dover Publications, New York.

Evans, M. and Swartz, T. (2000). Approximating Integrals via Monte Carlo and Deterministic Methods. Oxford University Press.

Liu, Q. and Pierce, D.A. (1994). A note on Gauss-Hermite quadrature. Biometrika 81, 624-629.

Louis, T. (1982). Finding the Observed Information Matrix when Using the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 44(2), 226-233.

See Also

EM_est

Examples

```
gen_data = generate_data(200, 1, 0.5, c(-0.5,1))
delta = gen_data$delta
Y = gen_data$Y
X = gen_data$X
res.trans = trans_m(X, delta,  Y, plot.Lamb = TRUE, show_res = FALSE)
```

Index

EM_est, 2, *8*

generate_data, 3

trans_alasso, 4
trans_m, 6