Package: cosso (via r-universe)

August 23, 2024

Version 2.1-2

Date 2023-03-07

Title Fit Regularized Nonparametric Regression Models Using COSSO Penalty

Description The COSSO regularization method automatically estimates and selects important function components by a soft-thresholding penalty in the context of smoothing spline ANOVA models. Implemented models include mean regression, quantile regression, logistic regression and the Cox regression models.

License GPL (>= 2)

Depends quadprog, Rglpk, parallel, glmnet

NeedsCompilation no

URL https://arxiv.org/abs/math/0702659

Author Hao Helen Zhang [aut, cph], Chen-Yen Lin [aut, cph], Isaac Ray [cre, ctb]

Maintainer Isaac Ray <null@stat.tamu.edu>

Repository CRAN

Date/Publication 2023-03-08 09:30:09 UTC

Contents

BUPA *BUPA Liver Disorder Data*

Description

345 male patients' blood test result and liver disorder status.

Usage

```
data(BUPA)
```

Format

CLASS	0: no liver disorder 1: liver disorder
MCV	mean corpuscular volume. minimum 65 and maximum 103 in original scale.
ALKPHOS	alkaline phosphotase. minimum 23 and maximum 138 in original scale.
SGPT	alamine aminotransferase. minimum 4 and maximum 155 in original scale.
SGOT	aspartate aminotransferase. minimum 5 and maximum 82 in original scale.
GAMMAGT	gamma-glutamyl transpeptidase. minimum 5 and maximum 297 in original scale.
DRINKS	number of alcoholic beverages drunk per day. minimum 0 and maximum 20 in original scale.

Details

All the variables, except for the response, have been scaled to [0,1] interval. To transform back to the original scale, use the formula:

$$x = min + (max - min) * z.$$

Source

Richard S. Forsyth at BUPA Medical Research Ltd.

cosso *Fit a generalized nonparametric model with cosso penalty*

Description

A comprehensive method for fitting various type of regularized nonparametric regression models using cosso penalty. Fits mean, logistic, Cox and quantile regression.

Usage

```
cosso(x,y,tau,family=c("Gaussian","Binomial","Cox","Quantile"),wt=rep(1,ncol(x)),
      scale=FALSE,nbasis,basis.id,cpus)
```

Arguments

x	input matrix; the number of rows is sample size, the number of columns is the data dimension. The range of input variables is scaled to [0,1] for continuous variables. Variables with less than 7 unique values will be considered as discrete variable.
y	response vector. Quantitative for `family="Gaussian"` or `family="Quantile"`. For `family="Binomial"` should be a vector with two levels. For `family="Cox"`, y should be a two-column matrix (or data frame) with columns named 'time' and 'status'
tau	the quantile to be estimated, a number strictly between 0 and 1. Argument required when `family="Quantile"`.
family	response type. Abbreviations are allowed.
wt	weights for predictors. Default is `rep(1,ncol(x))`
scale	if `TRUE`, continuous predictors will be rescaled to [0,1] interval. Default is `FALSE`.
nbasis	number of "knots" to be selected. Ignored when `basis.id` is provided.
basis.id	index designating selected "knots". Argument is not valid for `family="Quantile"`.
cpus	number of available processor units. Default is `1`. If `cpus>=2`, parallelize task using "parallel" package. Recommended when either sample size or number of covariates is large. Argument is only valid for `family="Cox"` or `family="Quantile"`.

Details

In the SS-ANOVA model framework, the regression function is assumed to have an additive form

$$\eta(x) = b + \sum_{j=1}^{p} \eta_j(x^{(j)}),$$

where b denotes intercept and η_j denotes the main effect of the j-th covariate.

For `"Gaussian"` response, the mean function is estimated by minimizing the objective function:

$$\sum_i (y_i - \eta(x_i))^2/nobs + \lambda_0 \sum_{j=1}^{p} \theta_j^{-1} w_j^2 ||\eta_j||^2, s.t. \sum_{j=1}^{p} \theta_j \le M.$$

For `"Binomial"` response, the log-odd function is estimated by minimizing the objective function:

$$-log - likelihood/nobs + \lambda_0 \sum_{j=1}^{p} \theta_j^{-1} w_j^2 ||\eta_j||^2, s.t. \sum_{j=1}^{p} \theta_j \le M.$$

For `"Quantile"` regression model, the quantile function, is estimated by minimizing the objective function:

$$\sum_i \rho_\tau(y_i - \eta(x_i))/nobs + \lambda_0 \sum_{j=1}^{p} \theta_j^{-1} w_j^2 ||\eta_j||^2, s.t. \sum_{j=1}^{p} \theta_j \le M.$$

For "Cox" regression model, the log-relative hazard function is estimated by minimizing the objective function:

$$-log - PartialLikelihood/nobs + \lambda_0 \sum_{j=1}^{p} \theta_j^{-1} w_j^2 ||\eta_j||^2, s.t. \sum_{j=1}^{p} \theta_j \leq M.$$

For identifiability sake, the intercept term in Cox model is absorbed into basline hazard, or equivalently set $b = 0$.

For large data sets, we can reduce the computational load of the optimization problem by selecting a subset of the observations of size *nbais* as "knots", which reduces the dimension of the kernel matrices from *nobs* to *nbasis*. Unless specified via `basis.id` or `nbasis`, the default number of "knots" is max(40,12*nobs^(2/9)) for "Gaussian" and "Binomial" and max(35,11 * nobs^(2/9)) for "Cox".

The weights can be specified based on either user's own discretion or adaptively computed from initial function estimates. See Storlie et al. (2011) for more discussions. One possible choice is to specify the weights as the inverse L_2 norm of initial function estimator, see `SSANOVAwt`.

Value

An object with S3 class "cosso".

`y`	the response vector.
`x`	the input matrix.
`Kmat`	a three-dimensional array containing kernel matrices for each input variables.
`wt`	weights for predictors.
`family`	type of regression model.
`basis.id`	indices of observations used as "knots".
`cpus`	number of cpu units used. Will be returned if `family="Cox"` or `family="Quantile"`.
`tau`	the quantile to be estimated. Will be returned if `family="Quantile"`.
`tune`	a list containing preliminary tuning result and L2-norm.

Author(s)

Hao Helen Zhang and Chen-Yen Lin

References

Lin, Y. and Zhang, H. H. (2006) "Component Selection and Smoothing in Smoothing Spline Analysis of Variance Models", Annals of Statistics, **34**, 2272–2297.

Leng, C. and Zhang, H. H. (2006) "Model selection in nonparametric hazard regression", Nonparametric Statistics, **18**, 417–429.

Zhang, H. H. and Lin, Y. (2006) "Component Selection and Smoothing for Nonparametric Regression in Exponential Families", Statistica Sinica, **16**, 1021–1041.

Storlie, C. B., Bondell, H. D., Reich, B. J. and Zhang, H. H. (2011) "Surface Estimation, Variable Selection, and the Nonparametric Oracle Property", Statistica Sinica, **21**, 679–705.

See Also

plot.cosso, predict.cosso, tune.cosso

Examples

```
## Gaussian
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*9,0,1),nc=9))
y=x[,1]+sin(2*pi*x[,2])+5*(x[,4]-0.4)^2+rnorm(200,0,1)
G.Obj=cosso(x,y,family="Gaussian")
plot.cosso(G.Obj,plottype="Path")

## Not run:
## Use all observations as knots
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*9,0,1),nc=9))
y=x[,1]+sin(2*pi*x[,2])+5*(x[,4]-0.4)^2+rnorm(200,0,1)
G.Obj=cosso(x,y,family="Gaussian",nbasis=200)
## Clean up
rm(list=ls())

## Binomial
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*9,0,1),nc=9))
trueProb=1/(1+exp(-x[,1]-sin(2*pi*x[,2])-5*(x[,4]-0.4)^2))
y=rbinom(200,1,trueProb)

B.Obj=cosso(x,y,family="Bin")
## Clean up
rm(list=ls())

## Cox
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*9,0,1),nc=9))
hazard=x[,1]+sin(2*pi*x[,2])+5*(x[,4]-0.4)^2
surTime=rexp(200,exp(hazard))
cenTime=rexp(200,exp(-hazard)*runif(1,4,6))
y=cbind(time=apply(cbind(surTime,cenTime),1,min),status=1*(surTime<cenTime))
C.obj=cosso(x,y,family="Cox",cpus=1)

## Try parallel computing
C.obj=cosso(x,y,family="Cox",cpus=4)

## Clean up
rm(list=ls())

## Quantile
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*7,0,1),nc=7))
y=x[,1]+sin(2*pi*x[,2])+5*(x[,4]-0.4)^2+rt(200,3)
Q.obj=cosso(x,y,0.3,family="Quan",cpus=1)
```

```
## Try parallel computing
Q.obj=cosso(x,y,0.3,family="Quan",cpus=4)

## End(Not run)
```

---

ozone *Ozone pollution data in Los Angels, 1976*

Description

This is the ozone data used in Breiman and Friedman (1985). This dataset contains 330 observations, and each observation is a daily measurement.

Usage

```
data(ozone)
```

Format

ozone	Ozone reading
temp	Temperature (degree C). minimum 25 and maximum 93 in original scale.
invHt	Inversion base height (feet). minimum 111 and maximum 5000 in original scale.
press	Pressure gradient (mm Hg). minimum -69 and maximum 107 in original scale.
vis	Visibility (miles). minimum 0 and maximum 350 in original scale.
milPress	500 millibar pressure height (m). minimum 5320 and maximum 5950 in original scale.
hum	Humidity (percent). minimum 19 and maximum 93.
invTemp	Inversion base temperature (degrees F). minimum -25 and maximum 332 in original scale.
wind	Wind speed (mph). minimum 0 and maximum 21 in original scale.

Details

All the variables, except for the response, have been scaled to [0,1] interval. To transform back to the original scale, use the formula:

$$x = min + (max - min) * z.$$

Source

Breiman, L. and Friedman, J. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation", Journal of the American Statistical Association, **80**, 580–598.

plot.cosso *Plot method for COSSO object*

Description

Plot L_2 norm solution path or main effects of selected functional components

Usage

```
## S3 method for class 'cosso'
plot(x,M,plottype =c("Path","Functionals"),eps=1e-7,...)
```

Arguments

x	a cosso object.
M	a smoothing parameter value. Argument required when plottype="Functionals".
plottype	either Path (default) or Functionals. The Path plot shows the L_2 norm path for each functional component as a function of smoothing parameter M. The Functional plot shows the estimated functional components for each input variable at a particular smoothing parameter M. Abbreviations are allowed.
eps	an effective zero, default is 1e-7.
...	additional arguments for plot generic.

Value

NULL

Author(s)

Hao Helen Zhang and Chen-Yen Lin

See Also

predict.cosso

Examples

```
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*7,0,1),nc=7))
trueProb=1/(1+exp(-x[,1]-sin(2*pi*x[,2])-5*(x[,4]-0.4)^2))
y=rbinom(200,1,trueProb)

B.Obj=cosso(x,y,family="Bin")
plot.cosso(B.Obj,plottype="Path")
plot.cosso(B.Obj,M=2,plottype="Func")
```

`predict.cosso` *Make predictions or extract coefficients from a cosso model*

Description

Make prediction for future observations or extract the model parameters at a particular smoothing parameter.

Usage

```
## S3 method for class 'cosso'
predict(object,xnew,M,type=c("fit","coefficients","nonzero"),eps=1e-7,...)
```

Arguments

`object`	a cosso object.
`xnew`	matrix of new values for x at which predictions are to be made. Object must be a matrix and have the same dimension as the training design points. Continuous variable will also have to be scaled to [0,1] interval.
`M`	a smoothing parameter value. M should be taken between 0 and p. If not provided, a cross-validation procedure will be carried out to select an appropriate value.
`type`	if `type="fit"` (default), fitted values will be returned. If `type="coefficients"`, model coefficients will be returned. Abbreviations are allowed.
`eps`	an effective zero, default is `1e-7`
`...`	additional arguments for predict function.

Value

The object returned depends on type.

When `type="fit"`, predicted

$$eta$$

function value will be given at the new design points `xnew`.

When `type="coefficients"`, three sets of coefficients will be returned.

`Intercept`	the estimated intercept. If `family="Cox"`, the intercept is zero.
`coefs`	the estimated coefficients for kernel representers.
`theta`	the estimated scale parameters for each functional component.

When `type="nonzero"`, a list of the indices of the nonconstant functional components will be returned.

Author(s)

Hao Helen Zhang and Chen-Yen Lin

See Also

plot.cosso

Examples

```
## Gaussian
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*7,0,1),nc=7))
y=x[,1]+sin(2*pi*x[,2])+5*(x[,4]-0.4)^2+rnorm(200,0,1)
G.Obj=cosso(x,y,family="Gaussian")
predict.cosso(G.Obj,M=2,type="nonzero")
predict.cosso(G.Obj,xnew=x[1:3,],M=2,type="fit")
## Clean up
rm(list=ls())

## Not run:
## Binomial
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*9,0,1),nc=9))
trueProb=1/(1+exp(-x[,1]-sin(2*pi*x[,2])-5*(x[,4]-0.4)^2))
y=rbinom(200,1,trueProb)

B.Obj=cosso(x,y,family="Bin")
f.hat=predict.cosso(B.Obj,xnew=x,M=2,type="fit")
prob.hat=1/(1+exp(-f.hat))
## Clean up
rm(list=ls())

## End(Not run)
```

SSANOVAwt *Compute adaptive weights by fitting a SS-ANOVA model*

Description

A preliminary estimate $\tilde{\eta}$ is first obtained by fitting a smoothing spline ANOVA model, and then use the inverse L_2-norm, $||\tilde{\eta}_j||^{-\gamma}$, as the initial weight for the j-th functional component.

Usage

```
SSANOVAwt(x,y,tau,family=c("Gaussian","Binomial","Cox","Quantile"),mscale=rep(1,ncol(x)),
          gamma=1,scale=FALSE,nbasis,basis.id,cpus)
```

Arguments

x	input matrix; the number of rows is sample size, the number of columns is the data dimension. The range of input variables is scaled to [0,1] for continuous variables.

y	response vector. Quantitative for family="Gaussian" or family="Quantile". For family="Binomial" should be a vector with two levels. For family="Cox", y should be a two-column matrix (data frame) with columns named 'time' and 'status'
tau	the quantile to be estimated, a number strictly between 0 and 1. Argument required when family="Quantile".
family	response type. Abbreviations are allowed.
mscale	scale parameter for the Gram matrix associated with each function component. Default is rep(1,ncol(x))
gamma	power of inverse L_2-norm. Default is 1.
scale	if TRUE, continuous predictors will be rescaled to [0,1] interval. Default is FALSE.
nbasis	number of "knots" to be selected. Ignored when basis.id is provided.
basis.id	index designating selected "knots". Argument is not valid if family="Quantile".
cpus	number of available processor units. Default is 1. If cpus>=2, parallelize task using "parallel" package. Recommended when either sample size or number of covariates is large. Argument is not valid if family="Gaussian" or family="Binomial".

Details

The initial mean function is estimated via a smoooothing spline objective function. In the SSANOVA model framework, the regression function is assumed to have an additive form

$$\eta(x) = b + \sum_{j=1}^{p} \eta_j(x^{(j)}),$$

where b denotes intercept and η_j denotes the main effect of the j-th covariate.

For "Gaussian" response, the mean regression function is estimated by minimizing the objective function:

$$\sum_i (y_i - \eta(x_i))^2/nobs + \lambda_0 \sum_{j=1}^{p} \alpha_j ||\eta_j||^2.$$

where RSS is residual sum of squares.

For "Binomial" response, the regression function is estimated by minimizing the objective function:

$$-log - likelihood/nobs + \lambda_0 \sum_{j=1}^{p} \alpha_j ||\eta_j||^2$$

For "Quantile" regression model, the quantile function, is estimated by minimizing the objective function:

$$\sum_i \rho(y_i - \eta(x_i))/nobs + \lambda_0 \sum_{j=1}^{p} \alpha_j ||\eta_j||^2.$$

For "Cox" regression model, the log-hazard function, is estimated by minimizing the objective function:

$$-log - PartialLikelihood/nobs + \lambda_0 \sum_{j=1}^{p} \alpha_j ||\eta_j||^2.$$

The smoothing parameter λ_0 is tuned by 5-fold Cross-Validation, if `family="Gaussian"`, `"Binomial"` or `"Quantile"`, and Approximate Cross-Validation, if `family="Cox"`. But the smoothing parameters α_j are given in the argument `mscale`.

The adaptive weights are then fiven by $||\tilde{\eta}_j||^{-\gamma}$.

Value

`wt` The adaptive weights.

Author(s)

Hao Helen Zhang and Chen-Yen Lin

References

Storlie, C. B., Bondell, H. D., Reich, B. J. and Zhang, H. H. (2011) "Surface Estimation, Variable Selection, and the Nonparametric Oracle Property", Statistica Sinica, **21**, 679–705.

Examples

```
## Adaptive COSSO Model
## Binomial
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*7,0,1),nc=7))
trueProb=1/(1+exp(-x[,1]-sin(2*pi*x[,2])-5*(x[,4]-0.4)^2))
y=rbinom(200,1,trueProb)

Binomial.wt=SSANOVAwt(x,y,family="Bin")
ada.B.Obj=cosso(x,y,wt=Binomial.wt,family="Bin")

## Not run:
## Gaussian
set.seed(20130310)
x=cbind(rbinom(200,1,.7),matrix(runif(200*7,0,1),nc=7))
y=x[,1]+sin(2*pi*x[,2])+5*(x[,4]-0.4)^2+rnorm(200,0,1)
Gaussian.wt=SSANOVAwt(designx,response,family="Gau")
ada.G.Obj=cosso(x,y,wt=Gaussian.wt,family="Gaussian")

## End(Not run)
```

---

tune.cosso *Tuning procedure for cosso*

Description

Compute K-fold cross-validated score and plot cross-validated score against a grid values of smooth parameter M.

Usage

```
tune.cosso(object,folds=5,plot.it=TRUE)
```

Arguments

object	a cosso object.
folds	number of folds for corss-validation. Default is 5. It is not recommended to use folds less than 4.
plot.it	if TRUE, plot the cross-validated score against a sequence values of M.

Value

OptM	the optimal smoothing parameter for M.
M	used tuning grid points.
cvm	the mean cross-validated error/minus log-likelihood.
cvsd	estimate of standard error of cvm.

Author(s)

Hao Helen Zhang and Chen-Yen Lin

See Also

cosso, predict.cosso

Examples

```
## Binomial
set.seed(20130310)
x=cbind(rbinom(150,1,.7),matrix(runif(150*5,0,1),nc=5))
trueProb=1/(1+exp(-x[,1]-sin(2*pi*x[,2])-5*(x[,4]-0.4)^2))
y=rbinom(150,1,trueProb)

B.Obj=cosso(x,y,family="Bin",nbasis=30)
tune.cosso(B.Obj,4,TRUE)
```

---

veteran *Veterans' Administration Lung Cancer study*

Description

Randomized trial of two treatment regimens for lung cancer.

Usage

```
data(veteran)
```

Format

time	survival time
status	censoring status
trt	0=standard 1=test
celltype	1=squamous, 2=smallcell, 3=adeno, 4=large.
karno	Karnofsky performance score. minimum 10 and maximum 99 in original scale.
diagtime	months from diagnosis to randomization. minimum 1 and maximum 87 in original scale.
age	in years. minimum 34 and maximum 81 in original scale.
prior	prior therapy 0=no, 1=yes.

Details

All the variables, except for the response, have been scaled to [0,1] interval. To transform back to the original scale, use the formula:

$$x = min + (max - min) * z.$$

Source

Kalbfleisch, J. and Prentice, R.L. (2002), The Statistical Analysis of Failure Time Data (Second Edition) Wiley: New Jersey.

Index