Package 'smog'

October 23, 2018

Type Package

Title Structural Modeling by using Overlapped Group Penalty

Version 1.0

Date 2018-10-12

Author Chong Ma, Shuangge Ma, Ray Liu, Kevin Galinsky

Maintainer Chong Ma <chong.ma@yale.edu>

Description This R package fits a linear non-penalized phynotype (demographic) variables and penalized groups of prognostic effect and predictive effect, by satisfying such hierarchy structures that if a predictive effect exists, its prognostic effect must also exist. This package can deal with continuous, binomial or multinomial, and survival response variables, underlying the assumption of Gaussian, binomial (multinomial), and cox proportional hazard models, respectively. It is implemented by combining the iterative shrinkage-thresholding algorithm (ISTA) and the alternating direction method of multipliers algorithms (ADMM). The main method is built in C++, and the complementary methods are written in R.

License GPL (>= 2)

Imports Rcpp (>= 0.12.18), coxed, foreach, doParallel

LazyData true

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.0.1

NeedsCompilation yes

Archs i386, x64

R topics documented:

cv.smog *Cross-valiation for smog*

Description

cv.smog conducts the nfolds cross-validations for the whole data, where one fold of the observations are used for model-testing, and the remaining data are used for model-building. It allows the nfolds to be processed in parallel, in order to speed up the cross-validation. However, it can only do the cross-validations for one user-specified lambda, because lambda is a three-dimensional vector, the optimal search for lambda is quite computationally expensive. The cv.smog outputs the Akaike's Information Criterion (AIC) for each testing.

Usage

```
cv.smog(x, y, g, v, lambda, hierarchy, family = "gaussian", nfolds = 10,
  parallel = FALSE, ncores = NULL, ...)
```

Arguments

x	a model matrix, or a data frame of dimensions n by p, in which the columns represents the predictor variables.
y	response variable, corresponds to the family description. When family is "gaussian" or "binomial", y ought to be a numeric vector of observations of length n; when family is "coxph", y represents the survival objects, containing the survival time and the censoring status. See Surv.
g	a vector of group labels for the predictor variables.
v	a vector of binary values, represents whether or not the predictor variables are penalized. Note that 1 indicates penalization and 0 for not penalization.
lambda	a numeric vector of three penalty parameters corresponding to L2 norm, squared L2 norm, and L1 norm, respectively.
hierarchy	a factor value in levels 0, 1, 2, which represent different hierarchical structure within groups, respectively. When hierarchy=0, λ_2 and λ_3 are forced to be zeroes; when hierarchy=1, λ_2 is forced to be zero; when hierarchy=2, there is no constraint on λ's. See more explainations under "Details".
family	a description of the distribution family for the response variable variable. For continuous response variable, family is "gaussian"; for multinomial or binary response variable, family is "binomial"; for survival response variable, family is "coxph", respectively.
nfolds	number of folds. One fold of the observations in the data are used as the testing, and the remaining are fitted for model training. Default is 10.
parallel	Whether or not process the nfolds cross-validations in parallel. If TRUE, use foreach to do each cross-validation in parallel. Default is FALSE.
ncores	number of cpu's for parallel computing. See makeCluster and registerDoParallel. Default is NULL.
...	other arguments that can be supplied to smog.

Details

The function runs `smog` `nfolds` times. Evenly split the whole data into `nfolds`, and one fold of the observations are used as the testing data, and the remaining are used for model training. After calculating the AIC for each fold of testing data, return the average of the AICs. Note that this method does NOT search for the optimal penalty parameters `lambda`, and a specific `lambda` should be supplied.

Value

the average of Akaike's Information Criterions (AICs) from the `nfolds` cross-validations. Note that AIC is -2log-likelihood+2p_l, where p_l is the number of non-zero predictor variables.

Author(s)

Chong Ma, <chong.ma@yale.edu>.

See Also

`smog.default`, `smog.formula`, `predict.smog`, `plot.smog`.

Examples

```
#require(plotly)

# generate design matrix x
set.seed(2018)
n=50;p=1000
s=10
x=matrix(0,n,1+2*p)
x[,1]=sample(c(0,1),n,replace = TRUE)
x[,seq(2,1+2*p,2)]=matrix(rnorm(n*p),n,p)
x[,seq(3,1+2*p,2)]=x[,seq(2,1+2*p,2)]*x[,1]

g=c(p+1,rep(1:p,rep(2,p)))  # groups
v=c(0,rep(1,2*p))           # penalization status

# generate beta
beta=c(rnorm(13,0,2),rep(0,ncol(x)-13))
beta[c(2,4,7,9)]=0

# generate y
data=x%*%beta
noise=rnorm(n)
snr=as.numeric(sqrt(var(data)/(s*var(noise))))
y=data+snr*noise

l1=l2=10^(-seq(1,3,0.2))
cvmod=matrix(0,length(l1),length(l2))
## Not run:
for(i in 1:length(l1)){
  for(j in i:length(l2)){
    cvmod[i,j] = cv.smog(x,y,g,v,lambda=c(l1[i],0,l2[j]),
                         hierarchy=1,family="gassian",nfolds=10,scale=TRUE)
  }
}
```

```
plot_ly(x=l1,y=l2,z=t(cvmod),type="contour",
        contours=list(showlabels=TRUE))%>%
        colorbar(title="aic")%>%
        layout(xaxis=list(title="lambda1"),
               yaxis=list(title="lambda2"))%>%
        config(mathjax='cdn')


## End(Not run)
```

`penalty` *Penalty function on the composite L2, L2-Square, and L1 penalties*

Description

Penalty function on the composite L2, L2-Square, and L1 penalties

Usage

```
penalty(x, lambda, hierarchy, d)
```

Arguments

`x`	A vector of two numeric values, in which x_1 represents the prognostic effect, and x_2 for the predictive effect, respectively.
`lambda`	a vector of three penalty parameters. λ_1 and λ_2 are L2 and L2-Square (ridge) penalties for x in a group level, and λ_3 is the L1 penalty for x_2, respectively.
`hierarchy`	a factor value in levels 0, 1, 2, which represent different hierarchical structure in x, respectively. When `hierarchy=0`, λ_2 and λ_3 are forced to be zeroes; when `hierarchy=1`, λ_2 is forced to be zero; when `hierarchy=2`, there is no constraint on λ's. See `smog`.
`d`	indices for overlapped variables in x.

`plot.smog` *plot method for objects of the class smog*

Description

`plot.smog` can produce a panel of plots for the primal errors, dual errors, and the penalized log-likelihood values, based on the provided fitted model (`x`) in the S3method of `smog`.

Usage

```
## S3 method for class 'smog'
plot(x, type = "l", xlab = "iteration",
  caption = list("primal error", "dual error", "log-likelihood"), ...)
```

Arguments

x	a fitted object of class inheriting from smog.
type, xlab	default line types and x axis labels for the panel of plots.
caption	a list of y axes labels for the panel of plots.
...	additional arguments that could be supplied to plot and par.

Details

For the panel of three plots, the xlab is "iterations" and the type is "l", by default. The ylab are "primal error", "dual error","log-likelihood", respectively. This panel of plots can reflect the convergence performance for the algorithm used in smog.

Author(s)

Chong Ma, <chong.ma@yale.edu>.

See Also

par, plot.default, smog.default, smog.formula, cv.smog.

---

predict.smog *predict method for objects of the class smog*

Description

predict.smog can produce the prediction for user-given new data, based on the provided fitted model (object) in the S3method of smog. If the newdata omitted, it would output the prediction for the fitted model itself. The yielded result should match with the family in the provided model. See smog.

Usage

```
## S3 method for class 'smog'
predict(object, newdata = NULL, family = "gaussian", ...)
```

Arguments

object	a fitted object of class inheriting from smog.
newdata	a data frame containing the predictor variables, which are used to predict. If omitted, the fitted linear predictors are used.
family	a description of distribution family for which the response variable is to be predicted.
...	additional arguments affecting the predictions produced.

Details

If newdata = NULL, the fitted.value based on the object is used for the prediction.

Value

If `family` = "gaussian", a vector of prediction for the response is returned. For `family` = "coxph", a vector of predicted survival probability is returned. When `family` = "binomial", it outputs a data frame containing the predicted group labels and the corresponding probabilies.

Author(s)

Chong Ma, <chong.ma@yale.edu>.

See Also

`smog.default`, `smog.formula`, `cv.smog`, `plot.smog`.

`prox` *proximal operator on the composite L2, L2-Square, and L1 penalties*

Description

proximal operator on the composite L2, L2-Square, and L1 penalties

Usage

```
prox(x, lambda, hierarchy, d)
```

Arguments

`x`	A numeric vector of two.
`lambda`	a vector of three penalty parameters. λ_1 and λ_2 are L2 and L2-Square (ridge) penalties for x in a group level, and λ_3 is the L1 penalty for x_2, respectively.
`hierarchy`	a factor value in levels 0, 1, 2, which represent different hierarchical structure in x, respectively. When `hierarchy=0`, λ_2 and λ_3 are forced to be zeroes; when `hierarchy=1`, λ_2 is forced to be zero; when `hierarchy=2`, there is no constraint on λ's. See `smog`.
`d`	indices for overlapped variables in x.

`smog.default` *Generalized linear model constraint on hierarchical structure by using overlapped group penalty*

Description

`smog` fits a linear non-penalized phynotype (demographic) variables such as age, gender, treatment, etc, and penalized groups of prognostic effect (main effect) and predictive effect (interaction effect), by satisfying the hierarchy structure: if a predictive effect exists, its prognostic effect must be in the model. It can deal with continuous, binomial or multinomial, and survival response variables, underlying the assumption of Gaussian, binomial (multinomial), and cox proportional hazard models, respectively. It can accept `formula`, and output coefficients table, fitted.values, and convergence information produced in the algorithm iterations.

Usage

```
## Default S3 method:
smog(x, y, g, v, lambda, hierarchy, family = "gaussian",
  subset = NULL, rho = 0.001, scale = FALSE, eabs = 0.001,
  erel = 0.001, LL = 1, eta = 1.25, maxitr = 500, ...)

## S3 method for class 'formula'
smog(formula, data = list(), g, v, lambda, hierarchy, ...)
```

Arguments

x	a model matrix, or a data frame of dimensions n by p, in which the columns represents the predictor variables.
y	response variable, corresponds to the family description. When family is "gaussian" or "binomial", y ought to be a numeric vector of observations of length n; when family is "coxph", y represents the survival objects, containing the survival time and the censoring status. See Surv.
g	a vector of group labels for the predictor variables.
v	a vector of binary values, represents whether or not the predictor variables are penalized. Note that 1 indicates penalization and 0 for not penalization.
lambda	a numeric vector of three penalty parameters corresponding to L2 norm, squared L2 norm, and L1 norm, respectively.
hierarchy	a factor value in levels 0, 1, 2, which represent different hierarchical structure within groups, respectively. When hierarchy=0, λ_2 and λ_3 are forced to be zeroes; when hierarchy=1, λ_2 is forced to be zero; when hierarchy=2, there is no constraint on λ's. See more explainations under "Details".
family	a description of the distribution family for the response variable variable. For continuous response variable, family is "gaussian"; for multinomial or binary response variable, family is "binomial"; for survival response variable, family is "coxph", respectively.
subset	an optional vector specifying a subset of observations to be used in the model fitting. Default is NULL.
rho	the penalty parameter used in the alternating direction method of multipliers (ADMM) algorithm. Default is 1e-3.
scale	whether or not scale the design matrix. Default is FALSE.
eabs	the absolute tolerance used in the ADMM algorithm. Default is 1e-3.
erel	the reletive tolerance used in the ADMM algorithm. Default is 1e-3.
LL	initial value for the Lipschitz continuous constant for approximation to the objective function in the Majorization- Minimization (MM) (or iterative shrinkage-thresholding algorithm (ISTA)). Default is 1.
eta	gradient stepsize for the backtrack line search for the Lipschitz continuous constant. Default is 1.25.
maxitr	the maximum iterations for convergence in the ADMM algorithm. Default is 500.
...	other relevant arguments that can be supplied to smog.
formula	an object of class "formula": a symbolic description of the model to be fitted. Should not include the intercept.
data	an optional data frame, containing the variables in the model.

Details

The formula has the form `response ~ 0 + terms` where `terms` is a series of predictor variables to be fitted for `response`. For `gaussian` family, the response is a continuous vector. For `binomial` family, the response is a factor vector, in which the last level denotes the "pivot". For `coxph` family, the response is a `Surv` object, containing the survival time and censoring status.

The terms contains the non-penalized predictor variables, and many groups of prognostic and predictive terms, where in each group the prognostic term comes first, followed by the predictive term.

The `hierarchy` denotes different hierachical structures within groups by adjusting the penalty parameters in the penalty function:

$$\Omega(\boldsymbol{\beta}) = \lambda_1||\boldsymbol{\beta}|| + \lambda_2||\boldsymbol{\beta}||^2 + \lambda_3|\beta_2|$$

Where $\boldsymbol{\beta} = (\beta_1, \beta_2)$. Note that β_1 denotes the prognostic effect (main effect), and β_2 for the predictive effect (interactive effect), respectively. When `hierachy=0`, λ_2 and λ_3 are forced to be zero, indicating no structure within groups. When `hierachy=1`, λ_2 is forced to be zero; and for `hierachy = 2`, there is no constraints on λ's. For `hierarchy` is either 1 or 2, they both admits the existence of the structure within groups.

`rho,eabs,erel,LL,eta` are the corresponding parameters used in the itervative shrinkage-thresholding algorithm (ISTA) and the alternating direction method of multipliers algorithm (ADMM).

Note that the missing values in the data are supposed to be dealt with in the data preprocessing, before applying the method.

Value

`smog` returns an object of class inhering from "smog". The generic accessor functions `coef`, `coefficients`, `fitted.value`, and `predict` can be used to extract various useful features of the value returned by `smog`.

An object of "smog" is a list containing at least the following components:

`coefficients`	a data frame containing the nonzero predictor variables' indexes, names, and estimates. When family is "binomial", the estimates have K-1 columns, each column representing the weights for the corresponding group. The last group behaves the "pivot".
`fitted.values`	the fitted mean values for the response variable, for family is "gaussian". When family is "binomial", the fitted.values are the probabilies for each class; when family is "coxph", the fitted.values are survival probabilities.
`loglike`	the penalized log-likelihood values for each iteration in the algorithm.
`PrimalError`	the averged norms $\|\|\beta - Z\|\|/\sqrt{p}$ for each iteration, in the ADMM algorithm.
`DualError`	the averaged norms $\|\|Z^{t+1} - Z^t\|\|/\sqrt{p}$ for each iteration, in the ADMM algorithm.
`converge`	the number of iterations processed in the ADMM algorithm.
`call`	the matched call.
`formula`	the formula supplied.

Author(s)

Chong Ma, <chong.ma@yale.edu>

See Also

cv.smog, predict.smog, plot.smog.

Examples

```
require(coxed)

n=50;p=1000
set.seed(2018)
# generate design matrix x
s=10
x=matrix(0,n,1+2*p)
x[,1]=sample(c(0,1),n,replace = TRUE)
x[,seq(2,1+2*p,2)]=matrix(rnorm(n*p),n,p)
x[,seq(3,1+2*p,2)]=x[,seq(2,1+2*p,2)]*x[,1]

g=c(p+1,rep(1:p,rep(2,p)))  # groups
v=c(0,rep(1,2*p))           # penalization status

# generate beta
beta=c(rnorm(13,0,2),rep(0,ncol(x)-13))
beta[c(2,4,7,9)]=0

# generate y
data1=x%*%beta
noise1=rnorm(n)
snr1=as.numeric(sqrt(var(data1)/(s*var(noise1))))
y1=data1+snr1*noise1
lfit1=smog(x,y=y1,g,v,hierarchy=1,lambda=c(0.02,0,0.001),family = "gaussian",scale=TRUE)

## generate binomial data
prob=exp(as.matrix(x)%*%as.matrix(beta))/(1+exp(as.matrix(x)%*%as.matrix(beta)))
y2=ifelse(prob>0.5,0,1)
lfit2=smog(x,y=y2,g,v,hierarchy=1,lambda=c(0.025,0,0.001),family = "binomial")

## generate survival data
data3=sim.survdata(N=n,T=100,X=x,beta=beta)
y3=data3$data[,c("y","failed")]
y3$failed=ifelse(y3$failed,1,0)
colnames(y3)=c("time","status")
lfit3=smog(x,y=y3,g,v,hierarchy = 1,lambda = c(0.075,0,0.001),family = "coxph",LL=10)
```

Index