

Package ‘glmtp’

October 15, 2021

Type Package

Title Generalized Linear Models with Truncated Lasso Penalty

Version 2.0.0

Date 2021-10-14

URL <https://yuyangyy.com/glmtp/>

Depends R (>= 3.5.0), foreach, parallel

Imports doParallel, ggplot2

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

Description Extremely efficient procedures for fitting regularization path with l0, l1, and truncated lasso penalty for linear regression and logistic regression models. This version is a completely new version compared with our previous version, which was mainly based on R. New core algorithms are developed and are now written in C++ and highly optimized.

Encoding UTF-8

License GPL-3

LazyData true

Author Chunlin Li [aut],
Yu Yang [aut, cre],
Chong Wu [aut]

Maintainer Yu Yang <yang6367@umn.edu>

RoxygenNote 7.1.1

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-10-15 09:10:09 UTC

R topics documented:

bin_data	2
cv.glmtp	3
gau_data	4
glmtp	5
plot.cv.glmtp	9
plot.glmtp	10
predict.cv.glmtp	12
predict.glmtp	13
Index	16

bin_data	<i>A simulated binomial dataset.</i>
----------	--------------------------------------

Description

A dataset simulated for illustrating logistic regression models.

Usage

```
data(bin_data)
```

Format

A list with two elements: design matrix X and y .

X design matrix

y response ...

Examples

```
data("bin_data")
cv.fit <- cv.glmtp(bin_data$X, bin_data$y, family = "binomial", penalty = "l1")
plot(cv.fit)
```

cv.glmtp

*Cross-validation for glmtp***Description**

Does k-fold cross-validation for glmtp, produces a plot, and returns a value for lambda (or kappa if penalty="l0").

Usage

```
cv.glmtp(X, y, ..., seed = NULL, nfolds = 10, obs.fold = NULL, ncores = 1)
```

Arguments

X	input matrix, of dimension nobs x nvars, as in glmtp.
y	response, of length nobs, as in glmtp.
...	Other arguments that can be passed to glmtp
seed	the seed for reproduction purposes
nfolds	number of folds; default is 10. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. The smallest value allowable is nfolds=3
obs.fold	an optional vector of values between 1 and nfolds identifying what fold each observation is in. If supplied, nfolds can be missing.
ncores	number of cores utilized; default is 1. If greater than 1, then use doParallel::foreach to fit each fold; if equal to 1, then use for loop to fit each fold. No need to register parallel outside.

Details

The function runs glmtp nfolds+1 times; the first to get the lambda or kappa sequence, and then the remainder to compute the fit with each of the folds omitted. The error is accumulated, and the average (haven't finished yet) error and standard deviation over the folds is computed.

Value

an object of class "cv.glmtp" is returned, which is a list with the ingredients of the cross-validation fit.

call	the function call
cv.mean	The mean cross-validated error - a vector of length length(kappa) if penalty = "l0" and a vector of length length{lambda} otherwise.
cv.se	estimate of standard error of cv.mean.
fit	a fitted glmtp object for the full data.
idx.min	the index of the lambda or kappa sequence that corresponding to the smallest cv mean error.

kappa	the values of kappa used in the fits, available when penalty = 'l0'.
kappa.min	the value of kappa that gives the minimum cv.mean, available when penalty is 'l0'.
lambda	the values of lambda used in the fits.
lambda.min	value of lambda that gives minimum cv.mean, available when penalty is 'l1' or 'tlp'.
null.dev	null deviance of the model.
obs.fold	the fold id for each observation used in the CV.

Author(s)

Chunlin Li, Yu Yang, Chong Wu
 Maintainer: Yu Yang <yang6367@umn.edu>

References

Shen, X., Pan, W., & Zhu, Y. (2012). *Likelihood-based selection and sharp parameter estimation*. *Journal of the American Statistical Association*, 107(497), 223-232.
 Yang, Y., & Zou, H. (2014). *A coordinate majorization descent algorithm for l1 penalized learning*. *Journal of Statistical Computation and Simulation*, 84(1), 84-95.
 Two R package Github: *ncvreg* and *glmnet*.

See Also

glm1p and plot, predict, and coef methods for "cv.glm1p" objects.

Examples

```
# Gaussian
X <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
cv.fit <- cv.glm1p(X, y, family = "gaussian", penalty = "l1", seed=2021)

# Binomial
X <- matrix(rnorm(100 * 20), 100, 20)
y <- sample(c(0,1), 100, replace = TRUE)
cv.fit <- cv.glm1p(X, y, family = "binomial", penalty = "l1", seed=2021)
```

gau_data

A simulated gaussian dataset.

Description

A dataset simulated for illustrating linear regression models.

Usage

```
data(gau_data)
```

Format

A list with five elements: design matrix X , response y , correlation structure of the covariates Σ , true beta β , and the noise level σ .

X design matrix

y response

Sigma correlation matrix of the covariates

beta true beta values

sigma the noise level ...

Examples

```
data("gau_data")
cv.fit <- cv.glmtp(gau_data$X, gau_data$y, family = "gaussian", penalty = "tlp")
plot(cv.fit)
```

 glmtp

fit a GLM with l0, l1, or tlp regularization

Description

Fit a generalized linear model via penalized maximum likelihood. The regularization path is computed for the l0, lasso, or truncated lasso penalty at a grid of values for the regularization parameter: number of non-zeros or lambda. Can deal with all shapes of data (not large sparse data matrices now). Fits linear, logistic, multinomial, and poisson models.

The sequence of models implied by lambda or kappa is fit by coordinate descent.

The package provides 3 penalties: l0, l1, and tlp and 3 distribution families: gaussian, binomial, and poisson.

Usage

```
glmtp(
  X,
  y,
  family = c("gaussian", "binomial"),
  penalty = c("l0", "l1", "tlp"),
  nlambdas = 100,
  lambda.min.ratio = ifelse(nobs < nvars, 0.01, 1e-04),
  lambda = NULL,
  kappa = NULL,
  tau = 0.3 * sqrt(log(nvars)/nobs),
```

```

delta = 2,
tol = 1e-04,
weights = NULL,
penalty.factor = rep(1, nvars),
standardize = TRUE,
dc.maxit = 20,
cd.maxit = 10000,
nr.maxit = 500,
...
)

```

Arguments

X	input matrix, of dimension nobs x nvars; each row is an observation vector. Currently, doesn't support the sparse matrix format (inherit from class "sparseMatrix" as in package Matrix).
y	response variable, of length nobs. Quantitative for family="gaussian", or family="poisson" (non-negative counts). For family="binomial", should be either a factor with two levels. For family="multinomial", (to be added).
family	a character string representing one of the built-in families. See Details section below.
penalty	a character string representing one of the built-in penalties. "l0" represents L ₀ penalty, "l1" represents lasso-type penalty, and "tlp" represents truncated lasso penalty.
nlambda	The number of lambda values. Default is 100.
lambda.min.ratio	Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size nobs relative to the number of variables nvars. If nobs > nvars, the default is 0.0001, close to zero. If nobs < nvars, the default is 0.01.
lambda	A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Avoid supplying a single value for lambda (for predictions after CV use predict() instead). Supply instead a decreasing sequence of lambda values. glmTLP relies on its warm starts for speed, and its often faster to fit a whole path than compute a single fit. When the penalty is 'l0', it is not recommended for the users to supply this parameter.
kappa	A user supplied kappa sequence. Typical usage is to have the program compute its own kappa sequence based on nvars and nobs. This sequence is used when penalty is 'l0'.
tau	A tuning parameter used in the TLP penalty. Default is $0.3 * \sqrt{\log(nvars)/nobs}$.
delta	A tuning parameter used in the coordinate majorization descent algorithm. See Yang, Y., & Zou, H. (2014) in the reference for detail.
tol	Tolerance level for all the iterative optimization algorithms.

weights	observation weights. Default is 1 for each observation
penalty.factor	Separate penalty factors can be applied to each coefficient. This is a number that multiplies lambda to allow differential shrinkage. Can be 0 for some variables, which implies no shrinkage, and that variable is always included in the model. Default is 1 for all variables.
standardize	Logical. Whether or not standardize the input matrix X; default is TRUE.
dc.maxit	Maximum number of iterations for the DC (Difference of Convex Functions) programming; default is 20.
cd.maxit	Maximum number of iterations for the coordinate descent algorithm; default is 10 ⁴ .
nr.maxit	Maximum number of iterations for the Newton-Raphson method; default is 500.
...	Additional argument. These include some of the original arguments to 'glmtp', and each must be named if used.

Details

The objective function for "gaussian" is

$$1/2RSS/nobs + \lambda * penalty,$$

and for the other models it is

$$-loglik/nobs + \lambda * penalty.$$

Note also that for "gaussian", glmtp standardizes y to have unit variance (using 1/(n-1) formula). The coefficients for any predictor variables with zero variance are set to zero for all values of lambda.

Details on 'family' option

glmtp currently only supports built-in families.

The built in families are specified via a character string. For all families, the object produced is a regularization path for fitting the generalized linear regression paths, by maximizing the appropriate penalized log-likelihood. Sometimes the sequence is truncated before nlambda values of lambda have been used, because of instabilities in the inverse link functions near a saturated fit. glmtp(..., family="binomial") fits a traditional logistic regression model for the log-odds. glmtp(..., family="multinomial") fits a symmetric multinomial model, where each class is represented by a linear model (on the log-scale). The penalties take care of redundancies. A two-class "multinomial" model will produce the same fit as the corresponding "binomial" model, except the pair of coefficient matrices will be equal in magnitude and opposite in sign, and half the "binomial" values.

Details on 'penalty' option

The built in penalties are specified by a character string. For l0 penalty, kappa sequence is used for generating the regularization path, while for the other penalties, lambda sequence is used for generating the regularization path.

Value

An object with S3 class "glm1tp".

beta	a $nvars \times \text{length}(\text{kappa})$ matrix of coefficients when penalty is 'l0'; a $nvars \times \text{length}(\text{lambda})$ matrix of coefficients when penalty is 'l1' and 'tlp'.
call	the call that produced this object.
family	the distribution family used in the model fitting.
intercept	the intercept vector, of $\text{length}(\text{kappa})$ or $\text{length}(\text{lambda})$.
lambda	The actual sequence of lambda values used. Note that the length may be smaller than the provided nlambda due to removal of saturated values.
penalty	the penalty type in the model fitting.
penalty.factor	the penalty factor for each coefficient used in the model fitting.
tau	the tuning parameter used in the model fitting, available when penalty = 'tlp'.
user.lambda	logical, whether or not the used lambda sequence is provided by the user.
user.kappa	logical, whether or not the used kappa sequence is provided by the user, available when penalty = 'l0'.

glm1tp functions

'glm1tp()', 'cv.glm1tp()'

Author(s)

Chunlin Li, Yu Yang, Chong Wu
 Maintainer: Yu Yang <yang6367@umn.edu>

References

Shen, X., Pan, W., & Zhu, Y. (2012). *Likelihood-based selection and sharp parameter estimation*. *Journal of the American Statistical Association*, 107(497), 223-232.
 Yang, Y., & Zou, H. (2014). *A coordinate majorization descent algorithm for l1 penalized learning*. *Journal of Statistical Computation and Simulation*, 84(1), 84-95.
 Two R package Github: *ncvreg* and *glmnet*.

See Also

print, predict, coef and plot methods, and the cv.glm1tp function.

Examples

```
# Gaussian
X <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
fit1 <- glm1tp(X, y, family = "gaussian", penalty = "l0")
fit2 <- glm1tp(X, y, family = "gaussian", penalty = "l1")
fit3 <- glm1tp(X, y, family = "gaussian", penalty = "tlp")
```



```
# Binomial

X <- matrix(rnorm(100 * 20), 100, 20)
y <- sample(c(0,1), 100, replace = TRUE)
fit <- glmtp(X, y, family = "binomial", penalty = "l1")
```

`plot.cv.glmtp`*plot the cross-validation curve produced by cv.glmtp*

Description

Plots the cross-validation curve, and upper and lower standard deviation curves, as a function of the lambda or kappa values used.

Usage

```
## S3 method for class 'cv.glmtp'
plot(x, vertical.line = TRUE, ...)
```

Arguments

<code>x</code>	Fitted "cv.glmtp" object
<code>vertical.line</code>	Logical. Whether to include a vertical line indicating the position of the index which gives the smallest CV error.
<code>...</code>	Other graphical parameters to plot

Details

A plot is produced, and a ggplot object is returned.

Author(s)

Chunlin Li, Yu Yang, Chong Wu
Maintainer: Yu Yang <yang6367@umn.edu>

References

Shen, X., Pan, W., & Zhu, Y. (2012). *Likelihood-based selection and sharp parameter estimation*. *Journal of the American Statistical Association*, 107(497), 223-232.
Yang, Y., & Zou, H. (2014). *A coordinate majorization descent algorithm for l1 penalized learning*. *Journal of Statistical Computation and Simulation*, 84(1), 84-95.
Two R package Github: *ncvreg* and *glmnet*.

Examples

```

X <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
cv.fit <- cv.glmtp(X, y, family = "gaussian", penalty = "t1p")
plot(cv.fit)
plot(cv.fit, vertical.line = FALSE)
cv.fit2 <- cv.glmtp(X, y, family = "gaussian", penalty = "l0")
plot(cv.fit2)
plot(cv.fit2, vertical.line = FALSE)

data("gau_data")
cv.fit <- cv.glmtp(gau_data$X, gau_data$y, family = "gaussian", penalty = "t1p")
plot(cv.fit)

data("bin_data")
cv.fit <- cv.glmtp(bin_data$X, bin_data$y, family = "binomial", penalty = "l1")
plot(cv.fit)

```

plot.glmtp

plot solution path from a "glmtp" object.

Description

this function generates a solution path plot from a fitted "glmtp" object.

Usage

```

## S3 method for class 'glmtp'
plot(
  x,
  xvar = c("lambda", "kappa", "deviance", "l1_norm", "log_lambda", "log_kappa"),
  xlab = iname,
  ylab = "Coefficients",
  title = "Solution Path",
  label = FALSE,
  label.size = 3,
  ...
)

```

Arguments

x	Fitted "glmtp" model object.
xvar	the x-axis variable to plot against, including "lambda", "kappa", "deviance", "l1_norm", and "log_lambda".
xlab	the x-axis label of the plot, default is "Lambda", "Kappa", "Fraction of Explained Deviance", "L1 Norm", and "Log Lambda".

ylab	the y-axis label of the plot, default is "Coefficients".
title	the main title of the plot, default is "Solution Path".
label	logical, whether or not attach the labels for the non-zero coefficients, default is FALSE.
label.size	the text size of the labels, default is 3.
...	Additional arguments.

Value

A ggplot object.

Author(s)

Chunlin Li, Yu Yang, Chong Wu
Maintainer: Yu Yang <yang6367@umn.edu>

References

Shen, X., Pan, W., & Zhu, Y. (2012). *Likelihood-based selection and sharp parameter estimation*. *Journal of the American Statistical Association*, 107(497), 223-232.
Yang, Y., & Zou, H. (2014). *A coordinate majorization descent algorithm for l1 penalized learning*. *Journal of Statistical Computation and Simulation*, 84(1), 84-95.
Two R package Github: *ncvreg* and *glmnet*.

See Also

print, predict, coef and plot methods, and the cv.glmtp function.

Examples

```
X <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
fit <- glmtp(X, y, family = "gaussian", penalty = "l1")
plot(fit, xvar = "lambda")
plot(fit, xvar = "log_lambda")
plot(fit, xvar = "l1_norm")
plot(fit, xvar = "log_lambda", label = TRUE)
fit2 <- glmtp(X, y, family = "gaussian", penalty = "l0")
plot(fit2, xvar = "kappa", label = TRUE)
```

predict.cv.glmtp *make predictions from a "cv.glmtp" object.*

Description

This function makes predictions from a cross-validated glmtp model, using the stored "glmtp" object, and the optimal value chosen for lambda.

Usage

```
## S3 method for class 'cv.glmtp'
predict(
  object,
  X,
  type = c("link", "response", "class", "coefficients", "numnz", "varnz"),
  lambda = NULL,
  kappa = NULL,
  which = object$idx.min,
  ...
)

## S3 method for class 'cv.glmtp'
coef(object, lambda = NULL, kappa = NULL, which = object$idx.min, ...)
```

Arguments

object	Fitted "cv.glmtp" object.
X	Matrix of new values for x at which predictions are to be made. Must be a matrix. See documentation for predict.glmtp.
type	Type of prediction required. Type "link" gives the linear predictors for "binomial" models; for "gaussian" models it gives the fitted values. Type "response" gives the fitted probabilities for "binomial"; for "gaussian" type "response" is equivalent to type "link". Type "coefficients" computes the coefficients at the requested values for lambda or kappa. Note that for "binomial" models, results are returned only for the class corresponding to the second level of the factor response. Type "class" applies only to "binomial" models, and produces the class label corresponding to the maximum probability. Type "numnz" returns the total number of non-zeros coefficients for each value of lambda or kappa. Type "varnz" returns a list of the indices of the nonzero coefficients for each value of lambda or kappa.
lambda	Value of the penalty parameter lambda at which predictions are required. Default is NULL.
kappa	Value of the penalty parameter kappa at which predictions are required. Default is NULL.
which	Index of the penalty parameter lambda or kappa sequence at which predictions are required. Default is the idx.min stored on the CV object.
...	Additional arguments.

Details

This function makes it easier to use the results of cross-validation to make a prediction.

Value

The object returned depends on the ... argument which is passed on to the predict method for glmtp objects.

Author(s)

Chunlin Li, Yu Yang, Chong Wu
Maintainer: Yu Yang <yang6367@umn.edu>

References

Shen, X., Pan, W., & Zhu, Y. (2012). *Likelihood-based selection and sharp parameter estimation*. *Journal of the American Statistical Association*, 107(497), 223-232.
Yang, Y., & Zou, H. (2014). *A coordinate majorization descent algorithm for l1 penalized learning*. *Journal of Statistical Computation and Simulation*, 84(1), 84-95.
Two R package Github: *ncvreg* and *glmnet*.

See Also

print, predict, coef and plot methods, and the cv.glmtp function.

Examples

```
X <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
cv.fit <- cv.glmtp(X, y, family = "gaussian", penalty = "l1")
predict(cv.fit, X = X[1:5, ])
coef(cv.fit)
predict(cv.fit, X = X[1:5, ], lambda = 0.1)
```

predict.glmtp	<i>make predictions from a "glmtp" object.</i>
---------------	--

Description

Similar to other predict methods, this functions predicts fitted values, logits, coefficients and more from a fitted "glmtp" object.

Usage

```
## S3 method for class 'glmtp'
predict(
  object,
  X,
  type = c("link", "response", "class", "coefficients", "numnz", "varnz"),
  lambda = NULL,
  kappa = NULL,
  which = 1:(ifelse(object$penalty == "l0", length(object$kappa),
    length(object$lambda))),
  ...
)

## S3 method for class 'glmtp'
coef(
  object,
  lambda = NULL,
  kappa = NULL,
  which = 1:(ifelse(object$penalty == "l0", length(object$kappa),
    length(object$lambda))),
  drop = TRUE,
  ...
)
```

Arguments

object	Fitted "glmtp" model object.
X	Matrix of new values for X at which predictions are to be made. Must be a matrix; can be sparse as in Matrix package. This argument is not used for type=c("coefficients", "nonzero")
type	Type of prediction required. Type "link" gives the linear predictors for "binomial" models; for "gaussian" models it gives the fitted values. Type "response" gives the fitted probabilities for "binomial"; for "gaussian" type "response" is equivalent to type "link". Type "coefficients" computes the coefficients at the requested values for lambda or kappa. Note that for "binomial" models, results are returned only for the class corresponding to the second level of the factor response. Type "class" applies only to "binomial" models, and produces the class label corresponding to the maximum probability. Type "numnz" returns the total number of non-zeros coefficients for each value of lambda or kappa. Type "varnz" returns a list of the indices of the nonzero coefficients for each value of lambda or kappa.
lambda	Value of the penalty parameter lambda at which predictions are required. Default is NULL.
kappa	Value of the penalty parameter kappa at which predictions are required. Default is NULL.
which	Index of the penalty parameter lambda or kappa sequence at which predictions are required. Default are the indices for the entire penalty parameter sequence.

... Additional arguments.
 drop Whether or not keep the dimension that is of length 1.

Details

The shape of the objects returned are different for "multinomial" objects. This function actually calls NextMethod(), and the appropriate predict method is invoked for each of the three model types. coef(...) is equivalent to predict(type="coefficients",...)

Value

The object returned depends on type.

Author(s)

Chunlin Li, Yu Yang, Chong Wu
 Maintainer: Yu Yang <yang6367@umn.edu>

References

Shen, X., Pan, W., & Zhu, Y. (2012). *Likelihood-based selection and sharp parameter estimation*. *Journal of the American Statistical Association*, 107(497), 223-232.
 Yang, Y., & Zou, H. (2014). *A coordinate majorization descent algorithm for l1 penalized learning*. *Journal of Statistical Computation and Simulation*, 84(1), 84-95.
 Two R package Github: *ncvreg* and *glmnet*.

See Also

print, predict, coef and plot methods, and the cv.glmtp function.

Examples

```
# Gaussian
X <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
fit <- glmtp(X, y, family = "gaussian", penalty = "l1")
predict(fit, X = X[1:5, ])
coef(fit)
predict(fit, X = X[1:5, ], lambda = 0.1)

# Binomial
X <- matrix(rnorm(100 * 20), 100, 20)
y <- sample(c(0,1), 100, replace = TRUE)
fit <- glmtp(X, y, family = "binomial", penalty = "l1")
coef(fit)
predict(fit, X = X[1:5, ], type = "response")
predict(fit, X = X[1:5, ], type = "response", lambda = 0.01)
predict(fit, X = X[1:5, ], type = "class", lambda = 0.01)
predict(fit, X = X[1:5, ], type = "numnz", lambda = 0.01)
```

Index

* datasets

bin_data, 2

gau_data, 4

* models

cv.glmTLP, 3

glmTLP, 5

plot.cv.glmTLP, 9

plot.glmTLP, 10

predict.cv.glmTLP, 12

predict.glmTLP, 13

* plot

plot.cv.glmTLP, 9

plot.glmTLP, 10

* regression

cv.glmTLP, 3

glmTLP, 5

predict.cv.glmTLP, 12

predict.glmTLP, 13

bin_data, 2

coef.cv.glmTLP(predict.cv.glmTLP), 12

coef.glmTLP(predict.glmTLP), 13

cv.glmTLP, 3

gau_data, 4

glmTLP, 5

plot.cv.glmTLP, 9

plot.glmTLP, 10

predict.cv.glmTLP, 12

predict.glmTLP, 13