

Package ‘MSTweedie’

March 26, 2018

Type Package

Title Multi-source Sparse Tweedie Modelling

Version 0.1.0

Author Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer Simon Fontaine <fontaines@dms.umontreal.ca>

Description Extending the HDtweedie package to multi-source data, this package implements multi-source sparse Tweedie modelling. Given non-negative data with excess zeroes (typically insurance claim data), the algorithm fits the Tweedie distribution while penalizing the coefficient. Flexible regularization is achieved by an adjustable penalty factor that can balance between-feature sparsity and between-sources sparsity within features.

License GPL-2

Depends R (>= 2.10)

Encoding UTF-8

LazyData true

Imports grDevices, stats, graphics, methods

RoxygenNote 6.0.1

Archs i386, x64

RemoteType github

RemoteHost <https://api.github.com>

RemoteRepo MSTweedie

RemoteUsername fontaine618

RemoteRef master

RemoteSha fb4ecd4da65bec37484775496a44e1c1661e63e1

GithubRepo MSTweedie

GithubUsername fontaine618

GithubRef master

GithubSHA1 fb4ecd4da65bec37484775496a44e1c1661e63e1

R topics documented:

AutoClaim	2
coef.cv.MSTweedie	4
coef.MSTweedie	5
cv.MSTweedie	6
kkt.check	9
MSTweedie	10
plot.cv.MSTweedie	13
plot.MSTweedie	14
predict.cv.MSTweedie	15
predict.MSTweedie	17
print.MSTweedie	18

Index	20
--------------	-----------

AutoClaim	<i>AutoClaim dataset</i>
-----------	--------------------------

Description

The motor insurance dataset is originally retrieved the [cplm](#) package. It contains insurance claim data as well as information on the policyholder. From the original dataset, only part of the variables are kept and some are transformed (see description below). Missing values are imputed via the [rrcovNA::impSeq](#) function.

Usage

```
data("AutoClaim")
```

Format

A data frame with 10296 observations on the following 35 variables.

CLM_AMT5 Aggregate claim loss of policy (in thousands)
 KIDSDRIV Number of child passengers
 TRAVTIME Commute time
 CAR_USE (1) Private or (2) commercial use
 BLUEBOOK (log) car value
 NPOLICY Number of policies
 RED_CAR Whether the color of the car is (2) car or (1) not
 REVOLKED Whether the policyholder's license was (2) revoked in the past or (1) not
 MVR_PTS Number of motor vehicule record points
 HOMEKIDS Number of children at home
 GENDER Gender of policyholder : (1) female or (2) male
 MARRIED Whether the policyholder is (2) married or (1) not
 PARENT1 Whether (2) the policyholder grew up in a single-parent family or (1) not
 AREA (1) Rural or (2) urban area

CAR_TYPE_2-6 (0-1 dummy variables) Type of car : (base) Panel Truck, (2) Pickup, (3) Sedan, (4) Sports Car, (5) SUV, (6) Van

JOBCLASS_2-9 (0-1 dummy variables) Job class of policyholder: (base) Unknown, (2) Blue Collar, (3) Clerical, (4) Doctor, (5) Home Maker, (6) Lawyer, (7) Manager, (8) Professionnal, (9) Student

MAX_EDUC_2-5 (0-1 dummy variables) Maximal level of education of policyholder: (base) less than High School, (2) Bachelors, (3) High School, (4) Masters, (5) PhD

AGE_CAT_2-5 (0-1 dummy variables) Age category of policyholder : (base) <30, (2) [30,40), (3) [40,50), (4) [50,60), (5) 60+].

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

Source

cp1m package.

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

Zhang, Y. (2013). "cp1m: Compound Poisson Linear Models." A vignette for R package cp1m. Available from <http://cran.r-project.org/web/packages/cp1m>.

Todorov, V. (2016). "rrcovNA: Scalable Robust Estimators with High Breakdown Point for Incomplete Data." A vignette for R package rrcovNA. Available from <https://cran.r-project.org/web/packages/rrcovNA>.

Examples

```
#import package
library(MSTweedie)

#load data
data(AutoClaim)

#display head of dataset
head(AutoClaim)

#classify the policies by REVOLKED and whether there was a claim or not
table(AutoClaim$REVOLKED, AutoClaim$CLM_AMT5 > 0)
```

coef.cv.MSTweedie	<i>Extract the estimated coefficients of a cv.MSTweedie object</i>
-------------------	--

Description

This function is a wrapper function for the coef.MSTweedie when applied to a cv.MSTweedie object.

Usage

```
## S3 method for class 'cv.MSTweedie'
coef(cv, s = c("lambda.1se", "lambda.min"))
```

Arguments

cv	cv.MSTweedie object
s	Either a vector of regularization parameters (must match those of fit), a vector of indices of regularization parameters. Default is the whole solution path or one of "lambda.min" or "lambda.1se". Default is "lambda.1se".

Details

Returns the estimated coefficient in the MSTweedie object within the cv.MSTweedie object at the specified s values of the regularization parameter. When "lambda.min" or "lambda.1se" is supplied for s, the respective value of the regularization parameter is used.

Value

A list of length ntasks of matrices of dimension nvars*length(s) containing the estimated coefficients at each values of s.

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[cv.MSTweedie](#), [coef.MSTweedie](#)

Examples

```
#import package
library(MSTweedie)

#load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
cv <- cv.MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')

# extract coefficients at lambda.1se
coef.cv.MSTweedie(cv)
```

coef.MSTweedie	<i>Extract the estimated coefficients of a MSTweedie object</i>
----------------	---

Description

This function extracts the estimated coefficients of a MSTweedie object at given regularization parameters and given sources.

Usage

```
## S3 method for class 'MSTweedie'
coef(fit, s = NULL, tasks)
```

Arguments

fit	MSTweedie object.
s	Either a vector of regularization parameters (must match those of fit) or a vector of indices of regularization parameters. Default is the whole solution path.
tasks	Indices of the sources of which coefficients must be returned. Default is all sources.

Value

A list of length length(tasks) of matrices of dimension nvars*length(s) containing the estimated coefficients at each values of s.

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.
 Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[MSTweedie](#), [plot.MSTweedie](#), [predict.MSTweedie](#)

Examples

```
# import package
library(MSTweedie)

# load data
data(AutoClaim)

# L1/Linf regularization
fit <- MSTweedie(x=AutoClaim, y=1, source=4, reg='Linf')

# extract coefficients at 34th to 36th lambda
coef.MSTweedie(fit, s=34:36)
```

cv.MSTweedie

Cross-Validation on the Multi-source sparse Tweedie model

Description

This function performs k-fold cross-validation (CV) of the multi-source sparse Tweedie model (MSTweedie) partly based on the [glmnet::cv](#) function. The Tweedie model deviance is the statistical criterion for model selection.

Usage

```
cv.MSTweedie(x, y, w, source, rho, nlambda = 100, lambda,
  lambda.min = 0.001, nfolds = 10, kktstop = F, foldid,
  adaptive = 0, x.normalize = TRUE, reg = c("L2", "Linf"),
  eps = 5e-04, sr = TRUE, maxit = 10000,
  pf = rep(1, nvars), alpha = 0, ...)
```

Arguments

x	A data frame containing the predictors, the responses (identifying the sources either by different columns in the simultaneous case or via an additional index column) and, optionally, the observation weights.
y	Either (1) a single integer identifying the column of x containing the response (requires source to be specified), (2) a vector of integers identifying which columns of x are the responses (simultaneous case).
w	A single integer identifying the column of x containing the observation weights. If this argument is missing, equal weight is assumed.
source	When y is a single integer, this argument identifies the column of x which indexes the different sources. Disregard if y is a vector or list of vectors.
rho	Power used for the mean-variance relation of the Tweedie distribution. Possible range is [1,2], default is 1.5.
nlambda	The length of the regularization path. Disregarded if lambda is specified, default if 100.

lambda.min	The fraction of the first regularization parameter (which is computed to be the smallest such that no predictors are included) defining the last regularization parameter. Disregarded if lambda is specified; possible range is (0,1), default is 1e-3.
lambda	(Optional) User specified sequence of regularization parameter with positive values. When omitted, the sequence is computed starting from the smallest value excluding all predictors from the model and decreasing to a fraction lambda.min of that starting value by logarithmic decrements.
nfolds	Number of folds in the k-folds CV. Default is 10.
kktstop	Logical flag for using the KKT conditions to stop the fit before the end of the regularization parameter sequence. Default is FALSE. Only applies to the preliminary fit.
foldid	(Optional) A list of vector of values between 1 and nfolds identifying what fold each observation is in. If supplied, nfolds can be missing. If missing, the folds are constructed randomly.
adaptive	Exponent of the adaptive penalty weights; suggested value is 1 (See reference for details.) When the argument is 0, no adaptation is performed.
x.normalize	Logical flag for standardization of the predictors prior to fitting the model. If TRUE, each predictors in each source is centered to zero and scaled to variance 1. After the fit of the model, the coefficients are returned on the original scale. Default is FALSE.
reg	Either "Linf" for using L_∞ -regularization in the fit or "L2" for the L_2 -regularization. Default is "Linf".
eps	Convergence threshold. Default is 5e-4.
sr	Logical flag for using the strong rule in the fit. Default is TRUE.
maxit	Maximum number of inner-loop iterations. Default is 10,000.
pf	Penalty weights in the penalty term by feature. Mostly used internally when the Adaptive Lasso is used in cross-validation. Expects a vector of length nvars, default is 1.
alpha	Parameter controlling the balance between across-feature and within-feature sparsity in the penalty term
	$(1 - \alpha) \beta _q + \alpha \beta _1.$
	Possible range is [0,1], default is 0.
...	Further arguments to be passed to MSTweedie.

Details

The function runs the MSTweedie function on the first dataset to get the sequence of regularization parameter and the coefficient estimates. Then, it performs CV along the solution path and computes the out-of-sample Tweedie deviance based on the predicted responses. For each value of the regularization parameter, the error is averaged over the number of folds used and the standard error is computed.

Value

lambda	A vector containing the sequence of regularization parameters.
cvm	A vector containing the Tweedie deviance mean across the folds along the solution path.

cvsd	A vector containing the Tweedie deviance standard error across the folds along the solution path.
cvupper	cvm+cvsd.
cvlo	cvm-cvsd.
reg	The type of regularization used in the algorithm.
alpha	The value of the argument alpha used.
name	"Tweedie Deviance", the loss function used.
MSTweedie.fit	The MSTweedie object (fitted on the whole dataset.)
time	Computing time.
lambda.min	Regularization parameter at minimum CV error.
lambda.1se	Largest regularization parameter within one standard error of the minimum.

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

Friedman, J., Hastie, T., Simon, N., Qian, J. and Tibshirani, R. (2017). "glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models." A vignette for R package glmnet. Available from <https://cran.r-project.org/web/packages/glmnet>.

glmnet package.

See Also

[MSTweedie](#), [coef.cv.MSTweedie](#), [plot.cv.MSTweedie](#), [predict.cv.MSTweedie](#)

Examples

```
#import package
library(MSTweedie)

#load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
cv<-cv.MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')
```


kkt.check

*KKT conditions plots for MSTweedie objects***Description**

This functions plots the values of the KKT conditions in the solution path of a MSTweedie object.

Usage

```
kkt.check(fit, eps = fit$eps, cond = c(1, 2),
          from = 1, to = length(fit$lambda), ...)
```

Arguments

<code>fit</code>	MSTweedie object.
<code>eps</code>	Value of the threshold to add to the plots. Default is the convergence threshold of the fit object, i.e. <code>fit\$eps</code> .
<code>cond</code>	A vector of integers between 1 and 4 indicating which conditions to plot. See "Details".
<code>from</code>	The index of the first regularization parameter to include. Default is 1.
<code>to</code>	The index of the last regularization parameter to include. Default is <code>length(fit\$lambda)</code> .
<code>...</code>	Additional graphical parameters to pass to plot.

Details

The reference contains detailed explanation of these KKT conditions.

If code contains 1, then the sequence of values for the non-zero KKT condition is plotted:

$$||U_j|| = \lambda v_j, \beta_j \neq 0.$$

If code contains 2, then the sequence of values for the zero KKT condition is plotted:

$$||U_j|| \leq \lambda v_j, \beta_j = 0.$$

If code contains 3, then the sequence of the detailed values for the non-zero KKT condition is plotted:

$$-U_j/\lambda v_j = 0, \beta_j \neq 0, k \notin M(\beta_j).$$

If code contains 4, then the sequence of the aggregate KKT condition is plotted:

$$\frac{-\sum_{j=1}^p U_j^\top \beta_j}{\sum_{j=1}^p v_j ||\beta_j||} = \lambda.$$

Value

Multiple plots are produced.

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[MSTweedie](#)

Examples

```
# import package
library(MSTweedie)

# load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
fit <- MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')

# plot the first two kkt conditions
par(mfrow=c(2,1))
kkt.check(fit)
```

MSTweedie

Regularization path for the Multi-source sparse Tweedie model

Description

This function fits the sparse Tweedie model on multi-source datasets along a sequence of regularization parameters lambda. The optimization is done by a Fortran95 routine.

Usage

```
MSTweedie(x, y, w, source, rho = 1.5,
  nlambda = 100, lambda.min, lambda, x.normalize = T,
  eps, sr = T, kktstop = F, reg = c("L2", "Linf"),
  alpha = 0, dfmax = nvars + 1, pmax = min(dfmax * 1.2, nvars),
  pf = rep(1, nvars), maxit = 10000)
```

Arguments

x Either (1) a data frame containing the predictors, the responses (identifying the sources either by different columns in the simultaneous case or via an additional index column) and, optionnaly, the observation weights or (2) a list of matrices containing only the predictors (mostly used internally for cross-validation.)

y	Either (1) a single integer identifying the column of x containing the response (requires source to be specified), (2) a vector of integers identifying which columns of x are the responses (simultaneous case) or (3) a list of vector of responses (mostly used internally for cross-validation.)
w	(Optional) Either (1) a single integer identifying the column of x containing the observation weights or (2) a list of vector of weights (mostly used internally for cross-validation.) If this argument is missing, equal weight is assumed.
source	When y is a single integer, this arguments identifies the column of x which indexes the different sources. Disregard is y is a vector or list of vectors.
rho	Power used for the mean-variance relation of the Tweedie distribution. Possible range is [1,2], default is 1.5.
nlambda	The length of the regularization path. Disregarded if lambda is specified, default if 100.
lambda.min	The fraction of the first regularization parameter (which is computed to be the smallest such that no predictors are included) defining the last regularization parameter. Disregarded if lambda is specified; possible range is (0,1), default is 1e-3.
lambda	(Optional) User specified sequence of regularization parameter with positive values. When omitted, the sequence is computed starting from the smallest value excluding all predictors from the model and decreasing to a fraction lambda.min of that starting value by logarithmic decrements.
x.normalize	Logical flag for stadardization of the predictors prior to fitting the model. If TRUE, each predictors in each source is centered to zero and scaled to variance 1. After the fit of the model, the coefficients are returned on the original scale. Default is FALSE.
eps	Convergence threshold. Default is 1e-3.
sr	Logical flag for using the strong rule in the fit. Default is TRUE.
kktstop	Logical flag for using the KKT conditions to stop the fit before the end of the regularization parameter sequence. Default is FALSE.
reg	Either "Linf" for using L_∞ -regularization in the fit or "L2" for the L_2 -regularization. Default is "Linf".
alpha	Parameter controlling the balance between across-feature and within-feature sparsity in the penalty term

$$(1 - \alpha)||\beta||_q + \alpha||\beta||_1.$$

Possible range is [0,1], default is 0.

dfmax	Maximum number of variables included in the model at a single time. Default is nvars+1.
pmax	Limits the number of features ever to be nonzero. The difference with dfmax, is that if, a variable eventually exits the model, it will still be counted here. Default is min(dfmax*1.2, nvars).
pf	Penalty weights in the penalty term by feature. Mostly used internaly when the Adaptive Lasso is used in cross-validation. Expects a vector of length nvars, default is 1.
maxit	Maximum number of inner-loop iterations. Default is 10,000.

Details

The sequence of regularization parameters implies a sequence of models fitted by the IRLS-BSUM algorithm described in the reference. For each value of the parameter, this function yield a model optimizing the penalized Tweedie log-likelihood of multi-source data. The type of sparsity can be controlled by the arguments `reg` and `alpha`.

The computation time is influence by the arguments `eps`, `nlambda`, `lambda.min` (or `lambda`) and `maxit`. Consider ajusting these parameters to speed up computation. Small values of regularization parameters are the often the longest to fit; the `kktstop` argument can stop the algorithm before the end if convergence is judged sufficient in term of KKT conditions.

To pass sources with missing features compared to other sources, simply add a column of zero instead.

Value

An object with S3 class `MSTweedie` :

<code>beta0</code>	A <code>ntaks*nlambda</code> matrix of parameter estimates for the intercept.
<code>beta</code>	A list of length <code>nlambda</code> containing <code>nvars*ntaks</code> matrix of parameter estimates for the features.
<code>df</code>	The number of included variables along the regularization path.
<code>lambda</code>	The sequence of regularization parameters.
<code>npasses</code>	The number of inner-loop iterations.
<code>idvars</code>	The index of the variables in order of inclusion in the model.
<code>dim</code>	The dimesions of the model (<code>nvars</code> , <code>ntasks</code>).
<code>call</code>	The original call that produce this object.
<code>pf</code>	The penalty factors for the features.
<code>eps</code>	The convergence threshold used in the algorithm.
<code>kkt</code>	A <code>nvars*ntasks*nlambda</code> array containing the values of the KKT conditions.
<code>norm</code>	A <code>nvars*nlambda</code> matrix containing the norm of the features along the regularization path.
<code>reg</code>	The type of regularization used in the algorithm.
<code>alpha</code>	The value of the argument <code>alpha</code> used.
<code>y</code>	A list of length <code>ntasks</code> containing the vectors of the responses for each source.
<code>x</code>	A list of length <code>ntasks</code> containing matrices of the features for each source.
<code>w</code>	A list of length <code>ntasks</code> containing the vectors of the observation weights for each source.
<code>rho</code>	The power of the mean-variance relation used in the algorithm.
<code>M</code>	A <code>nvars*ntasks*nlambda</code> array containing flags for the KKT conditions.
<code>time</code>	Computing time.

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[MSTweedie](#), [coef.MSTweedie](#), [print.MSTweedie](#), [plot.MSTweedie](#), [kkt.check](#), [predict.MSTweedie](#)

Examples

```
# import package
library(MSTweedie)

# load data
data(AutoClaim)

# fit the MSTweedie model with L1/Linf regularization
# y=1 sets CLM_AMT5 as the response
# source=4 sets REVOLKED as the source index
fit <- MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')
```

plot.cv.MSTweedie

Plot Cross-Validation curve of cv.MSTweedie object

Description

This function produces a plot of the mean Tweedie deviance against the logarithm of the solution path of a cv.MSTweedie object. The mean is accompanied by a one-standard-error interval. The regularization parameter value minimizing the CV deviance and the largest value within one standard error are indicated by vertical lines.

Usage

```
## S3 method for class 'cv.MSTweedie'
plot(x, sign.lambda = 1, ...)
```

Arguments

x	cv.MSTweedie object.
sign.lambda	Whether to plot from left-to-right or the opposite. The Default is 1 corresponding to right-to-left.
...	Additional parameters passed to plot .

Value

A plot is produced

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[cv.MSTweedie](#), [plot](#)

Examples

```
# import package
library(MSTweedie)

# load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
cv <- cv.MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')

# plot CV deviance mean and std. err., and lambda.min, lambda.1se
plot.cv.MSTweedie(cv)
```

plot.MSTweedie	<i>Plot the solution path of a MSTweedie object</i>
----------------	---

Description

This function produces coefficient profile plot(s) along the sequence of regularization parameters of a MSTweedie object.

Usage

```
## S3 method for class 'MSTweedie'
plot(x, log.lambda = TRUE, type.coef = c("coef", "norm"),
      lambda.min = NA, lambda.1se = NA, ...)
```

Arguments

x	MSTweedie object.
log.lambda	Logical flag: whether to plot against the logarithm of the regularization parameters (default) or against the reg. parameters.
type.coef	Whether to plot the coefficients in each source ("coef") or to plot the norm across all sources ("norm"), where the norm is given by x\$reg.
lambda.min	If supplied, this traces a vertical line at lambda.min (given by CV.)
lambda.1se	If supplied, this traces a vertical line at lambda.1se (given by CV.)
...	Additional parameters passed to plot .

Value

Plot(s) are produced.

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[MSTweedie](#), [plot](#), [coef.MSTweedie](#)

Examples

```
# import package
library(MSTweedie)

# load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
fit <- MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')

# plot solution path of the norm of the coefficients
plot.MSTweedie(fit, type.coef='norm')

# plot solution path of the the coefficients
par(mfrow=c(2,1))
plot.MSTweedie(fit, type.coef='coef')
```

predict.cv.MSTweedie *Prediction based on a cv.MSTweedie object*

Description

This function is a wrapper function for the predict.MSTweedie when applied to a cv.MSTweedie object.

Usage

```
## S3 method for class 'cv.MSTweedie'
predict(object, newx, s = c("lambda.1se", "lambda.min"))
```

Arguments

<code>object</code>	cv.MSTweedie object.
<code>newx</code>	A list of matrices all with <code>nvars</code> columns used to predict the responses. Each matrix is associated with a source from the original object MSTweedie object in the order of argument tasks. Default is the original dataset (<code>object\$MSTweedie.fit\$x</code>).
<code>s</code>	Either a vector of regularization parameters (must match those of <code>object</code>), a vector of indices of regularization parameters. Default is the whole solution path or one of "lambda.min" or "lambda.1se". Default is "lambda.1se".

Value

A list of length `ntasks` of vectors corresponding to the prediction of either the responses or the link function (see `type`).

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[cv.MSTweedie](#), [predict.MSTweedie](#)

Examples

```
# import package
library(MSTweedie)

# load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
cv <- cv.MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')

# extract coefficients at lambda.1se
head(predict.cv.MSTweedie(cv)[[1]])
```

predict.MSTweedie	<i>Prediction based on a MSTweedie object</i>
-------------------	---

Description

This function predicts the response (or the link function) given features at some regularization parameter.

Usage

```
## S3 method for class 'MSTweedie'
predict(fit, newx, tasks, s, type = c("response", "link"))
```

Arguments

fit	MSTweedie object.
newx	A list of matrices all with <code>nvars</code> columns used to predict the responses. Each matrix is associated with a source from the original <code>fit</code> MSTweedie object in the order of argument <code>tasks</code> . Default is the original dataset (<code>fit\$x</code>).
tasks	A vector of the same length as <code>newx</code> containing the index of the source associated with the respective matrix in <code>newx</code> . Default is all sources in original order.
s	Either an integer identifying the index of regularization parameter to use or the parameter itself (must match one of <code>fit</code> .)
type	Whether to return the "response" or the <code>"link"</code> (linear predictor).

Value

A list of length `length(tasks)` of vectors corresponding to the prediction of either the responses or the link function (see `type`).

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.

Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[MSTweedie](#), [coef.MSTweedie](#)

Examples

```
#import package
library(MSTweedie)

#load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
fit <- MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')

# predict first source at 34th lambda
head(predict.MSTweedie(fit, s=34L)[[1]])
```

print.MSTweedie	<i>Print the results of a MSTweedie object</i>
-----------------	--

Description

This function prints the number of variables included in the model and the value of the regularization parameter along the solution path.

Usage

```
## S3 method for class 'MSTweedie'
print(x, digits = max(3, getOption("digits") - 3))
```

Arguments

x	MSTweedie object.
digits	Number of significant digits in print display.

Value

A two column matrix with nlambda rows containing the number of variables included in each model with the value of the regularization parameter.

Author(s)

Simon Fontaine, Yi Yang, Bo Fan, Wei Qian and Yuwen Gu.
 Maintainer: Simon Fontaine <fontaines@dms.umontreal.ca>

References

Fontaine, S., Yang, Y., Fan, B., Qian, W. and Gu, Y. (2018). "A Unified Approach to Sparse Tweedie Model with Big Data Applications to Multi-Source Insurance Claim Data Analysis," to be submitted.

See Also

[MSTweedie](#)

Examples

```
# import package
library(MSTweedie)

# load data
data(AutoClaim)

# performs 10-folds CV with L1/Linf regularization
fit <- MSTweedie(x = AutoClaim, y=1, source=4, reg='Linf')

# prints number of selected variables along solution path
print.MSTweedie(fit)
```

Index

*Topic **datasets**

AutoClaim, [2](#)

AutoClaim, [2](#)

coef.cv.MSTweedie, [4](#), [8](#)

coef.MSTweedie, [4](#), [5](#), [13](#), [15](#), [17](#)

cplm, [2](#)

cv.MSTweedie, [4](#), [6](#), [14](#), [16](#)

glmnet::cv, [6](#)

kkt.check, [9](#), [13](#)

MSTweedie, [6](#), [8](#), [10](#), [10](#), [13](#), [15](#), [17](#), [18](#)

plot, [13–15](#)

plot.cv.MSTweedie, [8](#), [13](#)

plot.MSTweedie, [6](#), [13](#), [14](#)

predict.cv.MSTweedie, [8](#), [15](#)

predict.MSTweedie, [6](#), [13](#), [16](#), [17](#)

print.MSTweedie, [13](#), [18](#)

rrcovNA::impSeq, [2](#)