Package 'metafuse'

July 22, 2025

Type Package

Title Fused Lasso Approach in Regression Coefficient Clustering

Version 2.0-1

Date 2016-10-16

Author Lu Tang, Ling Zhou, Peter X.K. Song

Maintainer Lu Tang <lutang@umich.edu>

Description Fused lasso method to cluster and estimate regression coefficients of the same covariate across different data sets when a large number of independent data sets are combined. Package supports Gaussian, binomial, Poisson and Cox PH models.

License GPL-2

Imports glmnet, Matrix, MASS, evd

Repository CRAN

NeedsCompilation no

Date/Publication 2016-10-17 00:37:55

Contents

`metafuse-package` *Fused Lasso Approach in Regression Coefficient Clustering*

Description

Fused lasso method to cluster and estimate regression coefficients of the same covariate across different data sets when a large number of independent data sets are combined. Package supports Gaussian, binomial, Poisson and Cox PH models.

Details

Simple to use. Accepts `X`, `y`, and `sid` (numerica data source ID for which data entry belongs to) for regression models. Returns regression coefficient estimates and clusterings patterns of coefficients across different datasets, for each covariate. Provides visualization by fusogram, a dendrogram-type of presentation of coefficient clustering pattern across data sources.

Author(s)

Lu Tang, Ling Zhou, Peter X.K. Song
Maintainer: Lu Tang <lutang@umich.edu>

References

Lu Tang, and Peter X.K. Song. Fused Lasso Approach in Regression Coefficients Clustering - Learning Parameter Heterogeneity in Data Integration. *Journal of Machine Learning Research*, 17(113):1-23, 2016.

Fei Wang, Lu Wang, and Peter X.K. Song. Fused lasso with the adaptation of parameter ordering in combining multiple studies with repeated measurements. *Biometrics*, DOI:10.1111/biom.12496, 2016.

Examples

```
########### generate data ###########
n <- 200    # sample size in each dataset (can also be a K-element vector)
K <- 10     # number of datasets for data integration
p <- 3      # number of covariates in X (including the intercept)

# the coefficient matrix of dimension K * p, used to specify the heterogeneous pattern
beta0 <- matrix(c(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,   # beta_0 of intercept
                  0.0,0.0,0.0,0.0,0.0,1.0,1.0,1.0,1.0,1.0,   # beta_1 of X_1
                  0.0,0.0,0.0,0.0,0.5,0.5,0.5,1.0,1.0,1.0),  # beta_2 of X_2
                K, p)

# generate a data set, family=c("gaussian", "binomial", "poisson", "cox")
data <- datagenerator(n=n, beta0=beta0, family="gaussian", seed=123)
```

```
# prepare the input for metafuse
y       <- data$y
sid     <- data$group
X       <- data[,-c(1,ncol(data))]

########### run metafuse ###########
# fuse slopes of X1 (which is heterogeneous with 2 clusters)
metafuse(X=X, y=y, sid=sid, fuse.which=c(1), family="gaussian", intercept=TRUE, alpha=0,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)

# fuse slopes of X2 (which is heterogeneous with 3 clusters)
metafuse(X=X, y=y, sid=sid, fuse.which=c(2), family="gaussian", intercept=TRUE, alpha=0,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)

# fuse all three covariates
metafuse(X=X, y=y, sid=sid, fuse.which=c(0,1,2), family="gaussian", intercept=TRUE, alpha=0,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)

# fuse all three covariates, with sparsity penalty
metafuse(X=X, y=y, sid=sid, fuse.which=c(0,1,2), family="gaussian", intercept=TRUE, alpha=1,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)

# fit metafuse at a given lambda
metafuse.l(X=X, y=y, sid=sid, fuse.which=c(0,1,2), family="gaussian", intercept=TRUE,
          alpha=1, lambda=0.5)
```

`datagenerator` *simulate data*

Description

Simulate a dataset with data from K different sources, for demonstration of `metafuse`.

Usage

```
datagenerator(n, beta0, family, seed = NA)
```

Arguments

`n`	a vector of length K (the total number of datasets being integrated), specifying the sample sizes of individual datasets; can also be an scalar, in which case the function simulates K datasets of equal sample size
`beta0`	a coefficient matrix of dimension K * p, where K is the number of datasets being integrated and p is the number of covariates, including the intercept
`family`	the type of the response vector, `c("gaussian", "binomial", "poisson", "cox")`; `"gaussian"` for continuous response, `"binomial"` for binary response, `"poisson"` for count response, `"cox"` for observed time-to-event response, with censoring indicator
`seed`	the random seed for data generation, default is NA

Details

These datasets are artifical, and are used to demonstrate the features of `metafuse`. In the case when `family="cox"`, the response will contain two vectors, a time-to-event variable `time` and a censoring indicator `status`.

Value

Returns data frame with `n*K` rows (if `n` is a scalar), or `sum(n)` rows (if `n` is a `K`-element vector). The data frame contains columns "y", "x1", ..., "x_p-1" and "group" if `family="gaussian"`, `"binomial"` or `"poisson"`; or contains columns "time", "status", "x1", ..., "x_p-1" and "group" if `family="cox"`.

Examples

```
########### generate data ###########
n <- 200    # sample size in each dataset (can also be a K-element vector)
K <- 10     # number of datasets for data integration
p <- 3      # number of covariates in X (including the intercept)

# the coefficient matrix of dimension K * p, used to specify the heterogeneous pattern
beta0 <- matrix(c(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,   # beta_0 of intercept
                  0.0,0.0,0.0,0.0,0.0,1.0,1.0,1.0,1.0,1.0,   # beta_1 of X_1
                  0.0,0.0,0.0,0.0,0.5,0.5,0.5,1.0,1.0,1.0),  # beta_2 of X_2
                K, p)

# generate a data set, family=c("gaussian", "binomial", "poisson", "cox")
data <- datagenerator(n=n, beta0=beta0, family="gaussian", seed=123)
names(data)

# if family="cox", returned dataset contains columns "time"" and "status" instead of "y"
data <- datagenerator(n=n, beta0=beta0, family="cox", seed=123)
names(data)
```

`metafuse` *fit a GLM with fusion penalty for data integraion*

Description

Fit a GLM with fusion penalty on coefficients within each covariate across datasets, generate solution path and fusograms for visualization of the model selection.

Usage

```
metafuse(X = X, y = y, sid = sid, fuse.which = c(0:ncol(X)),
  family = "gaussian", intercept = TRUE, alpha = 0, criterion = "EBIC",
  verbose = TRUE, plots = FALSE, loglambda = TRUE)
```

Arguments

X	a matrix (or vector) of predictor(s), with dimensions of N*(p-1), where N is the total sample size of the integrated dataset
y	a vector of response, with length N; when family="cox", y is a data frame with cloumns time and status
sid	data source ID of length N, must contain integers numbered from 1 to K
fuse.which	a vector of integers from 0 to p-1, indicating which covariates are considered for fusion; 0 corresponds to the intercept; coefficients of covariates not in this vector are homogeneously estimated across all datasets
family	response vector type, "gaussian" if y is a continuous vector, "binomial" if y is binary vector, "poisson" if y is a count vector, "cox" if y is a data frame with cloumns time and status
intercept	if TRUE, intercept will be included, default is TRUE
alpha	the ratio of sparsity penalty to fusion penalty, default is 0 (i.e., no variable selection, only fusion)
criterion	"AIC" for AIC, "BIC" for BIC, "EBIC" for extended BIC,default is "BIC"
verbose	if TRUE, outputs whenever a fusion event happens, and returns the current value of lambda, default is TRUE
plots	if TRUE, create solution paths and fusogram plots to visualize the clustering of regression coefficients across datasets, default is FALSE
loglambda	if TRUE, lambda will be plotted in log-10 scale, default is TRUE

Details

Adaptive lasso penalty is used. See Zou (2006) for detail.

Value

A list containing the following items will be returned:

family	the response/model type
criterion	model selection criterion used
alpha	the ratio of sparsity penalty to fusion penalty
if.fuse	whether covariate is assumed to be heterogeneous (1) or homogeneous (0)
betahat	the estimated regression coefficients
betainfo	additional information about the fit, including degree of freedom, optimal lambda value, maximum lambda value to fuse all coefficients, and estimated friction of fusion

References

Lu Tang, and Peter X.K. Song. Fused Lasso Approach in Regression Coefficients Clustering - Learning Parameter Heterogeneity in Data Integration. *Journal of Machine Learning Research*, 17(113):1-23, 2016.

Fei Wang, Lu Wang, and Peter X.K. Song. Fused lasso with the adaptation of parameter ordering in combining multiple studies with repeated measurements. *Biometrics*, DOI:10.1111/biom.12496, 2016.

Examples

```
########### generate data ###########
n <- 200    # sample size in each dataset (can also be a K-element vector)
K <- 10     # number of datasets for data integration
p <- 3      # number of covariates in X (including the intercept)

# the coefficient matrix of dimension K * p, used to specify the heterogeneous pattern
beta0 <- matrix(c(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,   # beta_0 of intercept
                  0.0,0.0,0.0,0.0,0.0,1.0,1.0,1.0,1.0,1.0,   # beta_1 of X_1
                  0.0,0.0,0.0,0.0,0.5,0.5,0.5,1.0,1.0,1.0),  # beta_2 of X_2
                K, p)

# generate a data set, family=c("gaussian", "binomial", "poisson", "cox")
data <- datagenerator(n=n, beta0=beta0, family="gaussian", seed=123)

# prepare the input for metafuse
y       <- data$y
sid     <- data$group
X       <- data[,-c(1,ncol(data))]

########### run metafuse ###########
# fuse slopes of X1 (which is heterogeneous with 2 clusters)
metafuse(X=X, y=y, sid=sid, fuse.which=c(1), family="gaussian", intercept=TRUE, alpha=0,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)

# fuse slopes of X2 (which is heterogeneous with 3 clusters)
metafuse(X=X, y=y, sid=sid, fuse.which=c(2), family="gaussian", intercept=TRUE, alpha=0,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)

# fuse all three covariates
metafuse(X=X, y=y, sid=sid, fuse.which=c(0,1,2), family="gaussian", intercept=TRUE, alpha=0,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)

# fuse all three covariates, with sparsity penalty
metafuse(X=X, y=y, sid=sid, fuse.which=c(0,1,2), family="gaussian", intercept=TRUE, alpha=1,
          criterion="EBIC", verbose=TRUE, plots=TRUE, loglambda=TRUE)
```

`metafuse.l` *fit a GLM with fusion penalty for data integraion, with a fixed lambda*

Description

Fit a GLM with fusion penalty on coefficients within each covariate at given lambda.

Usage

```
metafuse.l(X = X, y = y, sid = sid, fuse.which = c(0:ncol(X)),
  family = "gaussian", intercept = TRUE, alpha = 0, lambda = lambda)
```

Arguments

`X`	a matrix (or vector) of predictor(s), with dimensions of `N*(p-1)`, where `N` is the total sample size of the integrated dataset
`y`	a vector of response, with length `N`; when `family="cox"`, `y` is a data frame with cloumns `time` and `status`
`sid`	data source ID of length `N`, must contain integers numbered from 1 to `K`
`fuse.which`	a vector of integers from 0 to `p-1`, indicating which covariates are considered for fusion; 0 corresponds to the intercept; coefficients of covariates not in this vector are homogeneously estimated across all datasets
`family`	response vector type, `"gaussian"` if `y` is a continuous vector, `"binomial"` if `y` is binary vector, `"poisson"` if `y` is a count vector, `"cox"` if `y` is a data frame with cloumns `time` and `status`
`intercept`	if `TRUE`, intercept will be included, default is `TRUE`
`alpha`	the ratio of sparsity penalty to fusion penalty, default is 0 (i.e., no variable selection, only fusion)
`lambda`	tuning parameter for fusion penalty

Details

Adaptive lasso penalty is used. See Zou (2006) for detail.

Value

A list containing the following items will be returned:

`family`	the response/model type
`alpha`	the ratio of sparsity penalty to fusion penalty
`if.fuse`	whether covariate is assumed to be heterogeneous (1) or homogeneous (0)
`betahat`	the estimated regression coefficients
`betainfo`	additional information about the fit, including degree of freedom, optimal lambda value, maximum lambda value to fuse all coefficients, and estimated friction of fusion

References

Lu Tang, and Peter X.K. Song. Fused Lasso Approach in Regression Coefficients Clustering - Learning Parameter Heterogeneity in Data Integration. *Journal of Machine Learning Research*, 17(113):1-23, 2016.

Fei Wang, Lu Wang, and Peter X.K. Song. Fused lasso with the adaptation of parameter ordering in combining multiple studies with repeated measurements. *Biometrics*, DOI:10.1111/biom.12496, 2016.

Examples

```
########### generate data ###########
n <- 200    # sample size in each dataset (can also be a K-element vector)
K <- 10     # number of datasets for data integration
p <- 3      # number of covariates in X (including the intercept)

# the coefficient matrix of dimension K * p, used to specify the heterogeneous pattern
beta0 <- matrix(c(0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,   # beta_0 of intercept
                  0.0,0.0,0.0,0.0,0.0,1.0,1.0,1.0,1.0,1.0,   # beta_1 of X_1
                  0.0,0.0,0.0,0.0,0.5,0.5,0.5,1.0,1.0,1.0),  # beta_2 of X_2
                K, p)

# generate a data set, family=c("gaussian", "binomial", "poisson", "cox")
data <- datagenerator(n=n, beta0=beta0, family="gaussian", seed=123)

# prepare the input for metafuse
y       <- data$y
sid     <- data$group
X       <- data[,-c(1,ncol(data))]

########### run metafuse ###########
# fit metafuse at a given lambda
metafuse.l(X=X, y=y, sid=sid, fuse.which=c(0,1,2), family="gaussian", intercept=TRUE,
          alpha=1, lambda=0.5)
```

Index