Package ‘mcen’

July 23, 2020

Type Package

Title Multivariate Cluster Elastic Net

Version 1.2

Date 2020-07-22

Author Ben Sherwood [aut, cre], Brad Price [aut]

Depends R (>= 3.0.0), glmnet, flexclust, Matrix, parallel, faraway, methods, stats

Maintainer Ben Sherwood <ben.sherwood@ku.edu>


ByteCompile TRUE

License MIT + file LICENSE

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-07-23 14:30:03 UTC

R topics documented:

beta_adjust .............................................................. 2
beta_adjust_bin .......................................................... 3
bin_horse ................................................................. 3
CalcHorseBin ............................................................ 4
CalcHorseEBin ........................................................... 5
cluster ................................................................. 5
cluster.vals ............................................................. 6
coeff.cv.mcen .......................................................... 7
coeff.mcen ............................................................. 7
cv.mcen ................................................................. 8
beta_adjust

Adjusts the value of the coefficients to account for the scaling of x and y.

Usage

beta_adjust(beta, sigma_x, sigma_y, mean_x, mean_y)

Arguments

- **beta**: The estimate of beta with scaled data.
- **sigma_x**: Sample standard deviations of the original predictors.
- **sigma_y**: Sample standard deviations of the original responses.
- **mean_x**: Sample means of the original predictors.
- **mean_y**: Sample means of the original responses.

Value

Returns the adjusted coefficients

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>
**beta_adjust_bin**

Adjusts the value of the binomial coefficients to account for the scaling of \(x\).

**Description**

Adjusts the value of the binomial coefficients to account for the scaling of \(x\).

**Usage**

\[
\text{beta_adjust_bin}(\beta, \sigma_x)
\]

**Arguments**

- \(\beta\): The estimate of \(\beta\) with scaled data.
- \(\sigma_x\): Sample standard deviations of the original predictors.

**Value**

Returns the adjusted coefficients.

**Author(s)**

Ben Sherwood &lt;ben.sherwood@ku.edu&gt;, Brad Price &lt;brad.price@mail.wvu.edu&gt;

**bin_horse**

The workhorse function for the binomial updates in mcen. It uses IRWLS glmnet updates to solve the regression problem.

**Description**

The workhorse function for the binomial updates in mcen. It uses IRWLS glmnet updates to solve the regression problem.

**Usage**

\[
\text{bin_horse}(Y, X, \delta, \gamma_y, y_{clusters}, set\_length, \epsilon, \text{maxiter})
\]
CalcHorseBin

Arguments

- Y: the matrix of responses
- X: the matrix of predictors with the intercept included
- delta: the tuning parameter for the lasso penalty
- gamma_y: the tuning parameter for the ridge fusion penalty
- y_clusters: the cluster assignments from the provided clustering algorithm
- set_length: the size of each cluster corresponding to a given response. r dimensions with each element containing the cluster size of that responses cluster.
- eps: the tolerance for conversion normally 1e-5
- maxiter: the maximum number of iterations

Value

Returns a matrix of coefficients

Author(s)

Brad Price <brad.price@mail.wvu.edu>

CalcHorseBin

Creates the working response for all responses for glmnet binomial family

Description

Creates the working response for all responses for glmnet binomial family

Usage

CalcHorseBin(Y, X, Beta)

Arguments

- Y: is the matrix of responses result is the list of vectors needed for the working responses in glmnet
- X: the matrix of predictors.
- Beta: current iteration of the regression coefficients

Author(s)

Brad Price <brad.price@mail.wvu.edu>
**CalcHorseEBin**

*Creates the probabilities and working response for the glmnet update for a given response with a binomial family*

**Description**

Creates the probabilities and working response for the glmnet update for a given response with a binomial family.

**Usage**

```
CalcHorseEBin(X, Beta, Y, r)
```

**Arguments**

- **X**: the matrix of predictors.
- **Beta**: current iteration of the regression coefficients.
- **Y**: is the matrix of responses.
- **r**: the response of interest. Result is a list of things needed for the working response in glmnet.

**Author(s)**

Brad Price <brad.price@mail.wvu.edu>

---

**cluster**

*Wrapper function for different clustering methods*

**Description**

Wrapper function for different clustering methods.

**Usage**

```
cluster(x, cNum, clusterMethod = "kmeans", clusterIterations = 100,
        clusterStartNum = 30)
```

**Arguments**

- **x**: data to be clustered. Clustering will be done on the columns.
- **cNum**: number of cluster centers.
- **clusterMethod**: "kmean" for kmeans function, "kmeanspp" for kcca implementation of kmeans++.
- **clusterIterations**: number of maximum iterations for clustering.
- **clusterStartNum**: random number of starting points used.
cluster.vals

Description

Returns the cluster values from a cv.mcen object.

Usage

cluster.vals(obj)

Arguments

obj The cv.mcen object.

Value

Returns the clusters from the model with the smallest cross-validation error.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

Examples

x <- matrix(rnorm(400),ncol=4)
beta <- beta <- matrix(c(1,1,0,0,0,-1,0,0,-1,1,0,0,-1,1,0,0),ncol=4)
y <- x%*%beta + rnorm(400)
mcen_fit <- cv.mcen(x,y,ky=2,gamma_y=3)
mcen_cluster <- cluster.vals(mcen_fit)
coef.cv.mcen

Returns the coefficients from the cv.mcen object with the smallest cross-validation error.

Description

Returns the coefficients from the cv.mcen object with the smallest cross-validation error.

Usage

## S3 method for class 'cv.mcen'
coef(object, ...)

Arguments

object The cv.mcen object.
...

Additional values to be passed.

Value

The matrix of coefficients for the best MCEN model as determined by cross-validation.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

Examples

x <- matrix(rnorm(400),ncol=4)
beta <- matrix(c(1,1,0,0,0,-1,0,0,-1,-1,1,1,0,0),ncol=4)
y <- x%*%beta + rnorm(400)
mcen_fit <- cv.mcen(x,y,ky=2,gamma_y=3)
best_coef <- coefficients(mcen_fit)

coef.mcen

Returns the coefficients from an mcen object.

Description

Returns the coefficients from an mcen object.

Usage

## S3 method for class 'mcen'
coef(object, delta = NULL, ...)

Examples

x <- matrix(rnorm(400),ncol=4)
beta <- matrix(c(1,1,0,0,0,-1,0,0,-1,-1,1,1,0,0),ncol=4)
y <- x%*%beta + rnorm(400)
mcen_fit <- cv.mcen(x,y,ky=2,gamma_y=3)
best_coef <- coefficients(mcen_fit)
Arguments

- **object**: The mcen object.
- **delta**: The L1 tuning parameter
- **...**: Additional values to pass on.

Value

The matrix of coefficients.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

Examples

```r
x <- matrix(rnorm(400),ncol=4)
beta <- beta <- matrix(c(1,1,0,0,0,0,-1,0,-1,0,-1,0,1,1,0,0),ncol=4)
y <- x%*%beta + rnorm(400)
mcen_fit <- mcen(x,y,ky=2,gamma_y=3,delta=c(1,2))
best_coef <- coefficients(mcen_fit,delta=1)
```

cv.mcen

Cross validation for mcen function

Description

Cross validation for mcen function

Usage

```r
cv.mcen(x, y, family = "mgaussian", ky = seq(2, 4), gamma_y = seq(0.1, 5.1, 0.5), nfolds = 10, folds = NULL, cluster_y = NULL, delta=NULL, n.cores = 1, ...)
```

Arguments

- **x**: Matrix set of predictors.
- **y**: Matrix set of responses.
- **family**: The exponential family the response corresponds to.
- **ky**: A vector with the number of possible clusters for y.
- **gamma_y**: Set of tuning parameter for clustering penalty in response categories.
- **nfolds**: Number of folds used in the cross-validation.
- **folds**: A vector of length n, where this identifies what fold of the kfold cross validation each observation belongs to.
get_best_cvm

- `cluster_y`  
  a priori definition of clusters. If clusters are provided they will remain fixed and are not estimated. Objective function is then convex.

- `delta`  
  Tuning parameter for the L1 penalty

- `n.cores`  
  Number of cores used for parallel processing.

- `...`  
  The variables passed to mcen

**Value**

Returns a cv.mcen object.

- `models`  
  A list of mcen objects.

- `cv`  
  Cross validation results.

- `ky`  
  The same value as the input ky.

- `gamma_y`  
  The same value as the input gamma_y.

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

**References**


**Examples**

```r
x <- matrix(rnorm(400), ncol=4)
beta <- matrix(c(1,1,0,0,0,0,-1,0,-1,0,-1,0,1,1,0,0), ncol=4)
y <- x%*%beta + rnorm(400)
cv_fit <- cv.mcen(x, y, ky=2)
```

---

**get_best_cvm**  
*Gets the index position for the model with the smallest cross-validation error.*

**Description**

Gets the index position for the model with the smallest cross-validation error.

**Usage**

```r
get_best_cvm(model)
```

**Arguments**

- `model`  
  The cv.mcen object.
matrix_multiply

Value

Returns the index for the model with the smallest cross-validation error.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

Examples

```r
x <- matrix(rnorm(400),ncol=4)
beta <- matrix(c(1,1,0,0,0,-1,0,-1,-1,1,0,0),ncol=4)
y <- x %*% beta + rnorm(400)
mcen_fit <- cv.mcen(x,y,ky=2,gamma_y=3)
get_best_cvm(mcen_fit)
```

matrix_multiply

Description

matrix multiply

Usage

`matrix_multiply(beta, x)`

Arguments

- `beta`: Matrix of coefficients.
- `x`: Design matrix.

Value

Returns x times beta

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>
mcen

Description

Fits an MCEN model

Usage

mcen(x, y, family = "mgaussian", ky = NULL, delta = NULL, gamma_y = 1,
ndelta = 25, delta.min.ratio = NULL, eps = 1e-05,
scale_x = TRUE, scale_y = TRUE, clusterMethod = "kmeans",
clusterStartNum = 30, clusterIterations = 10, cluster_y = NULL,
max_iter = 10, init_beta = NULL, n.cores = 1)

Arguments

x Matrix of predictors.
y Matrix of responses.
family Type of likelihood used two options "mgaussian" or "mbinomial".
ky Clusters for response.
delta L1 penalty.
gamma_y Penalty for with y clusters difference in predicted values.
ndelta Number of delta parameters.
delta.min.ratio Ratio between smallest and largest delta.
eps Convergence criteria.
scale_x Whether x matrix should be scaled, default is True.
scale_y Whether y matrix should be scaled, default is True.
clusterMethod K-means function used kmeans or kmeanspp.
clusterStartNum Number of random starting points for clustering.
clusterIterations Number of iterations for cluster convergence.
cluster_y An a priori definition of clusters. If clusters are provided they will remain fixed and are not estimated. Objective function is then convex.
max_iter Maximum number of iterations for coefficient estimates.
init_beta Clustering step requires an initial estimate, default is to use elastic net solution.
n.cores Number of cores used for calculation default is 1.
mcen.init

Value
returns a MCEN object

beta List of the coefficient estimates.
delta Value of delta.
gamma_y Value of gamma_y.
ky Value of ky.
y_clusters List of the clusters of y.

Author(s)
Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

References

Examples
x <- matrix(rnorm(400),ncol=4)
beta <- matrix(c(1,1,0,0,0,-1,0,0,-1,1),ncol=4)
y <- x%*%beta + rnorm(400)
mcen_fit <- mcen(x,y,ky=2,delta=1)

mcen.init Provides initial estimates for the mcen functionF

Description
Provides initial estimates for the mcen functionF

Usage
mcen.init(x, y, family = "mgaussian", delta = NULL, gamma_y = 1,
intercept = FALSE)

Arguments
x the n x p design matrix
y the n x y matrix of responses
family type of likelihood used two options "mgaussian" or "mbinomial"
delta sparsity tuning parameter
gamma_y tuning parameter for clustering responses
intercept whether an intercept should be included in the model
**mcen_bin_workhorse**

**Value**

matrix of coefficients

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

---

**mcen_bin_workhorse**

*Calculates cluster assignment and coefficient estimates for a binomial mcen.*

---

**Description**

Calculates cluster assignment and coefficient estimates for a binomial mcen.

**Usage**

```r
mcen_bin_workhorse(beta, delta = NULL, y, x, family = "mbinomial",
  ky = NULL, gamma_y = 1, eps = 1e-05, clusterMethod = "kmeans",
  clusterIterations = 100, clusterStartNum = 30, cluster_y = NULL,
  max_iter = 10)
```

**Arguments**

- **beta**: Initial estimate of coefficients.
- **delta**: Tuning parameter for L1 penalty.
- **y**: Matrix of responses.
- **x**: Matrix of predictors.
- **family**: type of likelihood used two options "mgaussian" or "mbinomial"
- **ky**: Number of clusters used for grouping response variables.
- **gamma_y**: Tuning parameter for the penalty between fitted values for responses in the same group.
- **eps**: Convergence criteria
- **clusterMethod**: Which clustering method was used, currently support kmeans or kmeanspp
- **clusterIterations**: Number of iterations for cluster convergence
- **clusterStartNum**: Number of random starting points for clustering
- **cluster_y**: An a priori definition of clusters. If clusters are provided they will remain fixed and are not estimated. Objective function is then convex.
- **max_iter**: The maximum number of iterations for estimating the coefficients

**Author(s)**

Brad Price <brad.price@mail.wvu.edu>
**mcen_workhorse**

Estimates the clusters and provides the coefficients for an mcen object

**Description**

Estimates the clusters and provides the coefficients for an mcen object

**Usage**

```r
mcen_workhorse(beta, delta = NULL, xx, xy, family = "mgaussian", ky = NULL, gamma_y = 0.5, eps = 1e-05, clusterMethod = "kmeans", clusterIterations = 100, clusterStartNum = 30, cluster_y = NULL, max_iter = 10, x = x)
```

**Arguments**

- `beta`: The initial value of the coefficients
- `delta`: The sparsity (L1) tuning parameter
- `xx`: Matrix of transpose of x times x.
- `xy`: Matrix of transpose of x times y.
- `family`: Type of likelihood used two options "mgaussian" or "mbinomial"
- `ky`: Number of clusters for the response
- `gamma_y`: Penalty for the y clusters difference in predicted values
- `eps`: Convergence criteria
- `clusterMethod`: Which clustering method was used, currently support kmeans or kmeanspp
- `clusterIterations`: Number of iterations for cluster convergence
- `clusterStartNum`: Number of random starting points for clustering
- `cluster_y`: An a priori definition of clusters. If clusters are provided they will remain fixed and are not estimated. Objective function is then convex.
- `max_iter`: The maximum number of iterations for estimating the coefficients
- `x`: The design matrix

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>
predict.cv.mcen

Makes predictions from the model with the smallest cross-validation error.

Description

Makes predictions from the model with the smallest cross-validation error.

Usage

## S3 method for class 'cv.mcen'
predict(object, newx, ...)

Arguments

object The cv.mcen object.
newx The X matrix of predictors.
... Additional parameters to be sent to predict.

Value

Returns the predicted values from the model with the smallest cross-validation error.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

Examples

x <- matrix(rnorm(400),ncol=4)
beta <- beta <- matrix(c(1,1,0,0,0,-1,-1,0,0,-1,1,1,0,0),ncol=4)
y <- x%*%beta + rnorm(400)
mcen_fit <- cv.mcen(x,y,ky=2,gamma_y=3)
new_x <- matrix(rnorm(12),ncol=4)
mcen_preds <- predict(mcen_fit, new_x)
predict.mcen

predictions from a mcen model

Description

predictions from a mcen model

Usage

## S3 method for class 'mcen'
predict(object, newx, ...)

Arguments

object       The mcen object.
newx         A matrix of new observations.
...           Additional variables to be sent to predict.

Value

Returns predictions for each beta of an mcen object

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

Examples

x <- matrix(rnorm(400),ncol=4)
beta <- beta <- matrix(c(1,1,0,0,0,0,-1,0,-1,1,1,0,0),ncol=4)
y <- x%*%beta + rnorm(400)
mcen_fit <- mcen(x,y,ky=2,delta=1)
new_x <- matrix(rnorm(12),ncol=4)
mcen_preds <- predict(mcen_fit, new_x)

pred_eval

Calculates the out of sample likelihood for an mcen object

Description

Calculates the out of sample likelihood for an mcen object

Usage

pred_eval(obj, test_x, test_y)
**pred_eval.mbinom_mcen**

**Arguments**

- `obj` The mbinom_mcen object.
- `test_x` The matrix of test predictors.
- `test_y` The matrix of test responses.

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

---

**Description**

Evaluates prediction error for multiple binomial responses.

**Usage**

```r
## S3 method for class 'mbinom_mcen'
pred_eval(obj, test_x, test_y)
```

**Arguments**

- `obj` The mbinom_mcen object.
- `test_x` A matrix of the test predictors.
- `test_y` A matrix of the test responses.

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

---

**pred_eval.mgauss_mcen**  
*Calculates the prediction error for a mgauss_mcen object.*

**Description**

Calculates the prediction error for a mgauss_mcen object.

**Usage**

```r
## S3 method for class 'mgauss_mcen'
pred_eval(obj, test_x, test_y)
```
Arguments

obj The mgauss mcen object.
test_x The matrix of test predictors.
test_y The matrix of test responses.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

print.cv.mcen

Prints nice output for a cv.mcen object.

Description

Prints nice output for a cv.mcen object.

Usage

## S3 method for class 'cv.mcen'
print(x, ...)

Arguments

x The cv.mcen object.
...

Additional parameters.

Value

Prints out information about where the cv.mcen object was minimized.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>
print.mcen

Prints nice output for an mcen object.

Description

Prints nice output for an mcen object.

Usage

```r
## S3 method for class 'mcen'
print(x, ...)  
```

Arguments

- `x`: The mcen object.
- `...`: Additional parameters.

Value

Prints out some basic information about the mcen object.

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

randomly_assign

randomly assign n samples to k groups

Description

randomly assign n samples to k groups

Usage

`randomly_assign(n, k)`

Arguments

- `n`: number of samples
- `k`: number of groups

Value

Returns assignments of n into k groups

Author(s)

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>
**SetEq**

*SetEq test set equivalence of two clustering sets*

**Description**

SetEq test set equivalence of two clustering sets

**Usage**

```r
SetEq(set1, set2)
```

**Arguments**

- `set1` is the cluster assignments of the previous iteration
- `set2` is the cluster assignments of the current clusters

**Value**

Returns a logical saying if the two clusterings are equal

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>

---

**squared_error**

*Calculates sum of squared error between two vectors or matrices*

**Description**

Calculates sum of squared error between two vectors or matrices

**Usage**

```r
squared_error(pred, test_y)
```

**Arguments**

- `pred` the predictions
- `test_y` the testing response values

**Value**

returns the sum of the squared differences between `pred` and `test_y`

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>, Brad Price <brad.price@mail.wvu.edu>
vl_binom

Calculates out of sample error on the binomial likelihood

Description

Calculates out of sample error on the binomial likelihood

Usage

vl_binom(pred, test_y)

Arguments

pred
  The predicted values.

test_y
  The test response values.

Author(s)

Brad Price <brad.price@mail.wvu.edu>
Index

beta_adjust, 2
beta_adjust_bin, 3
bin_horse, 3
CalcHorseBin, 4
CalcHorseEBin, 5
cluster, 5
cluster.vals, 6
coef.cv.mcen, 7
coef.mcen, 7
cv.mcen, 8
get_best_cvm, 9
matrix_multiply, 10
mcen, 11
mcen.init, 12
mcen_bin_workhorse, 13
mcen_workhorse, 14
pred_eval, 16
pred_eval.mbinom_mcen, 17
pred_eval.mgauss_mcen, 17
predict.cv.mcen, 15
predict.mcen, 16
print.cv.mcen, 18
print.mcen, 19
randomly_assign, 19
SetEq, 20
squared_error, 20
vl_binom, 21