

# Package ‘MWRidge’

October 12, 2022

**Type** Package

**Title** Two Stage Moving-Window Ridge Method for Prediction and Estimation

**Version** 1.0.0

**Date** 2016-12-12

**Author** Minli Bao

**Maintainer** Minli Bao <minli-bao@uiowa.edu>

**Description** A two stage moving-window Ridge method for coefficients estimation and model prediction. In the first stage, moving-window penalty and L1 penalty are applied. In the second stage, ridge regression is applied.

**Imports** glmnet

**License** GPL

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-12-17 10:48:12

## R topics documented:

MWRidge . . . . .	1
<b>Index</b>	<b>4</b>

---

MWRidge	<i>Two Stage Moving-Window Ridge Method for Prediction and Estimation</i>
---------	---

---

## Description

Find coefficients for a penalized regression model. Moving-window penalty is applied in the first stage and ridge regression is applied in the second stage.

**Usage**

```
MWRidge(X, Y, lambda, eta, phi, d, method='linear', epsilon = 1e-10, M = 100)
predict(X.test, X.train, Y.train, lambda, eta, phi, d, method='linear',
        epsilon = 1e-10, M = 100)
```

**Arguments**

X	The design matrix.
Y	The response variable.
lambda	The L1 penalty tuning parameter.
eta	The moving-window penalty tuning parameter.
phi	The L2 penalty tuning parameter.
d	The size of the moving-window penalty.
method	The regression method. Method can be linear or logistic.
X.test	The design matrix for the test data.
X.train	The design matrix for the training data.
Y.train	The response variable for the training data.
epsilon	Convergence criterion. The iteration will stop if the relative change is smaller than epsilon.
M	The maximum number of iterations.

**Details**

There is a two-stage regularized regression method. In the first stage, the function minimizes  $1/(2n)*SSE + \lambda*L1 + \eta/(2(d-1))*MW$ . Here SSE is the sum of squared error, L1 is the L1 penalty in Lasso and MW is the moving-window penalty. In the second stage, the function minimizes  $1/(2n)*SSE + \phi/2*L2$ . Here L2 is the L2 penalty in ridge regression.

**Value**

MWRidge returns:

beta            The coefficients estimates.

predict returns:

y.hat            The prediction of the test data based on the model trained on the training data.

**Author(s)**

Minli Bao <minli-bao@uiowa.edu>

**Examples**

```
n = 100
p = 200
set.seed(1)
X = matrix(rnorm(n * p, mean = 0, sd = 1), nrow = n, ncol = p)
beta = runif(p)
err = rnorm(n)
Y = X %*% beta + err
beta.hat = MWRidge(X, Y, lambda = 1, eta = 2, phi = 1, d = 2, method = 'linear')

prob = exp(X %*% beta)/(1 + exp(X %*% beta))
Y = rbinom(n, 1, as.vector(prob))
beta.hat = MWRidge(X, Y, lambda = 0.1, eta = 0.2, phi = 1, d = 2, method = 'logistic')

X.test = matrix(rnorm(50 * p, mean = 0, sd = 1), nrow = 50, ncol = p)
Y.hat = predict(X.test, X, Y, lambda = 0.1, eta = 0.2, phi = 1, d = 2, method = 'logistic')
```

# Index

\* **model fitting**

MWRidge, 1

MWRidge, 1

predict (MWRidge), 1