Package 'refund.wave'

February 20, 2015

Type Package

Title Wavelet-Domain Regression with Functional Data

Version 0.1

Date 2014-07-03

Author Lan Huo, Philip Reiss and Yihong Zhao

Maintainer Adam Ciarleglio <Adam.Ciarleglio@nyumc.org>

Depends R (>= 2.14.0), glmnet, wavethresh

Description Methods for regressing scalar responses on functional or image predictors, via transformation to the wavelet domain and back.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Collate 'decomp.R' 'decomp2d.R' 'decomp3d.R' 'reconstr.R' 'reconstr2d.R' 'reconstr3d.R' 'wcr.R' 'wcr.perm.R' 'wnet.R' 'wnet.perm.R' 'wUtil.R' 'plot.wnet-and-plot.wcr.R' 'Data2wd.R' 'wd2fhat.R'

NeedsCompilation no

Date/Publication 2014-07-11 23:41:47

R topics documented:

refund.wave-package *Wavelet-Domain Regression with Functional Data*

Description

Methods for regression with functional data in the wavelet domain. Various approaches to regression with scalar responses and functional predictors are implemented by `wcr` and `wnet`.

Details

For a complete list of functions type `library(help=refund.wave)`.

Author(s)

Lan Huo, Philip Reiss <phil.reiss@nyumc.org> and Yihong Zhao <yihong.zhao@nyumc.org>

gasoline *Octane numbers and NIR spectra of gasoline*

Description

Near-infrared reflectance spectra and octane numbers of 60 gasoline samples. Each NIR spectrum consists of log(1/reflectance) measurements at 401 wavelengths, in 2-nm intervals from 900 nm to 1700 nm. We thank Prof. John Kalivas for making this data set available.

Usage

```
data(gasoline)
```

Format

A data frame comprising

`octane` a numeric vector of octane numbers for the 60 samples.

`NIR` a 60 x 401 matrix of NIR spectra.

Source

Kalivas, John H. (1997). Two data sets of near infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, 37, 255–259.

References

For applications of functional principal component regression to this data set:

Reiss, P. T., and Ogden, R. T. (2007). Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102, 984–996.

Reiss, P. T., and Ogden, R. T. (2009). Smoothing parameter selection for a class of semiparametric linear models. *Journal of the Royal Statistical Society, Series B*, 71(2), 505–523.

`plot.wcr/wnet` *Default plotting for wavelet-domain scalar-on-function regression*

Description

Plots the coefficient function/image estimates produced by `wcr` and `wnet`.

Usage

```
## S3 method for class 'wcr'
plot(x, xlabel = "", ylabel = "Coefficient function", which.dim = 1, slices = NULL,
set.mfrow = TRUE, image.axes = FALSE, ...)
## S3 method for class 'wnet'
plot(x, xlabel = "", ylabel = "Coefficient function", which.dim = 1, slices = NULL,
set.mfrow = TRUE, image.axes = FALSE, ...)
```

Arguments

`x`	an object of class `"wcr"` or `"wnet"`.
`xlabel, ylabel`	for 1D functional predictors, x- and y-axis labels.
`which.dim, slices`	for 3D image predictors, the dimension (1, 2 or 3) and slices to use for plotting; see Details.
`set.mfrow`	logical value: for 3D predictors, if `TRUE` (the default), the function will try to set an appropriate value of the `mfrow` plotting parameter to display the number of slices specified. Otherwise you may wish to set `mfrow` outside the function call.
`image.axes`	for 2D and 3D predictors, the `axes` argument passed to `image`.
`...`	additional parameters passed to `plot` or `image`.

Details

As an example of how `which.dim` and `slices` are used, suppose we set `which.dim=2` and `slices=7:9`. Then three 2D slices of the coefficient image estimate `x$fhat` are displayed: `x$fhat[ , 7, ]`, `x$fhat[ , 8, ]`, and `x$fhat[ , 9, ]`.

Author(s)

Lan Huo

See Also

wcr and wnet; the latter includes examples.

---

pre-/post-process *Generic functions for transforming scalar-on-function regression to the wavelet domain and back*

Description

These two functions enable the user to transform the predictors to the wavelet domain (Data2wd); apply his/her favorite high-dimensional regression method to the wavelet coefficients (see the example); and inverse-transform the result to obtain a coefficient function estimate in the original domain (wd2fhat).

Usage

```
Data2wd(y, xfuncs, covt = NULL, min.scale = 0, nfeatures = NULL, filter.number = 10,
 wavelet.family = "DaubLeAsymm")

wd2fhat(est, info)
```

Arguments

y, xfuncs, covt, min.scale, nfeatures, filter.number, wavelet.family	see wnet.
est	an estimate produced by regressing on the wavelet-transformed predictors.
info	the wavelet infomation, output from Data2wd.

Value

Data2wd returns a list with components:

X	the design matrix, with columns of covariates and functional predictors converted to wavelet domain.
info	the infomation needed to reconstruct coefficient functions

wd2fhat returns the coefficient function. It could be a curve or a 2D/3D image.

Author(s)

Lan Huo and Philip Reiss <phil.reiss@nyumc.org>

See Also

wd

Examples

```
## Not run:
# MCP-penalized regression via ncvreg (which can also apply a SCAD penalty)
data(gasoline)
res <- Data2wd(gasoline$octane, xfuncs = gasoline$NIR[,1:256])
require(ncvreg)
obje = ncvreg::cv.ncvreg(res$X, gasoline$octane)
est = obje$fit$beta[,which.min(obje$cve)]
names(est) <- rownames(obje$fit$beta)
beta <- wd2fhat(est, res$info)

## End(Not run)
```

`refund.wave-internal` *Internal functions for the refund.wave package*

Description

These functions are ordinarily not to be called by the user, but if you contact the package authors with any questions about them, we'll do our best to clarify matters.

`wcr` *Principal component regression and partial least squares in the wavelet domain*

Description

Performs generalized linear scalar-on-function or scalar-on-image regression in the wavelet domain, by sparse principal component regression (PCR) and sparse partial least squares (PLS).

Usage

```
wcr(y, xfuncs, min.scale, nfeatures, ncomp, method = c("pcr", "pls"),
    mean.signal.term = FALSE, covt = NULL, filter.number = 10,
    wavelet.family = "DaubLeAsymm", family = "gaussian", cv1 = FALSE, nfold = 5,
    nsplit = 1, store.cv = FALSE, store.glm = FALSE, seed = NULL)
```

Arguments

`y`	scalar outcome vector.
`xfuncs`	functional predictors. For 1D predictors, an $n \times d$ matrix of signals, where n is the length of `y` and d is the number of sites at which each signal is defined. For 2D predictors, an $n \times d \times d$ array comprising n images of dimension $d \times d$. For 3D predictors, an $n \times d \times d \times d$ array comprising n images of dimension $d \times d \times d$. Note that d must be a power of 2.

min.scale	either a scalar, or a vector of values to be compared. Used to control the coarseness level of wavelet decomposition. Possible values are $0, 1, \ldots, log_2(d) - 1$.
nfeatures	number(s) of features, i.e. wavelet coefficients, to retain for prediction of y: either a scalar, or a vector of values to be compared.
ncomp	number(s) of principal components (if method="pcr") or PLS components (if method="pls"): either a scalar, or a vector of values to be compared.
method	either "pcr" (principal component regression) (the default) or "pls" (partial least squares).
mean.signal.term	logical: should the mean of each subject's signal be included as a covariate? By default, FALSE.
covt	covariates, if any: an n-row matrix, or a vector of length n.
filter.number	argument passed to function wd, imwd, or wd3D in the **wavethresh** package. Used to select the smoothness of wavelet in the decomposition.
wavelet.family	family of wavelets: passed to functions wd, imwd, orwd3D.
family	generalized linear model family. Current version supports "gaussian" (the default) and "binomial".
cv1	logical: should cross-validation be performed (to estimate prediction error) even if a single value is provided for each of min.scale, nfeatures and ncomp? By default, FALSE. Note that whenever multiple candidate values are provided for one or more of these tuning parameters, CV is performed to select the best model.
nfold	the number of validation sets ("folds") into which the data are divided.
nsplit	number of splits into nfold validation sets; CV is computed by averaging over these splits.
store.cv	logical: should the output include a CV result table?
store.glm	logical: should the output include the fitted glm?
seed	the seed for random data division. If seed = NULL, a random seed is used.

Details

Briefly, the algorithm works by (1) applying the discrete wavelet transform (DWT) to the functional/image predictors; (2) retaining only the nfeatures wavelet coefficients having the highest variance (for PCR; cf. Johnstone and Lu, 2009) or highest covariance with y (for PLS); (3) regressing y on the leading ncomp PCs or PLS components, along with any scalar covariates; and (4) applying the inverse DWT to the result to obtain the coefficient function estimate fhat.

This function supports only the standard DWT (see argument type in wd) with periodic boundary handling (see argument bc in wd).

For 2D predictors, setting min.scale=1 will lead to an error, due to a technical detail regarding imwd. Please contact the author if a workaround is needed.

See the Details for fpcr in refund for a note regarding decorrelation.

Value

An object of class "wcr". This is a list that, if store.glm = TRUE, includes all components of the fitted glm object. The following components are included even if store.glm = FALSE:

fitted.values	the fitted values.
param.coef	coefficients for covariates with decorrelation. The model is fitted after decorrelating the functional predictors from any scalar covariates; but for CV, one needs the "undecorrelated" coefficients from the training-set models.
undecor.coef	coefficients for covariates *without* decorrelation. See param.coef.
fhat	coefficient function estimate.
Rsq	coefficient of determination.
tuning.params	if CV is performed, a 2×4 table giving the indices and values of min.scale, nfeatures and ncomp chosen by CV.
cv.table	a table giving the CV criterion for each combination of min.scale, nfeatures and ncomp, if store.cv = TRUE; otherwise, the CV criterion only for the optimized combination of these parameters. Set to NULL if CV is not performed.
se.cv	if store.cv = TRUE, the standard error of the CV estimate for each combination of min.scale, nfeatures and ncomp.
family	generalized linear model family.

Author(s)

Lan Huo

References

Johnstone, I. M., and Lu, Y. (2009). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104, 682–693.

Reiss, P. T., Huo, L., Zhao, Y., Kelly, C., and Ogden, R. T. (2014). Wavelet-domain regression and predictive inference in psychiatric neuroimaging. Available at http://works.bepress.com/phil_reiss/29/

See Also

wnet

Examples

```
# See example for wnet
```

wcr/wnet.perm *Permutation test for wavelet-domain scalar-on-function regression*

Description

This function assesses statistical significance of a wcr or wnet fit by referring the cross-validation criterion to a permutation distribution.

Usage

```
wcr.perm(y, xfuncs, min.scale = 0, nfeatures, ncomp, method = c("pcr", "pls"),
         covt = NULL, nrep = 1, nsplit = 1, nfold = 5, nperm = 20,
         perm.method = NULL, family = "gaussian", seed.real = NULL,
         seed.perm = NULL, ...)
wnet.perm(y, xfuncs, min.scale = 0, nfeatures = NULL, alpha = 1, lambda,
         covt = NULL, nrep = 1, nsplit = 1, nfold = 5, nperm = 20,
         perm.method = NULL, family = "gaussian", seed.real = NULL,
         seed.perm = NULL, ...)
```

Arguments

y, xfuncs, min.scale, nfeatures, method, covt, family, nsplit, nfold	arguments passed to wcr or wnet.
ncomp	number of components; passed to wcr.
alpha, lambda	tuning parameters, passed to wnet.
nrep	number of replicates to average over, when computing the real-data CV criterion.
nperm	number of permutations. The default is suitable for toy applications only.
perm.method	either NULL or one of • "responses": permute the response vector y. • "y.residuals": permute the residuals upon regressing y ~ covt. • "x.residuals": permute the residuals upon regressing xfuncs ~ covt. See Details.
seed.real	the seed for random data division for real data. If seed.real = NULL, a random seed is used.
seed.perm	the seed for random data division for permuted data. If seed.perm = NULL, a random seed is used.
...	other arguments passed to wcr or wnet.

Details

Permutation tests of this type are discussed, in a classification setting, by Ojala and Garriga (2010). Permuting the responses (`perm.method="responses"`) is appropriate when regressing on functions/images only, with no scalar covariates. For linear regression with covariates, it is preferable to first regress on the covariates, and then permute the residuals (`perm.method="y.residuals"`). For logistic regression this is not feasible; but, following Potter (2005), one can instead permute the residuals from a regression of the functions/images on the covariates (`perm.method="x.residuals"`). When `perm.method=NULL` (the default), `"responses"` is used if `covt` is `NULL`, and `"x.residuals"` otherwise.

Value

`cv`	CV value for the real data (averaged over `nrep` replications).
`cv.perm`	CV values for the permuted data.
`pvalue`	p-value for the permutation test.

Author(s)

Lan Huo

References

Ojala, M., and Garriga, G. C. (2010). Permutation tests for studying classifier performance. *Journal of Machine Learning Research*, 11, 1833–1863.

Potter, D. M. (2005). A permutation test for inference in logistic regression with small- and moderate-sized data sets. *Statistics in Medicine*, 24, 693–708.

Reiss, P. T., Huo, L., Zhao, Y., Kelly, C., and Ogden, R. T. (2014). Wavelet-domain regression and predictive inference in psychiatric neuroimaging. Available at http://works.bepress.com/phil_reiss/29/

See Also

`wcr`, `wnet`

Examples

```
n = 200; d = 64

# Create true coefficient function
ftrue = matrix(0,d,d)
ftrue[40:46,34:38] = 1

# Generate random functional predictors, and scalar responses
ii = array(rnorm(n*d^2), dim=c(n,d,d))
iimat = ii; dim(iimat) = c(n,d^2)
yy = iimat %*% as.vector(ftrue) + rnorm(n, sd=.3)
## Not run:
obj.wnet.perm <- wnet.perm(yy, xfuncs = ii, min.scale = 4, nfeatures = 200, alpha = 1,
```

```
                          nperm = 10)

obj.wcr.perm <- wcr.perm(yy, xfuncs = ii, min.scale = 4, nfeatures = 20, ncomp = 6,
                         cv1 = TRUE, method = "pls", nperm = 10)

## End(Not run)
```

---

wnet *Generalized elastic net in the wavelet domain*

Description

Performs generalized linear scalar-on-function or scalar-on-image regression in the wavelet domain, by (naive) elastic net.

Usage

```
wnet(y, xfuncs, covt = NULL, min.scale = 0, nfeatures = NULL, alpha = 1,
     lambda = NULL, standardize = FALSE, pen.covt = FALSE, filter.number = 10,
     wavelet.family = "DaubLeAsymm", family = "gaussian", nfold = 5,
     nsplit = 1, store.cv = FALSE, store.glmnet = FALSE,  seed = NULL, ...)
```

Arguments

y	scalar outcome vector.
xfuncs	functional predictors. For 1D predictors, an $n \times d$ matrix of signals, where n is the length of y and d is the number of sites at which each signal is observed. For 2D predictors, an $n \times d \times d$ array comprising n images of dimension $d \times d$. For 3D predictors, an $n \times d \times d \times d$ array comprising n images of dimension $d \times d \times d$. Note that d must be a power of 2.
covt	covariates, if any: an n-row matrix, or a vector of length n.
min.scale	either a scalar, or a vector of candidate values to be compared. Used to control the coarseness level of the wavelet decomposition. Possible values are $0, 1, \ldots, log_2(d) - 1$.
nfeatures	number(s) of features, i.e. wavelet coefficients, to retain for prediction of y: either a scalar, or a vector of values to be compared.
alpha	elastic net mixing parameter, used by glmnet: either a scalar, or a vector of values to be compared. alpha=1 gives the lasso penalty, while alpha=0 yields the ridge penalty.
lambda	a vector of candidate regularization parameter values. If not supplied, glmnet automatically generates a sequence of candidate values.
standardize	logical, passed to glmnet: should the predictor variables be standardized? Defaults to FALSE, but either way, the coefficients are returned on the original scale.
pen.covt	logical: should the scalar covariates be penalized? If FALSE (the default), penalization is suppressed by setting the appropriate components of penalty.factor to 0 in the call to glmnet.

filter.number	passed to wd, imwd, or wd3D, in the **wavethresh** package, to select the smoothness of the wavelets.
wavelet.family	family of wavelets: passed to wd, imwd, or wd3D.
family	generalized linear model family. Current version supports "gaussian" (the default) and "binomial".
nfold	the number of validation sets ("folds") into which the data are divided.
nsplit	number of splits into nfold validation sets; CV is computed by averaging over these splits.
store.cv	logical: should the output include a CV result table?
store.glmnet	logical: should the output include the fitted glmnet?
seed	the seed for random data division. If seed = NULL, a random seed is used.
...	other arguments passed to glmnet.

Details

This function supports only the standard discrete wavelet transform (see argument type in wd) with periodic boundary handling (see argument bc in wd).

For 2D predictors, setting min.scale=1 will lead to an error, due to a technical detail regarding imwd. Please contact the authors if a workaround is needed.

Value

An object of class "wnet", which is a list with the following components:

glmnet	if store.glmnet = TRUE, the object returned by glmnet.
fitted.value	the fitted values.
coef.param	parametric coefficient estimates, for the scalar covariates.
fhat	coefficient function estimate.
Rsq	coefficient of determination.
lambda.table	array giving the candidate lambda values, chosen automatically by glmnet, for each combination of the other tuning parameters.
tuning.params	a 2×4 table giving the indices and values of min.scale, nfeatures, alpha and lambda that minimize the CV. For example, if alpha=c(0,0.5,1) is specified and the CV-minimizing tuning parameter combination takes alpha to be the 2nd of these values, then the third column of the table is c(2, 0.5).
cv.table	if store.cv = TRUE, a table giving the CV criterion for each combination of min.scale, nfeatures, alpha and lambda. Otherwise, just the minimized CV criterion.
se.cv	if store.cv = TRUE, the standard error of the CV estimate for each combination of min.scale, nfeatures, alpha and lambda.
family	generalized linear model family.

Author(s)

Lan Huo and Yihong Zhao

References

Zhao, Y., Ogden, R. T., and Reiss, P. T. (2012). Wavelet-based LASSO in functional linear regression. *Journal of Computational and Graphical Statistics*, 21(3), 600–617.

Zhao, Y., Chen, H., and Ogden, R. T. Wavelet-based Adaptive LASSO and screening approaches in functional linear regression. *Journal of Computational and Graphical Statistics*, to appear.

See Also

wcr, wnet.perm

Examples

```
## Not run:
### 1D functional predictor example ###

data(gasoline)

# input a single value of each tuning parameters
gas.wnet1 <- wnet(gasoline$octane, xfuncs = gasoline$NIR[,1:256],
             nfeatures= 20, min.scale = 0, alpha = 1)
gas.wpcr1 <- wcr(gasoline$octane, xfuncs = gasoline$NIR[,1:256], min.scale = 0,
                 nfeatures = 20, ncomp = 15)
gas.wpls1 <- wcr(gasoline$octane, xfuncs = gasoline$NIR[,1:256], min.scale = 0,
                 nfeatures = 20, ncomp = 15, method = "pls")
plot(gas.wnet1)
plot(gas.wpcr1)
plot(gas.wpls1)

# input vectors of candidate tuning parameter values
gas.wnet2 <- wnet(gasoline$octane, xfuncs = gasoline$NIR[,1:256],
                  nfeatures= 20, min.scale = 0:3, alpha = c(0.9, 1))
gas.wpcr2 <- wcr(gasoline$octane, xfuncs = gasoline$NIR[,1:256], min.scale = 0:3,
                 nfeatures = c(16, 18, 20), ncomp = 10:15)
gas.wpls2 <- wcr(gasoline$octane, xfuncs = gasoline$NIR[,1:256], min.scale = 0:3,
                 nfeatures = c(16, 18, 20), ncomp = 10:15, method = "pls")
plot(gas.wnet2)
plot(gas.wpcr2)
plot(gas.wpls2)

### 2D functional predictor example ###

n = 200; d = 64

# Create true coefficient function
ftrue = matrix(0,d,d)
ftrue[40:46,34:38] = 1

# Generate random functional predictors, and scalar responses
ii = array(rnorm(n*d^2), dim=c(n,d,d))
iimat = ii; dim(iimat) = c(n,d^2)
yy = iimat %*% as.vector(ftrue) + rnorm(n, sd=.3)
```

```
mm.wnet <- wnet(yy, xfuncs = ii, min.scale = 4, alpha = 1)

mm.wpls <- wcr(yy, xfuncs = ii, min.scale = 4, nfeatures = 20, ncomp = 6,
                method = "pls")

plot(mm.wnet)
plot(mm.wpls)

### 3D functional predictor example ###

n = 200; d = 16

# Create true coefficient function
ftrue = array(0,dim = rep(d, 3))
ftrue[10:16,12:15, 4:8] = 1

# Generate random functional predictors, and scalar responses
ii = array(rnorm(n*d^3), dim=c(n,rep(d,3)))
iimat = ii; dim(iimat) = c(n,d^3)
yy = iimat %*% as.vector(ftrue) + rnorm(n, sd=.3)

mmm.wnet <- wnet(yy, xfuncs = ii, min.scale = 2, alpha = 1)

mmm.wpls <- wcr(yy, xfuncs = ii, min.scale = 2, nfeatures = 20, ncomp = 6,
                method = "pls")
plot(mmm.wnet)
plot(mmm.wpls)

## End(Not run)
```

Index