Package ‘sdwd’

October 27, 2020

Type Package

Title Sparse Distance Weighted Discrimination

Version 1.0.5

Date 2020-10-24

Author Boxiang Wang and Hui Zou

Maintainer Boxiang Wang <boxiang-wang@uiowa.edu>

Description

Formulates a sparse distance weighted discrimination (SDWD) for high-dimensional classification and implements a very fast algorithm for computing its solution path with the L1, the elastic-net, and the adaptive elastic-net penalties. More details about the methodology SDWD is seen on Wang and Zou (2016) (<doi:10.1080/10618600.2015.1049700>).

Depends Matrix

Imports grDevices, graphics, stats, methods

License GPL-2

Repository CRAN

NeedsCompilation yes

Date/Publication 2020-10-27 14:00:06 UTC

R topics documented:

sdwd-package *Sparse Distance Weighted Discrimination*

Description

This package implements the generalized coordinate descent (GCD) algorithm to efficiently compute the solution path of the sparse distance weighted discrimination (DWD) at a given fine grid of regularization parameters. Sparse distance weighted discrimination is a high-dimensional margin-based classifier.

Details

Package:	sdwd
Type:	Package
Version:	1.0.3
Date:	2020-02-16
License:	GPL-2

Suppose `x` is the predictors and `y` is the binary response. With a fixed value `lambda2`, the package produces the solution path of the sparse DWD over a grid of `lambda` values. The value of `lambda2` can be further tuned by cross-validation.

The package `sdwd` contains five main functions:
`sdwd`
`cv.sdwd`
`coef.sdwd`
`plot.sdwd`
`plot.cv.sdwd`

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

Marron, J.S., Todd, M.J., Ahn, J. (2007) “Distance-Weighted Discrimination"", *Journal of the*

American Statistical Association, **102**(408), 1267–1271.
https://www.tandfonline.com/doi/abs/10.1198/016214507000001120

Tibshirani, Robert., Bien, J., Friedman, J.,Hastie, T.,Simon, N.,Taylor, J., and Tibshirani, Ryan. (2012) Strong Rules for Discarding Predictors in Lasso-type Problems, *Journal of the Royal Statistical Society, Series B*, **74**(2), 245–266.
https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2011.01004.x

Yang, Y. and Zou, H. (2013) "An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

---

coef.cv.sdwd *compute coefficients from a "cv.sdwd" object*

Description

Computes coefficients at chosen values of `lambda` from the `cv.sdwd` object.

Usage

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

Arguments

`object`	A fitted `cv.sdwd` object, obtained by conducting the cross-validation to the sparse DWD model.
`s`	Value(s) of the L1 tuning parameter `lambda` for computing coefficients. Default value is `"lambda.1se"`, which represents the largest `lambda` value achieving the cross-validation error within one standard error of the minimum. An alternative value is `"lambda.min"`, which is the `lambda` incurring the least cross-validation error. `s` can also be numeric, being taken as the value(s) to be used.
`...`	Other arguments that can be passed to `sdwd`.

Details

This function computes the coefficients at the values of `lambda` suggested by the cross-validation. This function is modified based on the `coef.cv` function from the `glmnet` and the `gcdnet` packages.

Value

The returned object depends on the choice of `s` and the `...` argument passed on to the `sdwd` method.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) "Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) "An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

See Also

`cv.sdwd` and `predict.cv.sdwd` methods.

Examples

```
data(colon)
colon$x = colon$x[ , 1:100] # this example only uses the first 100 columns
set.seed(1)
cv = cv.sdwd(colon$x, colon$y, lambda2=1, nfolds=5)
c1 = coef(cv, s="lambda.1se")
```

---

coef.sdwd — *compute coefficients for the sparse DWD*

Description

Computes the coefficients or returns the indices of nonzero coefficients at chosen values of `lambda` from a fitted `sdwd` object.

Usage

```
## S3 method for class 'sdwd'
coef(object, s=NULL, type=c("coefficients","nonzero"), ...)
```

Arguments

`object`	A fitted `sdwd` object.
`s`	Value(s) of the L1 tuning parameter `lambda` for computing coefficients. Default is the entire `lambda` sequence obtained by `sdwd`.
`type`	`"coefficients"` or `"nonzero"`? `"coefficients"` computes the coefficients at given values for `s`; `"nonzero"` returns a list of the indices of the nonzero coefficients for each value of `s`. Default is `"coefficients"`.
`...`	Not used. Other arguments to `predict`.

Details

`s` is the new vector at which predictions are requested. If `s` is not in the lambda sequence used for fitting the model, the `coef` function will use linear interpolation to make predictions. The new values are interpolated using a fraction of coefficients from both left and right `lambda` indices. This function is modified based on the `coef` function from the `gcdnet` and the `glmnet` packages.

Value

Either the coefficients at the requested values of `lambda`, or a list of the indices of the nonzero coefficients for each `lambda`.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang `<boxiang-wang@uiowa.edu>`

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) “An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22
https://www.jstatsoft.org/v33/i01/paper

See Also

`predict.sdwd`

Examples

```
data(colon)
fit = sdwd(colon$x, colon$y, lambda2=1)
c1 = coef(fit, type="coef",s=c(0.1, 0.005))
c2 = coef(fit, type="nonzero")
```

---

colon — *simplified gene expression data from Alon et al. (1999)*

Description

Gene expression data (2000 genes for 62 samples) from a DNA microarray experiments of colon tissue samples (Alon et al., 1999).

Usage

```
data(colon)
```

Details

This data set contains 62 colon tissue samples with 2000 gene expression levels. Among 62 samples, 40 are tumor tissues (coded 1) and 22 are normal tissues (coded -1).

Value

A list with the following elements:

`x`	A matrix of 2000 columns and 62 rows standing for 2000 gene expression levels and 62 colon tissue samples. Each row corresponds to a patient.
`y`	A numeric vector of length 62 representing the tissue type (1 for tumor; -1 for normal).

Source

The data were introduced in Alon et al. (1999).

References

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., and Levine, A.J. (1999). "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", *Proceedings of the National Academy of Sciences*, **96**(12), 6745–6750.

Examples

```
# load sdwd library
library(sdwd)

# load data set
data(colon)

# how many samples and how many predictors?
dim(colon$x)

# how many samples of class -1 and 1 respectively?
sum(colon$y == -1)
sum(colon$y == 1)
```

---

cv.sdwd *cross-validation for the sparse DWD*

Description

Conducts a k-fold cross-validation for sdwd and returns the suggested values of the L1 parameter `lambda`.

Usage

```
cv.sdwd(x, y, lambda = NULL, pred.loss = c("misclass", "loss"), nfolds = 5, foldid, ...)
```

Arguments

`x`	A matrix of predictors, i.e., the `x` matrix used in sdwd.
`y`	A vector of binary class labels, i.e., the `y` used in sdwd.
`lambda`	Default is `NULL`, and the sequence generated by sdwd is used. User can also provide a new `lambda` sequence to use in cross-validation.
`pred.loss`	`misclass` for classification error, `loss` for DWD loss.
`nfolds`	The number of folds. Default value is 5. The allowable range is from 3 to the sample size. Larger `nfolds` needs more timing.
`foldid`	An optional vector with values between 1 and `nfold`, representing the folder indices for each observation. If supplied, `nfold` can be missing.
`...`	Other arguments that can be passed to sdwd.

Details

This function runs sdwd to the sparse DWD by excluding every fold alternatively, and then computes the mean cross-validation error and the standard deviation. This function is modified based on the `cv` function from the `gcdnet` and the `glmnet` packages.

Value

A `cv.sdwd` object is returned, which includes the cross-validation fit.

`lambda`	The `lambda` sequence used in `sdwd`.
`cvm`	A vector of length `length(lambda)` for the mean cross-validated error.
`cvsd`	A vector of length `length(lambda)` for estimates of standard error of `cvm`.
`cvupper`	The upper curve: `cvm` + `cvsd`.
`cvlower`	The lower curve: `cvm` -`cvsd`.
`nzero`	Numbers of non-zero coefficients at each `lambda`.
`name`	"Mis-classification error", for plotting purposes.
`sdwd.fit`	A fitted `sdwd` object using the full data.
`lambda.min`	The `lambda` incurring the minimum cross validation error `cvm`.
`lambda.1se`	The largest value of `lambda` such that error is within one standard error of the minimum.
`cv.min`	The minimum cross-validation error.
`cv.1se`	The cross-validation error associated with `lambda.1se`.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) "Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) "An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

See Also

`sdwd`, `plot.cv.sdwd`, `predict.cv.sdwd`, and `coef.cv.sdwd` methods.

Examples

```
data(colon)
colon$x = colon$x[ , 1:100] # this example only uses the first 100 columns
n = nrow(colon$x)
set.seed(1)
id = sample(n, trunc(n/3))
cvfit = cv.sdwd(colon$x[-id, ], colon$y[-id], lambda2=1, nfolds=5)
plot(cvfit)
predict(cvfit, newx=colon$x[id, ], s="lambda.min")
```

`plot.cv.sdwd` *plot the cross-validation curve of the sparse DWD*

Description

Plots the cross-validation curve against a function of `lambda` values. The function also provides the upper and lower standard deviation curves.

Usage

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

Arguments

`x`	A fitted `cv.sdwd` object.
`sign.lambda`	Whether to plot against `log(lambda)` (default) or its negative if `sign.lambda=-1`.
`...`	Other graphical parameters to `plot`.

Details

This function depicts the cross-validation curves. This function is modified based on the `plot.cv` function from the `glmnet` and the `gcdnet` packages.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) “An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.

https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

See Also

`cv.sdwd`.

Examples

```
data(colon)
colon$x = colon$x[ , 1:100] # this example only uses the first 100 columns
set.seed(1)
cv = cv.sdwd(colon$x, colon$y, lambda2=1, nfolds=5)
plot(cv)
```

---

plot.sdwd — *plot coefficients for the sparse DWD*

Description

Plots the solution paths for a fitted `sdwd` object.

Usage

```
## S3 method for class 'sdwd'
plot(x, xvar=c("norm", "lambda"), color=FALSE, label=FALSE, ...)
```

Arguments

`x`	A fitted `sdwd` model.
`xvar`	Specifies the X-axis. If `xvar == "norm"`, plots against the L1-norm of the coefficients; if `xvar == "lambda"` against the log-lambda sequence.
`color`	If `TRUE`, plots the curves with rainbow colors; otherwise, with gray colors (default).
`label`	If `TRUE`, labels the curves with variable sequence numbers. Default is `FALSE`.
`...`	Other graphical parameters to `plot`.

Details

Plots the solution paths as a coefficient profile plot. This function is modified based on the `plot` function from the `gcdnet` and the `glmnet` packages.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) “An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

See Also

`print.sdwd`, `predict.sdwd`, `coef.sdwd`, `plot.sdwd`, and `cv.sdwd`.

Examples

```
data(colon)
fit = sdwd(colon$x, colon$y)
par(mfrow=c(1,3))
# plots against the L1-norm of the coefficients
plot(fit)
# plots against the log-lambda sequence
plot(fit, xvar="lambda", label=TRUE)
# plots with colors
plot(fit, color=TRUE)
```

---

predict.cv.sdwd — *make predictions from a "cv.sdwd" object*

Description

This function predicts the class labels of new observations by the sparse DWD at the `lambda` values suggested by `cv.sdwd`.

Usage

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

Arguments

object	A fitted cv.sdwd object.
newx	A matrix of new values for x at which predictions are to be made. Must be a matrix. See documentation for predict.sdwd.
s	Value(s) of the L1 tuning parameter lambda for making predictions. Default is the s="lambda.1se" saved on the cv.sdwd object. An alternative choice is s="lambda.min". s can also be numeric, being taken as the value(s) to be used.
...	Not used. Other arguments to predict.

Details

This function uses the cross-validation results to making predictions. This function is modified based on the predict.cv function from the glmnet and the gcdnet packages.

Value

Predicted class labels or fitted values, depending on the choice of s and the ... argument passed on to the sdwd method.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) “An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

See Also

cv.sdwd, and coef.cv.sdwd methods.

Examples

```
data(colon)
colon$x = colon$x[ , 1:100] # this example only uses the first 100 columns
set.seed(1)
cv = cv.sdwd(colon$x, colon$y, lambda2=1, nfolds=5)
```

```
predict(cv$sdwd.fit, newx=colon$x[2:5, ],
  s=cv$lambda.1se, type="class")
```

---

predict.sdwd *make predictions for the sparse DWD*

Description

This function predicts the binary class labels or the fitted values of an sdwd object.

Usage

```
## S3 method for class 'sdwd'
predict(object, newx, s=NULL, type=c("class", "link"), ...)
```

Arguments

object	A fitted sdwd object.
newx	A matrix of new values for x at which predictions are to be made. We note that newx must be a matrix, predict function does not accept a vector or other formats of newx.
s	Value(s) of the L1 tuning parameter lambda for computing coefficients. Default is the entire lambda sequence obtained by sdwd.
type	"class" or "link"? "class" produces the predicted binary class labels."link" returns the fitted values. Default is "class".
...	Not used. Other arguments to predict.

Details

s stands for the new lambda values for making predictions. If s is not in the original lambda sequence generated by sdwd, the predict.sdwd function will use linear interpolation by using a fraction of predicted values from the lambda values in the original sequence adjacent to the s to make predictions. The predict.sdwd function is modified based on the predict function from the glmnet and the gcdnet packages.

Value

Returns either the predicted class labels or the fitted values, depending on the choice of type.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) “An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

See Also

coef.sdwd

Examples

```
data(colon)
fit = sdwd(colon$x, colon$y, lambda2=1)
print(predict(fit ,type="class",newx=colon$x[2:5,]))
```

---

print.sdwd   *print an sdwd object*

Description

Print a summary of the sdwd solution paths.

Usage

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

Arguments

x	A fitted sdwd object.
digits	Specify the significant digits.
...	Additional print arguments.

Details

This function prints a two-column matrix with columns `Df` and `Lambda`, where the `Df` column exhibits the number of nonzero coefficients and the `Lambda` column displays the corresponding `lambda` value. This function is modified based on the `print` function from the `gcdnet` and the `glmnet` packages.

Value

A two-column matrix with one column of the number of nonzero coefficients and a second column of `lambda` values.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Yang, Y. and Zou, H. (2013) “An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

See Also

`print.sdwd`, `predict.sdwd`, `coef.sdwd`, `plot.sdwd`, and `cv.sdwd`.

Examples

```
data(colon)
fit = sdwd(colon$x, colon$y)
print(fit)
```

---

sdwd *fit the sparse DWD*

Description

Fits the sparse distance weighted discrimination (SDWD) model with imposing L1, elastic-net, or adaptive elastic-net penalties. The solution path is computed at a grid of values of tuning parameter `lambda`. This function is modified based on the `glmnet` and the `gcdnet` packages.

Usage

```
sdwd(x, y, nlambda=100,
     lambda.factor=ifelse(nobs < nvars, 0.01, 1e-04),
     lambda=NULL, lambda2=0, pf=rep(1, nvars),
     pf2=rep(1, nvars), exclude, dfmax=nvars + 1,
     pmax=min(dfmax * 1.2, nvars), standardize=TRUE,
     eps=1e-8, maxit=1e6, strong=TRUE)
```

Arguments

x	A matrix with N rows and p columns for predictors.
y	A vector of length p for binary responses. The element of y is either -1 or 1.
nlambda	The number of lambda values, i.e., length of the lambda sequence. Default is 100.
lambda.factor	The ratio of the smallest to the largest lambda in the sequence: lambda.factor = min(lambda) / max(lambda). max(lambda) is the least lambda to make all coefficients to be zero. The default value of lambda.factor is 0.0001 if $N >= p$ or 0.01 if $N < p$. Takes no effect when user specifies a lambda sequence.
lambda	An optional user-supplied lambda sequence. If lambda = NULL (default), the program computes its own lambda sequence based on nlambda and lambda.factor; otherwise, the program uses the user-specified one. Since the program will automatically sort user-defined lambda sequence in decreasing order, it is better to supply a decreasing sequence.
lambda2	The L2 tuning parameter λ_2.
pf	A vector of length p representing the L1 penalty weights to each coefficient of β for adaptive L1 or adaptive elastic net. pf can be 0 for some predictor(s), leading to including the predictor(s) all the time. One suggested choice of pf is $(\beta + 1/n)^{-1}$, where n is the sample size and β is the coefficents obtained by L1 DWD or enet DWD. Default is 1 for all predictors (and infinity if some predictors are listed in exclude).
pf2	A vector of length p for L2 penalty factor for adaptive L1 or adaptive elastic net. To allow different L2 shrinkage, user can set pf2 to be different L2 penalty weights for each coefficient of β. pf2 can be 0 for some variables, indicating no L2 shrinkage. Default is 1 for all predictors.
exclude	Whether to exclude some predictors from the model. This is equivalent to adopting an infinite penalty factor when excluding some predictor. Default is none.
dfmax	Restricts at most how many predictors can be incorporated in the model. Default is $p + 1$. This restriction is helpful when p is large, provided that a partial path is acceptable.
pmax	Restricts the maximum number of variables ever to be nonzero; e.g, once some β enters the model, it counts once. The count will not change when the β exits or re-enters the model. Default is min(dfmax*1.2,p).
standardize	Whether to standardize the data. If TRUE, sdwd normalizes the predictors such that each column has sum squares$\sum_{i=1}^N x_{ij}^2/N = 1$ of one. Note that x is always centered (i.e. $\sum_{i=1}^N x_{ij} = 0$) no matter standardize is TRUE or FALSE. sdwd always returns coefficient beta on the original scale. Default value is TRUE.

eps The algorithm stops when (i.e. $4 \max_j (\beta_j^{new} - \beta_j^{old})^2$ is less than eps, where $j = 0, \ldots, p$. Defaults value is 1e-8.

maxit Restricts how many outer-loop iterations are allowed. Default is 1e6. Consider increasing maxit when the algorithm does not converge.

strong If TRUE, adopts the strong rule to accelerate the algorithm.

Details

The sdwd minimizes the sparse penalized DWD loss function,

$$L(y, X, \beta)/N + \lambda_1 ||\beta||_1 + 0.5\lambda_2 ||\beta||_2^2,$$

where $L(u) = 1 - u$ if $u \leq 1/2$, $1/(4u)$ if $u > 1/2$ is the DWD loss. The value of lambda2 is user-specified.

To use the L1 penalty (lasso), set lambda2=0. To use the elastic net, set lambda2 as nonzero. To use the adaptive L1, set lambda2=0 and specify pf and pf2. To use the adaptive elastic net, set lambda2 as nonzero and specify pf and pf2 as well.

When the algorithm do not converge or run slow, consider increasing eps, decreasing nlambda, or increasing lambda.factor before increasing maxit.

Value

An object with S3 class sdwd.

b0 A vector of length length(lambda) representing the intercept at each lambda value.

beta A matrix of dimension p*length(lambda) representing the coefficients at each lambda value. The matrix is stored as a sparse matrix (Matrix package). To convert it into normal type matrix use as.matrix().

df The number of nonzero coefficients at each lambda.

dim The dimension of coefficient matrix, i.e., p*length(lambda).

lambda The lambda sequence that was actually used.

npasses Total number of iterations for all lambda values.

jerr Warnings and errors; 0 if no error.

call The call that produced this object.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang-wang@uiowa.edu>

References

Wang, B. and Zou, H. (2016) “Sparse Distance Weighted Discrimination", *Journal of Computational and Graphical Statistics*, **25**(3), 826–838.
https://www.tandfonline.com/doi/full/10.1080/10618600.2015.1049700

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization paths for generalized linear models via coordinate descent", *Journal of Statistical Software*, **33**(1), 1–22.
https://www.jstatsoft.org/v33/i01/paper

Marron, J.S., Todd, M.J., and Ahn, J. (2007) “Distance-Weighted Discrimination", *Journal of the American Statistical Association*, **102**(408), 1267–1271.
https://www.tandfonline.com/doi/abs/10.1198/016214507000001120

Tibshirani, Robert., Bien, J., Friedman, J.,Hastie, T.,Simon, N.,Taylor, J., and Tibshirani, Ryan. (2012) Strong Rules for Discarding Predictors in Lasso-type Problems, *Journal of the Royal Statistical Society, Series B*, **74**(2), 245–266.
https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2011.01004.x

Yang, Y. and Zou, H. (2013) “An Efficient Algorithm for Computing the HHSVM and Its Generalizations", *Journal of Computational and Graphical Statistics*, **22**(2), 396–415.
https://www.tandfonline.com/doi/full/10.1080/10618600.2012.680324

See Also

`print.sdwd`, `predict.sdwd`, `coef.sdwd`, `plot.sdwd`, and `cv.sdwd`.

Examples

```
# load the data
data(colon)
# fit the elastic-net penalized DWD with lambda2=1
fit = sdwd(colon$x, colon$y, lambda2=1)
print(fit)
# coefficients at some lambda value
c1 = coef(fit, s=0.005)
# make predictions
predict(fit, newx=colon$x[1:10, ], s=c(0.01, 0.005))
```

Index