Package 'kerndwd'

April 21, 2018

Type Package

Title Distance Weighted Discrimination (DWD) and Kernel Methods

Version 2.0.2

Date 2018-04-21

Author Boxiang Wang <boxiang@umn.edu>, Hui Zou <hzou@stat.umn.edu>

Maintainer Boxiang Wang <boxiang@umn.edu>

Description A novel implementation that solves the linear distance weighted discrimination and the kernel distance weighted discrimination.

Depends methods

Imports graphics, grDevices, stats, utils

License GPL-2

Repository CRAN

NeedsCompilation yes

Date/Publication 2018-04-21 20:21:36 UTC

R topics documented:

kerndwd-package *Kernel Distance Weighted Discrimination*

Description

Extremely novel efficient procedures for solving linear generalized DWD and kernel generalized DWD in reproducing kernel Hilbert spaces for classification. The algorithm is based on the majorization-minimization (MM) principle to compute the entire solution path at a given fine grid of regularization parameters.

Details

Suppose `x` is predictor and `y` is a binary response. The package computes the entire solution path over a grid of `lambda` values.

The main functions of the package `kerndwd` include:
`kerndwd`
`cv.kerndwd`
`tunedwd`
`predict.kerndwd`
`plot.kerndwd`
`plot.cv.kerndwd`

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang@umn.edu>

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244
Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004) "kernlab – An S4 Package for Kernel Methods in R", *Journal of Statistical Software*, **11**(9), 1–20.
http://www.jstatsoft.org/v11/i09/paper
Marron, J.S., Todd, M.J., Ahn, J. (2007) "Distance-Weighted Discrimination"", *Journal of the American Statistical Association*, **102**(408), 1267–1271.
https://faculty.franklin.uga.edu/jyahn/sites/faculty.franklin.uga.edu.jyahn/files/DWD3.pdf

BUPA *BUPA's liver disorders data*

Description

BUPA's liver disorders data: 345 male individuals' blood test result and liver disorder status.

Usage

```
data(BUPA)
```

Details

This data set consists of 345 observations and 6 predictors representing the blood test result liver disorder status of 345 patients. The three predictors are mean corpuscular volume (MCV), alkaline phosphotase (ALKPHOS), alamine aminotransferase (SGPT), aspartate aminotransferase (SGOT), gamma-glutamyl transpeptidase (GAMMAGT), and the number of alcoholic beverage drinks per day (DRINKS).

Value

A list with the following elements:

X	A numerical matrix for predictors: 345 rows and 6 columns; each row corresponds to a patient.
y	A numeric vector of length 305 representing the liver disorder status.

Source

The data set is available for download from UCI machine learning repository.
https://archive.ics.uci.edu/ml/datasets/Liver+Disorders

Examples

```
# load data set
data(BUPA)

# the number of samples predictors
dim(BUPA$X)

# the number of samples for each class
sum(BUPA$y == -1)
sum(BUPA$y == 1)
```

cv.kerndwd *cross-validation*

Description

Carry out a cross-validation for kerndwd to find optimal values of the tuning parameter lambda.

Usage

```
cv.kerndwd(x, y, kern, lambda, nfolds=5, foldid, wt, ...)
```

Arguments

x	A matrix of predictors, i.e., the matrix x used in kerndwd.
y	A vector of binary class labels, i.e., the y used in kerndwd. y has to be two levels.
kern	A kernel function.
lambda	A user specified lambda candidate sequence for cross-validation.
nfolds	The number of folds. Default value is 5. The allowable range is from 3 to the sample size.
foldid	An optional vector with values between 1 and nfold, representing the fold indices for each observation. If supplied, nfold can be missing.
wt	A vector of length n for weight factors. When wt is missing or wt=NULL, an unweighted DWD is fitted.
...	Other arguments being passed to kerndwd.

Details

This function computes the mean cross-validation error and the standard error by fitting kerndwd with every fold excluded alternatively. This function is modified based on the cv function from the glmnet package.

Value

A cv.kerndwd object including the cross-validation results is return..

lambda	The lambda sequence used in kerndwd.
cvm	A vector of length length(lambda): mean cross-validated error.
cvsd	A vector of length length(lambda): estimates of standard error of cvm.
cvupper	The upper curve: cvm + cvsd.
cvlower	The lower curve: cvm - cvsd.
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.
cvm.min	The cross-validation error corresponding to lambda.min, i.e., the least error.
cvm.1se	The cross-validation error corresponding to lambda.1se.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang@umn.edu>

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244
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.
http://www.jstatsoft.org/v33/i01/paper

See Also

kerndwd and plot.cv.kerndwd

Examples

```
set.seed(1)
data(BUPA)
BUPA$X = scale(BUPA$X, center=TRUE, scale=TRUE)
lambda = 10^(seq(3, -3, length.out=10))
kern = rbfdot(sigma=sigest(BUPA$X))
m.cv = cv.kerndwd(BUPA$X, BUPA$y, kern, qval=1, lambda=lambda, eps=1e-5, maxit=1e5)
m.cv$lambda.min
```

---

kerndwd *solve Linear DWD and Kernel DWD*

Description

Fit the linear generalized distance weighted discrimination (DWD) model and the generalized DWD on Reproducing kernel Hilbert space. The solution path is computed at a grid of values of tuning parameter lambda.

Usage

```
kerndwd(x, y, kern, lambda, qval=1, wt, eps=1e-05, maxit=1e+05)
```

Arguments

x	A numerical matrix with N rows and p columns for predictors.
y	A vector of length N for binary responses. The element of y is either -1 or 1.
kern	A kernel function; see dots.
lambda	A user supplied lambda sequence.
qval	The exponent index of the generalized DWD. Default value is 1.

wt	A vector of length n for weight factors. When wt is missing or wt=NULL, an unweighted DWD is fitted.
eps	The algorithm stops when (i.e. $\sum_j(\beta_j^{new} - \beta_j^{old})^2$ is less than eps, where $j = 0, \ldots, p$. Default value is 1e-5.
maxit	The maximum of iterations allowed. Default is 1e5.

Details

Suppose that the generalized DWD loss is $V_q(u) = 1 - u$ if $u \leq q/(q+1)$ and $\frac{1}{u^q}\frac{q^q}{(q+1)^{(q+1)}}$ if $u > q/(q+1)$. The value of λ, i.e., lambda, is user-specified.

In the linear case (kern is the inner product and N > p), the kerndwd fits a linear DWD by minimizing the L2 penalized DWD loss function,

$$\frac{1}{N}\sum_{i=1}^{n} V_q(y_i(\beta_0 + X_i'\beta)) + \lambda\beta'\beta.$$

If a linear DWD is fitted when N < p, a kernel DWD with the linear kernel is actually solved. In such case, the coefficient β can be obtained from $\beta = X'\alpha$.

In the kernel case, the kerndwd fits a kernel DWD by minimizing

$$\frac{1}{N}\sum_{i=1}^{n} V_q(y_i(\beta_0 + K_i'\alpha)) + \lambda\alpha' K\alpha,$$

where K is the kernel matrix and K_i is the ith row.

The weighted linear DWD and the weighted kernel DWD are formulated as follows,

$$\frac{1}{N}\sum_{i=1}^{n} w_i \cdot V_q(y_i(\beta_0 + X_i'\beta)) + \lambda\beta'\beta,$$

$$\frac{1}{N}\sum_{i=1}^{n} w_i \cdot V_q(y_i(\beta_0 + K_i'\alpha)) + \lambda\alpha' K\alpha,$$

where w_i is the ith element of wt. The choice of weight factors can be seen in the reference below.

Value

An object with S3 class kerndwd.

alpha	A matrix of DWD coefficients at each lambda value. The dimension is (p+1)*length(lambda) in the linear case and (N+1)*length(lambda) in the kernel case.
lambda	The lambda sequence.
npass	Total number of MM iterations for all lambda values.
jerr	Warnings and errors; 0 if none.
info	A list including parameters of the loss function, eps, maxit, kern, and wt if a weight vector was used.
call	The call that produced this object.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang@umn.edu>

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244
Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004) "kernlab – An S4 Package for Kernel Methods in R", *Journal of Statistical Software*, **11**(9), 1–20.
http://www.jstatsoft.org/v11/i09/paper
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.
http://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://faculty.franklin.uga.edu/jyahn/sites/faculty.franklin.uga.edu.jyahn/files/DWD3.pdf
Qiao, X., Zhang, H., Liu, Y., Todd, M., Marron, J.S. (2010) "Weighted distance weighted discrimination and its asymptotic properties", *Journal of the American Statistical Association*, **105**(489), 401–414.
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2996856/

See Also

predict.kerndwd, plot.kerndwd, and cv.kerndwd.

Examples

```
data(BUPA)
# standardize the predictors
BUPA$X = scale(BUPA$X, center=TRUE, scale=TRUE)

# a grid of tuning parameters
lambda = 10^(seq(3, -3, length.out=10))

# fit a linear DWD
kern = vanilladot()
DWD_linear = kerndwd(BUPA$X, BUPA$y, kern,
  qval=1, lambda=lambda, eps=1e-5, maxit=1e5)

# fit a DWD using Gaussian kernel
kern = rbfdot(sigma=1)
DWD_Gaussian = kerndwd(BUPA$X, BUPA$y, kern,
  qval=1, lambda=lambda, eps=1e-5, maxit=1e5)

# fit a weighted kernel DWD
kern = rbfdot(sigma=1)
```

```
weights = c(1, 2)[factor(BUPA$y)]
DWD_wtGaussian = kerndwd(BUPA$X, BUPA$y, kern,
  qval=1, lambda=lambda, wt = weights, eps=1e-5, maxit=1e5)
```

`kernel functions` *Kernel Functions*

Description

Kernel functions provided in the R package `kernlab`. Details can be seen in the reference below.
The Gaussian RBF kernel $k(x, x') = \exp(-\sigma\|x - x'\|^2)$
The Polynomial kernel $k(x, x') = (scale < x, x' > +offset)^{degree}$
The Linear kernel $k(x, x') =< x, x' >$
The Laplacian kernel $k(x, x') = \exp(-\sigma\|x - x'\|)$
The Bessel kernel $k(x, x') = (-\text{Bessel}^n_{(\nu+1)}\sigma\|x - x'\|^2)$
The ANOVA RBF kernel $k(x, x') = \sum_{1\leq i_1 \ldots < i_D \leq N} \prod_{d=1}^{D} k(x_{id}, {x'}_{id})$ where k(x, x) is a Gaussian RBF kernel.
The Spline kernel $\prod_{d=1}^{D} 1 + x_i x_j + x_i x_j \min(x_i, x_j) - \frac{x_i + x_j}{2} \min(x_i, x_j)^2 + \frac{\min(x_i, x_j)^3}{3}$. The parameter `sigma` used in `rbfdot` can be selected by `sigest()`.

Usage

```
rbfdot(sigma = 1)
polydot(degree = 1, scale = 1, offset = 1)
vanilladot()
laplacedot(sigma = 1)
besseldot(sigma = 1, order = 1, degree = 1)
anovadot(sigma = 1, degree = 1)
splinedot()
sigest(x)
```

Arguments

`sigma`	The inverse kernel width used by the Gaussian, the Laplacian, the Bessel, and the ANOVA kernel.
`degree`	The degree of the polynomial, bessel or ANOVA kernel function. This has to be an positive integer.
`scale`	The scaling parameter of the polynomial kernel function.
`offset`	The offset used in a polynomial kernel.
`order`	The order of the Bessel function to be used as a kernel.
`x`	The design matrix used in `kerndwd` when `sigest` is called to estimate `sigma` in `rbfdot()`.

Details

These R functions and descriptions are directly duplicated and/or adapted from the R package `kernlab`.

Value

Return an S4 object of class `kernel` which can be used as the argument of `kern` when fitting a `kerndwd` model.

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004) "kernlab – An S4 Package for Kernel Methods in R", *Journal of Statistical Software*, **11**(9), 1–20.
http://www.jstatsoft.org/v11/i09/paper

Examples

```
data(BUPA)
# generate a linear kernel
kfun = vanilladot()

# generate a Laplacian kernel function with sigma = 1
kfun = laplacedot(sigma=1)

# generate a Gaussian kernel function with sigma estimated by sigest()
kfun = rbfdot(sigma=sigest(BUPA$X))

# set kern=kfun when fitting a kerndwd object
data(BUPA)
BUPA$X = scale(BUPA$X, center=TRUE, scale=TRUE)
lambda = 10^(seq(-3, 3, length.out=10))
m1 = kerndwd(BUPA$X, BUPA$y, kern=kfun,
  qval=1, lambda=lambda, eps=1e-5, maxit=1e5)
```

---

plot.cv.kerndwd *plot the cross-validation curve*

Description

Plot cross-validation error curves with the upper and lower standard deviations versus log `lambda` values.

Usage

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

Arguments

x	A fitted cv.kerndwd object.
sign.lambda	Against log(lambda) (default) or its negative if sign.lambda=-1.
...	Other graphical parameters being passed to plot.

Details

This function plots the cross-validation error curves. This function is modified based on the plot.cv function of the glmnet package.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang@umn.edu>

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244

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.
http://www.jstatsoft.org/v33/i01/paper

See Also

cv.kerndwd.

Examples

```
set.seed(1)
data(BUPA)
BUPA$X = scale(BUPA$X, center=TRUE, scale=TRUE)
lambda = 10^(seq(-3, 3, length.out=10))
kern = rbfdot(sigma=sigest(BUPA$X))
m.cv = cv.kerndwd(BUPA$X, BUPA$y, kern,
  qval=1, lambda=lambda, eps=1e-5, maxit=1e5)
m.cv
```

`plot.kerndwd` *plot coefficients*

Description

Plot the solution paths for a fitted `kerndwd` object.

Usage

```
## S3 method for class 'kerndwd'
plot(x, color=FALSE, ...)
```

Arguments

`x`	A fitted "`kerndwd`"" model.
`color`	If `TRUE`, plots the curves with rainbow colors; otherwise, with gray colors (default).
`...`	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 `glmnet` package.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang@umn.edu>

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244
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.
http://www.jstatsoft.org/v33/i01/paper

See Also

`kerndwd`, `predict.kerndwd`, `coef.kerndwd`, `plot.kerndwd`, and `cv.kerndwd`.

Examples

```
data(BUPA)
BUPA$X = scale(BUPA$X, center=TRUE, scale=TRUE)
lambda = 10^(seq(-3, 3, length.out=10))
kern = rbfdot(sigma=sigest(BUPA$X))
m1 = kerndwd(BUPA$X, BUPA$y, kern, qval=1,
  lambda=lambda, eps=1e-5, maxit=1e5)
plot(m1, color=TRUE)
```

`predict.kerndwd` *predict class labels for new observations*

Description

Predict the binary class labels or the fitted values of an `kerndwd` object.

Usage

```
## S3 method for class 'kerndwd'
predict(object, kern, x, newx, type=c("class", "link"), ...)
```

Arguments

`object`	A fitted `kerndwd` object.
`kern`	The kernel function used when fitting the `kerndwd` object.
`x`	The predictor matrix, i.e., the `x` matrix used when fitting the `kerndwd` 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`.
`type`	`"class"` or `"link"`? `"class"` produces the predicted binary class labels and `"link"` returns the fitted values. Default is `"class"`.
`...`	Not used. Other arguments to `predict`.

Details

If `"type"` is `"class"`, the function returns the predicted class labels. If `"type"` is `"link"`, the result is $\beta_0 + x_i'\beta$ for the linear case and $\beta_0 + K_i'\alpha$ for the kernel case.

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@umn.edu>

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244

See Also

kerndwd

Examples

```
data(BUPA)
BUPA$X = scale(BUPA$X, center=TRUE, scale=TRUE)
lambda = 10^(seq(-3, 3, length.out=10))
kern = rbfdot(sigma=sigest(BUPA$X))
m1 = kerndwd(BUPA$X, BUPA$y, kern,
  qval=1, lambda=lambda, eps=1e-5, maxit=1e5)
predict(m1, kern, BUPA$X, tail(BUPA$X))
```

---

tunedwd *fast tune procedure for DWD*

Description

A fast implementaiton of cross-validation for kerndwd to find the optimal values of the tuning parameter lambda.

Usage

```
tunedwd(x, y, kern, lambda, qvals=1, eps=1e-5, maxit=1e+5, nfolds=5, foldid=NULL)
```

Arguments

x	A matrix of predictors, i.e., the matrix x used in kerndwd.
y	A vector of binary class labels, i.e., the y used in kerndwd. y has two levels.
kern	A kernel function.
lambda	A user specified lambda candidate sequence for cross-validation.
qvals	A vector containing the index of the generalized DWD. Default value is 1.
eps	The algorithm stops when (i.e. $\sum_j(\beta_j^{new} - \beta_j^{old})^2$ is less than eps, where $j = 0, \ldots, p$. Default value is 1e-5.
maxit	The maximum of iterations allowed. Default is 1e5.
nfolds	The number of folds. Default value is 5. The allowable range is from 3 to the sample size.
foldid	An optional vector with values between 1 and nfold, representing the fold indices for each observation. If supplied, nfold can be missing.

Details

This function returns the best tuning parameters q and lambda by cross-validation. An efficient tune method is employed to accelerate the algorithm.

Value

A tunedwd.kerndwd object including the cross-validation results is return.

lam.tune	The optimal lambda value.
q.tune	The optimal q value.

Author(s)

Boxiang Wang and Hui Zou
Maintainer: Boxiang Wang <boxiang@umn.edu>

References

Wang, B. and Zou, H. (2018) "Another Look at Distance Weighted Discrimination," *Journal of Royal Statistical Society, Series B*, **80**(1), 177–198.
https://rss.onlinelibrary.wiley.com/doi/10.1111/rssb.12244
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.
http://www.jstatsoft.org/v33/i01/paper

See Also

kerndwd.

Examples

```
set.seed(1)
data(BUPA)
BUPA$X = scale(BUPA$X, center=TRUE, scale=TRUE)
lambda = 10^(seq(-3, 3, length.out=10))
kern = rbfdot(sigma=sigest(BUPA$X))
ret = tunedwd(BUPA$X, BUPA$y, kern, qvals=c(1,2,10), lambda=lambda, eps=1e-5, maxit=1e5)
ret
```

Index