

Package ‘kmr’

August 4, 2017

Title Kernel multitask regression

Version 0.1.0

Description The package implements a kernel multitask regression algorithm to solve simultaneously several regression problems.

Depends R (≥ 3.2)

Imports parallel, graphics, stats

Suggests Hmisc, knitr, rmarkdown

License Artistic-2.0

Encoding UTF-8

LazyData true

URL <https://github.com/jpvert/kmr>

BugReports <https://github.com/jpvert/kmr/issues>

RoxygenNote 6.0.1

R topics documented:

cv.kmr	1
evalpred	3
gausskernel	4
kmr	4
plot.cv.kmr	6
predict.cv.kmr	6
predict.kmr	7
Index	8

cv.kmr	<i>Cross-validation for KMR</i>
--------	---------------------------------

Description

Does a k-fold cross-validation for kmr, and returns a fitted KMR model, CV performance scores and optimal values for the regularization parameter lambda.

Usage

```
cv.kmr(x, y, kx_type = c("linear", "gaussian", "precomputed"),
       kx_option = list(sigma = 1), kt_type = c("multitask", "empirical",
       "precomputed"), kt_option = list(alpha = 1, kt = NULL),
       lambda = 10^(-5:5), type.measure = c("ci", "mse", "cor"), nfolds = 5,
       nrepeats = 1, seed = 9182456, mc.cores = 1)
```

Arguments

x	x input matrix as in kmr.
y	y output matrix as in kmr.
kx_type	Kernel type for observations as in kmr.
kx_option	Optional list of parameters for the observation kernel as in kmr.
kt_type	Kernel type for tasks as in kmr.
kt_option	Optional list of parameters for the task kernel as in kmr.
lambda	Sequence of (more than one) values for lambda that must be tested. Default is $10^{(-5:5)}$.
type.measure	Measure type for evaluating performance. Possible options are "ci" (concordance index), "mse" (mean squared error), "cor" (pearson correlation). Default is "ci".
nfolds	Number of folds for cross-validation. Default is 5.
nrepeats	Number of times the k-fold cross-validation is performed Default is 1.
seed	A seed number for the random number generator (useful to have the same CV splits).
mc.cores	Number of parallelable CPU cores to use.

Value

An object (list) of class "cv.kmr", which can then be used to make predictions for the different tasks on new observations, as a list containing the following slots:

...	Outputs of a CV-fitted KMR model as in "kmr".
meanCV	A matrix of CV performance scores of dim ntask x nlambda.
bestlambda	A vector of lambdas of length ntask, each corresp to the underlying min CV score.
bestCV	A vector of min CV performance scores of length ntask.
lambda	Lambda sequence against which a model is tested.
type.measure	Measure type for evaluating performance.

References

Bernard, E., Jiao, Y., Scornet, E., Stoven, V., Walter, T., and Vert, J.-P. (2017). Kernel multitask regression for toxicogenetics. [bioRxiv-171298](#).

See Also

[kmr](#)

Examples

```
# Data
ntr <- 80
ntst <- 20
nt <- 50
p <- 20
xtrain <- matrix(rnorm(ntr*p),ntr,p)
xtest <- matrix(rnorm(ntst*p),ntst,p)
ytrain <- matrix(rnorm(ntr*nt),ntr,nt)
ytest <- matrix(rnorm(ntst*nt),ntst,nt)

# Train with Gaussian RBF kernel for x and multitask kernel for t
cvmo <- cv.kmr(x=xtrain, y=ytrain, kx_type="gaussian", kx_option=list(sigma=100),
               kt_type="multitask", kt_option=list(alpha=0.5), type.measure="mse")

# Plot
plot(cvmo)
# Predict
cvpred <- predict(cvmo, xtest)
# Evaluate
evalpred(cvpred, ytest, "mse")
```

evalpred

Evaluate prediction performance

Description

Computes performance scores of predicted against true responses.

Usage

```
evalpred(ypred, ytest, type.measure = c("ci", "mse", "cor"))
```

Arguments

ypred	Matrix or list of matrices (e.g. each corresponding to a tested lambda) of predicted reponses, of dimension nobs x ntask.
ytest	Matrix of true reponses, of dimension nobs x ntask.
type.measure	Measure type for evaluating performance. Possible options are "ci" (concordance index), "mse" (mean squared error), "cor" (pearson correlation). Default is "ci".

Value

A matrix of mean performance scores (averaged over observations), of dimension ntask x nlambda where nlambda denotes the length of the list ypred.

Note

ypred and ytest are not interchangeably equivalent for type.measure="ci", as ytest is treated as true responses and used to determine denominator in computing C-index.

gausskernel	<i>Gaussian RBF kernel</i>
-------------	----------------------------

Description

Computes the Gaussian RBF kernel $k(x, x') = \exp(-|x - x'|^2 / (2 * \sigma^2))$.

Usage

```
gausskernel(x, sigma = 1, x2)
```

Arguments

x	Input matrix of covariates with samples in rows.
sigma	Bandwidth of the Gaussian RBF kernel. Default is 1.
x2	(Optional) a second matrix of covariates with samples in rows.

Value

If x2 is provided, returns the kernel matrix crossing samples in x and samples in x2. If x2 is not provided, returns the kernel Gram matrix of x versus itself.

kmr	<i>Kernel multitask regression</i>
-----	------------------------------------

Description

Fits a kernel multitask regression (KMR) model.

Usage

```
kmr(x, y, kx_type = c("linear", "gaussian", "precomputed"),
    kx_option = list(sigma = 1), kt_type = c("multitask", "empirical",
    "precomputed"), kt_option = list(alpha = 1, kt = NULL))
```

Arguments

x	Input matrix of covariates, of dimension nobs x nvars; each row is an observation vector. If a precomputed kernel is used, then x is the square Gram matrix.
y	Output matrix of responses. y should be an nobs x ntasks matrix, where each row corresponds to an observation and each column to a task.
kx_type	Kernel for observations. kx_type="linear" is the linear kernel (default). kx_type="gaussian" is the Gaussian RBF kernel with bandwidth sigma=1 by default, or any other valued is passed as an element of the kx_option list. kx_type="precomputed" assumes that the x matrix provided is the kernel Gram matrix.
kx_option	An optional list of parameters for the observation kernel, including elements such as "sigma".

kt_type	Kernel for tasks. kt_type="multitask" (default) is the multitask kernel with parameter $0 \leq \alpha \leq 1$, which interpolates between the Dirac kernel for $\alpha = 1$ (default) and the constant kernel for $\alpha = 0$. The parameter <i>alpha</i> can be passed to the kt_option list as a field alpha. kt_type="empirical" takes the empirical correlation between outputs as kernel between the tasks. kt_type="precomputed" allows to provide a precomputed kernel as a field kt in the kt_type list.
kt_option	An optional list of parameters for the task kernel, including elements such as "alpha" and "kt".

Value

An object (list) of class "kmr", which can then be used to make predictions for the different tasks on new observations.

References

Bernard, E., Jiao, Y., Scornet, E., Stoven, V., Walter, T., and Vert, J.-P. (2017). Kernel multitask regression for toxicogenetics. [bioRxiv-171298](#).

Examples

```
# Data
ntr <- 80
ntst <- 20
nt <- 50
p <- 20
xtrain <- matrix(rnorm(ntr*p),ntr,p)
xtest <- matrix(rnorm(ntst*p),ntst,p)
ytrain <- matrix(rnorm(ntr*nt),ntr,nt)
ytest <- matrix(rnorm(ntst*nt),ntst,nt)

# Case I: with linear kernel for x and empirical kernel for t
# Train
mo1 <- kmr(x=xtrain, y=ytrain, kx_type="linear", kt_type="empirical")
# Predict
pred1 <- predict(mo1, xtest)
# Evaluate
evalpred(pred1, ytest, "mse")

# Case II: with precomputed kernel matrices
# Kernel matrices
kxtrain <- tcrossprod(xtrain)
kxtest <- tcrossprod(xtest,xtrain)
kttrain <- cor(ytrain)
# Train
mo2 <- kmr(x=kxtrain, y=ytrain, kx_type="precomputed",
           kt_type="precomputed", kt_option=list(kt=kttrain))
# Predict
pred2 <- predict(mo2, kxtest)
# Evaluate
evalpred(pred2, ytest, "mse")
```

plot.cv.kmr	<i>Plot the cross-validation curve produced by "cv.kmr"</i>
-------------	---

Description

Plots the cross-validation curve as a function of lambda values tested, as well as the optimal lambda selected by cross-validation for each task.

Usage

```
## S3 method for class 'cv.kmr'
plot(x, ...)
```

Arguments

x	Fitted "cv.kmr" model object.
...	Other graphical parameters to matplotlib.

Value

A plot is produced, and nothing returned.

See Also

[cv.kmr](#)

predict.cv.kmr	<i>Make prediction with a "cv.kmr" object</i>
----------------	---

Description

Similar to other predict methods, this function predicts fitted values from a fitted "cv.kmr" object.

Usage

```
## S3 method for class 'cv.kmr'
predict(object, newx, lambda = "lambda.opt", ...)
```

Arguments

object	Fitted "cv.kmr" model object.
newx	Matrix of new values for x or kernel matrix for new values crossing old values for x, at which predictions are to be made.
lambda	Value(s) of the regularization parameter, a single scalar or a sequence of values, or "lambda.opt" which allows different lambdas for each task tuned by cross-validation. Default is "lambda.opt".
...	Not used. Other arguments to predict.

Value

A matrix of predicted values for the new samples (in rows) and all tasks (in columns), corresponding to the regularization parameter `lambda`. If `lambda` is a list of specified values, the function returns a list of matrices, corresponding to the predictions for the different values of `lambda` in the list. If `lambda` is `"lambda.opt"`, a single matrix returned too.

Note

Note that `"lambda.opt"` will allow different `lambdas` for each task tuned by cross-validation; otherwise, specified value(s) of `lambda` will be fixed for all tasks.

See Also

[cv.kmr](#)

predict.kmr	<i>Make prediction with a "kmr" object</i>
-------------	--

Description

Similar to other predict methods, this function predicts fitted values from a fitted "kmr" object.

Usage

```
## S3 method for class 'kmr'
predict(object, newx, lambda = 1, ...)
```

Arguments

<code>object</code>	Fitted "kmr" model object.
<code>newx</code>	Matrix of new values for <code>x</code> or kernel matrix for new values crossing old values for <code>x</code> , at which predictions are to be made.
<code>lambda</code>	Value(s) of the regularization parameter, a single scalar or a sequence of values. Default is 1.
<code>...</code>	Not used. Other arguments to predict.

Value

A matrix of predicted values for the new samples (in rows) and all tasks (in columns), corresponding to the regularization parameter `lambda`. If `lambda` is a list of values, the function returns a list of matrices, corresponding to the predictions for the different values of `lambda` in the list.

See Also

[kmr](#)

Index

`cv.kmr`, [1](#), [6](#), [7](#)

`evalpred`, [3](#)

`gausskernel`, [4](#)

`kmr`, [2](#), [4](#), [7](#)

`plot.cv.kmr`, [6](#)

`predict.cv.kmr`, [6](#)

`predict.kmr`, [7](#)