

# Package ‘lasvmR’

August 27, 2015

**Type** Package

**Title** A Simple Wrapper for the LASVM Solver

**Version** 0.1.2

**Date** 2015-08-27

**Author** Aydin Demircioglu

**Maintainer** Aydin Demircioglu <aydin.demircioglu@ini.rub.de>

**Description** This is a simple wrapper for the LASVM Solver (see <http://leon.bottou.org/projects/lasvm>). LASVM is basically an online variant of the SMO solver.

**Copyright** Original C/C++ code (GPL) by Leon Bottou (<http://leon.bottou.org>), Yann Le Cun, AT&T Corp, NECI/NEC Labs America and Neuristique.

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.11.6), checkmate (>= 1.5.1)

**LinkingTo** Rcpp

**Suggests** knitr, testthat

**VignetteBuilder** knitr

**URL** <http://github.com/aydindemircioglu/lasvmR>

**BugReports** <http://github.com/aydindemircioglu/lasvmR/issues>

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-08-27 09:00:02

## R topics documented:

lasvmPredict . . . . .	2
lasvmPredictWrapper . . . . .	2
lasvmR . . . . .	3
lasvmTrain . . . . .	3
lasvmTrainWrapper . . . . .	5

**Index****6**


---

lasvmPredict	<i>lasvmPredict</i>
--------------	---------------------

---

**Description**

Use lasvm to train a given problem.

**Usage**

```
lasvmPredict(x, model, verbose = FALSE)
```

**Arguments**

x	data matrix
model	trained model
verbose	verbose output?

**Value**

a list consisting of predictions the predicted labels

**Examples**

```
model = lasvmR::lasvmTrain (x = as.matrix(iris[seq(1,150,2),1:4]),
y = (as.numeric(iris[seq(1,150,2),5]) %% 2)*2-1,
gamma = 1,
cost = 1,
kernel = 2)
ytrue = (as.numeric(iris[seq(2,150,2),5]) %% 2)*2-1
result = lasvmPredict (x = as.matrix(iris[seq(2,150,2),1:4]), model)
ypred = result$predictions
error = sum(abs(ypred - ytrue))/length(ytrue)
cat ("Error rate =", error*100)
```

---

lasvmPredictWrapper	<i>lasvmPredictWrapper</i>
---------------------	----------------------------

---

**Description**

Use lasvm to predict a given problem.

**Usage**

```
lasvmPredictWrapper(x, SV, elif, gamma, kdegree, kcoef0, bias, kerneltype,
verbose = FALSE)
```

**Arguments**

x	data matrix
SV	matrix of support vectors
elif	vector of alphas
gamma	gamma of RBF kernel
kdegree	degree for POLY kernel
kcoef0	coef0 for kernel
bias	bias term
kerneltype	type of kernel to use
verbose	verbose output?

**Value**

a list consisting of predictions just the predictions

---

lasvmR	<i>lasvmR: A Simple Wrapper for the LASVM Solver</i>
--------	------------------------------------------------------

---

**Description**

This is a simple wrapper for LASVM Solver (see <http://leon.bottou.org/projects/lasvm>). LASVM is an online variant of the SMO solver.

**lasvmR functions**

lasvmTrain lasvmPredict

---

lasvmTrain	<i>lasvmTrain</i>
------------	-------------------

---

**Description**

Use lasvm to train a given problem.

**Usage**

```
lasvmTrain(x, y, gamma = 1, cost = 1, degree = 3, coef0 = 0,
optimizer = 1, kernel = 2, selection = 0, termination = 0,
sample = 1e+08, cachesize = 256, bias = 1, epochs = 1,
epsilon = 0.001, verbose = FALSE)
```

**Arguments**

x	data matrix
y	labels
gamma	RBF kernel parameter
cost	regularization parameter
degree	degree for poly kernel
coef0	coefficient for poly kernel
optimizer	type of optimizer
kernel	kernel type
selection	selection strategy
termination	criterion for stopping
sample	time for stopping/number of iterations tec
cacheSize	size of kernel cache
bias	use bias?
epochs	number of epochs
epsilon	stopping criterion parameter
verbose	verbose output?

**Value**

a list consisting of alpha alpha for SVs as vector SV support vectors as matrix

**Examples**

```

model = lasvmR::lasvmTrain (x = as.matrix(iris[seq(1,150,2),1:4]),
y = (as.numeric(iris[seq(1,150,2),5]) %% 2)*2-1,
gamma = 1,
cost = 1,
kernel = 2)
ytrue = (as.numeric(iris[seq(2,150,2),5]) %% 2)*2-1
result = lasvmPredict (x = as.matrix(iris[seq(2,150,2),1:4]), model)
ypred = result$predictions
error = sum(abs(ypred - ytrue))/length(ytrue)
cat ("Error rate =", error*100)

```

---

lasvmTrainWrapper      *lasvmTrainWrapper*

---

### Description

Use lasvm to train a given problem.

### Usage

```
lasvmTrainWrapper(x, y, gamma, cost, degree = 3, coef0 = 0L,
  optimizer = 1L, kernel = 2L, selection = 0L, termination = 0L,
  sample = 0, cachesize = 256L, bias = 1L, epochs = 1L,
  epsilon = 0.001, verbose = FALSE)
```

### Arguments

x	data matrix
y	training labels
gamma	RBF kernel bandwidth
cost	regularization constant
degree	degree for poly kernel
coef0	coefficient for poly kernel
optimizer	type of optimizer
kernel	kernel type
selection	selection strategy
termination	criterion for stopping
sample	parameter for stopping criterion, e.g. seconds
cachesize	size of kernel cache
bias	use bias?
epochs	number of epochs
epsilon	stopping criterion parameter
verbose	verbose output?

### Value

a list consisting of SV matrix of support vectors alpha vector of alpha coefficients bias bias term

# Index

lasvmPredict, [2](#)  
lasvmPredictWrapper, [2](#)  
lasvmR, [3](#)  
lasvmR-package (lasvmR), [3](#)  
lasvmTrain, [3](#)  
lasvmTrainWrapper, [5](#)