Package: l1penSVMs (via r-universe)

October 21, 2026

Title Regularization Paths of L1-Penalized SVMs

Version 0.1.0

Description Computes regularization paths for L1-penalized support vector machine (SVM) models, including the standard SVM, receiver operating characteristic SVM (ROC-SVM), and fraud detection SVM (FD-SVM). The package implements efficient path algorithms to obtain solutions over a wide range of regularization parameter values. It also provides cross-validation for selecting an optimal regularization parameter, along with functions for prediction, visualization, and summarization of the fitted models. For more details, see Zhu et al. (2003) <https://proceedings.neurips.cc/paper_files/paper/2003/file/49d4b2faeb4b7b9e745775793141e2b2-Paper.pdf>, Kim et al. (2021) <doi:10.1002/sta4.400>, Park et al. (2023) <doi:10.1007/s42952-023-00207-6>.

License GPL-3

Config/roxygen2/version 8.1.0

Encoding UTF-8

Imports lpSolve, fastDummies, MASS

NeedsCompilation no

Author Hyungwoo Kim [aut, cre], Minhyoung Park [aut], Seung Jun Shin [aut]

Maintainer Hyungwoo Kim <khw7682@pknu.ac.kr>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-21 16:00:02 UTC

RemoteUrl https://github.com/cran/l1penSVMs

RemoteRef HEAD

RemoteSha 7b342ae65e7b763d21a1e39167f46038733258e5

Contents

cv.l1svm 2
l1.fdsvm.initialization 4
l1.fdsvmpath 5
l1.rocsvmpath 6
l1.svm.initialization 7
l1.svmpath 8
l1svmpath 9
lpfit.l1svm 11
plot.l1svmpath 12
predict.l1svmpath 13
print.cv.l1svm 14

Index **15**

`cv.l1svm` *Perform CV to choose the optimal t for three L1-penalized Support Vector Machine models*

Description

This function performs k-fold cross-validation to select the optimal for the L1-penalized Support Vector Machine (SVM), Receiver Operating Characteristic Support Vector Machine (ROC-SVM) and Fraud Detection Support Vector Machine (FD-SVM) based on the regularization path algorithm.

Usage

```
cv.l1svm(
  x,
  y,
  classification = c("SVM", "ROCSVM", "FDSVM"),
  nfold = NULL,
  sensitivity = NULL,
  init.t = NULL,
  max.iter = NULL,
  standardize = TRUE,
  incomp.size = NULL
)
```

Arguments

`x`	The data matrix (n x p) with n rows (observations) on p variables (columns); categorical variables are automatically one-hot encoded.
`y`	Response vector with class labels in {-1, 1}. Labels given as {0, 1} or as a two-level factor/character are automatically converted to this format.

`classification`	The classification method is to be applied. Either "SVM", "ROC-SVM" and "FD-SVM"
`nfold`	The number of folds in the cross-validation (CV), default is `nfold=5` for "SVM" and "ROC-SVM" and `nfold=2` for "FD-SVM"
`sensitivity`	The minimum proportion of actual fraud cases that are correctly classified as fraud for the "FD-SVM".
`init.t`	For "SVM" and "FD-SVM", the initial regularization parameter with sufficiently small t; default is 1e-5
`max.iter`	The maximum number of iterations (total across the entire path algorithm); default is 10000
`standardize`	Standardize each column of the data matrix to have zero mean and unit variance
`incomp.size`	Sample size for the incomplete U-statistic in ROC-SVM; default is the number of observations

Details

The `cv.l1.svm` determines the optimal regularization parameter t and yields the cofficient estimates at the optimal t

Value

An object `cv.l1svm` returns a list with following components:

t A vector of t values used in K-fold CV.

cv.err A vector of the averaged CV errors for different t values.

cv.AUC.risk A vector of the averaged CV AUC-risks for different t values.

cv.sens A vector of the averaged CV sensitivies for different t values.

cv.spec A vector of the averaged CV specificies for different t values.

t.opt Optimal value of t for thee L1-penalized SVM models.

sens.opt Optimal value of sensitivity at t=t.opt for the L1-penalized FD-SVM.

spec.opt Optimal value of specificity at t=t.opt for the L1-penalized FD-SVM.

params.opt The estimated coefficients at the optimal parameter t for three L1-penalized SVM.

Author(s)

Hyungwoo Kim, MinHyung Park, Seung Jun Shin

References

Ji Zhu, Saharon Rosset, Robert Tibshirani and Trevor Hastie, "1-norm Support Vector Machines", Advances in neural information processing systems, 16.

Hyungwoo Kim, Insuk Sohn and Seung Jun Shin, "Regularization paths of L1-penalized ROC Curve-Optimizing Support Vector Machines", Stat, 10(1), e400.

Minhyoung Park, Hyungwoo Kim and Seung Jun Shin, "L1-penalized Fraud Detection Support Vector Machines", under

See Also

l1.svmpath, l1.rocsvmpath, l1.fdsvmpath, plot.l1svmpath, predict.l1svmpath

Examples

```
library(l1penSVMs)
set.seed(2)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))
# SVM
cv.svm <- cv.l1svm(x, y, classification="SVM")

# ROCSVM
cv.rocsvm <- cv.l1svm(x, y, classification="ROCSVM")

# FDSVM
cv.fdsvm <- cv.l1svm(x, y, classification="FDSVM", sensitivity=0.9)
```

---

l1.fdsvm.initialization

Finding the initial solution for L1-penalized Fraud Detection Support Vector Machine

---

Description

This algorithm provides the initial solution of L1-penalized Fraud Detection Support Vector Machine (FD-SVM) for sufficiently small t.

Usage

```
l1.fdsvm.initialization(x, y, init.t)
```

Arguments

x	The data matrix after dummy variable encoding
y	The {-1, 1} valued response variable (Fraud +1, normal -1)
init.t	Initial regularization parameter with sufficiently small t; default is 1e-5

Value

The intercept and coefficients at initial regularization parameter

`l1.fdsvmpath` *Fit the entire solution paths for L1-penalized Fraud Detection Support Vector Machine*

Description

This algorithm computes the entire solution paths for L1-penalized FDSVM with a relatively low cost compared to linear programming problem.

Usage

```
l1.fdsvmpath(x, y, init.t, max.iter)
```

Arguments

`x`	The data matrix (n x p) with n rows (observations) on p variables (columns)
`y`	The `{-1, 1}` valued response variable (Fraud +1, normal -1)
`init.t`	Initial regularization parameter with sufficiently small t; default is 1e-5
`max.iter`	The maximum number of iterations (total across the entire path algorithm); default is 3000

Value

A 'l1.fdsvmpath' object is returned. It contains

t The sequence of regularization parameter values in the path

b0 The intercept sequence of length `length(t)`

beta A `p*length(t)` matrix of coefficients.

Author(s)

Seung Jun Shin, MinHyung Park, Hyungwoo Kim

References

Minhyoung Park, Hyungwoo Kim and Seung Jun Shin, "L1-penalized Fraud Detection Support Vector Machines", under

See Also

`l1.svmpath`, `l1.fdsvm.initialization`, `plot.l1svmpath`

Examples

```
library(l1penSVMs)
set.seed(1)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))
obj <- l1svmpath(x,y,classification="FDSVM")
```

`l1.rocsvmpath` *Fit the entire solution paths for L1-penalized Receiver Operating Characteristic Support Vector Machine*

Description

This algorithm computes the entire solution paths for L1-penalized Receiver Operating Characteristic Support Vector Machine (ROC-SVM)

Usage

```
l1.rocsvmpath(x, y, max.iter, incomp.size)
```

Arguments

`x`	The data matrix after dummy variable encoding
`y`	The `{-1, 1}` valued response variable
`max.iter`	The maximum number of iterations (total across the entire path algorithm); default is 3000.
`incomp.size`	Sample Size for the incomplete U-statistic in ROC-SVM; default is 2n.

Value

A 'l1.rocsvmpath' object is returned. It contains

beta A `p*length(t)` matrix of coefficients.

t The sequence of regularization parameter values in the path

x The data matrix after dummy variable encoding

y The `{-1, 1}` valued response variable

Author(s)

Seung Jun Shin, MinHyung Park, Hyungwoo Kim

References

Hyungwoo Kim, Insuk Sohn and Seung Jun Shin, "Regularization paths of L1-penalized ROC Curve-Optimizing Support Vector Machines", Stat, 10(1), e400.

See Also

`l1.svmpath`, `plot.l1svmpath`

Examples

```
library(l1penSVMs)
set.seed(1)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))
obj <- l1svmpath(x,y,classification="ROCSVM")
```

`l1.svm.initialization` *Finding the initial solution for L1-penalized SVM*

Description

This algorithm provides the initial solution of L1-penalized SVM for sufficiently small t.

Usage

```
l1.svm.initialization(x, y, init.t)
```

Arguments

`x`	The data matrix after dummy variable encoding
`y`	The `{-1, 1}` valued response variable
`init.t`	Initial regularization parameter with sufficiently small t; default is 1e-5

Value

The intercept and coefficients at initial regularization parameter

`l1.svmpath` *Fit the entire solution paths for L1-penalized Support Vector Machine*

Description

This algorithm computes the entire solution paths for L1-penalized Support Vector Machine (SVM)

Usage

```
l1.svmpath(x, y, init.t, max.iter)
```

Arguments

`x`	The data matrix after dummy variable encoding
`y`	The `{-1, 1}` valued response variable
`init.t`	The initial regularization parameter with sufficiently small t; default is 1e-5
`max.iter`	The maximum number of iterations (total across the entire path algorithm); default is 3000.

Value

An object of class `l1.svmpath`. It contains

b0 A vector of intercept estimates for different values of t

beta A `p*length(t)` matrix of coefficients

t The sequence of regularization parameter values in the path

x The data matrix after dummy variable encoding

y The `{-1, 1}` valued response variable

Author(s)

Seung Jun Shin, MinHyung Park, Hyungwoo Kim

References

Hyungwoo Kim, Insuk Sohn and Seung Jun Shin, "Regularization paths of L1-penalized ROC Curve-Optimizing Support Vector Machines", Stat, 10(1), e400.

See Also

`l1svmpath`, `plot.l1svmpath`

Examples

```
library(l1penSVMs)
set.seed(1)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))
obj <- l1svmpath(x,y,classification="SVM")
```

`l1svmpath` *The entire regularization path for L1-penalized Support Vector Machine models*

Description

This provides the entire solution paths for L1-penalized Support Vector Machine (SVM), Receiver Operating Characteristic Support Vector Machine (ROC-SVM) and Fraud Detection Support Vector Machine (FD-SVM) with a relatively low cost compared to linear programming problem.

Usage

```
l1svmpath(
  x,
  y,
  classification = c("SVM", "ROCSVM", "FDSVM"),
  init.t = NULL,
  max.iter = NULL,
  standardize = TRUE,
  incomp.size = NULL,
  fraud.label.fixed = FALSE
)
```

Arguments

`x`	The data matrix (n x p) with n rows (observations) on p variables (columns); categorical variables are automatically one-hot encoded.
`y`	Response vector with class labels in {-1, 1}. Labels given as {0, 1} or as a two-level factor/character are automatically converted to this format.
`classification`	The classification method is to be applied. Either "SVM", "ROCSVM" or "FDSVM"
`init.t`	For "SVM" and "FDSVM" models, the initial regularization parameter with sufficiently small t; default is 1e-5
`max.iter`	The maximum number of iterations (total across the entire path algorithm); default is 10000
`standardize`	Standardize each column of the data matrix to have zero mean and unit variance

`incomp.size` Sample size for the incomplete U-statistic in ROC-SVM; default is the number of observations

`fraud.label.fixed` If TRUE, preserves the existing {-1, +1} class encoding for FD-SVM.

Details

The algorithm used in `l1.svmpath()` for "SVM" is described in "1-norm Support Vector Machines" by Ji Zhu, Saharon Rosset, Robert Tibshirani and Trevor Hastie. "ROCSVM" is described in "Regularization paths of L1-penalized ROC Curve-Optimizing Support Vector Machines" by Hyungwoo Kim, Insuk Sohn and Seung Jun Shin. "FDSVM" is described in "L1-penalized Fraud Detection Support Vector Machines" by Minhyoung Park, Hyungwoo Kim and Seung Jun Shin. For more information, see references.

Value

A 'l1.svmpath' object is returned. It contains

t The sequence of regularization parameter values in the path

b0 For "SVM" and "FDSVM" models, the intercept sequence of length `length(t)`

beta A numeric matrix with `length(t)` rows and p columns, containing coefficient estimates along the regularization path.

x The data matrix after dummy variable encoding

y The `{-1, +1}` valued response variable (For "FDSVM", Fraud +1 and Normal -1)

Author(s)

Seung Jun Shin, MinHyung Park, Hyungwoo Kim

References

Ji Zhu, Saharon Rosset, Robert Tibshirani and Trevor Hastie, "1-norm Support Vector Machines", Advances in neural information processing systems, 16.

Hyungwoo Kim, Insuk Sohn and Seung Jun Shin, "Regularization paths of L1-penalized ROC Curve-Optimizing Support Vector Machines", Stat, 10(1), e400.

Minhyoung Park, Hyungwoo Kim and Seung Jun Shin, "L1-penalized Fraud Detection Support Vector Machines", under

See Also

`l1.svmpath`, `l1.rocsvmpath`, `l1.fdsvmpath`, `plot.l1svmpath`, `predict.l1svmpath`

Examples

```
library(l1penSVMs)
set.seed(1)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
```

```
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))

# SVM
obj <- l1svmpath(x,y,classification="SVM")

# ROCSVM
obj <- l1svmpath(x,y,classification="ROCSVM")

# FDSVM
obj <- l1svmpath(x,y,classification="FDSVM")
```

`lpfit.l1svm` *Fit the L1-penalized Support Vector Machine models at a specific value of t using the the* `lp` *function*

Description

It provides the estimated coefficients for L1-penalized Support Vector Machine (SVM) and Receiver Operating Characteristic Support Vector Machine (ROC-SVM) using linear programming (LP).

Usage

```
lpfit.l1svm(x, y, t, classification = c("SVM", "ROCSVM"))
```

Arguments

`x`	The data matrix (n x p) with n rows (observations) on p variables (columns)
`y`	The `{-1, 1}` valued response variable
`t`	The value of regularization parameter
`classification`	The classification method is to be applied. Either "SVM" (the default) or "ROCSVM"

Value

A numeric vector containing the estimated coefficients at the specified regularization parameter `t`. For SVM, the first element is the intercept. For ROC-SVM, only slope coefficients are returned.

Examples

```
set.seed(1)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))
obj <- lpfit.l1svm(x, y, t=1.5, classification="SVM")
```

`plot.l1svmpath` *Plot the Entire Regularization Paths for L1-penalized SVM models*

Description

Plot the whole regularization paths for L1-penalized Support Vector Machine (SVM), Receiver Operating Characteristic Support Vector Machine (ROC-SVM) and Fraud Detection Support Vector Machine (FD-SVM)

Usage

```
## S3 method for class 'l1svmpath'
plot(x, ...)
```

Arguments

`x`	An object of class `l1svmpath`.
`...`	Additional graphical arguments passed to plotting functions.

Details

This function is a method for the generic `plot()` function,

Value

Invisibly returns the input object of class `l1svmpath`. The function displays the regularization paths.

Examples

```
library(l1penSVMs)
set.seed(1)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))

# SVM
obj <- l1svmpath(x, y, classification="SVM")
plot(obj)
```

predict.l1svmpath *Predict the class label or classification score*

Description

Predict the class label or classification score

Usage

```
## S3 method for class 'l1svmpath'
predict(object, x, t = NULL, ...)
```

Arguments

`object`	A fitted regularization path object of class `l1svmpath`.
`x`	The test data matrix (n x p), with n rows (observations) on p variables (columns)
`t`	The value of regularization parameter
`...`	Additional arguments passed to the prediction method.

Details

This function returns the predicted class labels or the fitted classification scores for L1-penalized Support Vector Machine (SVM), Receiver Operating Characteristic Support Vector Machine (ROC-SVM) and Fraud Detection Support Vector Machine (FD-SVM)

Value

The predicted class label for the L1-penalized SVM and FDSVM, or the fitted classification scores for the L1-penalized ROCSVM.

Examples

```
library(l1penSVMs)
set.seed(1)
n <- 100; p <- 10
x <- matrix(rnorm(n*p),n,p)
beta  <- c(2,-1,0.5,rep(0,p-3))
fx  <- -2 + crossprod(t(x),beta)
y   <- sign(fx + rnorm(n,0,1))

# SVM
obj <- l1svmpath(x,y,classification="SVM")
predict(obj, x=x, t=1.5)
```

`print.cv.l1svm` *Print Cross-Validation Results*

Description

Print Cross-Validation Results

Usage

```
## S3 method for class 'cv.l1svm'
print(x, ...)
```

Arguments

`x`	An object of class `cv.l1svm`.
`...`	Additional arguments.

Value

Invisibly returns the input object of class "cv.l1svm". The function prints a summary of the cross-validation results.

Index

cv.l1svm, 2

l1.fdsvm.initialization, 4, *5*
l1.fdsvmpath, *4*, 5, *10*
l1.rocsvmpath, *4*, 6, *10*
l1.svm.initialization, 7
l1.svmpath, *4, 5*, *7*, 8, *10*
l1svmpath, *8*, 9
lpfit.l1svm, 11

plot.l1svmpath, *4, 5*, *7, 8*, *10*, 12
predict.l1svmpath, *4*, *10*, 13
print.cv.l1svm, 14