

Package ‘marginTree’

January 2, 2012

Title marginTree: margin trees for high-dimensional classification

Version 1.01

Author R. Tibshirani

Description Some functions for high-dimensional classification, useful for more than 2 classes

Maintainer Rob Tibshirani <tibs@stat.stanford.edu>

LazyLoad false

LazyData false

Depends e1071

License LGPL-2

Repository CRAN

Date/Publication 2010-01-04 20:32:18

R topics documented:

marginTree	2
marginTree.confusion	3
marginTree.cv	4
marginTree.getnonzero	6
marginTree.plclust	7
marginTree.plotcv	8
marginTree.predict	9

Index	11
--------------	-----------

marginTree

A function to train a margin tree classifier

Description

A function to train a margin tree classifier. This is a hierarchical version of the support vector classifier, useful for more than 2 classes.

Usage

```
marginTree(x,y, method="complete", n.threshold=20, predict.trainingset=TRUE)
```

Arguments

x	The input data of feature values, n samples by p features
y	Class labels- vector of length n
method	Clustering method- "complete" (default and recommended); "average" or "single"
n.threshold	Number of threshold values desired (default 10)
predict.trainingset	Predict the training set? Required for computing error rates and in preparation for cross-validation. Slows the computation down a bit for large datasets. Default TRUE

Details

marginTree fits a margin tree classifier. It is useful with more than 2 outcome classes, and when the number of features exceeds the number of observations, for example in genomic and proteomic applications. Details may be found in the paper on the website listed below. This function calls the svm function from library e1071.

Value

A list with components:

marg.obj	List with components marg- the matrix of pairwise margins, and svmfit, a list of svm classifiers used to find each pairwise margin
marg.tree	The margin tree. Same format as that produced by hclust.
svm.splitters	List a svm objects for classification at each junction in the margin tree.
plot.heights	Heights for plotting the margin tree
nclasses	Number of outcome classes
nlist	For internal use
ynams	Character names of outcome classes
threshold	Threshold values tried

err	Number of training errors for each threshold value
y	Training class labels
yhat	Matrix of predicted classes, one column per threshold value
nonzero	Average number of nonzero features per threshold value

Author(s)

Robert Tibshirani and Trevor Hastie

References

Rob Tibshirani and Trevor Hastie. Tech report. Feb. 2006. Margin trees for high-dimensional classification Available at <http://www-stat.stanford.edu/~tibs/research.html>

Examples

```
#generate some data with 5 classes and 100 features
set.seed(543)

x=matrix(rnorm(40*1000),nrow=40)
y=sort(rep(1:5,8))
x[y==2 | y==3, 1:50]=x[y==2|y==3, 1:50]+1
x[y==3,51:100]=x[y==3,51:100]+1
x[y==4|y==5,1:50]=x[y==4|y==5,1:50]-1
x[y==5, 51:100]=x[y==5,51:100]+1

#train the classifier

train.obj<- marginTree(x,y)

# print out results to see training error rates

train.obj
```

marginTree.confusion *A function giving a table of true versus predicted values, from a margin tree.*

Description

A function giving a table of true versus predicted values, from a margin tree

Usage

```
marginTree.confusion(train.obj, threshold, extra=TRUE)
```

Arguments

train.obj	The result of a call to marginTree.train or marginTree.cv
threshold	The desired threshold value
extra	Should the classwise and overall error rates be returned? Default TRUE

Details

marginTree.confusion Gives a cross-tabulation of true versus predicted classes for the train.obj returned by marginTree or marginTree.cv, at the specified threshold.

Author(s)

Robert Tibshirani and Trevor Hastie

Examples

```
#generate some data with 5 classes and 100 features
set.seed(543)

x=matrix(rnorm(40*1000),nrow=40)
y=sort(rep(1:5,8))
x[y==2 | y==3, 1:50]=x[y==2|y==3, 1:50]+1
x[y==3,51:100]=x[y==3,51:100]+1
x[y==4|y==5, 1:50]=x[y==4|y==5, 1:50]-1
x[y==5, 51:100]=x[y==5,51:100]+1

#train the classifier

train.obj<- marginTree(x,y)

# print out confusion matrix

marginTree.confusion(train.obj, threshold=.5)

# do cross-validation and print out CV confusion matrix
cv.obj<-marginTree.cv(x,y,train.obj)

marginTree.confusion(cv.obj, threshold=.5)
```

marginTree.cv

Cross-validation of the margin tree classifier

Description

A function to cross-validate the margin tree classifier

Usage

```
marginTree.cv(x, y, train.obj, nfold = min(table(y)), folds = NULL, threshold = NULL, n.threshold=20)
```

Arguments

x	Data of feature values, n samples by p features
y	Vector of class labels
train.obj	Output from call to marginTree
nfold	Number of cross-validation folds (default 10)
folds	A list with nfold components, each component a vector of indices of the samples in that fold. By default a (random) balanced cross-validation is used
threshold	Optional vector of threshold values, for feature selection
n.threshold	Number of threshold value for feature selection (default 10)

Details

marginTree.cv Does K-fold cross-validation of a margin tree classifier, to choose the threshold parameter for feature selection.

Value

A list with components:

threshold	Threshold values tried
error	CV misclassification errors for each threshold
size	Average number of features used at each threshold
size.indiv	Matrix of number of features used at each split (rows) and at each threshold (columns).
yhat	Matrix of cross-validated predictions.
y	Vector of outcome values
folds	Indices of CV folds
call	The calling sequence

Author(s)

Robert Tibshirani and Trevor Hastie

References

Rob Tibshirani and Trevor Hastie. Tech report. Feb. 2006. Margin trees for high-dimensional classification Available at <http://www-stat.stanford.edu/~tibs/research.html>

Examples

```
#generate some data with 5 classes and 100 features
set.seed(543)

x=matrix(rnorm(40*1000),nrow=40)
y=sort(rep(1:5,8))
x[y==2 | y==3, 1:50]=x[y==2|y==3, 1:50]+1
```

```
x[y==3,51:100]=x[y==3,51:100]+1
x[y==4|y==5,1:50]=x[y==4|y==5,1:50]-1
x[y==5, 51:100]=x[y==5,51:100]+1
```

```
#train the classifier

train.obj<- marginTree(x,y)

# do cross-validation
cv.obj<- marginTree.cv(x,y,train.obj)

# examine results

cv.obj
```

marginTree.getnonzero *Get important features at each split of the margin tree*

Description

A function to get important features at each split of the margin tree

Usage

```
marginTree.getnonzero(train.obj, threshold)
```

Arguments

train.obj	Output from call to marginTree
threshold	Threshold for feature selection: between 0 and 1

Details

marginTree.getnonzero Does hard thresholding of the weight vector at each split in the margin tree, to select features

Value

A list– one element per split in the tree– with components

feature.scores	The coefficient for the selected feature
left.classes	The outcome classes assigned to the left branch
right.classes	The outcome classes assigned to the right branch

Author(s)

Rob Tibshirani and Trevor Hastie

References

Rob Tibshirani and Trevor Hastie. Tech report. Feb. 2006. Margin trees for high-dimensional classification Available at <http://www-stat.stanford.edu/~tibs/research.html>

Examples

```
#generate some data with 5 classes and 100 features
set.seed(543)

x=matrix(rnorm(40*100),nrow=40)
y=sort(rep(1:5,8))
x[y==2 | y==3, 1:50]=x[y==2|y==3, 1:50]+1
x[y==3,51:100]=x[y==3,51:100]+1
x[y==4|y==5,1:50]=x[y==4|y==5,1:50]-1
x[y==5, 51:100]=x[y==5,51:100]+1

#train the classifier

train.obj<- marginTree(x,y)

# examine the selected features at each split

junk<- marginTree.getnonzero(train.obj,threshold=.5)

summary(junk)
```

marginTree.plclust *Plot a margin tree*

Description

A function to plot margin tree

Usage

```
marginTree.plclust(train.obj, ...)
```

Arguments

train.obj	Output from call to marginTree
...	Additional arguments to pass to the function plclust

Details

marginTree.plclustPlots a margin tree, with height of the vertical gap between each tree junction equal to the achieved margin for that split.

Author(s)

Rob Tibshirani and Trevor Hastie

References

Rob Tibshirani and Trevor Hastie. Tech report. Feb. 2006. Margin trees for high-dimensional classification Available at <http://www-stat.stanford.edu/~tibs/research.html>

Examples

```
#generate some data with 5 classes and 100 features
set.seed(543)

x=matrix(rnorm(40*1000),nrow=40)
y=sort(rep(1:5,8))
x[y==2 | y==3, 1:50]=x[y==2|y==3, 1:50]+1
x[y==3,51:100]=x[y==3,51:100]+1
x[y==4|y==5, 1:50]=x[y==4|y==5, 1:50]-1
x[y==5, 51:100]=x[y==5,51:100]+1

#train the classifier

train.obj<- marginTree(x,y)

# plot the margin tree
marginTree.plclust(train.obj)
```

marginTree.plotcv *A function to plot the cross-validated error curves from classifier a margin tree classifier*

Description

A function to plot the cross-validated error curves from a margin tree classifier

Usage

```
marginTree.plotcv(cv.obj)
```

Arguments

cv.obj The result of a call to marginTree.cv

Details

marginTree.plotcv plots the cross-validated misclassification error curves, from a margin tree classifier. An overall plot, and a plot by class, are produced.

Author(s)

Robert Tibshirani and Trevor Hastie

Examples

```
set.seed(543)

# generate some data
x=matrix(rnorm(40*1000),nrow=40)
y=sort(rep(1:5,8))
x[y==2 | y==3, 1:50]=x[y==2|y==3, 1:50]+1
x[y==3,51:100]=x[y==3,51:100]+1
x[y==4|y==5,1:50]=x[y==4|y==5,1:50]-1
x[y==5, 51:100]=x[y==5,51:100]+1

#train the classifier

train.obj<- marginTree(x,y)

# do cross-validation
cv.obj<- marginTree.cv(x,y,train.obj)

#make cv plot
marginTree.plotcv(cv.obj)
```

marginTree.predict *Prediction from the margin tree classifier*

Description

A function to predict from the margin tree classifier

Usage

```
marginTree.predict(train.obj,x, threshold=1)
```

Arguments

<code>train.obj</code>	Output from call to <code>marginTree</code>
<code>x</code>	Data of feature values, n samples by p features
<code>threshold</code>	Threshold values for feature selection. Between 1 and 0.)

Details

`marginTree.predict` yields class predictions from a margin tree classifier.

Value

Vector of predicted values

Author(s)

Robert Tibshirani and Trevor Hastie

References

Rob Tibshirani and Trevor Hastie. Tech report. Feb. 2006. Margin trees for high-dimensional classification Available at <http://www-stat.stanford.edu/~tibs/research.html>

Examples

```
#generate some data with 5 classes and 100 features

set.seed(543)
x=matrix(rnorm(40*1000),nrow=40)
y=sort(rep(1:5,8))
x[y==2 | y==3, 1:50]=x[y==2|y==3, 1:50]+1
x[y==3,51:100]=x[y==3,51:100]+1
x[y==4|y==5,1:50]=x[y==4|y==5,1:50]-1
x[y==5, 51:100]=x[y==5,51:100]+1

#generate some test data
xx=x+5*matrix(rnorm(40*1000),nrow=40)
yy=y

#train the classifier

train.obj<- marginTree(x,y)

# get predictions for test data
marginTree.predict(train.obj,xx)
```

Index

marginTree, 2
marginTree.confusion, 3
marginTree.cv, 4
marginTree.getnonzero, 6
marginTree.plclust, 7
marginTree.plotcv, 8
marginTree.predict, 9