Package 'SubLasso'

August 11, 2014

Type Package

Title Lasso-based feature selection for a gene expression profile/matrix, with a user-defined preselected feature subset.

Version 1.2

Depends R(>= 2.10), glmnet, psych, gplots

Date 2014-08-11

Author
Youxi Luo, Qinghan Meng, Ruiquan Ge, Guoqin Mai, Jikui Liu, Fengfeng Zhou(#corresponding)

Maintainer FengFeng Zhou `<fengfengzhou@gmail.com>`

Description Given a gene expression profile/matrix, this R package SubLasso develops a feature selection and classification algorithm by adding more features into a user-defined seed feature subset. This is per the frequent requests from biomedical researchers whether some well-known disease-biomarkers may work together with some additional features, to form an accurate classifier for a given disease. For example, whether there is a good classifier for breast cancer, by using the patterns of two biomarkers, i.e. BRCA1 and BRCA2, and a few others. The mathematical model is to fix some user-defined features in the finally chosen feature subset, with the optimized classification accuracy.

License GPL-2

URL `http://healthinformaticslab.org/ffzhou/`

R topics documented:

`SubLasso-package` *SubLasso package*

Description

SubLasso package

Details

Package:	SubLasso
Type:	Package
Version:	1.2
Date:	2014-08-11
License:	GPL-2

This package implemented a feature selection procedure with the optimized classification accuracy, and the chosen feature subset consists of the user-defined seed features. For the convenience of the users, the k-fold cross validation performance will also be calculated. The other user-friendly assets of this package include the minimum requirement for the parameter tuning, by automatic optimization.

Author(s)

Author: Youxi Luo, Qinghan Meng, Ruiquan Ge, Guoqin Mai, Jikui Liu, Fengfeng Zhou(corresponding)
Maintainer: Qinghan Meng <qinghan.meng@gmail.com>

`Colon example dataset` *Gene expression data from Alon et al. (1999)*

Description

Expression data of 2000 genes for 62 samples, which was generated from the microarray experiments of Colon tissue samples of Alon et al. (1999).

Usage

```
data(Colon)
```

Details

This data set contains 62 samples with 2000 genes: 40 tumor tissues, coded 1 and 22 normal tissues, coded 0.

Value

A list with the following elements:

`X` a (2000 x 62) matrix giving the expression levels of 2000 genes for the 62 Colon tissue samples. Each row corresponds to a gene, and each column to a patient/sample.

`y` a numeric vector of length 62 giving the type of tissue sample (tumor or normal).

Source

The data are described in Alon et al.(1999) and can be freely downloaded from `http://microarray.princeton.edu/oncology/affydata/index.html`.

References

Alon, U. and Barkai, N. and Notterman, D.A. and Gish, K. and Ybarra, S. and Mack, D. and Levine, A.J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA,**96**(12), 6745–6750.

Examples

```
library(SubLasso)

# load data set
data(Colon)

# how many samples and how many genes ?
dim(Colon$X)

# how many samples of class 0 and 1 respectively ?
sum(Colon$y==0)
sum(Colon$y==1)
```

`Golub_Merge example dataset` *Combined Training and Test Sets from the Golub Paper*

Description

The data are from Golub et al. These are the combined training samples and test samples. There are 47 patients with acute lymphoblastic leukemia (ALL) and 25 patients with acute myeloid leukemia (AML).

Usage

```
data(Golub_Merge)
```

Value

X matrix giving the expression levels.

y giving the type of tissue sample.

References

Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, Science, 531-537, 1999, T. R. Golub and D. K. Slonim and P. Tamayo and C. Huard and M. Gaasenbeek and J. P. Mesirov and H. Coller and M.L. Loh and J. R. Downing and M. A. Caligiuri and C. D. Bloomfield and E. S. Lander

Examples

```
# load SubLasso library
library(SubLasso)

# load data set
data(Golub_Merge)
```

predict.SubLasso *Predict method for SubLasso fits.*

Description

Similar to other predict methods, this functions predicts fitted values, logits, coefficients and more from a fitted "SubLasso" object.

Usage

```
## S3 method for class 'SubLasso'
predict(object, xpred, type, s, ...)
```

Arguments

object Fitted "SubLasso" model object.

xpred Matrix of new values for x at which predictions are to be made. Must have the same number of rows with x

type type=c("link","response","class"). Default is "class";Type of prediction required. Type "link" gives the linear predictors; Type "response" gives the fitted probabilities; Type "class" produces the class label corresponding to the maximum probability.

s Value(s) of the penalty parameter lambda at which predictions are required. Default is obtained by CV method.

... not used currently.

Value

`Predy` it depends on type.

SubLasso *Gene selection using Lasso for gene expression profile matrix with user-defined genes fixed in model.*

Description

This package implemented a feature selection procedure with the optimized classification accuracy, and the chosen feature subset consists of the user-defined seed features. For the convenience of the users, the k-fold cross validation performance will also be calculated. The other user-friendly assets of this package include the minimum requirement for the parameter tuning, by automatic optimization.

Usage

```
SubLasso(X, y, subset, nfold)
```

Arguments

`X` The gene expression matrix, row is sample, column is for the expression levels of genes (probe sets).

`y` The category vector, 1 (positive, illness) or 0 (negative, normal).

`subset` The vector of gene names (probe sets) must belong to the genes in the model. Default is null set, meaning no pre-fixed genes.

`nfold` The number of cross-validation. Default is 5.

Value

`selname` The vector of features selected by the model.

`valid` The performance measurements sensitivity (Sn), specificity (Sp), Accuracy (Acc), and Matthews correlation coefficient (Mcc).

`description` A descriptive summarization of selected features.

`correlation` The correlations among the selected features.

`w` The coefficient (weight) of each feature in the model.

`lambda` The actual penalty parameter values used or optimized.

`cv.predp` The predict probability in cross-validation.

`cv.predy` The predict class label in cross-validation.

`fit` The fitted model when doing feature selection.

`show.boxplot` A boxplot produced by the linear prediction function of logisitic model. X-axis is the sample group and Y-axis is the prediction function score of each subject. The up and down real line of box is the 0.75 and 0.25 quantile respectively. The bold real line in the middle of box is the median.

show.heatmap The hierarchical clustering of samples based on expression patterns of the selected features. Each row corresponds to a feature and each column corresponds to a sample. The status of illness or normal for each subject is shown with the above bar. Gene expression value is indicated by different color in the medial matrix.

Author(s)

Youxi Luo, Qinghan Meng, Ruiquan Ge, Guoqin Mai, Jikui Liu, Fengfeng Zhou(#corresponding)

References

[1] Friedman, J., Hastie, T. and Tibshirani, R. (2008) Regularization Paths for Generalized Linear Modelsvia Coordinate Descent, http://www.stanford.edu/~hastie/Papers/glmnet.pdf Journal of Statistical Software, Vol. 33(1), 1-22 Feb 2010. http://www.jstatsoft.org/v33/i01/
[2] Simon, N., Friedman, J., Hastie, T., Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5) 1-13 http://www.jstatsoft.org/v39/i05/

See Also

glmnet

Examples

```
##### Example 1
library(SubLasso)
data(Golub_Merge)
X <- Golub_Merge$X
y <- Golub_Merge$y
f1=SubLasso(X,y,nfold=10)
## The linear discriminating function
cat(f1$intercept, " + ", paste(f1$w,f1$selname,collapse=" + ",sep="*"))


## predict.SubLasso(f1,X[1:10,]) ##error predicted x
predy=predict.SubLasso(f1,X)
predy=predict.SubLasso(f1,X,type="class")
predy=predict.SubLasso(f1,X,type="link")
predy=predict.SubLasso(f1,X,type="response")
predy=predict.SubLasso(f1,X,type="response",s=0.05)
subset=f1$selname
f2=SubLasso(X,y,subset,nfold=10)
cat(f2$intercept, " + ", paste(f2$w,f2$selname,collapse=" + ",sep="*"))
f2$show.boxplot()
f2$show.heatmap()

subset=row.names(X)[1:10]
f3=SubLasso(X,y,subset,nfold=10)
predy=predict.SubLasso(f3,X)
predy=predict.SubLasso(f3,X,type="class")
```

```
predy=predict.SubLasso(f3,X,type="link")
predy=predict.SubLasso(f3,X,type="response")
predy=predict.SubLasso(f3,X,type="response",s=0.05)
cat(f3$intercept, " + ", paste(f3$w,f3$selname,collapse=" + ",sep="*"))
f3$show.heatmap()

###Example 2
library(SubLasso)
data(Colon)
X<-Colon$X
y<-ifelse(Colon$y == 1,1,0)
f1=SubLasso(X,y,nfold=10)
subset=f1$selname
## The linear discriminating function
cat(f1$intercept, " + ", paste(f1$w,f1$selname,collapse=" + ",sep="*"))

f2=SubLasso(X,y,subset,nfold=10)
subset=row.names(X)[30:40]
cat(f2$intercept, " + ", paste(f2$w,f1$selname,collapse=" + ",sep="*"))

f3=SubLasso(X,y,subset,nfold=10)
cat(f3$intercept, " + ", paste(f3$w,f1$selname,collapse=" + ",sep="*"))
f3$show.boxplot()
f3$show.heatmap()
```

Index