Package 'sodavis'

November 16, 2015

Type Package

Title SODA: Main and Interaction Effects Selection for Discriminant Analysis and Logistic Regression

Version 0.1

Depends R (>= 3.0.0), nnet, MASS

Date 2015-11-07

Author Yang Li, Jun S. Liu

Maintainer Yang Li <yli01@fas.harvard.edu>

Description Variable and interaction selection are essential to classification in high-dimensional setting. In this package, we provide the implementation of SODA procedure, which is a forward-backward algorithm that selects both main and interaction effects under quadratic discriminant analysis and logistic regression model.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-11-16 08:20:20

R topics documented:

mich_lung — *Gene expression data for Michigan lung cancer study in Beer et al. (2002)*

Description

Gene expression data of 5217 genes for n = 86 subjects, with 62 subjects in "good outcomes" (class 1) and 24 subjects in "poor outcomes" (class 2), from the microarray study of Beer et al. (2002).

Usage

```
data(mich_lung)
```

Format

Response variable vector and design matrix on 86 observations for expression of 5217 genes.

References

Beer et al. (1999) Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature medicine*, 286(8): 816-824.

soda — *SODA algorithm for variable and interaction selection*

Description

SODA is a forward-backward variable and interaction selection algorithm under logistic regression model with second-order terms. In the forward stage, a stepwise procedure is conducted to screen for important predictors with both main and interaction effects, and in the backward stage SODA remove insignificant terms so as to optimize the extended BIC (EBIC) criterion. SODA is applicable for variable selection for logistic regression, linear/quadratic discriminant analysis and other discriminant analysis with generative model being in exponential family.

Usage

```
soda(xx, yy, norm = FALSE, debug = FALSE, gam = 0, minF = 3)
```

Arguments

xx	The design matrix, of dimensions n * p, without an intercept. Each row is an observation vector.
yy	The response vector of dimension n * 1.
norm	Logical flag for xx variable quantile normalization to standard normal, prior to performing SODA algorithm. Default is norm=FALSE. Quantile-normalization is suggested if the data contains obvious outliers.

debug	Logical flag for printing debug information.
gam	Tuning paramter gamma in extended BIC criterion. EBIC for selected set S: EBIC = -2 * log-likelihood + |S| * log(n) + 2 * |S| * gamma * log(p)
minF	Minimum number of steps in forward interaction screening. Default is minF=3.

Value

EBIC	Trace of extended Bayesian information criterion (EBIC) score.
Type	Trace of step type ("Forward (Main)", "Forward (Int)", "Backward").
Var	Trace of selected variables.
Term	Trace of selected main and interaction terms.
final_EBIC	Final selected term set EBIC score.
final_Var	Final selected variables.
final_Term	Final selected main and interaction terms.

Author(s)

Yang Li, Jun S. Liu

References

Li Y, Liu JS. (2015). Robust variable and interaction selection for high-dimensional classification via logistic regression. *Technical Report*.

Examples

```
# simulation study with 1 main effect and 2 interactions (uncomment the code to run)
#N = 250;
#p = 1000;
#r = 0.5;
#s = 1;
#H = abs(outer(1:p, 1:p, "-"))
#S = s * r^H;
#S[cbind(1:p, 1:p)] = S[cbind(1:p, 1:p)] * s

#xx = as.matrix(data.frame(mvrnorm(N, rep(0,p), S)));
#zz = 1 + xx[,1] - xx[,10]^2 + xx[,10]*xx[,20];
#yy = as.numeric(runif(N) < exp(zz) / (1+exp(zz)))

#res_SODA = soda(xx, yy, gam=0.5);
#cv_SODA  = soda_trace_CV(xx, yy, res_SODA)
#cv_SODA

# Michigan lung cancer dataset (uncomment the code to run)
#data(mich_lung);
#res_SODA = soda(mich_lung_xx, mich_lung_yy, gam=0.5);
#cv_SODA  = soda_trace_CV(mich_lung_xx, mich_lung_yy, res_SODA)
#cv_SODA
```

soda_trace_CV *Calculate a trace of cross-validation error rate for SODA forward-backward procedure*

Description

This function takes a SODA result variable as input, and calculates the cross-validation error for each step of the SODA procedure.

Usage

```
soda_trace_CV(xx, yy, res_SODA)
```

Arguments

xx	The design matrix, of dimensions n * p, without an intercept. Each row is an observation vector.
yy	The response vector of dimension n * 1.
res_SODA	SODA result varaible. See example below.

Author(s)

Yang Li, Jun S. Liu

Examples

```
# Michigan lung cancer dataset (uncomment the code to run)
#data(mich_lung);
#res_SODA = soda(mich_lung_xx, mich_lung_yy, gam=0.5);
#cv_SODA  = soda_trace_CV(mich_lung_xx, mich_lung_yy, res_SODA)
#cv_SODA
```

Index