

Package ‘RAMP’

January 27, 2015

Type Package

Title Regularized Generalized Linear Models with Interaction Effects

Version 1.0

Date 2014-12-24

Author Yang Feng, Ning Hao and Hao Helen Zhang

Maintainer Yang Feng <yangfeng@stat.columbia.edu>

Description This package provides an efficient procedure for fitting the entire solution path for high-dimensional regularized generalized linear models with interactions effects under the strong heredity constraint.

License GPL-2

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-12-31 01:11:46

R topics documented:

predict.RAMP	1
print.RAMP	3
RAMP	4

Index	6
--------------	----------

predict.RAMP	<i>Model prediction based on a fitted RAMP object.</i>
--------------	--

Description

Similar to the usual predict methods, this function returns predictions from a fitted "RAMP" object.

Usage

```
## S3 method for class 'RAMP'
predict(object, newdata = NULL, type = c("link", "response", "class"),
        allpath = FALSE,...)
```

Arguments

object	Fitted "RAMP" model object.
newdata	Matrix of new values for x at which predictions are to be made, without the intercept term.
type	Type of prediction required. Type "response" gives the fitted values for "gaussian", fitted probabilities for "binomial", fitted mean for "poisson", and the fitted relative risk for "cox". Type "link" returns the linear predictors for "binomial", "poisson" and "cox" models; for "gaussian" models it is equivalent to type "response". Type "class" applies only to "binomial" models, and produces the class label corresponding to the maximum probability (0-1 labels).
allpath	allpath = T will output all the predictions on the solution path. allpath = FALSE will only output the one the criterion selected in the "RAMP" object.
...	Not used. Other arguments to predict.

Value

The object returned depends on type.

Author(s)

Yang Feng, Ning Hao and Hao Helen Zhang.

See Also

[RAMP](#)

Examples

```
set.seed(0)
n = 500
p = 100 #Can be changed to a much larger number say 100000
x = matrix(rnorm(2*n*p),2*n,p)
eta = 1 * x[,1] + 2 * x[,3] + 3*x[,6] + 4*x[,1]*x[,3] + 5*x[,1]*x[,6]
y = eta + rnorm(2*n)
xtr = x[1:n, ]
ytr = y[1:n]

xte = x[(n+1):(2*n), ]
yte = y[(n+1):(2*n)]

fit1 = RAMP(xtr, ytr)
ypred = predict(fit1,xte)
```

```

####binary prediction
y = rbinom(2*n, 1, 1/(1+exp(-eta)))
xtr = x[1:n, ]
ytr = y[1:n]

xte = x[(n+1):(2*n), ]
yte = y[(n+1):(2*n)]

fit2 = RAMP(xtr, ytr,family='binomial')
ypred = predict(fit2,xte,type='class')
mean(ypred!=yte) ##classification error on the test set

ypred = predict(fit2,type='class')
mean(ypred!=ytr) ##classification error on the training set

```

```
print.RAMP
```

Result summary of a fitted RAMP object.

Description

Similar to the usual print methods, this function summarize results from a fitted "RAMP" object.

Usage

```
## S3 method for class 'RAMP'
print(x,digits = max(3, getOption("digits") - 3),...)
```

Arguments

x	Fitted "RAMP" model object.
digits	The number of significant digits for the coefficient estimates.
...	Not used. Other arguments to predict.

Value

No value is returned.

Author(s)

Yang Feng, Ning Hao and Hao Helen Zhang.

See Also

[RAMP](#)

RAMP *Regularization Algorithm under Marginality Principle (RAMP) for high dimensional generalized quadratic regression.*

Description

RAMP

Usage

```
RAMP(X, y, family = "gaussian", penalty = "LASSO", gamma, inter = TRUE,
      eps = 1e-15, tune = "EBIC", lam.list, lambda.min.ratio, max.iter = 100,
      max.num, n.lambda = 100, ebic.gamma = 1, refit = TRUE, trace = FALSE)
```

Arguments

X	input matrix, of dimension nobs x nvars; each row is an observation vector.
y	response variable, of dimension nobs x 1. non-negative counts for family="poisson", binary for family="binomial".
family	response type. Default is "gaussian".
penalty	Currently, only LASSO is implemented.
gamma	concavity parameter. Experimental feature.
inter	whether to select interaction effects. Default is TRUE.
eps	the precision used to test the convergence. Default is 1e-15.
tune	tuning parameter selection method. "AIC", "BIC" and "EBIC" are available options. Default is EBIC.
lam.list	a user supplied λ sequence. typical usage is to have the program compute its own lambda sequence based on lambda.min.ratio and n.lambda. supplying a value of λ overrides this.
lambda.min.ratio	optional input. smallest value for lambda, as a fraction of max.lam, the (data derived) entry value. the default depends on the sample size n relative to the number of variables p. if $n > p$, the default is 0.0001. otherwise, the default is 0.01.
max.iter	maximum number of iteration in the computation. Default is 100.
max.num	optional input. maximum number of nonzero coefficients.
n.lambda	the number of lambda values. Default is 100.
ebic.gamma	the gamma parameter value in the EBIC criteria. Default is 1.
refit	whether to perform a MLE refit on the selected model. Default is TRUE.
trace	whether to trace the fitting process. Default is FALSE.

Value

a0 intercept vector of length(lambda).
beta nvar x length(lambda) matrix of coefficients.

Author(s)

Yang Feng, Ning Hao and Hao Helen Zhang.

See Also

[predict.RAMP](#)

Examples

```
set.seed(0)
n = 500
p = 100 #Can be changed to a much larger number say 100000
x = matrix(rnorm(n*p),n,p)
eta = 1 * x[,1] + 2 * x[,3] + 3*x[,6] + 4*x[,1]*x[,3] + 5*x[,1]*x[,6]
y = eta + rnorm(n)

fit1 = RAMP(x, y)
fit1    ###examine the results

y = rbinom(n, 1, 1/(1+exp(-eta)))
fit2 = RAMP(x, y, family='binomial')    ###for binary response
```

Index

*Topic **function**

predict.RAMP, 1

print.RAMP, 3

RAMP, 4

*Topic **prediction**

predict.RAMP, 1

predict.RAMP, 1, 5

print.RAMP, 3

RAMP, 2, 3, 4