**Package ‘hierNet’**

February 20, 2015

**Title**  A Lasso for Hierarchical Interactions

**Version**  1.6

**Author**  Jacob Bien and Rob Tibshirani

**Description**  Fits sparse interaction models for continuous and binary responses subject to the strong (or weak) hierarchy restriction that an interaction between two variables only be included if both (or at least one of) the variables is included as a main effect. For more details, see Bien, J., Taylor, J., Tibshirani, R., (2013) “A Lasso for Hierarchical Interactions.” Annals of Statistics. 41(3). 1111-1141.

**Maintainer**  Jacob Bien <jbien@cornell.edu>

**License**  GPL-2

**NeedsCompilation**  yes

**Repository**  CRAN

**Date/Publication**  2014-04-29 08:50:35

**R topics documented:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>hierNet</td>
<td>2</td>
</tr>
<tr>
<td>hierNet.cv</td>
<td>4</td>
</tr>
<tr>
<td>hierNet.logistic</td>
<td>5</td>
</tr>
<tr>
<td>hierNet.logistic.path</td>
<td>8</td>
</tr>
<tr>
<td>hierNet.path</td>
<td>10</td>
</tr>
<tr>
<td>hierNet.varimp</td>
<td>12</td>
</tr>
<tr>
<td>predict.hierNet</td>
<td>13</td>
</tr>
<tr>
<td>predict.hierNet.logistic</td>
<td>14</td>
</tr>
<tr>
<td>predict.hierNet.path</td>
<td>15</td>
</tr>
</tbody>
</table>

**Index**  17
Description

One of the main functions in the hierNet package. Builds a regression model with hierarchically constrained pairwise interactions. Required inputs are an x matrix of features (the columns are the features) and a y vector of values. Reasonably fast for moderate sized problems (100-200 variables). We are currently working on an alternate algorithm for large scale problems.

Usage

```r
hiernet(x, y, lamL=1e-8, strongL=FALSE, diagonalL=TRUE, aaL=NULL, zzL=NULL,
        centerL=TRUE, stand.mainL=TRUE, stand.intL=FALSE,
        rhoL=nrow(x), niterL=100, sym.epsL=1e-3,
        stepL=1, maxiterL=2000, backtrackL=0.2, tolL=1e-5, traceL=0)
```

Arguments

- **x**: A matrix of predictors, where the rows are the samples and the columns are the predictors
- **y**: A vector of observations, where length(y) equals nrow(x)
- **lam**: Regularization parameter (>0). L1 penalty param is lam * (1-delta).
- **delta**: Elastic Net parameter. Squared L2 penalty param is lam * delta. Not a tuning parameter: Think of as fixed and small. Default 1e-8.
- **strong**: Flag specifying strong hierarchy (TRUE) or weak hierarchy (FALSE). Default FALSE.
- **diagonal**: Flag specifying whether to include "pure" quadratic terms, th_jjX_j^2, in the model. Default TRUE.
- **aa**: An *optional* argument, a list with results from a previous call
- **zz**: An *optional* argument, a matrix whose columns are products of features, computed by the function compute.interactions.c
- **center**: Should features be centered? Default TRUE; FALSE should rarely be used. This option is available for special uses only
- **stand.main**: Should main effects be standardized? Default TRUE.
- **stand.int**: Should interactions be standardized? Default FALSE.
- **rho**: ADMM parameter: tuning parameter (>0) for ADMM. If there are convergence problems, try decreasing rho. Default n.
- **niter**: ADMM parameter: number of iterations
- **sym.eps**: ADMM parameter: threshold for symmetrizing with strong=TRUE
- **step**: Stepsize for generalized gradient descent
- **maxiter**: Maximum number of iterations for generalized gradient descent
hierNet

backtrack  Backtrack parameter for generalized gradient descent

Error tolerance parameter for generalized gradient descent

Output option; trace=1 gives verbose output

Value

bp  p-vector of estimated "positive part" main effect (p=# features)

bn  p-vector of estimated "negative part" main effect; overall main effect estimated coefficients are bp-bn

th  Matrix of estimated interaction coefficients, of dimension p by p. Note: when output from hierNet is printed, th is symmetrized (set to (th+t(th))/2) for simplicity.

obj  Value of objective function at minimum.

lam  Value of lambda used

type  Type of model fit- "gaussian" or "logistic" (binomial)

mx  p-vector of column means of x

sx  p-vector of column standard deviations of x

my  mean of y

mzz  column means of feature product matrix

szz  column standard deviations of feature product matrix

call  The call to hierNet

Author(s)

Jacob Bien and Robert Tibshirani

References


See Also

predict.hierNet, hierNet.cv, hierNet.path

Examples

set.seed(12)
# fit a single hierNet model
x=matrix(rnorm(100*10),ncol=10)
x=scale(x,TRUE,TRUE)
y=x[,1]+2*x[,2]+ x[,1]*x[,2]+3*rnorm(100)
fit=hierNet(x,y,lambda=50)
print(fit)

# try strong (rather than weak) hierarchy
fit=hiernet(x,y,lam=50, strong=TRUE)
print(fit)

# a typical analysis including cross-validation
set.seed(12)
x=matrix(rnorm(100*10),ncol=10)
x=scale(x,TRUE,TRUE)
y=x[,1]+2*x[,2]+ x[,1]*x[,2]+3*ranorm(100)
fit=hiernet.path(x,y)
fitcv=hiernet.cv(fit,x,y)
print(fitcv)

lamhat=fitcv$lamhat.ise
fit2=hiernet(x,y,lam=lamhat)
yhat=predict.hierNet(fit2,x)

---

**hierNet.cv**  
*Cross-validation function for hierNet*

**Description**

Uses cross-validation to estimate the regularization parameter for hierNet

**Usage**

`hierNet.cv(fit, x, y, nfolds=10, folds=NULL, trace=0)`

**Arguments**

- **fit**: Object returned from call to hierNet.path or hierNet.logistic.path. All parameter settings will be taken from this object.
- **x**: A matrix of predictors, where the rows are the samples and the columns are the predictors
- **y**: A vector of observations, where length(y) equals nrow(x)
- **nfolds**: Number of cross-validation folds
- **folds** (Optional) user-supplied cross-validation folds. If provided, nfolds is ignored.
- **trace**: Verbose output? 0=no, 1=yes
- **...**: Additional arguments to be passed to hierNet.path or hierNet.logistic.path

**Value**

- **lamlist**: Vector of lambda values tried
- **cv.err**: Estimate of cross-validation error
- **cv.se**: Estimated standard error of cross-validation estimate
- **lamhat**: lambda value minimizing cv.err
hierNet.logistic

- **lamhat.1se**: largest lambda value with cv.err less than or equal to \( \min(\text{cv.err}) + \text{SE} \)
- **folds**: Indices of folds used in cross-validation
- **yhat**: \( n \) by \( n_{\lambda} \) matrix of predicted values. Here, \( i \)th prediction is based on training on all folds that do not include the \( i \)th data point.
- **nonzero**: Vector giving number of non-zero coefficients for each lambda value
- **call**: The call to hierNet.cv

**Author(s)**

Jacob Bien and Robert Tibshirani

**References**


**See Also**

hierNet, hierNet.path, hierNet.logistic, hierNet.logistic.path

**Examples**

```r
set.seed(12)
x = matrix(rnorm(100*10), ncol=10)
x = scale(x, TRUE, TRUE)
y = x[,1] + 2*x[,2] + x[,1]*x[,2] + 3*rnorm(100)
fit = hierNet.path(x, y)
fitcv = hierNet.cv(fit, x, y)
print(fitcv)
plot(fitcv)

x = matrix(rnorm(100*10), ncol=10)
x = scale(x, TRUE, TRUE)
y = x[,1] + 2*x[,2] + x[,1]*x[,2] + 3*rnorm(100)
y = 1*(y > 0)
fit = hierNet.logistic.path(x, y)
fitcv = hierNet.cv(fit, x, y)
print(fitcv)
plot(fitcv)
```

---

**hierNet.logistic**: A logistic regression Lasso for interactions
**Description**

One of the main functions in the hierNet package. Builds a logistic regression model with hierarchically constrained pairwise interactions. Required inputs are an x matrix of features (the columns are the features) and a y vector of values. Reasonably fast for moderate sized problems (100-200 variables). We are currently working on a alternate algorithm for large scale problems.

**Usage**

```r
hierNet.logistic(x, y, lam, delta=1e-8, diagonal=TRUE, strong=FALSE, aa=NULL, zz=NULL, center=TRUE, stand.main=TRUE, stand.int=FALSE, rho=nrow(x), niter=100, sym.eps=1e-3,# ADMM params
step=1, maxiter=2000, backtrack=0.2, tol=1e-5, trace=1)
```

**Arguments**

- `x`: A matrix of predictors, where the rows are the samples and the columns are the predictors.
- `y`: A vector of observations, with values 0 or 1, where length(y) equals nrow(x).
- `lam`: Regularization parameter (>0). L1 penalty param is `lam * (1-delta)`.  
- `diagonal`: Flag specifying whether to include "pure" quadratic terms, th_jjX_j^2, in the model. Default TRUE.
- `strong`: Flag specifying strong hierarchy (TRUE) or weak hierarchy (FALSE). Default FALSE.
- `aa`: An *optional* argument, a list with results from a previous call.
- `zz`: An *optional* argument, a matrix whose columns are products of features, computed by the function `compute.interactions.c`.
- `center`: Should features be centered? Default TRUE; FALSE should rarely be used. This option is available for special uses only.
- `stand.main`: Should main effects be standardized? Default TRUE.
- `stand.int`: Should interactions be standardized? Default FALSE.
- `rho`: ADMM parameter: tuning parameter (>0) for ADMM. If there are convergence problems, try decreasing `rho`. Default n.
- `niter`: ADMM parameter: number of iterations.
- `sym.eps`: ADMM parameter Thresholding for symmetrizing with strong=TRUE
- `step`: Stepsize for generalized gradient descent.
- `maxiter`: Maximum number of iterations for generalized gradient descent.
- `backtrack`: Backtrack parameter for generalized gradient descent.
- `tol`: Error tolerance parameter for generalized gradient descent.
- `trace`: Output option; trace=1 gives verbose output.
Value

- **b0**: Intercept
- **bp**: p-vector of estimated "positive part" main effect (p=#features)
- **bn**: p-vector of estimated "negative part" main effect; overall main effect estimated coefficients are bp-bn
- **th**: Matrix of estimated interaction coefficients, of dimension p by p
- **obj**: Value of objective function at minimum.
- **lam**: Value of lambda used
- **type**: Type of model fit- "gaussian" or "logistic" (binomial)
- **mx**: p-vector of column means of x
- **my**: Mean of y
- **sx**: p-vector of column standard deviations of x
- **mzz**: column means of feature product matrix
- **call**: The call to hierNet

Author(s)

Jacob Bien and Robert Tibshirani

References


See Also

predict.hierNet.logistic, linkhierNet.logistic.path

Examples

```r
set.seed(12)
x = matrix(rnorm(100*10), ncol=10)
x = scale(x, TRUE, TRUE)
y = x[,1]*2*x[,2] + x[,1]*x[,2] + 3*ranorm(100)
y = 1*(y>0)
fit = hierNet.logistic(x, y, lam=5)
print(fit)
```
**hierNet.logistic.path**  
*Fit a path of logistic hierNet models- lasso models with interactions*

**Description**
One of the main functions in the hierNet package. Fits a logistic path of hierNet models over different values of the regularization parameter. Calls hierNet.logistic, which builds a regression model with hierarchically constrained pairwise interactions. Required inputs are an x matrix of features (the columns are the features) and a y vector of values. Reasonably fast for moderate sized problems (100-200 variables). We are currently working on an alternate algorithm for large scale problems.

**Usage**
```r
hierNet.logistic.path(x, y, lamlist = NULL, delta=1e-8, minlam = NULL, maxlam = NULL, flmin=.01, nlam = 20, diagonal = TRUE, strong = FALSE, aa = NULL, zz = NULL, stand.main = TRUE, stand.int = FALSE, rho = nrow(x), niter = 100, sym.eps = 0.001, step = 1, maxiter = 2000, backtrack = 0.2, tol = 1e-05, trace = 0)
```

**Arguments**
- **x**: A matrix of predictors, where the rows are the samples and the columns are the predictors.
- **y**: A vector of observations equal to 0 or 1, where length(y) equals nrow(x).
- **lamlist**: Optional vector of values of lambda (the regularization parameter). L1 penalty param is lambda * (1-delta).
- **delta**: Elastic Net parameter. Squared L2 penalty param is lambda * delta. Not a tuning parameter: Think of as fixed and small. Default 1e-8.
- **minlam**: Optional minimum value for lambda.
- **maxlam**: Optional maximum value for lambda.
- **flmin**: Fraction of maxlam; minlam= flmin*maxlam. If computation is slow, try increasing flmin to focus on the sparser part of the path.
- **nlam**: Number of values of lambda to be tried.
- **diagonal**: Flag specifying whether to include "pure" quadratic terms, th_jjX_j^2, in the model. Default TRUE.
- **stand.main**: Should main effects be standardized? Default TRUE.
- **stand.int**: Should interactions be standardized? Default FALSE.
- **strong**: Flag specifying strong hierarchy (TRUE) or weak hierarchy (FALSE). Default FALSE.
- **aa**: An *optional* argument, a list with results from a previous call.
zz  An *optional* argument, a matrix whose columns are products of features, computed by the function compute.interactions.c

rho  ADMM parameter: tuning parameter (>0) for ADMM. If there are convergence problems, try decreasing rho. Default n.

niter  ADMM parameter: number of iterations

sym. eps  ADMM parameter Thresholding for symmetrizing with strong=TRUE

step  Stepsize for generalized gradient descent

maxiter  Maximum number of iterations for generalized gradient descent

backtrack  Backtrack parameter for generalized gradient descent

tol  Error tolerance parameter for generalized gradient descent

trace  Output option; trace=1 gives verbose output

Value

bp  p by nlam matrix of estimated "positive part" main effects (p=#features)

bn  p by nlam matrix of estimated "negative part" main effects

th  p by p by nlam array of estimated interaction coefficients

obj  nlam values of objective function, one per lambda value

lamlist  Vector of values of lambda used

mx  p-vector of column means of x

sx  p-vector of column standard deviations of x

my  mean of y

mzz  column means of feature product matrix

szz  column standard deviations of feature product matrix

Author(s)

Jacob Bien and Robert Tibshirani

References


See Also

hierNet,predict.hierNet, hierNet.cv

Examples

set.seed(12)
x=matrix(rnorm(100*10),ncol=10)
x=scale(x,TRUE,TRUE)
y=x[,1]+2*x[,2]+ x[,1]*x[,2]+3*rnorm(100)
y=1*(y>0)
fit=hierNet.logistic.path(x,y)
print(fit)
hierNet.path

**Fit a path of hierNet models: lasso models with interactions**

**Description**

One of the main functions in the hierNet package. Fits a path of hierNet models over different values of the regularization parameter. Calls hierNet, which builds a regression model with hierarchically constrained pairwise interactions. Required inputs are an x matrix of features (the columns are the features) and a y vector of values. Reasonably fast for moderate sized problems (100-200 variables). We are currently working on an alternate algorithm for large scale problems.

**Usage**

```r
hierNet.path(x, y,
            lamlist = NULL, delta=1e-8, minlam = NULL, maxlam = NULL, nlam=20, flmin=.01,
            diagonal = TRUE, strong = FALSE, aa = NULL, zz = NULL,
            stand.main = TRUE, stand.int = FALSE,
            rho = nrow(x), niter = 100, sym.eps = 0.001,
            step = 1, maxiter = 2000, backtrack = 0.2, tol = 1e-05, trace = 0)
```

**Arguments**

- **x**: A matrix of predictors, where the rows are the samples and the columns are the predictors
- **y**: A vector of observations, where length(y) equals nrow(x)
- **lamlist**: Optional vector of values of lambda (the regularization parameter). L1 penalty param is lambda * (1-delta).
- **delta**: Elastic Net parameter. Squared L2 penalty param is lambda * delta. Not a tuning parameter: Think of as fixed and small. Default 1e-8.
- **minlam**: Optional minimum value for lambda
- **maxlam**: Optional maximum value for lambda
- **nlam**: Number of values of lambda to be tried
- **flmin**: Fraction of maxlam; minlam= flmin*maxlam. If computation is slow, try increasing flmin to focus on the sparser part of the path
- **diagonal**: Flag specifying whether to include “pure” quadratic terms, th_jjX_j^2, in the model. Default TRUE.
- **strong**: Flag specifying strong hierarchy (true) or weak hierarchy (false). Default false
- **aa**: An *optional* argument, a list with results from a previous call
- **zz**: An *optional* argument, a matrix whose columns are products of features, computed by the function compute.interactions.c
- **stand.main**: Should main effects be standardized? Default TRUE
- **stand.int**: Should interactions be standardized? Default FALSE
rhuNet.path

rho ADMM parameter: tuning parameter (>0) for ADMM. If there are convergence problems, try decreasing rho. Default n.
niter ADMM parameter: number of iterations
sym. eps ADMM parameter Thresholding for symmetrizing with strong=TRUE
step Stepsize for generalized gradient descent
maxiter Maximum number of iterations for generalized gradient descent
backtrack Backtrack parameter for generalized gradient descent
tol Error tolerance parameter for generalized gradient descent
trace Output option; trace=1 gives verbose output

Value
bp p by nlam matrix of estimated "positive part" main effects (p=#variables)
bn p by nlam matrix of estimated "negative part" main effects
th p by p by nlam array of estimated interaction coefficients
obj nlam values of objective function, one per lambda value
lamlist Vector of values of lambda used
mx p-vector of column means of x
sx p-vector of column standard deviations of x
my mean of y
mzz column means of feature product matrix
szz column standard deviations of feature product matrix

Author(s)
Jacob Bien and Robert Tibshirani

References

See Also
hierNet, predict.hierNet, hierNet.cv

Examples
set.seed(12)
x=matrix(rnorm(100*10),ncol=10)
x=scale(x,TRUE,TRUE)
y=x[,1]+2*x[,2]+ x[,1]*x[,2]+3*rnorm(100)
fit=hierNet.path(x,y)
print(fit)
Variable importance for hierNet.

Description

(This is an experimental function.) Calculates a measure of the importance of each variable.

Usage

`hierNet.varimp(fit, x, y, ...)`

Arguments

- `fit`: The results of a call to the "hierNet"
- `x`: The training set feature matrix used in call produced "fit"
- `y`: The training set response vector used in call produced "fit"
- `...`: additional arguments (not currently used)

Value

Table of variable importance.

Author(s)

Jacob Bien and Robert Tibshirani

References


See Also

`hierNet, hierNet.path`

Examples

```r
set.seed(12)
x = matrix(rnorm(100*10), ncol=10)
x = scale(x, TRUE, TRUE)
y = x[,1] + 2*x[,2] + x[,1]*x[,2]*rnorm(100)
newx = matrix(rnorm(100*10), ncol=10)
fit = hierNet(x, y, lam=50)
yhat = predict.hierNet(fit, newx)

fit = hierNet.path(x, y)
yhat = predict.hierNet(fit, newx)
```
predict.hierNet

Prediction function for hierNet and hierNet.logistic.

Description
A function to perform prediction, using an x matrix and the output of the "hierNet" or "hierNet.logistic" function.

Usage
## S3 method for class 'hierNet'
predict(object, newx, newzz=NULL, ...)  

Arguments
- **object**: The results of a call to the "hierNet" or "hierNet.path" or function. The coefficients that are part of this object will be used for making predictions.
- **newx**: The new x at which predictions should be made. Can be a vector or a matrix (one observation per row).
- **newzz**: Optional matrix of products of columns of newx, computed by compute.interactions.c
- **...**: Additional arguments (not currently used)

Value
- **yhat**: Vector of predictions for each observation. For logistic model, these are the estimated probabilities.

Author(s)
Jacob Bien and Robert Tibshirani

References

See Also
hierNet, hierNet.path

Examples
set.seed(12)
x=matrix(rnorm(100*10),ncol=10)
x=scale(x,TRUE,TRUE)
y=x[,1]+2*x[,2]+ x[,1]*x[,2]+3*rnorm(100)
newx=matrix(rnorm(100*10),ncol=10)
fit=hierNet(x,y,lam=50)
predict.hierNet.logistic

**Prediction function for hierNet.logistic.**

---

**Description**

A function to perform prediction, using an x matrix and the output of the "hierNet.logistic" function or "hierNet.logistic.path".

**Usage**

```r
## S3 method for class 'hierNet.logistic'
predict(object, newx, newzz=NULL,...)
```

**Arguments**

- `object` The results of a call to the "hierNet.logistic" or "hierNet.logistic.path" or function. The coefficients that are part of this object will be used for making predictions.
- `newx` The new x at which predictions should be made. Can be a vector or a matrix (one observation per row).
- `newzz` Optional matrix of products of columns of newx, computed by `compute.interactions.c` additional arguments (not currently used)
- `...` additional arguments (not currently used)

**Value**

`yhat` Matrix of predictions (probabilities), one row per observation

**Author(s)**

Jacob Bien and Robert Tibshirani

**References**


**See Also**

hierNet.logistic, hierNet.logistic.path
Examples

```r
set.seed(12)
x = matrix(rnorm(100*10), ncol=10)
x = scale(x, TRUE, TRUE)
y = x[,1] + 2*x[,2] + x[,1]*x[,2] + 3*rnorm(100)
y = 1*(y>0)
newx = matrix(rnorm(100*10), ncol=10)
fit = hierNet.logistic(x, y, lam=5)
yhat = predict.hierNet.logistic(fit, newx)
```

```r
fit = hierNet.logistic.path(x, y)
yhat = predict.hierNet.logistic.path(fit, newx)
```

---

**predict.hierNet.path**  
*Prediction function for hierNet.path and hierNet.logistic.path.*

**Description**

A function to perform prediction, using an x matrix and the output of the "hierNet.path" or "hierNet.logistic.path" functions.

**Usage**

```r
## S3 method for class 'hierNet.path'
predict(object, newx, newzz=NULL, ...)
```

**Arguments**

- `object`: The results of a call to the "hierNet" or "hierNet.path" or function. The coefficients that are part of this object will be used for making predictions.
- `newx`: The new x at which predictions should be made. Can be a vector or a matrix (one observation per row).
- `newzz`: Optional matrix of products of columns of newx, computed by `compute.interactions.c`
- `...`: additional arguments (not currently used)

**Value**

- `yhat`: Matrix of predictions, one row per observation. For logistic model, these are the estimated probabilities.

**Author(s)**

Jacob Bien and Robert Tibshirani

**References**

See Also

hierNet, hierNet.path

Examples

```r
set.seed(12)
x = matrix(rnorm(100*10), ncol=10)
x = scale(x, TRUE, TRUE)
y = x[,1] + 2*x[,2] + x[,1]*x[,2] + 3*rnorm(100)
newx = matrix(rnorm(100*10), ncol=10)
fit = hierNet(x, y, lam=50)
yhat = predict.hierNet(fit, newx)

fit = hierNet.path(x, y)
yhat = predict.hierNet(fit, newx)
```
Index

hierNet. 2, 5, 9, 11–13, 16
hierNet.cv. 3, 4, 9, 11
hierNet.logistic. 5, 5, 14
hierNet.logistic.path. 5, 8, 14
hierNet.path. 3, 5, 10, 12, 13, 16
hierNet.varimp. 12

predict.hierNet. 3, 9, 11, 13
predict.hierNet.logistic. 7, 14
predict.hierNet.path. 15