

# Package ‘LogicForest’

February 14, 2012

**Type** Package

**Title** Logic Forest

**Version** 2.0.0

**Date** 2011-09-20

**Author** Bethany Wolf

**Maintainer** Bethany Wolf <wolfb@musc.edu>

**Depends** LogicReg, CircStats, gtools, plotrix

**Description** Two classification ensemble methods based on logic regression models. Logforest uses a bagging approach to construct an ensemble of logic regression models. LBoost uses a combination of boosting and cross-validation to construct and ensemble of logic regression models. Both methods are used for classification of binary responses based on binary predictors and for identification of important variables and variable interactions predictive of a binary outcome.

**License** GPL-2

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2011-09-21 19:22:01

## R topics documented:

LogicForest-package . . . . .	2
BoostVimp.plot . . . . .	2
LBoost . . . . .	4
LBoost.fit . . . . .	6
LF.data . . . . .	6
LF.testdata . . . . .	7

logforest . . . . .	8
logforest.fit . . . . .	10
persistence.plot . . . . .	10
predict.LBoost . . . . .	12
predict.logforest . . . . .	13
print.LBoost . . . . .	14
print.LFprediction . . . . .	15
print.logforest . . . . .	16
submatch.plot . . . . .	17
vimp.plot . . . . .	18

<b>Index</b>	<b>20</b>
--------------	-----------

---

LogicForest-package    *Logic Forest*

---

## Description

Two classification ensemble methods based on logic regression models. Logforest uses a bagging approach to construct an ensemble of logic regression models. LBoost uses a combination of boosting and cross-validation to construct an ensemble of logic regression models. Both methods are used for classification of binary responses based on binary predictors and for identification of important variables and variable interactions predictive of a binary outcome.

## Details

Package:	LogicForest
Type:	Package
Version:	2.0.0
Date:	2011-09-20
License:	GPL-2
LazyLoad:	yes

## Author(s)

Bethany Wolf

Maintainer: Bethany Wolf <wolfb@musc.edu>

---

BoostVimp.plot    *Variable and Interaction Importance Plots for a LBoost Model*

---

**Description**

Dot chart of variable and/or interaction importance for the variables/interactions with the largest magnitude variable importance scores.

**Usage**

```
BoostVimp.plot(fit, num=10, pred, norm=TRUE, titles=TRUE)
```

**Arguments**

<code>fit</code>	an object of class <code>LBoost</code> .
<code>num</code>	number of variables/interactions to be included on plot.
<code>pred</code>	logical. If <code>TRUE</code> , a variable importance plot is constructed for individual variables.
<code>norm</code>	logical. If <code>TRUE</code> , variable/interaction importance scores are normalized such that the largest importance score takes value one and all other values are scaled accordingly.
<code>titles</code>	logical. If <code>FALSE</code> , titles are not included on the plot.

**Value**

Plots up to three separate plots of the predictors/interactions with the largest magnitude variable importance score. If `pred=TRUE` is specified, one plot will be of the largest magnitude individual variable importance scores. Note, `pred.imp` must also have been specified as `TRUE` when running `LBoost` to be able to generate this plot. A plot for each type of interaction importance measure will also be generated if `PI.imp="Both"` when running `LBoost`. If only "Permutation" or "AddRemove" was specified for `PI.imp`, one plot will be generated for the interaction importance type specified in `LBoost`.

**Author(s)**

Bethany Wolf [wolfb@musc.edu](mailto:wolfb@musc.edu)

**References**

Wolf, B.J., Slate, E.H., Hill, E.G. (2010) Logic Forest: An ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics*.

**See Also**

[persistence.plot](#), [submatch.plot](#)

**Examples**

```
data(LBoost.fit)

#Plot of top 10 predictors based on variable importance from the LBoost
#model LBoost.fit
BoostVimp.plot(fit=LBoost.fit, num=10, pred=TRUE, norm=TRUE, titles=TRUE)
```

LBoost

*LBoost***Description**

Constructs an ensemble of logic regression models using boosting for classification and identification of important predictors and predictor interactions

**Usage**

```
LBoost(resp, Xs, anneal.params, nBS = 100, kfold = 5, nperm = 1,
       PI.imp = NULL, pred.imp = FALSE)
```

**Arguments**

<code>resp</code>	numeric vector of binary response values.
<code>Xs</code>	matrix or data frame of zeros and ones for all predictor variables.
<code>anneal.params</code>	a list containing the parameters for simulated annealing. See the help file for the function <code>logreg.anneal.control</code> in the <code>LogicReg</code> package. If missing, default annealing parameters are set at <code>start=1</code> , <code>end=-2</code> , and <code>iter=50000</code> .
<code>nBS</code>	number of logic regression trees to be fit in the LBoost model.
<code>kfold</code>	The number of times the data are to be split in constructing the ensemble.
<code>nperm</code>	If measuring predictor importance of interaction importance using the permutation based measure, <code>nperm</code> is the number of permutations to be done in determining predictor of interaction importance.
<code>PI.imp</code>	A character string describing which measure of interaction importance will be used. Possible values include "Permutation", "AddRemove", and "Both". Using "Permutation" will provide the permutation based measure of interaction importance, "AddRemove" will provide the add-in/leave-out based measure of interaction importance, and "Both" provides both measures of importance.
<code>pred.imp</code>	logical. If FALSE, predictor importance scores will not be measured.

**Value**

An object of class "LBoost" which is a list including values

<code>CVmod</code>	A list of all logic regression fits and the associated information in the LBoost model. Each item in the list also gives a list of LR fits for a specific kfold data set, a matrix of weights given to each LR fit for that kfold data set, a matrix of the kfold training data used to construct the list of fits.
<code>CVmisc</code>	a list including the mean cross-validation misclassification rate for the models and a list of vectors giving the predictions for each of the kfold test data sets.
<code>AddRemove.PIimport</code>	If <code>PI.imp</code> is specified as either "AddRemove" or "Both, this is a vector of add-in/leave-out importance scores for all interactions that occur in the LBoost model. If <code>PI.imp</code> is not specified or is "Permutation", this will state "Not measured".

Perm.PIimport	If PI.imp is specified as either "Permutation" or "Both, this is a vector of add-in/leave-out importance scores for all interactions that occur in the LBoost model. If PI.imp is not specified or is "AddRemove", this will state "Not measured".
Pred.import	If pred.imp is specified as TRUE, a vector of importance scores for all predictors in the data.
Pred.freq	a vector frequency of predictors occurring in individual logic regression in the LBoost model.
PI.frequency	a vector frequency of interactions occurring in individual logic regression in the LBoost model.
wt.mat	a list containing kfold matrices of observation weights for each tree for the kfold training data sets.
alphas	a list containing kfold vectors of tree specific weights for trees constructed from each of the kfold training data sets.
data	A matrix of the original data used to construct the LBoost model.
PIimp	A character string describing which interaction importance measure was used.
PredImp	logical. If TRUE predictor importance was measured.

**Author(s)**

Bethany Wolf wolfb@musc.edu

**References**

Wolf, B.J., Slate, E.H., Hill, E.G. (2010) Logic Forest: An ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics*.

**See Also**

[print.LBoost](#), [predict.LBoost](#), [BoostVimp.plot](#), [submatch.plot](#), [persistence.plot](#)

**Examples**

```
data(LF.data)

#Set using annealing parameters using the logreg.anneal.control
#function from LogicReg package
newanneal<-logreg.anneal.control(start=1, end=-2, iter=2500)

#typically more than 2500 iterations (>25000) would be used for
#the annealing algorithm. A typical LBoost models also contains at
#least 100 trees. These parameters were set to allow for faster
#run time

#The data set LF.data contains 50 binary predictors and a binary response Ybin
#Looking at only the Permutation Measure
LBfit.1<-LBoost(resp=LF.data$Ybin, Xs=LF.data[,1:50], nBS=20, kfold=4,
anneal.params=newanneal, nperm=2, PI.imp="Permutation")
print(LBfit.1)
```

```
#Looking at only the Add-in/Leave-out importance measure
LBfit.2<-LBoost(resp=LF.data$Ybin, Xs=LF.data[,1:50], nBS=20, kfold=4,
anneal.params=newanneal, PI.imp="AddRemove")
print(LBfit.2)

#Looking at both measures of importance plus predictor importance
LBfit.3<-LBoost(resp=LF.data$Ybin, Xs=LF.data[,1:50], nBS=20, kfold=4,
anneal.params=newanneal, nperm=2, PI.imp="Both", pred.imp=TRUE)
print(LBfit.3)
```

---

LBoost.fit

*Example of a LBoost Model*


---

### Description

The LBoost model for data with a binary response with error following the underlying logical relationship

$L=(X4 \text{ and } X5) \text{ or } (X9 \text{ and } X10)$

The LBoost model used 5-fold cross validation and includes 100 logic regression trees. Predictor importance and two measures of interaction (prime implicant) importance were also run as part of the model.

### Usage

```
data(LBoost.fit)
```

### Format

An object of class LBoost.

### Examples

```
data(LBoost.fit)
```

---

LF.data

*Example Data for Logic Forest and LBoost*


---

**Description**

LF.data is simulated case control data where the ratio of cases:controls is 1:1. The dataset contains 200 observations and 50 binary predictors. Columns 1,2,.....,50 are the binary predictors. Column 51 is the true binary response and column 52 is a noisy binary response. Each predictor is simulated as an independent Bernoulli random variable with success probability 0.5. The response variable is simulated from the model

$$Y=(X4 \text{ and } X5) \text{ or } (X9 \text{ and } X10)$$

The true response in column 51 takes value 1 if either of these conditions occurs and takes value 0 otherwise. Column 52 represents the response with some error. The probability that the noisy response (column 52) has the same value as the true response is 90

**Usage**

```
LF.data
```

**Examples**

```
data(LF.data)
```

---

LF.testdata

*Example of New Test Data for Prediction in Logic Forest and LBoost*

---

**Description**

LF.testdata is simulated case control data where the ratio of cases:controls is 1:1. The dataset contains 100 observations and 50 binary predictors. Columns 1,2,.....,50 are the binary predictors. Column 51 is the true binary response and column 52 is a noisy binary response. Each predictor is simulated as an independent Bernoulli random variable with success probability 0.5. The response variable is simulated from the model

$$L=(X4 \text{ and } X5) \text{ or } (X9 \text{ and } X10)$$

The true response in column 51 takes value 1 if either of these conditions occurs and takes value 0 otherwise. The probability that the noisy response (column 52) has the same value as the true response is 90

**Usage**

```
LF.testdata
```

**Examples**

```
data(LF.testdata)
```

logforest

*Logic Forest***Description**

Constructs an ensemble of logic regression models using bagging for classification and identification of important predictors and predictor interactions

**Usage**

```
logforest( resp, Xs, nBSXVars, anneal.params, nBS = 100, h = 0.5,
norm = TRUE, numout = 5)
```

**Arguments**

resp	numeric vector of binary response values.
Xs	matrix or dataframe of zeros and ones for all predictor variables.
nBSXVars	integer for the number of predictors used to construct each logic regression model. The default value is all predictors in the data.
anneal.params	a list containing the parameters for simulated annealing. See the help file for the function <code>logreg.anneal.control</code> in the <code>LogicReg</code> package. If missing, default annealing parameters are set at <code>start=1</code> , <code>end=-2</code> , and <code>iter=50000</code> .
nBS	number of logic regression trees to be fit in the logic forest model.
h	a number between 0 and 1 for the minimum proportion of trees in the logic forest that must predict a 1 for the prediction to be one.
norm	logical. If <code>FALSE</code> , predictor and interaction scores in model output are not normalized to range between zero and one.
numout	number of predictors and interactions to be included in model output

**Value**

An object of class "logforest" which is a list including values

AllFits	A list of all logic regression fits in the logic forest model.
Top5.PI	a vector of the 5 interactions with the largest magnitude variable importance score.
Predictor.importance	a vector of importance scores for all predictors that occur in the logic forest.
PI.importance	a vector of importance scores for all interactions that occur in the logic forest.
Predictor.frequency	a vector frequency of predictors occurring in individual logic regression in the logic forest.
PI.frequency	a vector frequency of interactions occurring in individual logic regression in the logic forest.

ModelPI.import	a list on interaction importance measures for each logic regression model in the logic forest.
OOBmisclass	out-of-bag error estimate for the logic forest.
OOBprediction	a matrix. Column one is the out-of-bag prediction for responses in original data. Columns 2 is the proportion of out-of-bag trees that predicted class value to be one.
IBdata	a list of all in-bag data sets for the logic forest model.
OOBdata	a list of all out-of-bag data sets for the logic forest model.
norm	logical. If TRUE the normalized predictor and interaction importance scores are returned.
numout	the number of predictors and interactions (based on the variable importance measure) to be returned by logforest.
predictors	number of predictor variables in the data used to construct the logic forest.

**Author(s)**

Bethany Wolf wolfb@musc.edu

**References**

Wolf, B.J., Slate, E.H., Hill, E.G. (2010) Logic Forest: An ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics*.

**See Also**

[print.logforest](#), [predict.logforest](#), [vimp.plot](#), [submatch.plot](#), [persistence.plot](#)

**Examples**

```
data(LF.data)

#Set using annealing parameters using the logreg.anneal.control
#function from LogicReg package

newanneal<-logreg.anneal.control(start=1, end=-2, iter=2500)

#typically more than 2500 iterations (iter>25000) would be used for
#the annealing algorithm. A typical forest also contains at
#least 100 trees. These parameters were set to allow for faster
#run times

#The data set LF.data contains 50 binary predictors and a binary
#response Ybin
LF.fit1<-logforest(resp=LF.data$Ybin, Xs=LF.data[,1:50], nBS=20,
anneal.params=newanneal)
print(LF.fit1)
predict(LF.fit1)
```

```
#Changing print parameters
LF.fit2<-logforest(resp=LF.data$Ybin, Xs=LF.data[,1:50], nBS=20,
anneal.params=newanneal, norm=TRUE, numout=10)
print(LF.fit2)
```

---

logforest.fit                    *Example of a Logic Forest Model*

---

### **Description**

The Logic Forest model for data with a binary response with error following the underlying logical relationship

$L=(X4 \text{ and } X5) \text{ or } (X9 \text{ and } X10)$

The logic forest includes 100 logic regression trees.

### **Usage**

```
data(logforest.fit)
```

### **Format**

An object of class logforest.

### **Examples**

```
data(logforest.fit)
```

---

persistence.plot                    *Plot Persistence of a Variable of Interaction*

---

### **Description**

Plot the persistence of a particular variable or interaction term in a logic forest.

### **Usage**

```
persistence.plot(fit, preds, PI, title)
```

### **Arguments**

fit	an object of class logforest.
preds	a numeric value representing the number of predictors in the original data used to construct the logic forest model.
PI	a character string representing the variable or interaction term to be plotted.
title	an optional title for the plot

## Details

Persistence plots can be generated for any variable or interaction term that occurs in a logic forest or LBoost model. Plots can also be generated for variables and interaction terms that do not occur in the model exactly as given in the argument PI assuming that at least one subset match for that variable/interaction exists.

## Value

Invisibly all subset matches for the variable or interaction term defined by the argument PI.

## Author(s)

Bethany Wolf wolfb@musc.edu

## References

Wolf, B.J., Slate, E.H., Hill, E.G. (2010) Logic Forest: An ensemble classifier for discovering logical combinations of binary markers. Bioinformatics.

## See Also

[vimp.plot](#), [submatch.plot](#)

## Examples

```
data(logforest.fit, LBoost.fit)

#Persistence of the predictor X5 in the logic forest model logforest.fit
persistence.plot(fit=logforest.fit, preds=50, PI="X5",
title="logforest Persistence for X5")
#Same plot for the LBoost model LBoost.fit
persistence.plot(fit=LBoost.fit, preds=50, PI="X5",
title="LBoost Persistence for X5")

#Persistence of the interaction term (X4 & X5) in the logic forest model
#logforest.fit
persistence.plot(fit=logforest.fit, preds=50, PI="X4 & X5",
title="logforest Persistence for interaction X4 & X5")

#Same plot for the LBoost model LBoost.fit
persistence.plot(fit=LBoost.fit, preds=50, PI="X4 & X5",
title="LBoost persistence for interaction X4 & X5")
```

---

predict.LBoost            *Prediction of Response Using LBoost*

---

### Description

Computes predicted values for newdata for as LBoost model fitted using LBoost.

### Usage

```
## S3 method for class 'LBoost'  
predict(object, newdata, ...)
```

### Arguments

object	an object of class LBoost.
newdata	a matrix or data frame containing new predictor values. If not given, predictions will be made for the original data used to construct the model.
...	not currently used.

### Value

If new data are given, predict returns an object of class "LBoost.prediction", that includes a list of elements

prediction	a numeric vector of predicted responses.
weighted.prop	a numeric vector of the weighted proportion of trees in the forest that predict a class value of one.

If no new data are given, the function returns the LBoost model prediction for the original data.

### Author(s)

Bethany Wolf wolfb@musc.edu

### See Also

[LBoost](#)

### Examples

```
data(LF.data, LF.testdata, LBoost.fit)
```

```
#NOTE- this is the LBoost model prediction for the original data  
predict(object=LBoost.fit)
```

```
#predicting classes for new test data  
predict(object=LBoost.fit, newdata=LF.testdata[,1:50])
```

---

predict.logforest	<i>Prediction of Response Using Logic Forest</i>
-------------------	--------------------------------------------------

---

**Description**

Computes predicted values for newdata or the out-of-bag prediction for a logic forest model fitted using logforest.

**Usage**

```
## S3 method for class 'logforest'  
predict(object, newdata, cutoff, ...)
```

**Arguments**

object	an object of class logforest.
newdata	a matrix or data frame containing new predictor values. If not given, predictions will be made for the original data used to construct the model.
cutoff	a number between 0 and 1 for the minimum proportion of trees in the logic forest that must predict a 1 for the prediction to be one.
...	not currently used.

**Value**

If new data are given, predict returns an object of class "LFprediction", that includes a list of elements

LFprediction	a numeric vector of predicted responses.
proportion_one	a numeric vector of the proportion of trees in the forest that predict a class value of one.
AllTrees	a matrix/vector containing the predicted values of each tree in a logic forest for each response, the proportion of trees that predict a one, and the predicted class for each observations.

If no new data are given, the function returns the out-of-bag prediction for the original data. The function returns a matrix. The first column is the out-of-bag prediction for all observations from the original data and the second column is the proportion of out-of-bag trees the predict a one.

**Author(s)**

Bethany Wolf wolfb@musc.edu

**See Also**

[logforest](#)

**Examples**

```

data(LF.data, LF.testdata, logforest.fit)

#predicting classes for the original data
#NOTE- this is the out-of-bag prediction
predict(object=logforest.fit)

#predicting classes for new test data
predict(object=logforest.fit, newdata=LF.testdata[,1:50])

```

---

<code>print.LBoost</code>	<i>Prints Output for and LBoost Model</i>
---------------------------	-------------------------------------------

---

**Description**

Prints a matrix of the most important predictors and interactions, their variable importance scores, and frequency of occurrence for objects fitted by logforest.

**Usage**

```

## S3 method for class 'LBoost'
print(x, num, ...)

```

**Arguments**

<code>x</code>	object of class logforest.
<code>num</code>	number of predictors/prime implicants to be included in the print out of the model. The default value is 5.
<code>...</code>	other options are ignored.

**Value**

If `pred.imp` was set equal to `TRUE` when constructing the LBoost model, a matrix including the most important predictors, their normalized importance scores and the frequency of occurrence in the LBoost model will be included.

If `PI.imp` was set equal to "Permutation" when constructing the LBoost model, a matrix including the most important predictors, their normalized permutation based importance scores and the frequency of occurrence in the LBoost model will be included. If `PI.imp` was set equal to "AddRemove" when constructing the LBoost model, a matrix including the most important predictors, their normalized add-in/leave-out based importance scores and the frequency of occurrence in the LBoost model will be included. If `PI.imp` was set equal to "Both" when constructing the LBoost model, two matrices including the most important predictors, their normalized importance scores (by each method) and the frequency of occurrence in the LBoost model will be included.

**Author(s)**

Bethany Wolf wolfb@musc.edu

**See Also**

[LBoost](#)

**Examples**

```
data(LBoost.fit)
print(LBoost.fit)
```

---

`print.LFprediction`      *Prints Logic Forest Prediction Output*

---

**Description**

Prints predictions from a logic forest model.

**Usage**

```
## S3 method for class 'LFprediction'
print(x, ...)
```

**Arguments**

`x`                    an object of class `LF.prediction`.  
`...`                  other options are ignored.

**Value**

Returns a numeric vector of predicted classes for all observations in `newdata` or the out-of-bag predictions for data used to construct the logic forest and a numeric vector of the proportion of trees in a logic forest that predict a class of one.

**Author(s)**

Bethany Wolf wolfb@musc.edu

**See Also**

[predict.logforest](#)

**Examples**

```
data(logforest.fit, LF.testdata)

#OOB prediction on original data
predict(logforest.fit)

#Prediction for new test data
predict(logforest.fit, newdata=LF.testdata[,1:50])
```

---

<code>print.logforest</code>	<i>Prints Output for a Logic Forest Model</i>
------------------------------	-----------------------------------------------

---

**Description**

Prints a matrix of the most important predictors and interactions, their variable importance scores, and frequency of occurrence for objects fitted by logforest.

**Usage**

```
## S3 method for class 'logforest'
print(x, ...)
```

**Arguments**

<code>x</code>	object of class logforest.
<code>...</code>	other options are ignored.

**Value**

If `x$norm=TRUE`, all variable importance scores are normalized such that the largest score take value one and all other scores are scaled accordingly.

**Author(s)**

Bethany Wolf [wolfb@musc.edu](mailto:wolfb@musc.edu)

**See Also**

[logforest](#)

**Examples**

```
data(logforest.fit)
print(logforest.fit)
```

---

submatch.plot	<i>Plot of Variable/Interaction Frequency</i>
---------------	-----------------------------------------------

---

**Description**

Plot of frequency of occurrence of variables/interactions in increasingly larger interactions identified in a logic forest model.

**Usage**

```
submatch.plot(fit, pred.nms, pis, preds, size, color = FALSE)
```

**Arguments**

fit	an object of class logforest.
pred.nms	an optional vector of predictor names considered for subset matching. If missing, all variable names from the original data used to construct the logic forest model are considered.
pis	a numeric value representing the number of variables/interactions to be included on the plot. Variables/interactions are sorted according to frequency of occurrence as subset matches.
preds	a numeric value representing the number of predictors in the original dataset.
size	a numeric value for the size of the interaction terms represented on the plot.
color	logical. If TRUE, plots are generated using color, otherwise they are in grayscale.

**Details**

The center plot shows the most frequently occurring interactions of the size specified in the arguments. Circles on the plot represent the relative frequency of occurrence on the interactions on the y-axis in interactions of the same size and all larger sizes for those interactions identified in a logic forest model. The histogram at the top of the plot represents the total number of interaction terms of the size indicated on the x-axis identified by the logic forest model. The dark gray in this histogram represents the number of interactions of that size that are accounted for by the interactions shown of the y-axis. The histogram on the right side of the plot represents the frequency of occurrence of all interactions identified in a logic forest model of the size specified in the arguments. Dark gray bars in the histogram represent the frequency of occurrence of those interactions on the y-axis.

**Value**

Invisibly the number of subset matches at each size for those interactions given in the plot.

**Author(s)**

Bethany Wolf wolfb@musc.edu

**References**

Wolf, B.J., Slate, E.H., Hill, E.G. (2010) Logic Forest: An ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics*.

**See Also**

[vimp.plot](#), [link{persistence.plot}](#)

**Examples**

```
data(logforest.fit, LBoost.fit)

#Plot of subset matches for the 5 most frequently occurring interactions
#of size 2 in the logic forest model logforest.fit
submatch.plot(fit=logforest.fit, pis=5, preds=50, size=2, color=FALSE)

#Same plot for the LBoost model LBoost.fit
submatch.plot(fit=LBoost.fit, pis=5, preds=50, size=2, color=FALSE)
```

---

vimp.plot

---

*Variable and Interaction Importance Plots for a Logic Forest Model*


---

**Description**

Dot chart of variable and/or interaction importance for the variables/interactions with the largest magnitude variable importance scores.

**Usage**

```
vimp.plot(fit, num=10, type=2, norm=TRUE, titles=TRUE)
```

**Arguments**

fit	an object of class logforest.
num	number of variables/interactions to be included on plot.
type	type of plot to be constructed: type=0 generates a plot of the individual variables with the largest importance score, type=1 generated a plot of the interaction terms with the largest importance score, and type=2 generates side by side plot of the most important single variable and the most important interactions from a logic forest model.
norm	logical. If TRUE, variable/interaction importance scores are normalized such that the largest importance score takes value one and all other values are scaled accordingly.
titles	logical. If FALSE, titles are not included on the plot.

**Value**

Invisibly the predictors/interactions with the largest magnitude variable importance score.

**Author(s)**

Bethany Wolf wolfb@musc.edu

**References**

Wolf, B.J., Slate, E.H., Hill, E.G. (2010) Logic Forest: An ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics*.

**See Also**

[persistence.plot](#), [submatch.plot](#)

**Examples**

```
data(logforest.fit)

#Plot of top 10 predictors based on variable importance from logforest
vimp.plot(fit=logforest.fit, type=0, norm=FALSE)

#Plot of top 10 interactions based on variable importance from
vimp.plot(fit=logforest.fit, num=10, type=1, norm=FALSE)

#Plots of top 10 predictors and interactions based on variable
#importance from logforest
vimp.plot(fit=logforest.fit, num=10, type=2, norm=FALSE)

#Plots of top 10 predictors and interactions based on normalized
#variable importance from logforest
vimp.plot(fit=logforest.fit, num=10, type=2)
```

# Index

## \*Topic **datasets**

LBoost.fit, [6](#)

LF.data, [6](#)

LF.testdata, [7](#)

logforest.fit, [10](#)

## \*Topic **package**

LogicForest-package, [2](#)

BoostVimp.plot, [2](#), [5](#)

LBoost, [3](#), [4](#), [12](#), [15](#)

LBoost.fit, [6](#)

LF.data, [6](#)

LF.testdata, [7](#)

logforest, [8](#), [13](#), [16](#)

logforest.fit, [10](#)

LogicForest (LogicForest-package), [2](#)

LogicForest-package, [2](#)

persistence.plot, [3](#), [5](#), [9](#), [10](#), [19](#)

predict.LBoost, [5](#), [12](#)

predict.logforest, [9](#), [13](#), [15](#)

print.LBoost, [5](#), [14](#)

print.LFprediction, [15](#)

print.logforest, [9](#), [16](#)

submatch.plot, [3](#), [5](#), [9](#), [11](#), [17](#), [19](#)

vimp.plot, [9](#), [11](#), [18](#), [18](#)