

# Package ‘stepgbm’

April 4, 2023

**Title** Stepwise Variable Selection for Generalized Boosted Regression Modeling

**Version** 1.0.1

**Date** 2023-04-03

**Description** An introduction to a couple of novel predictive variable selection methods for generalised boosted regression modeling (gbm). They are based on various variable influence methods (i.e., relative variable influence (RVI) and knowledge informed RVI (i.e., KIRVI, and KIRVI2)) that adopted similar ideas as AVI, KIAVI and KIAVI2 in the 'steprf' package, and also based on predictive accuracy in stepwise algorithms. For details of the variable selection methods, please see: Li, J., Siwabessy, J., Huang, Z. and Nichol, S. (2019) <[doi:10.3390/geosciences9040180](https://doi.org/10.3390/geosciences9040180)>. Li, J., Alvarez, B., Siwabessy, J., Tran, M., Huang, Z., Przeslawski, R., Radke, L., Howard, F., Nichol, S. (2017). <[DOI:10.13140/RG.2.2.27686.22085](https://doi.org/10.13140/RG.2.2.27686.22085)>.

**Depends** R (>= 4.0)

**Imports** spm, steprf

**License** GPL (>= 2)

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**Suggests** knitr, rmarkdown, reshape2, lattice

**NeedsCompilation** no

**Author** Jin Li [aut, cre]

**Maintainer** Jin Li <[jinli68@gmail.com](mailto:jinli68@gmail.com)>

**Repository** CRAN

**Date/Publication** 2023-04-04 08:20:02 UTC

## R topics documented:

cran-comments	2
stepgbm	2
stepgbmRVI	5

<b>Index</b>	<b>9</b>
--------------	----------

---

 cran-comments

*Note on notes*


---

### Description

This is an updated and extended version of ‘spm’ package. The change in package name from ‘spm’ to ‘spm2’ is due to the change in Author’s support from Geoscience Australia to Data2Action Australia.

## R CMD check results 0 errors | 0 warnings | 0 notes

### Author(s)

Jin Li

---

 stepgbm

*Select predictive variables for generalized boosted regression modeling (gbm) by various variable influence methods and predictive accuracy in a stepwise algorithm*


---

### Description

This function is to select predictive variables for generalized boosted regression modeling (gbm) based on various variable influence methods (i.e., relative variable influence (RVI) and knowledge informed RVI (i.e., KIRVI, and KIRVI2)) and predictive accuracy. It is implemented via the functions ‘stepgbmRVI’ and ‘steprf::steprfAVIPredictors’. It should be noted that this function can be slow, may taking several minutes, hours or even days depending on the number of the predictive variables to be selected and the specifications of relevant arguments.

### Usage

```
stepgbm(
  trainx,
  trainy,
  method = "KIRVI",
  var.monotone = rep(0, ncol(trainx)),
  family = "gaussian",
  n.trees = 3000,
  learning.rate = 0.001,
  interaction.depth = 2,
  bag.fraction = 0.5,
  train.fraction = 1,
  n.minobsinnode = 10,
  cv.fold = 10,
  weights = rep(1, nrow(trainx)),
  keep.data = FALSE,
```

```

    verbose = TRUE,
    n.cores = 6,
    rpt = 2,
    predacc = "VEcv",
    min.n.var = 2,
    delta.predacc = 0.001,
    rseed = 1234,
    ...
)

```

## Arguments

trainx	a dataframe or matrix contains columns of predictive variables.
trainy	a vector of response, must have length equal to the number of rows in trainx.
method	a variable selection method for 'GBM'; can be: "RVI", "KIRVI" and "KIRVI2". If "RVI" is used, it would produce the same results as 'stepgbmRVI'. By default, "KIRVI" is used.
var.monotone	an optional vector, the same length as the number of predictors, indicating which variables have a monotone increasing (+1), decreasing (-1), or arbitrary (0) relationship with the outcome. By default, a vector of 0 is used.
family	either a character string specifying the name of the distribution to use or a list with a component name specifying the distribution and any additional parameters needed. See gbm for details. By default, "gaussian" is used.
n.trees	the total number of trees to fit. This is equivalent to the number of iterations and the number of basis functions in the additive expansion. By default, 3000 is used.
learning.rate	a shrinkage parameter applied to each tree in the expansion. Also known as step-size reduction. By default, 0.001 is used.
interaction.depth	the maximum depth of variable interactions. 1 implies an additive model, 2 implies a model with up to 2-way interactions, etc. By default, 2 is used.
bag.fraction	the fraction of the training set observations randomly selected to propose the next tree in the expansion. By default, 0.5 is used.
train.fraction	The first train.fraction * nrow(data) observations are used to fit the gbm and the remainder are used for computing out-of-sample estimates of the loss function.
n.minobsinnode	minimum number of observations in the trees terminal nodes. Note that this is the actual number of observations not the total weight. By default, 10 is used.
cv.fold	integer; number of cross-validation folds to perform within gbm. if > 1, then apply n-fold cross validation; the default is 10, i.e., 10-fold cross validation that is recommended.
weights	an optional vector of weights to be used in the fitting process. Must be positive but do not need to be normalized. If keep.data = FALSE in the initial call to gbm then it is the user's responsibility to resupply the weights to gbm.more. By default, a vector of 1 is used.

keep.data	a logical variable indicating whether to keep the data and an index of the data stored with the object. Keeping the data and index makes subsequent calls to <code>gbm.more</code> faster at the cost of storing an extra copy of the dataset. By default, 'FALSE' is used.
verbose	If TRUE, <code>gbm</code> will print out progress and performance indicators. By default, 'TRUE' is used.
n.cores	The number of CPU cores to use. See <code>gbm</code> for details. By default, 6 is used.
rpt	iteration of cross validation.
predacc	"VEcv" for <code>vecv</code> in function <code>pred.acc</code> .
min.n.var	minimum number of predictive variables remained in the final predictive model the default is 1.
delta.predacc	minimum changes between the accuracy of two consecutive predictive models. By default, 0.01 is used.
rseed	random seed. By default, 1234 is used.
...	other arguments passed on to <code>gbm</code> .

### Value

A list with the following components: 1) `stepgbmPredictorsFinal`: the variables selected for the last GBM model, whether it is of the highest predictive accuracy need to be confirmed using 'max.predictive.accuracy' that is listed next; 2) `max.predictive.accuracy`: the predictive accuracy of the most accurate GBM model for each run of 'stepgbmRVI', which can be used to confirm the model with the highest accuracy, 3) `numberruns`: number of runs of 'stepgbmRVI'; 4) `laststepRVI`: the outputs of last run of 'stepgbmRVI'; 5) `stepgbmRVIOutputsAll`: the outputs of all 'stepgbmRVI' produced during the variable selection process; 6) `stepgbmPredictorsAll`: the outputs of 'stepgbmRVIPredictors' for all 'stepgbmRVI' produced during the variable selection process; 7) `KIRVIPredictorsAll`: predictors used for all 'stepgbmRVI' produced during the variable selection process; for a method "RVI", if the variables are different from those in the training dataset, it suggests that these variables should be tested if the predictive accuracy can be further improved.

### Author(s)

Jin Li

### References

- Li, J. (2022). *Spatial Predictive Modeling with R*. Boca Raton, Chapman and Hall/CRC.
- Li, J., Siwabessy, J., Huang, Z., Nichol, S. (2019). "Developing an optimal spatial predictive model for seabed sand content using machine learning, geostatistics and their hybrid methods." *Geosciences* 9 (4):180.
- Li, J., Alvarez, B., Siwabessy, J., Tran, M., Huang, Z., Przeslawski, R., Radke, L., Howard, F., Nichol, S. (2017). "Application of random forest, generalised linear model and their hybrid methods with geostatistical techniques to count data: Predicting sponge species richness." *Environmental Modelling & Software* 97: 112-129.
- Li, J., Alvarez, B., Siwabessy, J., Tran, M., Huang, Z., Przeslawski, R., Radke, L., Howard, F., Nichol, S. (2017). Selecting predictors to form the most accurate predictive model for count data. *International Congress on Modelling and Simulation (MODSIM) 2017*, Hobart.

**Examples**

```

library(spm)

data(petrel)

stepgbm1 <- stepgbm(trainx = petrel[, c(1,2, 6:9)], trainy =
log(petrel[, 5] + 1), method = "KIRVI", family = "gaussian", rpt = 2,
predacc = "VEcv", cv.fold = 5, min.n.var = 2, n.cores = 2,
delta.predacc = 0.01, rseed = 1234)
names(stepgbm1)
stepgbm1$stepgbmPredictorsFinal$variables.most.accurate
stepgbm1$max.predictive.accuracy
stepgbm1$stepgbmPredictorsAll[[1]]

# The variables selected can be derived with
stepgbm1$stepgbmPredictorsAll[[1]]$variables.most.accurate

data(sponge)
stepgbm2 <- stepgbm(trainx = sponge[, -3], trainy = sponge[, 3], method = "KIRVI",
family = "poisson", rpt = 2, cv.fold = 5, predacc = "VEcv", min.n.var = 2,
n.cores = 2, delta.predacc = 0.01, rseed = 1234)
stepgbm2
stepgbm2$max.predictive.accuracy

# The variables selected can be derived with
stepgbm2$stepgbmPredictorsAll[[1]]$variables.most.accurate

```

---

stepgbmRVI

*Select predictive variables for generalized boosted regression modeling (gbm) by relative variable influence (rvi) and accuracy in a step-wise algorithm*

---

**Description**

This function is to select predictive variables for generalized boosted regression modeling (gbm) by their relative variable influence that is calculated for each model after excluding the least influence variable, and corresponding predictive accuracy. It is also developed for 'stepgbm' function.

**Usage**

```

stepgbmRVI(
  trainx,
  trainy,
  var.monotone = rep(0, ncol(trainx)),
  family = "gaussian",
  n.trees = 3000,

```

```

learning.rate = 0.001,
interaction.depth = 2,
bag.fraction = 0.5,
train.fraction = 1,
n.minobsinnode = 10,
cv.fold = 10,
weights = rep(1, nrow(trainx)),
keep.data = FALSE,
verbose = TRUE,
n.cores = 6,
rpt = 2,
predacc = "VEcv",
min.n.var = 2,
rseed = 1234,
...
)

```

### Arguments

<code>trainx</code>	a dataframe or matrix contains columns of predictive variables.
<code>trainy</code>	a vector of response, must have length equal to the number of rows in <code>trainx</code> .
<code>var.monotone</code>	an optional vector, the same length as the number of predictors, indicating which variables have a monotone increasing (+1), decreasing (-1), or arbitrary (0) relationship with the outcome. By default, a vector of 0 is used.
<code>family</code>	either a character string specifying the name of the distribution to use or a list with a component name specifying the distribution and any additional parameters needed. See <code>gbm</code> for details. By default, "gaussian" is used.
<code>n.trees</code>	the total number of trees to fit. This is equivalent to the number of iterations and the number of basis functions in the additive expansion. By default, 3000 is used.
<code>learning.rate</code>	a shrinkage parameter applied to each tree in the expansion. Also known as step-size reduction.
<code>interaction.depth</code>	the maximum depth of variable interactions. 1 implies an additive model, 2 implies a model with up to 2-way interactions, etc. By default, 2 is used.
<code>bag.fraction</code>	the fraction of the training set observations randomly selected to propose the next tree in the expansion. By default, 0.5 is used.
<code>train.fraction</code>	The first <code>train.fraction * nrow(data)</code> observations are used to fit the <code>gbm</code> and the remainder are used for computing out-of-sample estimates of the loss function.
<code>n.minobsinnode</code>	minimum number of observations in the trees terminal nodes. Note that this is the actual number of observations not the total weight. By default, 10 is used.
<code>cv.fold</code>	integer; number of cross-validation folds to perform within <code>gbm</code> . if > 1, then apply n-fold cross validation; the default is 10, i.e., 10-fold cross validation that is recommended.

weights	an optional vector of weights to be used in the fitting process. Must be positive but do not need to be normalized. If keep.data = FALSE in the initial call to gbm then it is the user's responsibility to resupply the weights to gbm.more. By default, a vector of 1 is used.
keep.data	a logical variable indicating whether to keep the data and an index of the data stored with the object. Keeping the data and index makes subsequent calls to gbm.more faster at the cost of storing an extra copy of the dataset. By default, 'FALSE' is used.
verbose	If TRUE, gbm will print out progress and performance indicators. By default, 'TRUE' is used.
n.cores	The number of CPU cores to use. See gbm for details. By default, 6 is used.
rpt	iteration of cross validation.
predacc	"VEcv" for vecv in function pred.acc.
min.n.var	minimum number of predictive variables remained in the final predictive model the default is 1.
rseed	random seed. By default, 1234 is used.
...	other arguments passed on to gbm.

### Value

A list with the following components: variable removed based on avi (variable.removed), averaged predictive accuracy of the model after excluding variable.removed (predictive.accuracy), contribution to accuracy by each variable.removed (delta.accuracy), and predictive accuracy matrix of the model after excluding variable.removed for each iteration (predictive.accuracy2)

### Author(s)

Jin Li

### References

- Li, J., Siwabessy, J., Huang, Z., Nichol, S. (2019). "Developing an optimal spatial predictive model for seabed sand content using machine learning, geostatistics and their hybrid methods." *Geosciences* 9 (4):180.
- Li, J., Alvarez, B., Siwabessy, J., Tran, M., Huang, Z., Przeslawski, R., Radke, L., Howard, F., Nichol, S. (2017). "Application of random forest, generalised linear model and their hybrid methods with geostatistical techniques to count data: Predicting sponge species richness." *Environmental Modelling & Software* 97: 112-129.
- Li, J., Alvarez, B., Siwabessy, J., Tran, M., Huang, Z., Przeslawski, R., Radke, L., Howard, F., Nichol, S. (2017). Selecting predictors to form the most accurate predictive model for count data. *International Congress on Modelling and Simulation (MODSIM) 2017*, Hobart.
- Chang, W. 2021. Cookbook for R. <http://www.cookbook-r.com/>.

**Examples**

```

library(spm)
data(petrel)
stepgbm1 <- stepgbmRVI(trainx = petrel[, c(1,2, 6:9)], trainy = log(petrel[, 5] + 1),
  cv.fold = 5, min.n.var = 2, n.cores = 2, rseed = 1234)
stepgbm1

#plot stepgbm1 results
library(reshape2)
pa1 <- as.data.frame(stepgbm1$predictive.accuracy2)
names(pa1) <- stepgbm1$variable.removed
pa2 <- melt(pa1, id = NULL)
names(pa2) <- c("Variable","VEcv")
library(lattice)
with(pa2, boxplot(VEcv~Variable, ylab="VEcv (%)", xlab="Predictive variable removed"))

barplot(stepgbm1$delta.accuracy, col = (1:length(stepgbm1$variable.removed)),
  names.arg = stepgbm1$variable.removed, main = "Predictive accuracy vs variable removed",
  font.main = 4, cex.names=1, font=2, ylab="Increase rate in VEcv (%)")

# Extract names of the selected predictive variables by stepgbm
library(steprf)
steprfAVIPredictors(stepgbm1, trainx = petrel[, c(1,2, 6:9)])

data(sponge)
set.seed(1234)
stepgbm2 <- stepgbmRVI(trainx = sponge[, -3], trainy = sponge[, 3],
  family = "poisson", cv.fold = 5, min.n.var = 2, n.cores = 2)
stepgbm2

#plot stepgbm2 results
library(reshape2)
pa1 <- as.data.frame(stepgbm2$predictive.accuracy2)
names(pa1) <- stepgbm2$variable.removed
pa2 <- melt(pa1, id = NULL)
names(pa2) <- c("Variable","VEcv")
library(lattice)
with(pa2, boxplot(VEcv~Variable, ylab="VEcv (%)", xlab="Predictive variable removed"))

barplot(stepgbm2$delta.accuracy, col = (1:length(stepgbm2$variable.removed)),
  names.arg = stepgbm2$variable.removed, main = "Predictive accuracy vs variable removed",
  font.main = 4, cex.names=1, font=2, ylab="Increase rate in VEcv (%)")

# Extract names of the selected predictive variables by stepgbm
steprfAVIPredictors(stepgbm2, trainx = sponge[, -3])

```



# Index

cran-comments, [2](#)

stepgbm, [2](#)

stepgbmRVI, [5](#)