Package ‘BAGofT’

November 15, 2019

Type Package
Title A Binary Regression Adaptive Goodness-of-Fit Test (BAGofT)
Version 0.1.0
Description Performs goodness-of-fit test for binary regression models with at least 1 continuous covariate. The implemented method BAGofT is from Zhang, Ding and Yang (2019) <arXiv:1911.03063>.
Depends R (>= 3.6.0)
License GPL-3
Encoding UTF-8
LazyData true
Imports stringr (>= 1.4.0)
Suggests nnet (>= 7.3.12), randomForest (>= 4.6.14)
RoxygenNote 6.1.1
NeedsCompilation no
Author Jiawei Zhang [aut, cre],
Jie Ding [aut],
Yuhong Yang [aut]
Maintainer Jiawei Zhang <zhan4362@umn.edu>
Repository CRAN
Date/Publication 2019-11-15 12:30:02 UTC

R topics documented:

BAGofT ................................................................. 2

Index 6
Description

BAGofT is used to test the goodness-of-fit of binary regression models with at least one continuous covariate. The test statistic is constructed based on the results from multiple splits. In each split, the test first splits the data into a training set and test set. Then it adaptively selects candidate partitions based on the training set and performs chi square tests with necessary corrections on the test set. The selection algorithm is the tree-based greedy adaptive partition scheme from Zhang, Ding and Yang (2019). Current version supports goodness-of-fit tests for logistic regression, probit regression and complementary log-log regression. R package "stringr" is required.

Usage

BAGofT(formula, data, link = "logit", Ctv = NULL, Dsv = NULL, g = 5

, nsplits = 100, spp = 1/2.3, min.Obsr = 10, adj = TRUE,

partition.Method = "xqt")

Arguments

formula an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to test.
data a data frame containing the covariates used in the model and the other covariates not in the model but considered to form the partition.
link a specification for the model link function. Can be one of "logit", "probit", "cloglog".
Ctv a character vector of the names of the continuous covariates to choose the partition.
Dsv a character vector of the names of the discrete covariates to choose the partition.
g number of maximum groups for the tree-based greedy adaptive partition selection.
nsplits number of splits.
spp decides the number of observations in the test set. Test set size can be calculated by \( \text{floor}(g \times n^{spp}) \), where \( n \) is the total data size.
min.Obsr decides the minimum number of observations in each groups. Minimum group size can be calculated by \( \text{floor}(n/min.Obsr) \), where \( n \) is the total data size.
adj whether apply finite sample correction. It is recommended to specify this argument to be TRUE to guarantee the correct size of the test.
partition.Method options include:
"xqt" (using the quantiles of the covariates specified in Ctv and the distinct values in Dsv to partition),
"neu_fit" (using the fitted probabilities from neural network based on variables in Ctv and Dsv to partition, requires R package "nnet"),
"rf_fit" (using the fitted probabilities from random forest based on variables in Ctv and Dsv to partition, requires R package "randomForest"),
"p_fit" (using the fitted probabilities from the model to assess to partition, no need to specify Ctv and Dsv in this case).

Value

p.dat 
the single split BAGofT p values from ‘nsplits’ number of splits. It is used to construct the final test statistic ‘test.stat’.

test.stat 
the value of the test statistic. Calculated from the median of ‘p.dat’ from ‘nsplits’ number of splits.

p.value 
the p value of the ‘test.stat’ compared to \( N(0.5, 1/12\text{nsplits}) \). The significance level is 0.05. P value is less than 0.05 when ‘test.stat’ is less than the 0.05 quantile of \( N(0.5, 1/12\text{nsplits}) \).

chisq.dat 
the chi square statistics from ‘nsplits’ number of splits. It is used to construct an alternative final test statistic ‘test.stat3’.

p.value2 
compares the ‘test.stat’ to an alternative distribution \( \text{Beta}((\text{nsplits}+1)/2, (\text{nsplits}+1)/2) \). The significance level is 0.05. P value is less than 0.05 when test.stat is less than the 0.05 quantile of \( \text{Beta}((\text{nsplits} + 1)/2, (\text{nsplits} + 1)/2) \).

test.stat3 
an alternative final test statistic. Calculated by taking the average of the ‘chisq.dat’ from ‘nsplits’ number of splits.

p.value3 
compares ‘test.stat3’ to \( N(g, 2g/\text{nsplits}) \). The significance level is 0.05. P value is less than 0.05 when ‘test.stat3’ is greater than the 0.95 quantile of \( N(g, 2g/\text{nsplits}) \).

maxgpCtList_Sum 
counts number for each covariates in Ctv and Dsv used to partition in the group with the largest contribution in all nsplits number of splits. Available when ‘partition.Method’ = "xqt".

allgpCtList_Sum 
counts number for each covariates in Ctv and Dsv used to partition in all of the groups in all nsplits number of splits. Available when ‘partition.Method’ = "xqt".

singleSplit.results 
a list containing details in each split. To check the elements inside, specify singleSplit.results[[s]]$chisq (chi square value in the s th split), singleSplit.results[[s]]$p.value (p value in the s th split), singleSplit.results[[s]]$ngp (number of groups in the s th split), singleSplit.results[[s]]$leafs (partition selected in the s th split), singleSplit.results[[s]]$contri (contribution of each group in the s th split. We get the unadjusted chi square after summing them up), singleSplit.results[[s]]$maxgp (the group with the largest contribution in the s th split),
singleSplit.results[[s]]$maxgpCt (the count of each variables in Ctv and Dsv used in the group with largest contribution in the s th split), singleSplit.results[[s]]$maxgpCt (the count of each variables in Ctv and Dsv used in all of the groups in the s th split), singleSplit.results[[s]]$maxleaf (details of the group with largest contribution in the s th split).

References


Examples

# A simple example with 3 continuous covariates.
# The logistic regression model used to generate data contains 3 covariates. We consider whether the model "y ~ x1 + x2" fits the data well conditional on x1, x2 and x3.

n <- 500
x1dat <- runif(n, -6, 6)
x2dat <- rnorm(n, 0, sqrt(2.25))
x3dat <- rchisq(n, 4)
lindat <- x1dat * 0.267 + x2dat * 0.267 + x3dat * 0.5
pdat <- 1/(1 + exp(-lindat))
ydat <- sapply(pdat, function(x) rbinom(1, 1, x))
dat <- data.frame(y = ydat, x1 = x1dat, x2 = x2dat, x3 = x3dat)

test1 <- BAGofT(y ~ x1 + x2, data = dat, Ctv = c("x1", "x2", "x3"))

# show the diagnosis. It indicates probably we miss the main effect of x3.
print(test1$maxgpCtList_Sum)

## Not run:

# An example with 6 continuous covariates and 1 discrete covariate. The logistic regression model used to generate data contains a 4th order term of x7.
# We consider whether the model "y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7" fits the data well conditional on all of these covariates.

n <- 500
x1dat <- runif(n, -3, 3)
\begin{verbatim}
x2dat <- runif(n, -3, 3)
x3dat <- rnorm(n, 0, sqrt(2.25))
x4dat <- rnorm(n, 0, sqrt(2.25))
x5dat <- rchisq(n, 8)
x6dat <- rbinom(n, 1, 0.5)
x7dat <- rnorm(n, 0, sqrt(4))

lindat <- x1dat * 0.3 + x2dat * 0.3 + x3dat*0.1 + x4dat*0.2 + x5dat*0.2 + (x6dat-0.5) * 0.3 + x7dat*0.3 + x7dat^4*3

pdat <- 1/(1 + exp(-lindat))
ydat <- sapply(pdat, function(x) rbinom(1, 1, x))

dat <- data.frame(y = ydat, x1 = x1dat, x2 = x2dat, x3 = x3dat, x4 = x4dat, x5 = x5dat, x6 = x6dat, x7 = x7dat)

# BAGofT that generates partitions by quantiles of covariates in Ctv and distinct values in Dsv

test2 <- BAGofT(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = dat, link = "logit",
                  Ctv = c("x1", "x2", "x3", "x4", "x5", "x7"), Dsv = c("x6"),
                  g = 5, nsplits = 100, spp = 1/2.3,
                  min.Obsr = 10, adj = TRUE, partition.Method = "xqt")

# BAGofT that generates partitions by quantiles of fitted probabilities from neural network on
# covariates in Ctv and Dsv

test3 <- BAGofT(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = dat, link = "logit",
                  Ctv = c("x1", "x2", "x3", "x4", "x5", "x7"), Dsv = c("x6"),
                  g = 5, nsplits = 100, spp = 1/2.3,
                  min.Obsr = 10, adj = TRUE, partition.Method = "neu_fit")

# BAGofT that generates partitions by quantiles of fitted probabilities from random forest on
# covariates in Ctv and Dsv

test4 <- BAGofT(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = dat, link = "logit",
                  Ctv = c("x1", "x2", "x3", "x4", "x5", "x7"), Dsv = c("x6"),
                  g = 5, nsplits = 100, spp = 1/2.3,
                  min.Obsr = 10, adj = TRUE, partition.Method = "rf_fit")

# BAGofT that generates partitions by quantiles of fitted probabilities from model to assess
# test5 <- BAGofT(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = dat, link = "logit",
#                 g = 5, nsplits = 100, spp = 1/2.3,
#                 min.Obsr = 10, adj = TRUE, partition.Method = "p_fit")

# print the partition results from test with "xqt" in split 1
print(test1$singleSplit.results[[1]]$leafs)

## End(Not run)
\end{verbatim}
Index

*Topic htest
   BAGofT, 2

BAGofT, 2

formula, 2