Package ‘VLMCX’

February 8, 2024

Type Package

Title Variable Length Markov Chain with Exogenous Covariates

Version 1.0

Date 2024-02-01

Imports graphics, nnet, berryFunctions, stats, utils

Description Models categorical time series through a Markov Chain when a) covariates are predictors for transitioning into the next state/symbol and b) when the dependence in the past states has variable length. The probability of transitioning to the next state in the Markov Chain is defined by a multinomial regression whose parameters depend on the past states of the chain and, moreover, the number of states in the past needed to predict the next state also depends on the observed states themselves. See Zambom, Kim, and Garcia (2022) <doi:10.1111/jtsa.12615>.

License GPL (>= 2)

NeedsCompilation no

Author Adriano Zanin Zambom Developer [aut, cre, cph],
Seonjin Kim Developer [aut],
Nancy Lopes Garcia Developer [aut]

Maintainer Adriano Zanin Zambom Developer <adriano.zambom@gmail.com>

Repository CRAN

Date/Publication 2024-02-08 21:10:07 UTC

R topics documented:

`AIC` *Akaike Information Criteria for* `VLMCX` *objects that compose Variable Length Markov Chains with Exogenous Covariates*

Description

Computes the Akaike Information Criteria for the data using the estimated parameters of the multinomial logistic regression in the VLMCX fit.

Usage

```
AIC(fit)
```

Arguments

`fit`	a betaVLMC object.

Value

a numeric value with the corresponding AIC.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
set.seed(1)
n = 1000
d = 2

X = cbind(rnorm(n), rnorm(n))
p = 1/(1 + exp(0.5 + -2*X[,1] - 3.5*X[,2]))

y = c(sample(1:0,1), rbinom(n,1, p))

fit = maximum.context(y[1:n], X, max.depth = 3, n.min = 25)
draw(fit)
AIC(fit)
##[1] 563.5249

fit = VLMCX(y[1:n], X, alpha.level = 0.001, max.depth = 3, n.min = 25)
draw(fit)
```

```
AIC(fit)
##[1] 559.4967
```

BIC — *Bayesian Information Criteria for for* `VLMCX` *objects that compose Variable Length Markov Chains with Exogenous Covariates*

Description

Computes the Bayesian Information Criteria for the data using the estimated parameters of the multinomial logistic regression in the VLMCX fit.

Usage

```
BIC(fit)
```

Arguments

`fit`	a betaVLMC object.

Value

a numeric value with the corresponding BIC.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
set.seed(1)
n = 1000
d = 2

X = cbind(rnorm(n), rnorm(n))
p = 1/(1 + exp(0.5 + -2*X[,1] - 3.5*X[,2]))

y = c(sample(1:0,1), rbinom(n,1, p))

fit = maximum.context(y[1:n], X, max.depth = 3, n.min = 25)
draw(fit)
BIC(fit)
##[1] 696.0343

fit = VLMCX(y[1:n], X, alpha.level = 0.001, max.depth = 3, n.min = 25)
```

```
draw(fit)
BIC(fit)
##[1] 588.9432
```

coef — *Coefficients from a Variable Length Markov Chain with Exogenous Covariates*

Description

Extracts the estimated coefficients from a VLMCX object for a specific context (sequence of states in the past used to predict the next state/symbol of the chain).

Usage

```
coef(fit, context)
```

Arguments

`fit`	a VLMCX object.
`context`	the context whose coefficients are desired.

Value

an object with two items:

`alpha`	a vector with coefficients corresponding to the intercept for the transition into the states in the state space of `y`.
`beta`	a 3 dimensional-array of estimated coefficients corresponding to [steps in the past, number of covariate, symbol (in the state space) to transition into].

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
set.seed(1)
n = 1000
d = 2

X = cbind(rnorm(n), rnorm(n))

y = rbinom(n,1,.5)
fit = maximum.context(y, X)
```

```
coef(fit, c(0,0,1,0))
## context in the order: y_{t-1} = 0, y_{t-2} = 0, y_{t-3} = 1, y_{t-4} = 0
```

`context.algorithm` *Context Algorithm using exogenous covariates*

Description

Prunes the given tree according to the significance of the covariates and the contexts that are determined by a multinomial regression.

Usage

```
context.algorithm(fit, node, alpha.level = 0.05, max.depth = 5, n.min = 5, trace = FALSE)
```

Arguments

`fit`	a VLMCX object
`node`	The top most node up to which the prunning is allowed.
`alpha.level`	the alpha level for rejection of each hypothesis in the algorithm.
`max.depth`	the maximum depth of the initial "maximal" tree.
`n.min`	minimum number of observations for each parameter needed in the estimation of that context
`trace`	if trace == TRUE then information is printed during the running of the prunning algorithm.

Value

context.algorithm returns an object of class `"VLMCX"`. The generic functions `coef`, `AIC`,`BIC`, `draw`, and `LogLik` extract various useful features of the fitted object returned by *VLMCX*.

An object of class `"VLMCX"` is a list containing at least the following components:

`y`	the time series data corresponding to the states inputed by the user.
`X`	the time series covariates data inputed by the user.
`tree`	the estimated rooted tree estimated by the algorithm. Each node contains the `context`, the intercept (`alpha`) and regression parameters (`beta`) corresponding to the covariates of that regression and a list `child`, whose entries are `nodes` with the same structure.
`LogLik`	the log-likelihood of the data using the estimated context tree.
`baseline.state`	the state used as a baseline fore the multinomial regression.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
n = 500

X = cbind(rnorm(n), rnorm(n))
y = rbinom(n,1,.5)

fit = maximum.context(y, X, max.depth = 3)
pruned.fit = context.algorithm(fit, fit$tree)
draw(pruned.fit)
```

---

draw *Draw the Variable Length Markov Chain estimated model*

Description

Draws the rooted tree corresponding to the estimated contexts in a `VLMCX` object.

Usage

```
draw(fit, title = "VLMCX Context Tree", print.coef = TRUE)
```

Arguments

`fit`	a VLMCX object.
`title`	the title in the graph.
`print.coef`	It TRUE the algorithm prints in the console the list of all contexts and their corresponding alpha and beta coefficients for the multinomial regression. If FALSE, the algorithm prints in the console a text version of the rooted context tree.

Details

The graph contains circles corresponding to the estimated nodes of the contexts estimated by the algorithm but does not include the structure and covariate parameter vectors.

Value

No return value, called for plotting only.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
n = 1000
d = 2
set.seed(1)
X = cbind(rnorm(n), rnorm(n))

y = rbinom(n,1,.2)
fit = maximum.context(y, X)

draw(fit)

fit = VLMCX(y, X, alpha.level = 0.0001, max.depth = 3, n.min = 15, trace = TRUE)
draw(fit)

draw(fit, print.coef = FALSE)
```

`estimate` *Estimation of Variable Length Markov Chain with Exogenous Covariates*

Description

Estimates the parameters of the multinomial logistic model in the VLMCX tree for each context in the tree.

Usage

```
estimate(VLMCXtree, y, X)
```

Arguments

`VLMCXtree`	a VLMCX tree
`y`	a "time series" vector (numeric, charachter, or factor)
`X`	Numeric matrix of predictors with rows corresponding to the `y` observations (over time) and columns corresponding to covariates.

Value

A tree from an object of type VLMCX. The tree contains the items

`context`	the context, or sequence of symbols.
`alpha`	a vector with coefficients corresponding to the intercept for the transition into the states in the state space of `y`.

beta	a 3 dimensional-array of estimated coefficients corresponding to [steps in the past, number of covariate, symbol (in the state space) to transition into].
child	list whose entries are nodes with the same structure.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
set.seed(1)
n = 4000
d = 2

X = cbind(rnorm(n), rnorm(n))
alphabet = 0:2 ### state space
y = sample(alphabet,2, replace = TRUE)
for (i in 3:n)
{
if (identical(as.numeric(y[(i-1):(i-2)]), c(0,0)))
value =  c(exp(-0.5 + -1*X[i-1,1] + 2.5*X[i-1,2]),
           exp(0.5 + -2*X[i-1,1] - 3.5*X[i-1,2]))
else if (identical(as.numeric(y[(i-1):(i-2)]), c(0,1)))
value = c(exp(-0.5),     exp(0.5))
else
value = c(runif(1,0,3), runif(1,0,3))


    prob = c(1,value)/(1 + sum(value)) ## compute probs with baseline state probability
    y[i] = sample(alphabet,1,prob=prob)

}


tree = NULL
tree$context = "x"  ## this is the root
tree$alpha = NULL
tree$beta = NULL
tree$child = list()
this_child = NULL
this_child$context = "0"
this_child$alpha = 0
this_child$child = list()
tree$child[[1]] = this_child

this_grandchild = NULL
this_grandchild$context = c(0, 0)
this_grandchild$alpha = 0
this_grandchild$beta = array(c(0,0,0,0),c(1, 2, 2)) ## steps, d, alphabet (state space)
this_grandchild$child = list()
```

```
tree$child[[1]]$child[[1]] = this_grandchild

this_other_grandchild = NULL
this_other_grandchild$context = c(0, 1)
this_other_grandchild$alpha = 0
this_other_grandchild$beta = NULL
this_other_grandchild$child = list()

tree$child[[1]]$child[[2]] = this_other_grandchild

estimate(tree, y, X)

fit = VLMCX(y, X, alpha.level = 0.0001, max.depth = 2, n.min = 15, trace = TRUE)
estimate(fit$tree, y, X)
```

LogLik *Log Likelihood for Variable Length Markov Chains with Exopgenous Covariates*

Description

Computes the log-likelihood of the data using the estimated parameters of the multinomial logistic regression based on contexts of variable length, that is, a finite suffix of the past, called "context", is used to predict the next symbol, which can have different lengths depending on the past observations themselves.

Usage

```
LogLik(fit)
```

Arguments

fit	a VLMCX object.

Value

a numeric value with the corresponding log-likelihood

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
n = 1000
d = 2

X = cbind(rnorm(n), rnorm(n))

y = rbinom(n,1,.5)
fit = maximum.context(y, X)

LogLik(fit)
```

---

maximum.context *Maximum Context Tree*

Description

Build the largest context tree, which is the biggest context tree such that all elements in it have been observed at least `n.min` times.

Usage

```
maximum.context(y, X, max.depth = 5, n.min = 5)
```

Arguments

`y`	a "time series" vector (numeric, charachter, or factor)
`X`	Numeric matrix of predictors with rows corresponding to the y observations (over time) and columns corresponding to covariates.
`max.depth`	Maximum depth of the desired tree.
`n.min`	Minimum number of observations per coefficient to be estimated.

Value

maximum.context returns an object of class `"VLMCX"`. The generic functions `coef`, `AIC`,`BIC`, `draw`, and `LogLik` extract various useful features of the value returned by *VLMCX*.

An object of class `"VLMCX"` is a list containing at least the following components:

`y`	the time series data corresponding to the states inputed by the user.
`X`	the time series covariates data inputed by the user.
`tree`	the estimated rooted tree estimated by the algorithm. Each node contains the `context`, the intercept (`alpha`) and regression parameters (`beta`) corresponding to the covariates of that regression and a list `child`, whose entries are `nodes` with the same structure.
`LogLik`	the log-likelihood of the data using the estimated context tree.
`baseline.state`	the state used as a baseline fore the multinomial regression.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
n = 1000
d = 2

X = cbind(rnorm(n), rnorm(n))

y = rbinom(n,1,.5)
fit = maximum.context(y, X)
```

---

predict — *Prediction of the next state of the Markov Chain/Categorical Time series*

Description

Uses the estimated coefficients from a VLMCX object to estimate the next state of the Markov Chain either using new data or the original data with which the model was fit.

Usage

```
predict(fit, new.y = NULL, new.X = NULL)
```

Arguments

`fit`	a VLMCX object.
`new.y`	the new sequency of observations of the "time series" as a vector (numeric, charachter, or factor). The values of y.new must be of the same type as the ones used to fit the VLMCX object. If new.y is NULL (or if new.X is NULL) the algorithm uses the original data used to fit the VLMCX object.
`new.X`	Numeric matrix of predictors with rows corresponding to the new.y observations (over time) and columns corresponding to covariates.

Value

a value of the predicted symbol of the next state of the Markoc Chain corresponding to the type of the imput (numeric, charachter, or factor).

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
set.seed(1)
n = 1000

X = cbind(rnorm(n))

y = rbinom(n,1,.5)
fit = maximum.context(y, X)

## using the original data
predict(fit)

## using new data
predict(fit, new.y = c(0,0,1,0,0), new.X = c(2.3, 1.1, -.2, -3,1))
```

---

simulate — *Simulate a Variable Length Markov Chain with Exogenous covariates*

Description

Simulate the states of a Markov Chain based on VLMCX model.

Usage

```
simulate(VLMCXtree, nsim = 500, X = NULL, seed = NULL, n.start = 100)
```

Arguments

`VLMCXtree`	a VLMCX tree (a VLMCX object can also be used, in which case its tree is used).
`nsim`	non-negative integer, giving the length of the result.
`X`	A vector or matrix of exogenous variables. If vector, its length must be equal to nsim+n.start, if matrix, its first dimension must be of length nsim+n.start, if NULL a univariate independent and identically distributed Normal vector is used.
`seed`	random seed initializer.
`n.start`	the number of initial values to be discarded (because of initial effects).

Value

a vector with nsim simulated states.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

Examples

```
tree = NULL
  tree$context = "x"  ## this is the root
  tree$alpha = NULL
  tree$beta = NULL
  tree$child = list()
  this_child = NULL
  this_child$context = "left"
  this_child$alpha = 0.5
  this_child$child = list()
  tree$child[[1]] = this_child

  this_grandchild = NULL
  this_grandchild$context = c("left", "left")
  this_grandchild$alpha = 0.6
  this_grandchild$beta = array(c(1.9, 1.6, 2.6, -1.6),c(2, 2, 1)) ## steps, d, alphabet
  this_grandchild$child = list()

  tree$child[[1]]$child[[1]] = this_grandchild

  this_other_grandchild = NULL
  this_other_grandchild$context = c("left", "right")
  this_other_grandchild$alpha = -0.6
  this_other_grandchild$beta = array(c(-1.3, -1.5, 2.3, -1.2),c(2, 2, 1))
  this_other_grandchild$child = list()

  tree$child[[1]]$child[[2]] = this_other_grandchild

  other_child = NULL
  other_child$context = "right"
  other_child$alpha = -0.7
  other_child$beta = array(c(1,-.3),c(1, 2, 1)) ## steps, d, alphabet
  other_child$child = list()
  tree$child[[2]] = other_child

  set.seed(1)
  X = cbind(rnorm(1100), rnorm(1100))
  simulated.data = simulate(tree, nsim = 1000, X, seed = 1, n.start = 100)
  fit = VLMCX(simulated.data$y, simulated.data$X, alpha.level = 0.001,
               max.depth = 4, n.min = 20, trace = TRUE)
  draw(fit)
  fit
```

VLMCX *Variable Length Markov Chain with Exogenous Covariates*

Description

Estimates a Variable Length Markov Chain model, which can also be seen as a categorical time series model, where exogenous covariates can compose the multinomial regression that predicts the next state/symbol in the chain. This type of approach is a parsimonious model where only a finite suffix of the past, called "context", is enough to predict the next symbol. The length of the each context can differ depending on the past observations themselves.

Usage

```
VLMCX(y, X, alpha.level = 0.05, max.depth = 5, n.min = 5, trace = FALSE)
```

Arguments

`y`	a "time series" vector (numeric, charachter, or factor)
`X`	Numeric matrix of predictors with rows corresponding to the y observations (over time) and columns corresponding to covariates.
`alpha.level`	the alpha level for rejection of each hypothesis in the algorithm.
`max.depth`	the maximum depth of the initial "maximal" tree.
`n.min`	minimum number of observations for each parameter needed in the estimation of that context
`trace`	if trace == TRUE then information is printed during the running of the prunning algorithm.

Details

The algorithm is a backward selection procedure that starts with the maximal context, which is the biggest context tree such that all elements in it have been observed at least `n.min` times. Then, final nodes (past most state in each context) are prunned according to the p-value from the likelihood ratio test for removing the covariates corresponding to that node and the significance of that node itself. The algorithm continues iteratively prunning until nodes cannot be prunned because the covariates or the node context itself is significant.

Value

VLMCX returns an object of class `"VLMCX"`. The generic functions `coef`, `AIC`,`BIC`, `draw`, and `LogLik` extract various useful features of the fitted object returned by *VLMCX*.

An object of class `"VLMCX"` is a list containing at least the following components:

`y`	the time series data corresponding to the states inputed by the user.
`X`	the time series covariates data inputed by the user.
`tree`	the estimated rooted tree estimated by the algorithm. Each node contains the `context`, the intercept (`alpha`) and regression parameters (`beta`) corresponding to the covariates of that regression and a list `child`, whose entries are `nodes` with the same structure.
`LogLik`	the log-likelihood of the data using the estimated context tree.
`baseline.state`	the state used as a baseline fore the multinomial regression.

Author(s)

Adriano Zanin Zambom <adriano.zambom@csun.edu>

References

Zambom, Kim, Garcia (2022) Variable length Markov chain with exogenous covariates. Journal of Time Series Analysis, 43, 321-328.

Examples

```
#### Example 1

set.seed(1)
n = 3000
d = 2

X = cbind(rnorm(n), rnorm(n))
alphabet = 0:2
y = sample(alphabet,2, replace = TRUE)
for (i in 3:n)
{
  if (identical(as.numeric(y[(i-1):(i-2)]), c(0,0)))
    value =  c(exp(-0.5 + -1*X[i-1,1] + 2.5*X[i-1,2]),
           exp(0.5 + -2*X[i-1,1] - 3.5*X[i-1,2]))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(0,1)))
    value = c(exp(-0.5),     exp(0.5))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(0,2)))
    value = c(exp(1),     exp(1))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(2,0)))
    value = c(exp(0.5 + 1.2*X[i-1,1] + 0.5*X[i-1,2] + 2*X[i-2,1] + 1.5*X[i-2,2]),
              exp(-0.5  -2*X[i-1,1] - .5*X[i-1,2] +1.3*X[i-2,1] + 1.5*X[i-2,2]))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(2,1)))
    value  = c(exp(-1 + -X[i-1,1] + 2.5*X[i-1,2]),
               exp(0.1 + -0.5*X[i-1,1] - 1.5*X[i-1,2]))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(2,2)))
    value = c(exp(-0.5 + -X[i-1,1] - 2.5*X[i-1,2]),
              exp(0.5 + -2*X[i-1,1] - 3.5*X[i-1,2]))
  else
    value = c(runif(1,0,3), runif(1,0,3))


    prob = c(1,value)/(1 + sum(value)) ## compute probs with baseline state probability
    y[i] = sample(alphabet,1,prob=prob)

}

fit = VLMCX(y, X, alpha.level = 0.001, max.depth = 4, n.min = 15, trace = TRUE)
draw(fit)
```

```
## Note the only context that was estimated but not in the true
## model is (1): removing it or not does not change the likelihood,
## so the algorithm keeps it.
coef(fit, c(0,2))

predict(fit,new.y = c(0,0), new.X = matrix(c(1,1,1,1), nrow=2))
#[1] 0.2259747309 0.7738175143 0.0002077548

predict(fit,new.y = c(0,0,0), new.X = matrix(c(1,1,1,1,1,1), nrow=3))
# [1] 0.2259747309 0.7738175143 0.0002077548


#### Example 2

set.seed(1)
n = 2000
d = 1

X = rnorm(n)
alphabet = 0:1
y = sample(alphabet,2, replace = TRUE)
for (i in 3:n)
{
  if (identical(as.numeric(y[(i-1):(i-3)]), c(0,0, 0)))
    value =  c(exp(-0.5 -1*X[i-1]  + 2*X[i-2]))
  else if (identical(as.numeric(y[(i-1):(i-3)]), c(0, 0, 1)))
    value = c(exp(-0.5))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(1,0)))
    value = c(exp(0.5 + 1.2*X[i-1] + 2*X[i-2] ))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(1,1)))
    value  = c(exp(-1 + -X[i-1] +2*X[i-2]))
  else
    value = c(runif(1,0,3))

    prob = c(1,value)/(1 + sum(value)) ## compute probs with baseline state probability
    y[i] = sample(alphabet,1,prob=prob)
}
fit = VLMCX(y, X, alpha.level = 0.001, max.depth = 4, n.min = 15, trace = TRUE)
draw(fit)
coef(fit, c(1,0))



#### Example 3

set.seed(1)
n = 4000
d = 1

X = cbind(rnorm(n))
alphabet = 0:3
y = sample(alphabet,2, replace = TRUE)
for (i in 3:n)
```

```
{
  if (identical(as.numeric(y[(i-1):(i-2)]), c(3, 3)))
    value =  c(exp(-0.5 -1*X[i-1] + 2.5*X[i-2]),
           exp(0.5 -2*X[i-1] - 3.5*X[i-2]),
           exp(0.5 +2*X[i-1] + 3.5*X[i-2]))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(3, 1)))
    value = c(exp(-0.5 + X[i-1]),
              exp(0.5 -1.4*X[i-1]),
              exp(0.9 +1.4*X[i-1]))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(1, 0)))
    value = c(exp(-.5),
              exp(.5),
              exp(.8))
  else if (identical(as.numeric(y[(i-1):(i-2)]), c(1, 2)))
    value = c(exp(.4),
              exp(-.5),
              exp(.8))
  else
    value = c(runif(1,0,3), runif(1,0,3), runif(1,0,3))


    prob = c(1,value)/(1 + sum(value)) ## compute probs with baseline state probability
    y[i] = sample(alphabet,1,prob=prob)

}

fit = VLMCX(y, X, alpha.level = 0.00001, max.depth = 3, n.min = 15, trace = TRUE)
## The context (0, 1) was not identified because the
draw(fit)
coef(fit, c(3,1))
```

Index