Package ‘ordinalbayes’

October 14, 2022

Title Bayesian Ordinal Regression for High-Dimensional Data

Version 0.1.1

Date 2022-03-30

Description Provides a function for fitting various penalized Bayesian cumulative link ordinal response models when the number of parameters exceeds the sample size. These models have been described in Zhang and Archer (2021) <doi:10.1186/s12859-021-04432-w>.

License MIT + file LICENSE

Depends DESeq2, R (>= 2.10), SummarizedExperiment

Imports coda, devtools, dclone, runjags

Suggests knitr, Biobase, testthat (>= 3.0.0), rmarkdown

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

SystemRequirements JAGS (>= 4.0.0)

VignetteBuilder knitr

URL `https://github.com/kelliejarcher/ordinalbayes`

BugReports `https://github.com/kelliejarcher/ordinalbayes/issues`

NeedsCompilation no

Author Kellie J. Archer [aut, cre] (<`https://orcid.org/0000-0003-1555-5781`>),
Anna Seffernick [ctb],
Shuai Sun [ctb],
Yiran Zhang [aut]

Maintainer Kellie J. Archer <`archer.43@osu.edu`>

Repository CRAN

Date/Publication 2022-04-06 15:10:02 UTC

R topics documented:

`cesc` *Data Frame of Smaller Subset of The Cancer Genome Atlas Cervical Cancer HTSeq Data.*

Description

This data.frame is a subset that includes 41 transcripts and four phenotypic variables for 242 subjects with squamous cell cervical cancer.

Usage

```
cesc
```

Format

This `data.frame` includes the following variables:

age_at_index a numeric vector storing age at cervical cancer diagnosis.

cigarettes_per_day a numeric vector storing the number of cigarettes smoked per day.

race a character vector representing race of the subject.

Stage an ordered factor representing stage of cervical cancer.

ENSG00000076344 a numeric vector representing expression of ENSG00000076344.

ENSG00000077274 a numeric vector representing expression of ENSG00000077274.

ENSG00000101888 a numeric vector representing expression of ENSG00000101888.

ENSG00000115548 a numeric vector representing expression of ENSG00000115548.

ENSG00000122884 a numeric vector representing expression of ENSG00000122884.

ENSG00000125430 a numeric vector representing expression of ENSG00000125430.

ENSG00000131370 a numeric vector representing expression of ENSG00000131370.

ENSG00000135443 a numeric vector representing expression of ENSG00000135443.

ENSG00000136457 a numeric vector representing expression of ENSG00000136457.

ENSG00000138398 a numeric vector representing expression of ENSG00000138398.

ENSG00000150636	a numeric vector representing expression of ENSG00000150636.
ENSG00000161277	a numeric vector representing expression of ENSG00000161277.
ENSG00000163510	a numeric vector representing expression of ENSG00000163510.
ENSG00000164485	a numeric vector representing expression of ENSG00000164485.
ENSG00000164651	a numeric vector representing expression of ENSG00000164651.
ENSG00000166091	a numeric vector representing expression of ENSG00000166091.
ENSG00000166342	a numeric vector representing expression of ENSG00000166342.
ENSG00000171121	a numeric vector representing expression of ENSG00000171121.
ENSG00000177173	a numeric vector representing expression of ENSG00000177173.
ENSG00000180229	a numeric vector representing expression of ENSG00000180229.
ENSG00000188817	a numeric vector representing expression of ENSG00000188817.
ENSG00000197360	a numeric vector representing expression of ENSG00000197360.
ENSG00000203601	a numeric vector representing expression of ENSG00000203601.
ENSG00000225449	a numeric vector representing expression of ENSG00000225449.
ENSG00000230201	a numeric vector representing expression of ENSG00000230201.
ENSG00000233996	a numeric vector representing expression of ENSG00000233996.
ENSG00000236138	a numeric vector representing expression of ENSG00000236138.
ENSG00000236819	a numeric vector representing expression of ENSG00000236819.
ENSG00000250602	a numeric vector representing expression of ENSG00000250602.
ENSG00000253923	a numeric vector representing expression of ENSG00000253923.
ENSG00000256980	a numeric vector representing expression of ENSG00000256980.
ENSG00000259083	a numeric vector representing expression of ENSG00000259083.
ENSG00000259134	a numeric vector representing expression of ENSG00000259134.
ENSG00000260484	a numeric vector representing expression of ENSG00000260484.
ENSG00000263612	a numeric vector representing expression of ENSG00000263612.
ENSG00000264049	a numeric vector representing expression of ENSG00000264049.
ENSG00000264954	a numeric vector representing expression of ENSG00000264954.
ENSG00000265579	a numeric vector representing expression of ENSG00000265579.
ENSG00000271711	a numeric vector representing expression of ENSG00000271711.
ENSG00000272071	a numeric vector representing expression of ENSG00000272071.
ENSG00000276517	a numeric vector representing expression of ENSG00000276517.

Source

TCGAbiolinks TCGA-CESC

Examples

```
data("cesc")
head(cesc)
```

`coef.ordinalbayes` *Extract Model Coefficients*

Description

Extract Model Coefficients

Usage

```
## S3 method for class 'ordinalbayes'
coef(object, method = mean, ...)
```

Arguments

`object`	an `ordinalbayes` object.
`method`	The default is `method=mean` which estimates the mean of each parameter in the MCMC chain. Other options are `method=median` or any other relevant summary function.
`...`	other arguements.

Value

`alpha`	Summary estimates for the thresholds
`zeta`	Summary estimates for the unpenalized covariates. Only available if unpenalized covariates were included in the fitted model.
`beta`	Summary estimates for the penalized covariates
`gamma`	Summary estimates for the variable inclusion indicators. Not available when `model="lasso"`

See Also

`ordinalbayes`, `print.ordinalbayes`, `summary.ordinalbayes`, `predict.ordinalbayes`

Examples

```

data("cesc")
fit<-ordinalbayes(Stage~1, data=cesc, x=cesc[,5:45],
         model="regressvi", gamma.ind="fixed", pi.fixed=0.99,
         adaptSteps=1000, burnInSteps=1000, nChains=2,
         numSavedSteps=2000, thinSteps=2, seed=26)
coef(fit)
```

`finalSet` *Subset of The Cancer Genome Atlas Cervical Cancer HTSeq Data.*

Description

This subset includes 2,009 transcripts and four phenotypic variables for 242 subjects with squamous cell cervical cancer.

Usage

```
finalSet
```

Format

The format is of formal class 'DESeqTransform' (package "DESeq2") with the following extractor functions:

assay includes rld transformed transcript expression data

colData includes phenotypic data: `age_at_index`, `cigarettes_per_day`, `race`, and `Stage`

Source

TCGAbiolinks TCGA-CESC

`ordinalbayes` *Ordinal Bayesian Regression Models for High-Dimensional Data*

Description

Ordinal Bayesian Regression Models for High-Dimensional Data

Usage

```
ordinalbayes(
  formula,
  data,
  x = NULL,
  subset,
  center = TRUE,
  scale = TRUE,
  a = 0.1,
  b = 0.1,
  model = "regressvi",
  gamma.ind = "fixed",
  pi.fixed = 0.05,
  c.gamma = NULL,
```

```
  d.gamma = NULL,
  alpha.var = 10,
  sigma2.0 = NULL,
  sigma2.1 = NULL,
  coerce.var = 10,
  lambda0 = NULL,
  nChains = 3,
  adaptSteps = 5000,
  burnInSteps = 5000,
  numSavedSteps = 9999,
  thinSteps = 3,
  parallel = TRUE,
  seed = NULL,
  quiet = FALSE
)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The left side of the formula is the ordinal outcome while the variables on the right side of the formula are the covariates that are not included in the penalization process. Note that if all variables in the model are to be penalized, an intercept only model formula should be specified.
`data`	an optional data.frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model.
`x`	an optional matrix of predictors that are to be penalized in the model fitting process.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.
`center`	logical, if TRUE the penalized predictors are centered.
`scale`	logical, if TRUE the penalized predictors are scaled.
`a`	hyperprior for the penalty parameter lambda which is Gamma with parameters `a` and `b`.
`b`	hyperprior for the penalty parameter lambda which is Gamma with parameters `a` and `b`.
`model`	Specify which penalized ordinal model to fit as "regressvi", "lasso", "dess", or "normalss".
`gamma.ind`	indicates whether prior for the variable inclusion indicators is "fixed" or "random" (for models "regressvi", "dess", or "normalss").
`pi.fixed`	constant prior for the variable inclusion indicators is when `gamma.ind="fixed"`.
`c.gamma`	hyperprior for the variable inclusion indicators is when `gamma.ind="random"`.
`d.gamma`	hyperprior for the variable inclusion indicators is when `gamma.ind="random"`.
`alpha.var`	variance for alpha_k thresholds in the MCMC chain (default 10).

`sigma2.0`	variance for the spike when `model="normalss"` (set to some small positive value).
`sigma2.1`	variance for the slab when `model="normalss"` (set to some large positive value).
`coerce.var`	variance associated with any unpenalized predictors in the MCMC chain (default 10).
`lambda0`	parameter value for the spike when `model="dess"`.
`nChains`	number of parallel chains to run (default 3)
`adaptSteps`	number of iterations for adaptation (default 5,000).
`burnInSteps`	number of iterations of the Markov chain to run (default 5,000).
`numSavedSteps`	number of saved steps per chain (default 9,999).
`thinSteps`	thinning interval for monitors (default 3).
`parallel`	logical, run the MCMC on multiple processors (default TRUE).
`seed`	integer, seed to ensure reproducibility.
`quiet`	logical, when TRUE, suppress output of JAGS (or rjags) when updating models

Value

`results` An object of class runjags

`call` Model call

`model` Name of the ordinal model that was fit

`a` Value the user specified for `a`

`b` Value the user specified for `b`

`featureNames` Names of the penalized predictors

`center` Value the user specified for `center`

`scale` Value the user specified for `scale`

`y` Observed ordinal response

`x` Matrix of penalized predictors used in model fitting

`w` Matrix of unpenalized predictors used in model fitting

`gamma.ind` Value the user specified for `gamma.ind`

`pi.fixed` Value the user specified for `pi.fixed` if `gamma.ind="fixed"`

`c.gamma` Value the user specified for `c.gamma` if `gamma.ind="random"`

`d.gamma` Value the user specified for `d.gamma` if `gamma.ind="random"`

`sigma2.0` Value the user specified for `sigma2.0` if `model="normalss"`

`sigma2.1` Value the user specified for `sigma2.1` if `model="normalss"`

`lambda0` value the user specified for `lambda0` if `model="dess"`

See Also

`print.ordinalbayes`, `summary.ordinalbayes`, `coef.ordinalbayes`

Examples

```
# The number of adaptSteps, burnInSteps, and numSavedSteps was reduced for package testing
data("cesc")
data(reducedSet)
fit<-ordinalbayes(Stage~1, data=cesc, x=cesc[,5:45],
         model="regressvi", gamma.ind="fixed", pi.fixed=0.99,
         adaptSteps=1000, burnInSteps=1000, nChains=2,
         numSavedSteps=2000, thinSteps=2, seed=26)
```

`plot.ordinalbayes` *Trace Plot and/or Density Plot of MCMC Output.*

Description

Trace Plot and/or Density Plot of MCMC Output.

Usage

```
## S3 method for class 'ordinalbayes'
plot(x, trace = TRUE, density = FALSE, ...)
```

Arguments

`x`	an `ordinalBayes` object.
`trace`	a logical value. If TRUE, trace plots are produced for each variable in the chain.
`density`	a logical value. If TRUE, density plots are produced for each variable in the chain.
`...`	other arguments.

Value

No returned value, called for side effects

Examples

```
data("cesc")
fit<-ordinalbayes(Stage~1, data=cesc, x=cesc[,5:45],
     model="regressvi",gamma.ind="fixed",
     pi.fixed=0.99, adaptSteps=1000, burnInSteps=1000, nChains=2,
     numSavedSteps=2000, thinSteps=2, seed=26)
 plot(fit)
```

`predict.ordinalbayes` *Predicted Probabilities and Class for an Ordinal Bayes Fit.*

Description

Predicted Probabilities and Class for an Ordinal Bayes Fit.

Usage

```
## S3 method for class 'ordinalbayes'
predict(
  object,
  neww = NULL,
  newdata,
  newx = NULL,
  model.select = "average",
  ...
)
```

Arguments

`object`	an `ordinalBayes` fitted object.
`neww`	an optional formula that includes the unpenalized variables to use for predicting the response. If omitted, the training data are used.
`newdata`	an optional data.frame that minimally includes the unpenalized variables to use for predicting the response. If omitted, the training data are used.
`newx`	an optional matrix of penalized variables to use for predicting the response. If omitted, the training data are used.
`model.select`	when `"average"` (default) is used, the mean coefficient values over the MCMC chain are used to estimate fitted probabilities; when `"median"` is used, the median coefficient values over the MCMC chain are used to estimate fitted probabilities; when `"max.predicted.class"` is used, each step in the chain is used to calculate fitted probabilities and the class. The predicted class is that attaining the maximum fitted probability.
`...`	other arguments.

Value

`predicted`	a matrix of predicted probabilities from the fitted model.
`class`	a vector containing the predicted class taken as that class having the largest predicted probability.

See Also

`ordinalbayes`, `coef.ordinalbayes`, `summary.ordinalbayes`, `print.ordinalbayes`

Examples

```
data("cesc")
fit<-ordinalbayes(Stage~1, data=cesc, x=cesc[,5:45],,
     model="regressvi", gamma.ind="fixed", pi.fixed=0.99,
     adaptSteps=1000, burnInSteps=1000, nChains=2, numSavedSteps=2000,
     thinSteps=2, seed=26)
phat<-predict(fit)
table(phat$class, cesc$Stage)
```

`print.ordinalbayes` *Print MCMC Summary Statistics*

Description

Print MCMC Summary Statistics

Usage

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

Arguments

`x`	A fitted ordinalbayes object.
`...`	other arguments.

Value

Matrix with the summaries from the `x$results` object which of class 'runjags'. Columns include Lower95, Median, Upper95, Mean, SD, Mode, MCerr, MC%ofSD, SSeff, AC.20, and psrf

See Also

`ordinalbayes`, `summary.ordinalbayes`, `coef.ordinalbayes`, `predict.ordinalbayes`

Examples

```
data("cesc")
fit<-ordinalbayes(Stage~1, data=cesc, x=cesc[,5:45],
model="regressvi", gamma.ind="fixed", pi.fixed=0.99, adaptSteps=1000,
burnInSteps=1000, nChains=2, numSavedSteps=2000, thinSteps=2)
print(fit)
```

`reducedSet` *Smaller Subset of The Cancer Genome Atlas Cervical Cancer HTSeq Data.*

Description

This subset includes 41 transcripts and four phenotypic variables for 242 subjects with squamous cell cervical cancer.

Usage

```
reducedSet
```

Format

The format is of formal class 'DESeqTransform' (package "DESeq2") with the following extractor functions:

assay includes rld transformed transcript expression data

colData includes phenotypic data: `age_at_index`, `cigarettes_per_day`, `race`, and `Stage`

Source

TCGAbiolinks TCGA-CESC

`summary.ordinalbayes` *Summarize an Ordinal Bayes Object.*

Description

Summarize an Ordinal Bayes Object.

Usage

```
## S3 method for class 'ordinalbayes'
summary(object, epsilon = 0.1, ...)
```

Arguments

`object`	A fitted `ordinalbayes` object.
`epsilon`	a small positive value that is close to 0 for testing an interval null hypothesis for the beta parameters.
`...`	other arguments.

Value

`alphamatrix`	The MCMC output for the threshold parameters.
`betamatrix`	The MCMC output for the penalized parameters.
`zetamatrix`	The MCMC output for the unpenalized parameters (if included).
`gammamatrix`	The MCMC output for the variable inclusion parameters (not available for lasso).
`gammamean`	The posterior mean of the variable inclusion indicators (not available for lasso)for the variable inclusion indicators (not available for lasso).
`gamma.BayesFactor`	Bayes factor for the variable inclusion indicators (not available for lasso).
`Beta.BayesFactor`	Bayes factor for the penalized parameters where the interval null is tested using epsilon.
`lambdamatrix`	The MCMC output for the penalty parameter (not available for normalss).

See Also

`ordinalbayes`, `print.ordinalbayes`, `coef.ordinalbayes`, `predict.ordinalbayes`

Examples

```
data("cesc")
fit<-ordinalbayes(Stage~1, data=cesc, x=cesc[,5:45],
         model="regressvi", gamma.ind="fixed", pi.fixed=0.99,
         adaptSteps=1000, burnInSteps=1000, nChains=2,
         numSavedSteps=2000, thinSteps=2, seed=26)
summary.fit<-summary(fit)
names(summary.fit)
names(which(summary.fit$Beta.BayesFactor>5))
names(which(summary.fit$gamma.BayesFactor>5))
```

Index