Package ‘UHM’

January 20, 2025

Type Package

Title Unified Zero-Inflated Hurdle Regression Models

Version 0.3.0

Maintainer Taban Baghfalaki <t.baghfalaki@gmail.com>

URL https://github.com/tbaghfalaki/UHM

Description Run a Gibbs sampler for hurdle models to analyze data showing an excess of zeros, which is common in zero-inflated count and semi-continuous models. The package includes the hurdle model under Gaussian, Gamma, inverse Gaussian, Weibull, Exponential, Beta, Poisson, negative binomial, logarithmic, Bell, generalized Poisson, and binomial distributional assumptions. The models described in Ganjali et al. (2024).

License GPL (>= 2.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.0

Imports stats, jagsUI, numbers

Depends R (>= 4.0.0)

SystemRequirements JAGS 4.x.y

NeedsCompilation no

Author Taban Baghfalaki [cre, aut] (<https://orcid.org/0000-0002-2100-4532>),
Mojtaba Ganjali [aut] (<https://orcid.org/0000-0002-8574-1750>),
Narayanaswamy Balakrishnan [aut]

Repository CRAN

Date/Publication 2024-03-08 21:20:02 UTC

Contents

`dataB` *Simulated data from zero-inflated Beta regression model*

Description

Simulated data was generated with x1 following a Bernoulli distribution with a success probability of 0.4, x2 following a standard normal distribution, and y following a zero-inflated Beta regression model.

Usage

```
dataB
```

Format

A data frame which contains x1, x2 and y.

y the response variable

x1 Binary covariate

x2 Continuous covariate

See Also

`UHM`,`ZIHR`

`dataC` *Simulated data from zero-inflated Gaussian regression model*

Description

Simulated data was generated with x1 following a Bernoulli distribution with a success probability of 0.4, x2 following a standard normal distribution, and y following a zero-inflated Gaussian regression model.

Usage

```
dataC
```

Format

A data frame which contains x1, x2 and y.

y the response variable

x1 Binary covariate

x2 Continuous covariate

See Also

UHM,ZIHR

`dataD` *Simulated data from zero-inflated Poisson regression model*

Description

Simulated data was generated where x1 follows a Bernoulli distribution with a success probability of 0.2, x2 follows a standard normal distribution, and y follows a zero-inflated Poisson regression model.

Usage

```
dataD
```

Format

A data frame which contains x1, x2 and y.

y the response variable

x1 Binary covariate

x2 Continuous covariate

See Also

UHM,ZIHR

dataI *Simulated data from zero-inflated exponential regression model*

Description

Simulated data was generated with x1 following a Bernoulli distribution with a success probability of 0.4, x2 following a standard normal distribution, and y following a zero-inflated inverse Gaussian regression model.

Usage

```
dataI
```

Format

A data frame which contains x1, x2 and y.

y the response variable

x1 Binary covariate

x2 Continuous covariate

See Also

UHM,ZIHR

dataP *Simulated data from zero-inflated exponential regression model*

Description

Simulated data was generated with x1 following a Bernoulli distribution with a success probability of 0.4, x2 following a standard normal distribution, and y following a zero-inflated exponential regression model.

Usage

```
dataP
```

Format

A data frame which contains x1, x2 and y.

y the response variable

x1 Binary covariate

x2 Continuous covariate

See Also

UHM,ZIHR

Prediction *Prediction of new observations*

Description

Computing a prediction for new observations

Usage

```
Prediction(object, data)
```

Arguments

object	an object inheriting from class ZIHR
data	dataset of observed variables with the same format as the data in the object

Details

It provides a summary of the output of the ZIHR function, including parameter estimations.

Value

Estimation, standard errors and 95% credible intervals for predictions

Author(s)

Taban Baghfalaki <t.baghfalaki@gmail.com>, Mojtaba Ganjali <m-ganjali@sbu.ac.ir>

See Also

ZIHR

Examples

```
# Example 1
data(dataD)
index <- 1:(dim(dataD)[1])
IND_new <- sample(index, .5 * length(index))
datat <- dataD[IND_new, ]
datav <- dataD[-IND_new, ]
modelY <- y~x1 + x2
modelZ <- z~x1
D1 <- ZIHR(modelY, modelZ,
           data = datat, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Poisson"
```

```
)


SummaryZIHR(D1)
Prediction(D1, data = datav)


D2 <- ZIHR(modelY, modelZ,
           data = datat, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Bell"
)
SummaryZIHR(D2)



# Example 2
data(dataC)
modelY <- y~x1 + x2
modelZ <- z~x1
C <- ZIHR(modelY, modelZ,
          data = dataC, n.chains = 2, n.iter = 1000,
          n.burnin = 500, n.thin = 1, family = "Gaussian"
)
SummaryZIHR(C)

Prediction(C, data = datav)



# Example 3
data(dataP)
modelY <- y~x1 + x2
modelZ <- z~x1
P1 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Exponential"
)
SummaryZIHR(P1)

P2 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Gamma"
)
SummaryZIHR(P2)

P3 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Weibull"
)
SummaryZIHR(P3)


# Example B
```

```
data(dataB)
modelY <- y~x1 + x2
modelZ <- z~x1
P <- ZIHR(modelY, modelZ,
          data = dataB, n.chains = 2, n.iter = 1000,
          n.burnin = 500, n.thin = 1, family = "Beta"
)
SummaryZIHR(P)

# Example C
data(dataI)
modelY <- y~x1 + x2
modelZ <- z~x1
P4 <- ZIHR(modelY, modelZ,
           data = dataI, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "inverse.gaussian"
)
SummaryZIHR(P4)
```

SummaryZIHR *Summary of ZIHR*

Description

Computing a summary of the outputs of the ZIHR function

Usage

```
SummaryZIHR(object)
```

Arguments

object	an object inheriting from class ZIHR

Details

It provides a summary of the output of the ZIHR function, including parameter estimations.

Value

Estimation list of posterior summary includes estimation, standard deviation, lower and upper bounds for 95% credible intervals, and Rhat (when n.chain > 1). DIC deviance information criterion LPML Log Pseudo Marginal Likelihood (LPML) criterion

Author(s)

Taban Baghfalaki <t.baghfalaki@gmail.com>, Mojtaba Ganjali <m-ganjali@sbu.ac.ir>

See Also

ZIHR

Examples

```
# Example 1
data(dataD)
index <- 1:(dim(dataD)[1])
IND_new <- sample(index, .5 * length(index))
datat <- dataD[IND_new, ]
datav <- dataD[-IND_new, ]
modelY <- y~x1 + x2
modelZ <- z~x1
D1 <- ZIHR(modelY, modelZ,
           data = datat, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Poisson"
)


  SummaryZIHR(D1)
  Prediction(D1, data = datav)


  D2 <- ZIHR(modelY, modelZ,
             data = datat, n.chains = 2, n.iter = 1000,
             n.burnin = 500, n.thin = 1, family = "Bell"
  )
  SummaryZIHR(D2)



  # Example 2
  data(dataC)
  modelY <- y~x1 + x2
  modelZ <- z~x1
  C <- ZIHR(modelY, modelZ,
            data = dataC, n.chains = 2, n.iter = 1000,
            n.burnin = 500, n.thin = 1, family = "Gaussian"
  )
  SummaryZIHR(C)

  Prediction(C, data = datav)



  # Example 3
  data(dataP)
  modelY <- y~x1 + x2
  modelZ <- z~x1
  P1 <- ZIHR(modelY, modelZ,
             data = dataP, n.chains = 2, n.iter = 1000,
             n.burnin = 500, n.thin = 1, family = "Exponential"
```

```
)
SummaryZIHR(P1)

P2 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Gamma"
)
SummaryZIHR(P2)

P3 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Weibull"
)
SummaryZIHR(P3)


# Example B
data(dataB)
modelY <- y~x1 + x2
modelZ <- z~x1
P <- ZIHR(modelY, modelZ,
          data = dataB, n.chains = 2, n.iter = 1000,
          n.burnin = 500, n.thin = 1, family = "Beta"
)
SummaryZIHR(P)

# Example C
data(dataI)
modelY <- y~x1 + x2
modelZ <- z~x1
P4 <- ZIHR(modelY, modelZ,
           data = dataI, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "inverse.gaussian"
)
SummaryZIHR(P4)
```

`UHM` *UHM Package*

Description

Run a Gibbs sampler for hurdle models. The package includes the hurdle generalized linear model under Gaussian, exponential, Gamma, Weibull, inverse Gaussian, Poisson, negative binomial, logarithmic, logistic, and binomial distributional assumptions. The package also considers hurdle generalized Poisson models and hurdle Beta regression models. For model comparison, Deviance Information Criterion (DIC) and Log Pseudo Marginal Likelihood (LPML) are presented.

Author(s)

Taban Baghfalaki <t.baghfalaki@gmail.com>, Mojtaba Ganjali <m-ganjali@sbu.ac.ir>, Narayanaswamy Balakrishnan <bala@mcmaster.ca>

References

1. Ganjali, M., Baghfalaki, T. & Balakrishnan, N. (2024). A Unified Bayesian approach for Modeling Zero-Inflated count and continuous outcomes.

See Also

Useful links:

- https://github.com/tbaghfalaki/UHM

ZIHR *Zero-inflation hurdle regression models*

Description

Fits zero-inflated hurdle regression models

Usage

```
ZIHR(
  modelY,
  modelZ,
  data,
  n.chains = n.chains,
  n.iter = n.iter,
  n.burnin = n.burnin,
  n.thin = n.thin,
  family = "Gaussian"
)
```

Arguments

modelY	a formula for the mean of the count response. This argument is identical to the one in the "glm" function.
modelZ	a formula for the probability of zero. This argument is identical to the one in the "glm" function.
data	data set of observed variables.
n.chains	the number of parallel chains for the model; default is 1.
n.iter	integer specifying the total number of iterations; default is 1000.
n.burnin	integer specifying how many of n.iter to discard as burn-in ; default is 5000.
n.thin	integer specifying the thinning of the chains; default is 1.

`family` Family objects streamline the specification of model details for functions like glm. They cover various distributions like "Gaussian", "Exponential", "Weibull", "Gamma", "Beta", "inverse.gaussian", "Poisson", "NB", "Logarithmic", "Bell", "GP", and "Binomial". Specifically, "NB" and "GP" are tailored for hurdle negative binomial and hurdle generalized Poisson models, respectively, while the others are utilized for the corresponding models based on their names.

Details

A function utilizing the 'JAGS' software to estimate the linear hurdle regression model.

Value

- MCMC chains for the unknown parameters
- Est list of posterior mean for each parameter
- SD list of standard error for each parameter
- L_CI list of 2.5th percentiles of the posterior distribution serves as the lower bound of the Bayesian credible interval
- U_CI list of 97.5th percentiles of the posterior distribution serves as the lower bound of the Bayesian credible interval
- Rhat Gelman and Rubin diagnostic for all parameter
- beta the regression coefficients of mean of the hurdle model
- alpha the regression coefficients of probability of the hurdle model
- The variance, over-dispersion, dispersion, or scale parameters of models depend on the family used
- DIC deviance information criterion
- LPML Log Pseudo Marginal Likelihood (LPML) criterion

Author(s)

Taban Baghfalaki <t.baghfalaki@gmail.com>, Mojtaba Ganjali <m-ganjali@sbu.ac.ir>

Examples

```
# Example 1
data(dataD)
index <- 1:(dim(dataD)[1])
IND_new <- sample(index, .5 * length(index))
datat <- dataD[IND_new, ]
datav <- dataD[-IND_new, ]
modelY <- y~x1 + x2
modelZ <- z~x1
D1 <- ZIHR(modelY, modelZ,
           data = datat, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Poisson"
)
```

```
SummaryZIHR(D1)
Prediction(D1, data = datav)


D2 <- ZIHR(modelY, modelZ,
           data = datat, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Bell"
)
SummaryZIHR(D2)



# Example 2
data(dataC)
modelY <- y~x1 + x2
modelZ <- z~x1
C <- ZIHR(modelY, modelZ,
          data = dataC, n.chains = 2, n.iter = 1000,
          n.burnin = 500, n.thin = 1, family = "Gaussian"
)
SummaryZIHR(C)

Prediction(C, data = datav)



# Example 3
data(dataP)
modelY <- y~x1 + x2
modelZ <- z~x1
P1 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Exponential"
)
SummaryZIHR(P1)

P2 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Gamma"
)
SummaryZIHR(P2)

P3 <- ZIHR(modelY, modelZ,
           data = dataP, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "Weibull"
)
SummaryZIHR(P3)


# Example B
data(dataB)
modelY <- y~x1 + x2
```

```
modelZ <- z~x1
P <- ZIHR(modelY, modelZ,
          data = dataB, n.chains = 2, n.iter = 1000,
          n.burnin = 500, n.thin = 1, family = "Beta"
)
SummaryZIHR(P)

# Example C
data(dataI)
modelY <- y~x1 + x2
modelZ <- z~x1
P4 <- ZIHR(modelY, modelZ,
           data = dataI, n.chains = 2, n.iter = 1000,
           n.burnin = 500, n.thin = 1, family = "inverse.gaussian"
)
SummaryZIHR(P4)
```

Index