Package ‘AsynchLong’

November 24, 2023

Type Package

Title Regression Analysis of Sparse Asynchronous Longitudinal Data

Version 2.3

Date 2023-11-24

Author Hongyuan Cao, Donglin Zeng, Jialiang Li, Jason P. Fine, and Shannon T. Holloway

Maintainer Shannon T. Holloway <shannon.t.holloway@gmail.com>

Description Estimation of regression models for sparse asynchronous longitudinal observations, where time-dependent response and covariates are mismatched and observed intermittently within subjects. Kernel weighted estimating equations are used for generalized linear models with either time-invariant or time-dependent coefficients. Cao, H., Li, J., and Fine, J. P. (2016) <doi:10.1214/16-EJS1141>. Cao, H., Zeng, D., and Fine, J. P. (2015) <doi:10.1111/rssb.12086>.

Depends compiler, parallel, stats, graphics, methods

License GPL-2

NeedsCompilation no

RoxygenNote 7.2.1

Repository CRAN

Date/Publication 2023-11-24 16:40:02 UTC

R topics documented:

`AsynchLong-package` *Regression Analysis of Sparse Asynchronous Longitudinal Data*

Description

Estimation of regression models for sparse asynchronous longitudinal observations, where time-dependent response and covariates are mismatched and observed intermittently within subjects. Kernel weighted estimating equations are used for generalized linear models with either time-invariant or time-dependent coefficients.

Details

Package:	AsynchLong
Type:	Package
Version:	2.2
Date:	2022-06-05
License:	GPL-2

Author(s)

Hongyuan Cao, Jason P. Fine, Jialiang Li, Donglin Zeng, and Shannon T. Holloway Maintainer: Shannon T. Holloway <shannon.t.hollowa@gmail.com>

References

Cao, H., Zeng, D., and Fine, J. P. (2015) Regression Analysis of sparse asynchronous longitudinal data. Journal of the Royal Statistical Society: Series B, 77, 755-776.

Cao, H., Li, Jialiang, and Fine, J. P. (2016). On last observation carried forward and asynchronous longitudinal regression analysis. Electronic Journal of Statistics, 10, 1155-1180.

See Also

`asynchTD`, `asynchTI`, `asynchLV`

`asynchDataTD` *Generated Asynchronous Longitudinal Data with Time-Dependent Coefficients*

Description

For the purposes of the package examples, the data set was adapted from the numerical simulations of the original manuscript. Specifically, data was generated for 400 subjects. The number of observation times for the response was Poisson distributed with intensity rate 5, and similarly for the number of observation times for the covariates. Observation times are generated from a uniform distribution Unif(0,1) independently. The covariate process is Gaussian, with values at fixed time points being multivariate normal with mean 0, variance 1 and correlation $\exp(-|t_{ij} - t_{ik}|)$. The responses were generated from $Y(t) = \beta_0 + X(t) * \beta_1 + \epsilon(t)$, where β_0 = 0.5, β_1 = 0.4t + 0.5, and $\epsilon(t)$ is Gaussian with mean 0, variance 1 and $\mathrm{cov}(\epsilon(s), \epsilon(t)) = 2^{-|t-s|}$. Covariates are stored as TD.x. Responses are stored as TD.y.

Format

TD.x is a data frame with 4052 observations on the following 3 variables.

`ID` patient identifier, there are 400 patients.

`t` the covariate observation times

`X1` the covariate measured at observation time t

TD.y is a data frame with 3939 observations on the following 3 variables.

`ID` patient identifier, there are 400 patients.

`t` the response observation times.

`Y` the response measured at time t.

Source

Generated by Shannon T. Holloway in R.

References

Cao, H., Zeng, D., and Fine, J. P. (2015) Regression Analysis of sparse asynchronous longitudinal data. Journal of the Royal Statistical Society: Series B, 77, 755-776.

`asynchDataTI` *Generated Asynchronous Longitudinal Data with Time-Invariant Coefficients*

Description

For the purposes of the package examples, the data set was adapted from the numerical simulations of the original manuscript. Specifically, data was generated for 400 subjects. The number of observation times for the response was Poisson distributed with intensity rate 5, and similarly for the number of observation times for the covariates. Observation times are generated from a uniform distribution Unif(0,1) independently. The covariate process is Gaussian, with values at fixed time points being multivariate normal with mean 0, variance 1 and correlation $\exp(-|t_{ij} - t_{ik}|)$. The responses were generated from $Y(t) = \beta_0 + X(t) * \beta_1 + \epsilon(t)$, where β_0 = 0.5, β_1 = 1.5, and $\epsilon(t)$ is Gaussian with mean 0, variance 1 and $\mathrm{cov}(\epsilon(s), \epsilon(t)) = 2^{-|t-s|}$. Covariates are stored as TI.x. Responses are stored as TI.y.

Format

TI.x is a data frame with 2014 observations on the following 3 variables.

`ID` patient identifier, there are 400 patients.

`t` the covariate observation times

`X1` the covariate measured at observation time t

TI.y is a data frame with 2101 observations on the following 3 variables.

`ID` patient identifier, there are 400 patients.

`t` the response observation times.

`Y` the response measured at time t.

Source

Generated by Shannon T. Holloway in R.

References

Cao, H., Zeng, D., and Fine, J. P. (2015) Regression Analysis of sparse asynchronous longitudinal data. Journal of the Royal Statistical Society: Series B, 77, 755-776.

`asynchHK` *Regression Analysis for Time-Invariant Coefficients Using Half-Kernel Estimation*

Description

Estimation of regression models for sparse asynchronous longitudinal observations using a half-kernel estimation approach with time-invariant coefficients.

Usage

```
asynchHK(data.x, data.y, kType = "epan", lType = "identity", bw = NULL,
         nCores = 1, verbose = TRUE, ...)
```

Arguments

`data.x` A data.frame of covariates. The structure of the data.frame must be {patient ID, time of measurement, measurement(s)}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].

`data.y` A data.frame of response measurements. The structure of the data.frame must be {patient ID, time of measurement, measurement}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].

`kType`	An object of class character indicating the type of smoothing kernel to use in the estimating equation. Must be one of {"epan", "uniform", "gauss"}, where "epan" is the Epanechnikov kernel and "gauss" is the Gaussian kernel.
`lType`	An object of class character indicating the type of link function to use for the regression model. Must be one of {"identity","log","logistic"}.
`bw`	If provided, bw is an object of class numeric or a numeric vector containing the bandwidths for which parameter estimates are to be obtained. If NULL, an optimal bandwidth will be determined using an adaptive selection procedure. The range of the bandwidth search space is taken to be $2*(Q3-Q1)*n^{-0.7} to 2*(Q3-Q1)*n^{-0.3}$, where Q3 is the 0.75 quantile and Q1 is the 0.25 quantile of the pooled sample of measurement times for the covariate and response, and n is the number of patients. See original reference for details of the selection procedure.
`nCores`	A numeric object. For auto-tune method, the number of cores to employ for calculation. If nCores > 1, the bandwidth search space will be distributed across the cores using parallel's parLapply.
`verbose`	An object of class logical. TRUE results in screen prints.
`...`	Ignored.

Details

For lType = "log" and lType = "logistic", parameter estimates are obtained by minimizing the estimating equation using optim() with method="Nelder-Mead"; all other arguments take their default values.

For lType = "identity", parameter estimates are obtained using solve().

Value

A list is returned. If bandwidths are provided, each element of the list is a matrix, where the ith row corresponds to the ith bandwidth of argument “bw" and the columns correspond to the model parameters. If the bandwidth is determined automatically, each element is a named vector calculated at the optimal bandwidth.

`betaHat`	The estimated model coefficients.
`stdErr`	The standard error for each coefficient.
`zValue`	The estimated z-value for each coefficient.
`pValue`	The p-value for each coefficient.

If the bandwidth is determined automatically, two additional list elements are returned:

`optBW`	The estimated optimal bandwidth for each coefficient.
`minMSE`	The mean squared error at the optimal bandwidth for each coefficient.

Author(s)

Hongyuan Cao, Jialiang Li, Jason P. Fine, and Shannon T. Holloway

References

Cao, H., Li, Jialiang, and Fine, J. P. (2016). On last observation carried forward and asynchronous longitudinal regression analysis. Electronic Journal of Statistics, 10, 1155–1180.

Examples

```
data(asynchDataTI)

res <- asynchHK(data.x = TI.x,
                data.y = TI.y,
                bw = c(0.05, 0.03),
                kType = "epan",
                lType = "identity")
```

---

asynchLV   *Regression Analysis Using Last Value Carried Forward*

Description

Estimation of regression models for sparse asynchronous longitudinal observations using the last value carried forward approach.

Usage

```
asynchLV(data.x, data.y, lType = "identity", verbose = TRUE, ...)
```

Arguments

`data.x`	A data.frame of covariates. The structure of the data.frame must be {patient ID, time of measurement, measurement(s)}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].
`data.y`	A data.frame of response measurements. The structure of the data.frame must be {patient ID, time of measurement, measurement}. Patient IDs must be of class integer or be able to be coerced to integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].
`lType`	An object of class character indicating the type of link function to use for the regression model. Must be one of {"identity","log","logistic"}.
`verbose`	An object of class logical. TRUE results in screen prints.
`...`	Ignored.

Details

For lType = "log" and lType = "logistic", parameter estimates are obtained by minimizing the estimating equation using R's optim() with method="Nelder-Mead"; all other settings take their default values.

For lType = "identity", parameter estimates are obtained use solve().

Value

A list is returned, the elements of which are named vectors:

`betaHat`	The estimated model coefficients.
`stdErr`	The standard error for each coefficient.
`zValue`	The estimated z-value for each coefficient.
`pValue`	The p-value for each coefficient.

Author(s)

Hongyuan Cao, Donglin Zeng, Jason P. Fine, and Shannon T. Holloway

References

Cao, H., Zeng, D., and Fine, J. P. (2015) Regression Analysis of sparse asynchronous longitudinal data. Journal of the Royal Statistical Society: Series B, 77, 755-776.

Examples

```
data(asynchDataTI)

res <- asynchLV(data.x = TI.x,
                data.y = TI.y,
                lType = "identity")
```

---

asynchTD — *Regression Analysis for Time-Dependent Coefficients*

Description

Estimation of regression models for sparse asynchronous longitudinal observations with time-dependent coefficients.

Usage

```
asynchTD(data.x, data.y, times, kType = "epan", lType = "identity",
                bw=NULL, nCores = 1, verbose = TRUE, ...)
```

Arguments

`data.x` A data.frame of covariates. The structure of the data.frame must be {patient ID, time of measurement, measurement(s)}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].

`data.y` A data.frame of response measurements. The structure of the data.frame must be {patient ID, time of measurement, measurement}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].

`kType` An object of class character indicating the type of smoothing kernel to use in the estimating equation. Must be one of {"epan", "uniform", "gauss"}, where "epan" is the Epanechnikov kernel and "gauss" is the Gaussian kernel.

`lType` An object of class character indicating the type of link function to use for the regression model. Must be one of {"identity","log","logistic"}.

`bw` If provided, bw is an object of class numeric containing a single bandwidth at which parameter estimates are to be obtained. If NULL, an “optimal" bandwidth will be determined for each time point using an adaptive selection procedure. The range of the bandwidth search space is taken to be $2*(Q3-Q1)*n^{-0.7} to 2*(Q3-Q1)*n^{-0.3}$, where Q3 is the 0.75 quantile and Q1 is the 0.25 quantile of the pooled sample of measurement times for the covariate and response, and n is the number of patients. For each time point, the optimal bandwidth(s) is taken to be that which minimizes the mean squared error. See original reference for details of the selection procedure.

`times` A vector object of class numeric. The time points at which the coefficients are to be estimated.

`nCores` A numeric object. For auto-tune method, the number of cores to employ for calculation. If nCores > 1, the bandwidth search space will be distributed across the cores using parallel’s parLapply.

`verbose` An object of class logical. TRUE results in screen prints.

`...` Ignored.

Details

For lType = "log" and lType = "logistic", parameter estimates are obtained by minimizing the estimating equation using optim() with method="Nelder-Mead"; all other arguments take their default values.

For lType = "identity", parameter estimates are obtained using solve().

Upon completion, a single plot is generating showing the time-dependence of each coefficient.

Value

A list is returned. Each element of the list is a matrix, where the ith row corresponds to the ith time point of input argument “times" and the columns correspond to the model parameters.

The returned values are estimated using either the provided bandwidth or the “optimal" bandwidth as determined using the adaptive selection procedure.

`betaHat`	The estimated model coefficients.
`stdErr`	The standard errors for each coefficient.
`zValue`	The estimated z-values for each coefficient.
`pValue`	The p-values for each coefficient.

If the bandwidth is determined automatically, two additional list elements are returned:

`optBW`	The estimated optimal bandwidth for each coefficient.
`minMSE`	The mean squared error at the optimal bandwidth for each coefficient.

Author(s)

Hongyuan Cao, Donglin Zeng, Jason P. Fine, and Shannon T. Holloway

References

Cao, H., Zeng, D., and Fine, J. P. (2014) Regression Analysis of sparse asynchronous longitudinal data. Journal of the Royal Statistical Society: Series B, 77, 755-776.

Examples

```
data(asynchDataTD)

res <- asynchTD(data.x = TD.x,
                data.y = TD.y,
                times = c(0.25, 0.50, 0.75),
                bw = 0.05,
                kType = "epan",
                lType = "identity")
```

`asynchTI` *Regression Analysis for Time-Invariant Coefficients*

Description

Estimation of regression models for sparse asynchronous longitudinal observations with time-invariant coefficients.

Usage

```
asynchTI(data.x, data.y, kType = "epan", lType = "identity", bw = NULL,
         nCores = 1, verbose = TRUE, ...)
```

Arguments

`data.x` A data.frame of covariates. The structure of the data.frame must be {patient ID, time of measurement, measurement(s)}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].

`data.y` A data.frame of response measurements. The structure of the data.frame must be {patient ID, time of measurement, measurement}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].

`kType` An object of class character indicating the type of smoothing kernel to use in the estimating equation. Must be one of {"epan", "uniform", "gauss"}, where "epan" is the Epanechnikov kernel and "gauss" is the Gaussian kernel.

`lType` An object of class character indicating the type of link function to use for the regression model. Must be one of {"identity","log","logistic"}.

`bw` If provided, bw is an object of class numeric or a numeric vector containing the bandwidths for which parameter estimates are to be obtained. If NULL, an optimal bandwidth will be determined using an adaptive selection procedure. The range of the bandwidth search space is taken to be $2*(Q3-Q1)*n^{-0.7} to 2*(Q3-Q1)*n^{-0.3}$, where Q3 is the 0.75 quantile and Q1 is the 0.25 quantile of the pooled sample of measurement times for the covariate and response, and n is the number of patients. See original reference for details of the selection procedure.

`nCores` A numeric object. For auto-tune method, the number of cores to employ for calculation. If nCores > 1, the bandwidth search space will be distributed across the cores using parallel's parLapply.

`verbose` An object of class logical. TRUE results in screen prints.

`...` Ignored.

Details

For lType = "log" and lType = "logistic", parameter estimates are obtained by minimizing the estimating equation using optim() with method="Nelder-Mead"; all other arguments take their default values.

For lType = "identity", parameter estimates are obtained using solve().

Value

A list is returned. If bandwidths are provided, each element of the list is a matrix, where the ith row corresponds to the ith bandwidth of argument “bw" and the columns correspond to the model parameters. If the bandwidth is determined automatically, each element is a named vector calculated at the optimal bandwidth.

`betaHat` The estimated model coefficients.

`stdErr` The standard error for each coefficient.

`zValue` The estimated z-value for each coefficient.

pValue The p-value for each coefficient.

If the bandwidth is determined automatically, two additional list elements are returned:

optBW The estimated optimal bandwidth for each coefficient.

minMSE The mean squared error at the optimal bandwidth for each coefficient.

Author(s)

Hongyuan Cao, Donglin Zeng, Jason P. Fine, and Shannon T. Holloway

References

Cao, H., Zeng, D., and Fine, J. P. (2015) Regression Analysis of sparse asynchronous longitudinal data. Journal of the Royal Statistical Society: Series B, 77, 755-776.

Examples

```
data(asynchDataTI)

res <- asynchTI(data.x = TI.x,
                data.y = TI.y,
                bw = c(0.05, 0.03),
                kType = "epan",
                lType = "identity")
```

---

asynchWLV *Weighted Last Observation Carried Forward Regression Analysis for Time-Invariant Coefficients*

Description

Estimation of regression models for sparse asynchronous longitudinal observations using the weighted last value carried forward approach with time-invariant coefficients.

Usage

```
asynchWLV(data.x, data.y, kType = "epan", lType = "identity", bw = NULL,
          nCores = 1, verbose = TRUE, ...)
```

Arguments

`data.x`	A data.frame of covariates. The structure of the data.frame must be {patient ID, time of measurement, measurement(s)}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].
`data.y`	A data.frame of response measurements. The structure of the data.frame must be {patient ID, time of measurement, measurement}. Patient IDs must be of class integer or be able to be coerced to class integer without loss of information. Missing values must be indicated as NA. All times will automatically be rescaled to [0,1].
`kType`	An object of class character indicating the type of smoothing kernel to use in the estimating equation. Must be one of {"epan", "uniform", "gauss"}, where "epan" is the Epanechnikov kernel and "gauss" is the Gaussian kernel.
`lType`	An object of class character indicating the type of link function to use for the regression model. Must be one of {"identity","log","logistic"}.
`bw`	If provided, bw is an object of class numeric or a numeric vector containing the bandwidths for which parameter estimates are to be obtained. If NULL, an optimal bandwidth will be determined using an adaptive selection procedure. The range of the bandwidth search space is taken to be $2*(Q3-Q1)*n^{-0.7} to 2*(Q3-Q1)*n^{-0.3}$, where Q3 is the 0.75 quantile and Q1 is the 0.25 quantile of the pooled sample of measurement times for the covariate and response, and n is the number of patients. See original reference for details of the selection procedure.
`nCores`	A numeric object. For auto-tune method, the number of cores to employ for calculation. If nCores > 1, the bandwidth search space will be distributed across the cores using parallel's parLapply.
`verbose`	An object of class logical. TRUE results in screen prints.
`...`	Ignored.

Details

For lType = "log" and lType = "logistic", parameter estimates are obtained by minimizing the estimating equation using optim() with method="Nelder-Mead"; all other arguments take their default values.

For lType = "identity", parameter estimates are obtained using solve().

Value

A list is returned. If bandwidths are provided, each element of the list is a matrix, where the ith row corresponds to the ith bandwidth of argument “bw" and the columns correspond to the model parameters. If the bandwidth is determined automatically, each element is a named vector calculated at the optimal bandwidth.

`betaHat`	The estimated model coefficients.
`stdErr`	The standard error for each coefficient.
`zValue`	The estimated z-value for each coefficient.

`pValue` The p-value for each coefficient.

If the bandwidth is determined automatically, two additional list elements are returned:

`optBW` The estimated optimal bandwidth for each coefficient.

`minMSE` The mean squared error at the optimal bandwidth for each coefficient.

Author(s)

Hongyuan Cao, Jialiang Li, Jason P. Fine, and Shannon T. Holloway

References

Cao, H., Li, Jialiang, and Fine, J. P. (2016). On last observation carried forward and asynchronous longitudinal regression analysis. Electronic Journal of Statistics, 10, 1155-1180.

Examples

```
data(asynchDataTI)

res <- asynchWLV(data.x = TI.x,
                 data.y = TI.y,
                 bw = c(0.05, 0.03),
                 kType = "epan",
                 lType = "identity")
```

Index