

Package ‘timedelay’

May 16, 2018

Version 1.0.8

Date 2018-05-16

Title Time Delay Estimation for Stochastic Time Series of
Gravitationally Lensed Quasars

Author Hyungsuk Tak, Kaisey Mandel, David A. van Dyk, Vinay L. Kashyap, Xiao-
Li Meng, and Aneta Siemiginowska

Maintainer Hyungsuk Tak <hyungsuk.tak@gmail.com>

Depends R (>= 2.2.0), MASS

Description We provide a toolbox to estimate the time delay between the brightness time series of gravitationally lensed quasar images via Bayesian and profile likelihood approaches. The model is based on a state-space representation for irregularly observed time series data generated from a latent continuous-time Ornstein-Uhlenbeck process. Our Bayesian method adopts scientifically motivated hyper-prior distributions and a Metropolis-Hastings within Gibbs sampler, producing posterior samples of the model parameters that include the time delay. A profile likelihood of the time delay is a simple approximation to the marginal posterior distribution of the time delay. Both Bayesian and profile likelihood approaches complement each other, producing almost identical results; the Bayesian way is more principled but the profile likelihood is easier to implement.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-16 16:50:57 UTC

R topics documented:

bayesian	2
entirelogprofilelikelihood	5
simple	8
timedelay	9

Index	13
--------------	-----------

bayesian

*Estimating the time delay via the Bayesian method***Description**

bayesian produces posterior samples of all the model parameters one of which is the time delay. This function has options for two MCMC techniques, ancillarity-sufficiency interweaving strategy (ASIS) and adaptive MCMC, to improve the convergence rate of the MCMC.

Usage

```
bayesian(data.lcA, data.lcB, data.flux, theta.ini,
         delta.ini, delta.uniform.range, delta.proposal.scale,
         tau.proposal.scale, tau.prior.shape, tau.prior.scale,
         sigma.prior.shape, sigma.prior.scale, asis = TRUE, micro, multimodality = FALSE,
         adaptive.frequency, adaptive.delta = TRUE, adaptive.delta.factor,
         adaptive.tau = TRUE, adaptive.tau.factor,
         sample.size, warmingup.size)
```

Arguments

data.lcA	A (n by 3) matrix for image A (light curve A); the first column has n observation times, the second column contains n flux (or magnitude) values, the third column includes n measurement errors.
data.lcB	A (w by 3) matrix for image B (light curve B); the first column has w observation times, the second column contains w flux (or magnitude) values, the third column includes w measurement errors.
data.flux	"True" if data are recorded on flux scale or "FALSE" if data are on magnitude scale. If your observed time series can take on negative values, then data.flux = FALSE.
theta.ini	Initial values of theta = (mu, sigma, tau) for MCMC.
delta.ini	Initial values of the time delay for MCMC.
delta.uniform.range	The range of the Uniform prior distribution for the time delay. The feasible entire support is $c(\min(\text{simple}[, 1]) - \max(\text{simple}[, 1]), \max(\text{simple}[, 1]) - \min(\text{simple}[, 1]))$.
delta.proposal.scale	The proposal scale of the Metropolis step for the time delay.
tau.proposal.scale	The proposal scale of the Metropolis-Hastings step for tau.
tau.prior.shape	The shape parameter of the Inverse-Gamma hyper-prior distribution for tau.
tau.prior.scale	The scale parameter of the Inverse-Gamma hyper-prior distribution for tau.

<code>sigma.prior.shape</code>	The shape parameter of the Inverse-Gamma hyper-prior distribution for σ^2 .
<code>sigma.prior.scale</code>	The scale parameter of the Inverse-Gamma hyper-prior distribution for σ^2 . If no prior information is available, we recommend using $2 * 10^{-7}$.
<code>asis</code>	(Optional) "TRUE" if we use the ancillarity-sufficiency interweaving strategy (ASIS) for c (always recommended). Default is "TRUE".
<code>micro</code>	It determines the order of a polynomial regression model that accounts for the difference between microlensing trends. Default is 3. When zero is assigned, the Bayesian model fits a curve-shifted model.
<code>multimodality</code>	(Optional) "TRUE" if we use a repelling-attracting Metropolis algorithm for sampling Δ . When it is "TRUE" it is recommended that <code>adaptive.delta = FALSE</code> so that the adaptive MCMC is not used in the presence of multimodality. Default is "FALSE".
<code>adaptive.frequency</code>	(If "adaptive.delta = TRUE" or "adaptive.tau = TRUE") The adaptive MCMC is applied for every specified frequency. If it is specified as 500, the adaptive MCMC is applied to every 500th iteration.
<code>adaptive.delta</code>	(Optional) "TRUE" if we use the adaptive MCMC for the time delay. Default is "TRUE".
<code>adaptive.delta.factor</code>	(If "adaptive.delta = TRUE") The factor, $\exp(\text{adaptive.delta.factor})$ or $\exp(-\text{adaptive.delta.factor})$, multiplied to the proposal scale of the time delay for adaptive MCMC.
<code>adaptive.tau</code>	(Optional) "TRUE" if we use the adaptive MCMC for τ . Default is "TRUE".
<code>adaptive.tau.factor</code>	(If "adaptive.tau = TRUE") The factor, $\exp(\text{adaptive.tau.factor})$ or $\exp(-\text{adaptive.tau.factor})$, multiplied to the proposal scale of τ for adaptive MCMC.
<code>sample.size</code>	The number of the posterior samples of each model parameter.
<code>warmup.size</code>	The number of burn-ins for MCMC.

Details

The function `bayesian` produces posterior samples of the model parameters one of which is the time delay.

Value

The outcome of `bayesian` comprises of:

<code>delta</code>	Posterior samples of the time delay
<code>X</code>	Posterior samples of the latent magnitudes
<code>beta</code>	Posterior samples of the polynomial regression coefficients, β
<code>mu</code>	Posterior samples of the mean parameter of the O-U process, μ
<code>sigma</code>	Posterior samples of the short term variability of the O-U process, σ
<code>tau</code>	Posterior samples of the mean reversion time of the O-U process, τ

```
tau.accept.rate
    The acceptance rate of the MCMC for tau
delta.accept.rate
    The acceptance rate of the MCMC for the time delay
```

Author(s)

Hyungsuk Tak

References

Hyungsuk Tak, Kaisey Mandel, David A. van Dyk, Vinay L. Kashyap, Xiao-Li Meng, and Aneta Siemiginowska (2017+). "Bayesian Estimates of Astronomical Time Delays between Gravitationally Lensed Stochastic Light Curves," tentatively accepted in *Annals of Applied Statistics* (ArXiv 1602.01462). Hyungsuk Tak, Xiao-Li Meng, David A. van Dyk (2017+). "A Repelling-Attracting Metropolis Algorithm for Multimodality," submitted to *Journal of Computational and Graphical Statistics* (ArXiv 1601.05633).

Examples

```
# Loading datasets
data(simple)

# A typical quasar data set
head(simple)

# Subset (data for image A) of the typical quasar data set
lcA <- simple[, 1 : 3]

# Another subset (data for image B) of the typical quasar data set
# The observation times for image B are not necessarily the same as those for image A
lcB <- simple[, c(1, 4, 5)]

# The two subsets do not need to have the same number of observations
# For example, here we add one more observation time for image B
lcB <- rbind(lcB, c(290, 1.86, 0.006))

dim(lcA)
dim(lcB)

#####
# Time delay estimation via Bayesian approach #
#####

# Cubic microlensing model (m = 3)
output = bayesian(data.lcA = lcA, data.lcB = lcB,
                  data.flux = FALSE, theta.ini = c(0, 0.01, 200),
                  delta.ini = 50, delta.uniform.range = c(0, 100),
                  delta.proposal.scale = 1,
                  tau.proposal.scale = 3,
                  tau.prior.shape = 1, tau.prior.scale = 1,
```

```

sigma.prior.shape = 1, sigma.prior.scale = 2 / 10^7,
asis = TRUE, micro = 3,
sample.size = 100, warmingup.size = 50)

names(output)

# hist(output$delta, 20)
# plot(output$delta, type = "l")
# acf(output$delta)
# output$delta.accept
# output$tau.accept

# If multimodality exists, then please add two arguments:
# multimodality = TRUE, adaptive.delta = FALSE
# This is to prevent the Markov chain from adapting to a local mode.
# If we know the distance between the most distant modes,
# try the smallest value of delta.proposal.scale that still makes
# the Markov chain jump between those modes frequently.
# For example, when the distance is d, then try
# delta.proposal.scale = d * 0.7, decreasing or increasing 0.7.

# Graphical model checking
# beta.hat <- colMeans(output$beta)
# delta.hat <- mean(output$delta)
# time.x <- lcB[, 1] - delta.hat
# time.covariate <- cbind(rep(1, length(time.x)), time.x, time.x^2, time.x^3)
##### This time.covariate is when micro = 3, a third-order polynomial regression.
##### With micro = 0, "time.covariate <- rep(1, length(time.x))" is used.
# predicted <- time.covariate %*% beta.hat
# plot(lcA[, 1], lcA[, 2], col = 2)
##### Adjust the range of the x-axis by the argument of plot.
##### For example "xlim = c(-1, 2)"
# points(lcB[, 1] - delta.hat, lcB[, 2] - predicted, col = 4, pch = 1)
# for (i in 1 : length(output$delta)) {
#   temp.time <- c(lcA[, 1], lcB[, 1] - output$delta[i])
#   points(sort(temp.time), output$X[i, ],
#         col = "gray", cex = 0.5, pch = 1, lwd = 0.01)
# }
# points(lcA[, 1], lcA[, 2], pch = 0, col = 2, lwd = 1)
# points(lcB[, 1] - delta.hat, lcB[, 2] - predicted,
#       col = 4, pch = 1, lwd = 1)

```

entirelogprofilelikelihood

Calculating the entire profile likelihood curve over the given grid values of the time delay

Description

entirelogprofilelikelihood calculates the entire profile likelihood curve over the given grid values of the time delay.

Usage

```
entirelogprofilelikelihood(data.lcA, data.lcB, grid,
                           initial, data.flux,
                           delta.uniform.range, micro)
```

Arguments

data.lcA	A (n by 3) matrix; the first column has n observation times of light curve A, the second column has n flux (or magnitude) values of light curve A, the third column has n measurement errors of light curve A.
data.lcB	A (w by 3) matrix; the first column has w observation times of light curve B, the second column has w flux (or magnitude) values of light curve B, the third column has w measurement errors of light curve B.
grid	A vector containing values of the time delay on which the profile likelihood values are calculated. We recommend using the grid interval equal to 0.1.
initial	The initial values of the other model parameters (μ , $\log(\sigma)$, $\log(\tau)$, β). We take log on σ and τ for numerical stability.
data.flux	"True" if data are recorded on flux scale or "FALSE" if data are on magnitude scale.
delta.uniform.range	The range of the Uniform prior distribution for the time delay. The feasible entire support is $c(\min(\text{simple}[, 1]) - \max(\text{simple}[, 1]), \max(\text{simple}[, 1]) - \min(\text{simple}[, 1]))$.
micro	It determines the order of a polynomial regression model that accounts for the difference between microlensing trends. Default is 3. When zero is assigned, the Bayesian model fits a curve-shifted model.

Details

The function entirelogprofilelikelihood is used to obtain the entire profile likelihood curve over the given grid values of the time delay.

Value

The outcome of entirelogprofilelikelihood is the values of the log profile likelihood function over the given grid values of the time delay.

Author(s)

Hyungsuk Tak

References

Hyungsuk Tak, Kaisey Mandel, David A. van Dyk, Vinay L. Kashyap, Xiao-Li Meng, and Aneta Siemiginowska (2016+). "Bayesian Estimates of Astronomical Time Delays between Gravitationally Lensed Stochastic Light Curves," tentatively accepted in *Annals of Applied Statistics* (ArXiv 1602.01462).

Examples

```
# Loading datasets
data(simple)
head(simple)

#####
# Time delay estimation via profile likelihood #
#####

# Subset (data for image A) of the typical quasar data set
lcA <- simple[, 1 : 3]

# Another subset (data for image B) of the typical quasar data set
# The observation times for image B are not necessarily the same as those for image A
lcB <- simple[, c(1, 4, 5)]

# The two subsets do not need to have the same number of observations
# For example, here we add one more observation time for image B
lcB <- rbind(lcB, c(290, 1.86, 0.006))

dim(lcA)
dim(lcB)

##### The entire profile likelihood values on the grid of values of the time delay.

# Cubic microlensing model
ti1 <- lcB[, 1]
ti2 <- lcB[, 1]^2
ti3 <- lcB[, 1]^3
ss <- lm(lcB[, 2] - mean(lcA[, 2]) ~ ti1 + ti2 + ti3)

initial <- c(mean(lcA[, 2]), log(0.01), log(200), ss$coefficients)
delta.uniform.range <- c(0, 100)
grid <- seq(0, 100, by = 0.1)
# grid interval "by = 0.1" is recommended,
# but users can set a finer grid of values of the time delay.

### Running the following codes takes more time than CRAN policy
### Please type the following lines without "#" to run the function and to see the results

# logprof <- entirelogprofilelikelihood(data.lcA = lcA, data.lcB = lcB, grid = grid,
#                                     initial = initial, data.flux = FALSE,
#                                     delta.uniform.range = delta.uniform.range, micro = 3)
```

```
# plot(grid, logprof, type = "l",
#       xlab = expression(bold(Delta)),
#       ylab = expression(bold(paste("log L"[prof], "(", Delta, ")"))))
# prof <- exp(logprof - max(logprof)) # normalization
# plot(grid, prof, type = "l",
#       xlab = expression(bold(Delta)),
#       ylab = expression(bold(paste("L"[prof], "(", Delta, ")"))))
```

simple

Simulated simple data of a doubly-lensed quasar

Description

We simulated a simple data set of a doubly-lensed quasar with generative parameter values (Δ , β_0 , μ , σ , τ) equal to (50, 2, 0, 0.03, 100). The number of epochs is 78 and measurement errors are the same to 0.005.

Usage

```
data(simple)
```

Format

A simple data set of a doubly-lensed quasar:

time observation times

x magnitudes of light A

se.x measurement errors of x

y magnitudes of light B

se.y measurement errors of y

Examples

```
data(simple)
```

`timedelay`*Time Delay Estimation for Stochastic Time Series of Gravitationally Lensed Quasars*

Description

The R package **timedelay** provides a toolbox to estimate the time delay between the brightness time series of gravitationally lensed quasar images via Bayesian and profile likelihood approaches. The model is based on a state-space representation for irregularly observed time series data generated from a latent continuous-time Ornstein-Uhlenbeck process. Our Bayesian method adopts scientifically motivated hyper-prior distributions and a Metropoli-Hastings within Gibbs sampler, producing posterior samples of the model parameters that include the time delay. A profile likelihood of the time delay is a simple approximation to the marginal posterior distribution of the time delay. Both Bayesian and profile likelihood approaches complement each other, producing almost identical results; the Bayesian way is more principled but the profile likelihood is easier to be implemented.

Details

Package:	timedelay
Type:	Package
Version:	1.0.8
Date:	2018-05-16
License:	GPL-2
Main functions:	bayesian , entirelogprofilelikelihood

Author(s)

Hyungsuk Tak, Kaisey Mandel, David A. van Dyk, Vinay L. Kashyap, Xiao-Li Meng, and Aneta Siemiginowska

Maintainer: Hyungsuk Tak <hyungsuk.tak@gmail.com>

References

Hyungsuk Tak, Kaisey Mandel, David A. van Dyk, Vinay L. Kashyap, Xiao-Li Meng, and Aneta Siemiginowska (2016+). "Bayesian Estimates of Astronomical Time Delays between Gravitationally Lensed Stochastic Light Curves," tentatively accepted in *Annals of Applied Statistics* (ArXiv 1602.01462).

Examples

```
# Loading datasets
data(simple)
```

```

head(simple)

# Subset (data for image A) of the typical quasar data set
lcA <- simple[, 1 : 3]

# Another subset (data for image B) of the typical quasar data set
# The observation times for image B are not necessarily the same as those for image A
lcB <- simple[, c(1, 4, 5)]

# The two subsets do not need to have the same number of observations
# For example, here we add one more observation time for image B
lcB <- rbind(lcB, c(290, 1.86, 0.006))

dim(lcA)
dim(lcB)

#####
# Time delay estimation via Bayesian approach #
#####

# Cubic microlensing model (m = 3)
output <- bayesian(data.lcA = lcA, data.lcB = lcB,
                  data.flux = FALSE, theta.ini = c(0, 0.01, 200),
                  delta.ini = 50, delta.uniform.range = c(0, 100),
                  delta.proposal.scale = 1,
                  tau.proposal.scale = 3,
                  tau.prior.shape = 1, tau.prior.scale = 1,
                  sigma.prior.shape = 1, sigma.prior.scale = 2 / 10^7,
                  asis = TRUE, micro = 3,
                  sample.size = 100, warmingup.size = 50)

names(output)
# hist(output$delta)
# plot(output$delta, type = "l")
# acf(output$delta)

### Argument description

# data.lcA: An n1 by three matrix for image A (light curve A) with
#           n1 observation times in the first column,
#           n1 observed magnitudes (or in flux) in the second column,
#           n1 measurement errors for image A in the third column.

# data.lcB: An n2 by three matrix for image B (light curve B) with
#           n2 observation times in the first column,
#           n2 observed magnitudes (or in flux) in the second column,
#           n2 measurement errors for image A in the third column.

# data.flux: "True" if data are recorded on flux scale or "FALSE" if data are on magnitude scale.
# If your observed time series can take on negative values, then data.flux = FALSE.

# theta.ini: Initial values of theta = (mu, sigma, tau) for MCMC.

```

```

# delta.ini: Initial values of the time delay for MCMC.
# delta.uniform.range: The range of the Uniform prior distribution for the time delay.
#
#           The feasible entire support is
#           c(min(simple[, 1]) - max(simple[, 1]), max(simple[, 1]) - min(simple[, 1]))

# delta.proposal.scale: The proposal scale of the Metropolis step for the time delay.
# tau.proposal.scale: The proposal scale of the Metropolis-Hastings step for tau.

# tau.prior.shape: The shape parameter of the Inverse-Gamma hyper-prior distribution for tau.
# tau.prior.scale: The scale parameter of the Inverse-Gamma hyper-prior distribution for tau.
# sigma.prior.shape: The shape parameter of
#                   the Inverse-Gamma hyper-prior distribution for sigma^2.
# sigma.prior.scale: The scale parameter of
#                   the Inverse-Gamma hyper-prior distribution for sigma^2.
# micro: It determines the order of a polynomial regression model that accounts
#        for the difference between microlensing trends. Default is 3.
#        When zero is assigned, the Bayesian model fits a curve-shifted model.

# asis: "TRUE" if we use the ancillarity-sufficiency interweaving strategy (ASIS)
#       for c (always recommended)
# multimodality: "TRUE" if we use the repelling-attracting Metropolis algorithm
#               for sampling Delta in the presence of multimodality.
#               Please make sure that "adaptive.delta = FALSE" so that
#               adaptive MCMC for Delta is not used in the presence of multimodality.

# adaptive.frequency: The adaptive MCMC is applied for every specified frequency.
#                   If it is specified as 100,
#                   the adaptive MCMC is applied to every 100th iteration.
# adaptive.delta: "TRUE" if we use the adaptive MCMC for the time delay.
# adaptive.delta.factor: The factor, exp(adaptive.delta.factor) or exp(-adaptive.delta.factor),
#                       multiplied to the proposal scale of the time delay for adaptive MCMC.
#                       Default is 0.01.

# adaptive.tau: "TRUE" if we use the adaptive MCMC for tau.
# adaptive.tau.factor: The factor, exp(adaptive.tau.factor) or exp(-adaptive.tau.factor),
#                     multiplied to the proposal scale of tau for adaptive MCMC.

# sample.size: The number of posterior samples for each parameter
# warmingup.size: The number of burn-ins

# Type ?bayesian for further details on graphical model checking.

#####
# Time delay estimation via profile likelihood #
#####

##### The entire profile likelihood values on the grid of values of the time delay.

# Cubic microlensing model
ti1 <- lcB[, 1]
ti2 <- lcB[, 1]^2
ti3 <- lcB[, 1]^3

```

```

ss <- lm(lcB[, 2] - mean(lcA[, 2]) ~ ti1 + ti2 + ti3)

initial <- c(mean(lcA[, 2]), log(0.01), log(200), ss$coefficients)
delta.uniform.range <- c(0, 100)
grid <- seq(0, 100, by = 0.1)
# grid interval "by = 0.1" is recommended for accuracy,
# but users can set a finer grid of values of the time delay.

### Running the following codes takes more time than CRAN policy
### Please type the following lines without "#" to run the function and to see the results

# logprof <- entirelogprofilelikelihood(data.lcA = lcA, data.lcB = lcB, grid = grid,
#                                       initial = initial, data.flux = FALSE,
#                                       delta.uniform.range = delta.uniform.range, micro = 3)

# plot(grid, logprof, type = "l",
#       xlab = expression(bold(Delta)),
#       ylab = expression(bold(paste("log L"[prof], "(, Delta, ")"))))
# prof <- exp(logprof - max(logprof)) # normalization
# plot(grid, prof, type = "l",
#       xlab = expression(bold(Delta)),
#       ylab = expression(bold(paste("L"[prof], "(, Delta, ")"))))

### Argument description

# data.lcA: The data set (n by 3 matrix) for light curve A
# (1st column: observation times, 2nd column: values of fluxes or magnitudes,
# 3rd column: measurement errors)
# data.lcB: The data set (w by 3 matrix) for light curve B
# (1st column: observation times, 2nd column: values of fluxes or magnitudes,
# 3rd column: measurement errors)
# grid: the vector of grid values of the time delay
#       on which the log profile likelihood values are calculated.
# initial: The initial values of the other model parameters (mu, log(sigma), log(tau), beta)
# data.flux: "True" if data are recorded on flux scale or "FALSE" if data are on magnitude scale.
# delta.uniform.range: The range of the Uniform prior distribution for the time delay.
# micro: It determines the order of a polynomial regression model that accounts
#         for the difference between microlensing trends. Default is 3.
#         When zero is assigned, the Bayesian model fits a curve-shifted model.

```

Index

*Topic **datasets**

simple, 8

*Topic **methods**

bayesian, 2

entirelogprofilelikelihood, 5

*Topic **package**

timedelay, 9

bayesian, 2, 9

entirelogprofilelikelihood, 5, 9

simple, 8

timedelay, 9

timedelay-package (timedelay), 9