Package 'beast'

November 29, 2017

Type Package

Title Bayesian Estimation of Change-Points in the Slope of Multivariate Time-Series

Version 1.0

Date 2017-11-29

Author Panagiotis Papastamoulis

Maintainer Panagiotis Papastamoulis <papapast@yahoo.gr>

Description The beast package deals with Bayesian Estimation of chAnge-points in the Slope of multivariate Time-series. Assume that a temporal process is composed of contiguous segments with differing slopes and replicated noise-corrupted time series measurements are observed. The unknown mean of the data generating process is modelled as a piecewise linear function of time with an unknown number of change-points. The package infers the joint posterior distribution of the number and position of change-points as well as the unknown mean parameters per time-series by MCMC sampling. A-priori, the proposed model uses an overfitting number of mean parameters but, conditionally on a set of change-points, only a subset of them influences the likelihood. An exponentially decreasing prior distribution on the number of change-points gives rise to a posterior distribution concentrating on sparse representations of the underlying sequence, but also available is the Poisson distribution.

License GPL-2

Imports RColorBrewer

Depends R (>= 2.10)

NeedsCompilation no

R topics documented:

`beast-package` *Bayesian Estimation of Change-Points in the Slope of Multivariate Time-Series*

Description

The beast package deals with Bayesian Estimation of chAnge-points in the Slope of multivariate Time-series. Assume that a temporal process is composed of contiguous segments with differing slopes and replicated noise-corrupted time series measurements are observed. The unknown mean of the data generating process is modelled as a piecewise linear function of time with an unknown number of change-points. The package infers the joint posterior distribution of the number and position of change-points as well as the unknown mean parameters per time-series by MCMC sampling. A-priori, the proposed model uses an overfitting number of mean parameters but, conditionally on a set of change-points, only a subset of them influences the likelihood. An exponentially decreasing prior distribution on the number of change-points gives rise to a posterior distribution concentrating on sparse representations of the underlying sequence, but also available is the Poisson distribution.

Details

The `beast` package implements the Bayesian method introduced by Papastamoulis et al (2017) for detecting an unknown number of change-points in the slope of multivariate time-series. For a given period $t = 1, \ldots, T$ we observe multiple time series which are assumed independent, each one consisting of multiple measurements (replicates). Each time series is assumed to have its own segmentation, which is common among its replicates. Thus, different time series have distinct mean parameters in the underlying normal distribution. The variance, which is assumed known, can be either shared between different time series or not and in practice it is estimated at a pre-processing stage.

Our model infers the joint posterior distribution of the number and location of change-points by MCMC sampling. For this purpose a Metropolis-Hastings MCMC sampler is used. The main function of the package is `beast`.

Assume that the observed data consists of N time-series, each one consisting of R variables (or replicates) measured at T consecutive time-points. The input of the main function `beast` should be given in the form of a `list` `myDataList` with the following attributes:

- `length(myDataList)` should be equal to R, that is, the number of variables (or replicates)
- `dim(myDataList)[[1]]`, ..., `dim(myDataList)[[R]]` should be all $T \times N$, that is, rows and columns should correspond to time-points and different series, respectively.

Then, a basic usage of the package consists of the following commands:

- `beastRun <- beast( myDataList = myDataList )`
- `print(beastRun)`
- `plot(beastRun)`

which correspond to running the MCMC sampler, printing and plotting output summaries, respectively.

Author(s)

Panagiotis Papastamoulis

Maintainer: Panagiotis Papastamoulis <papapast@yahoo.gr>

References

Papastamoulis P., Furukawa T., van Rhijn N., Bromley M., Bignell E. and Rattray M. (2017). Bayesian detection of piecewise linear trends in replicated time-series with application to growth data modelling. arXiv:**1709.06111** [stat.AP]

See Also

`beast`, `print.beast.object`, `plot.beast.object`

Examples

```
## Not run:
# This example illustrates the package using a subset of four
#      time-series of the fungal dataset.
library('beast') # load package
data("FungalGrowthDataset") # load dataset
myIndex <- c(392, 62, 3, 117) # run the sampler only for the
#                                  specific subset of time-series
set.seed(1) # optional
# Run MCMC sampler:
run_mcmc <- beast(myDataList = FungalGrowthDataset, subsetIndex = myIndex,
zeroNormalization = TRUE)
# Print output:
print(run_mcmc)
# Plot output to file: "beast_plot.pdf"
plot(run_mcmc, fileName = "beast_plot.pdf", timeScale=1/6, xlab = "hours", ylab = "growth")
# NOTE 1: for a complete analysis remove the `subsetIndex = myIndex` argument.
# NOTE 2: `zeroNormalization = TRUE` is an optional argument that forces all
```

```
#    time-series to start from zero. It is not supposed to be used
#    for other applications.

## End(Not run)
```

---

beast *Main function*

Description

This is the main function of the package, implementing the MCMC sampler described in Papastamoulis et al (2017).

Usage

```
beast(myDataList, burn, nIter, mhPropRange, mhSinglePropRange, startPoint,
    timeScale, savePlots, zeroNormalization, LRange, tau,
    gammaParameter, nu0, alpha0, beta0, subsetIndex, saveTheta, sameVariance,
    Prior
)
```

Arguments

`myDataList` Observed data in the form of a `list` with length R, denoting the dimensionality of the multivariate time-series data. For each $r = 1, \ldots, R$, `myDataList[[r]]` should correspond to $T \times N$ array, with `myDataList[[r]][t, n]` corresponding to the observed data for time-series $n = 1, \ldots, N$ and time-point $t = 1, \ldots, T$.

`burn` Number of iterations that will be discarder as burn-in period. Default value: `burn = 20000`.

`nIter` Number of MCMC iterations. Default value: `nIter = 70000`.

`mhPropRange` Positive integer corresponding to the parameter d_1 of MCMC Move 3.a of Papastamoulis et al (2017). Default value: `mhPropRange = 1`.

`mhSinglePropRange`
Positive integer denoting the parameter d_2 of Papastamoulis et al (2017). Default value: `mhPropRange = 40`.

`startPoint` An (optional) positive integer pointing at the minimum time-point where changes are allowed to occur. Default value: `startPoint = 2` (all possible values are taken into account).

`timeScale` Null.

`savePlots` Character denoting the name of the folder where various plots will be saved to.

`zeroNormalization`
Logical value denoting whether to normalize to zero all time time-series for $t = 1$. Default: `zeroNormalization = FALSE`.

`LRange` Range of possible values for the number of change-points. Default value: `LRange = 0:30`.

tau	Positive real number corresponding to parameter c in Move 2 of Papastamoulis et al (2017). Default value: `tau = 0.05`.
gammaParameter	Positive real number corresponding to parameter α of the exponential prior distribution. Default value: `gammaParameter = 2`.
nu0	Positive real number corresponding to prior parameter ν_0 in Papastamoulis et al (2017). Default value: `nu0 = 0.1`.
alpha0	Positive real number corresponding to prior parameter α_0 in Papastamoulis et al (2017). Default value: `alpha0 = 1`.
beta0	Positive real number corresponding to prior parameter β_0 in Papastamoulis et al (2017). Default value: `beta0 = 1`.
subsetIndex	Optional subset of integers corresponding to time-series indexes. If not null, the sampler will be applied only to the specified subset.
saveTheta	Logical value indicating whether to save the MCMC output. Default: True.
sameVariance	Logical value indicating whether to assume the same variance per time-point across time-series. Default value: `sameVariance = TRUE`.
Prior	Character string specifying the prior distribution of the number of change-points. Allowed values: `Prior = "complexity"` (default) or `Prior = "Poisson"` (not suggested).

Value

The output of the sampler is returned as a list, with the following features:

`Cutpoint_posterior_median`
: The estimated medians per change-point, conditionally on the most probable number of change-points per time-series.

`Cutpoint_posterior_variance`
: The estimated variances per change-points, conditionally on the most probable number of change-points per time-series.

`NumberOfCutPoints_posterior_distribution`
: Posterior distributions of number of change-points per time-series.

`NumberOfCutPoints_MAP`
: The most probable number of change-points per time-series.

`Metropolis-Hastings_acceptance_rate`
: Acceptance of the MCMC move-types.

`subject_ID`
: the identifier of individual time-series.

`Cutpoint_mcmc_trace_map`
: The sampled values of each change-point per time series, conditionally on the MAP values.

`theta`
: The sampled values of the means per time-series, conditionally on the MAP values.

`nCutPointsTrace`
: The sampled values of the number of change-points, per time-series.

Note

The complexity prior distribution with parameter `gammaParameter = 2` is the default prior assumption imposed on the number of change-points. Smaller (larger) values of `gammaParameter` will a-priori support larger (respectively: smaller) number of change-points.

For completeness purposes, the Poisson distribution is also allowed in the `Prior`. In this latter case, the `gammaParameter` denotes the rate parameter of the Poisson distribution. Note that in this case the interpretation of `gammaParameter` is reversed: Smaller (larger) values of `gammaParameter` will a-priori support smaller (respectively: larger) number of change-points.

Author(s)

Panagiotis Papastamoulis

References

Papastamoulis P., Furukawa T., van Rhijn N., Bromley M., Bignell E. and Rattray M. (2017). Bayesian detection of piecewise linear trends in replicated time-series with application to growth data modelling. arXiv:1709.06111 [stat.AP]

Examples

```
## Not run:
# This example illustrates the package using a subset of four
#      time-series of the fungal dataset.
library('beast') # load package
data("FungalGrowthDataset") # load dataset
myIndex <- c(392, 62, 3, 117) # run the sampler only for the
#                                  specific subset of time-series
set.seed(1) # optional
# Run MCMC sampler:
run_mcmc <- beast(myDataList = FungalGrowthDataset, subsetIndex = myIndex,
zeroNormalization = TRUE)
# Print output:
print(run_mcmc)
# Plot output to file: "beast_plot.pdf"
plot(run_mcmc, fileName = "beast_plot.pdf", timeScale=1/6, xlab = "hours", ylab = "growth")
# NOTE 1: for a complete analysis remove the `subsetIndex = myIndex` argument.
# NOTE 2: `zeroNormalization = TRUE` is an optional argument that forces all
#    time-series to start from zero. It is not supposed to be used
#    for other applications.

## End(Not run)
```

`birthProbs` *Birth Probabilities*

Description

This function defines the probability of proposing an addition of a change-point.

Usage

```
birthProbs(LRange)
```

Arguments

LRange	The range of possible values for the number of change-points.

Value

probs	Vector of probabilities

Author(s)

Panagiotis Papastamoulis

`complexityPrior` *Complexity prior distribution*

Description

This function computes the complexity prior distribution on the number of change-points, defined as $f(\ell) = P(\ell_n = \ell) \propto e^{-\alpha\ell\log(bT/\ell)}, a, b > 0; \ell = 0, 1, 2, \ldots$. Note that this distribution has exponential decrease (Castillo and van der Vaart, 2012) when $b > 1 + e$, so we set $b = 3$.

Usage

```
complexityPrior(Lmax = 20, gammaParameter, nTime)
```

Arguments

Lmax	maximum number of change-points (default = 20).
gammaParameter	positive real number, corresponding to α.
nTime	positive integer denoting the total number of time-points.

Value

logPrior	Prior distribution values in the log-scale.

Author(s)

Panagiotis Papastamoulis

References

Castillo I. and van der Vaart A (2012). Needles and Straw in a Haystack: Posterior concentration for possibly sparse sequences. The Annals of Statistics, 40(4), 2069–2101.

computeEmpiricalPriorParameters
Compute the empirical mean.

Description

This function computes the empirical mean of the time-series.

Usage

```
computeEmpiricalPriorParameters(myDataList, nu0 = 1, alpha0 = 1, beta0 = 1)
```

Arguments

`myDataList`	Observed multivariate time-series.
`nu0`	positive real number.
`alpha0`	positive real number.
`beta0`	positive real number.

Value

`mu0`	Empirical mean

Author(s)

Panagiotis Papastamoulis

computePosteriorParameters
Compute empirical posterior parameters

Description

Compute empirical posterior parameters

Usage

```
computePosteriorParameters(myDataList, priorParameters)
```

Arguments

`myDataList`	Observed data.
`priorParameters`	Prior parameters.

Value

list of posterior parameters

Author(s)

Panagiotis Papastamoulis

`computePosteriorParametersFree` *Posterior parameters*

Description

Posterior parameters

Usage

```
computePosteriorParametersFree(myDataList, priorParameters)
```

Arguments

`myDataList` observed data.

`priorParameters` list of prior parameters.

Value

list of posterior parameters.

Author(s)

Panagiotis Papastamoulis

`FungalGrowthDataset` *Fungal Growth Dataset*

Description

Time-series dataset with $N \times R \times T$ growth levels for $R = 3$ replicates of $N = 411$ objects (mutants) measured every 10 minutes for $T = 289$ time-points. See Papastamoulis et al (2017) for a detailed description.

Usage

```
FungalGrowthDataset
```

Format

Time-series data.

References

Papastamoulis P., Furukawa T., van Rhijn N., Bromley M., Bignell E. and Rattray M. (2017). Bayesian detection of piecewise linear trends in replicated time-series with application to growth data modelling. arXiv:**1709.06111** [stat.AP]

`localProposal` *Move 3.b*

Description

Implements Move 3.b of the Metropolis-Hastings MCMC sampler.

Usage

```
localProposal(cutPoints, nTime, mhPropRange, startPoint)
```

Arguments

`cutPoints`	Current configuration of change-points.
`nTime`	Total number of time-points.
`mhPropRange`	Parameter d_2.
`startPoint`	Integer, at least equal to 2.

Value

`newState`	Candidate state of the chain.
`propRatio`	Proposal ratio.

Author(s)

Panagiotis Papastamoulis

`logLikelihoodFullModel` *Log-likelihood function.*

Description

Log-likelihood function.

Usage

```
logLikelihoodFullModel(myData, cutPoints, theta, startPoint)
```

Arguments

`myData`	data
`cutPoints`	change-points.
`theta`	means.
`startPoint`	optional integer at least equal to 2.

Value

log-likelihood value.

Author(s)

Panagiotis Papastamoulis

`logPrior` *Log-prior.*

Description

Logarithm of the prior distribution on the number of change-points.

Usage

```
logPrior(cutPoints, nTime, startPoint)
```

Arguments

`cutPoints`	change-points.
`nTime`	number of time-points.
`startPoint`	optional integer, at least equal to 2.

Value

logarithm of the prior distribution.

Author(s)

Panagiotis Papastamoulis

---

mcmcSampler *MCMC sampler*

Description

This function implements the Metropolis-Hastings MCMC sampler for individual time-series.

Usage

```
mcmcSampler(myData, nIter, finalIterationPdf, modelVariance, mhPropRange,
mhSinglePropRange, movesRange, startPoint, postPar, dName, timeScale,
burn, iterPerPlotPrefix, priorParameters, L = 3, LRange, tau,
gammaParameter, saveTheta, Prior = "complexity")
```

Arguments

`myData`	observed data.
`nIter`	number of mcmc iterations
`finalIterationPdf`	output folder
`modelVariance`	null
`mhPropRange`	positive integer
`mhSinglePropRange`	positive integer
`movesRange`	null
`startPoint`	positive integer
`postPar`	list of emprirically estimated parameters
`dName`	subject ID
`timeScale`	null
`burn`	burn-in period.
`iterPerPlotPrefix`	null
`priorParameters`	prior parameters.
`L`	null
`LRange`	range of possible values of the number of change-points.

`tau`	real.
`gammaParameter`	real.
`saveTheta`	TRUE.
`Prior`	character.

Value

MCMC output.

Author(s)

Panagiotis Papastamoulis

`myUnicodeCharacters` *Printing*

Description

Printing various unicode symbols.

Usage

```
myUnicodeCharacters()
```

Value

printed symbol

`normalizeTime0` *Zero normalization*

Description

Zero normalization at 1st time-point

Usage

```
normalizeTime0(myDataList)
```

Arguments

`myDataList`	data

Value

null

Author(s)

Panagiotis Papastamoulis

`plot.beast.object` *Plot function*

Description

This function plots objects returned by the `beast` function. All output is diverted to a pdf file provided in the `fileName` argument.

Usage

```
## S3 method for class 'beast.object'
plot(x, fileName, width, height, pointsize, ylab, xlab, timeScale, myPal, ...)
```

Arguments

`x`	An object of class `beast.object`, which is returned by the `beast` function.
`fileName`	Name of the output pdf file.
`width`	Width of pdf file. Default: 9
`height`	Height pdf file. Default: 6
`pointsize`	Pointsize. Default: 12
`ylab`	y-axis label. Default: x.
`xlab`	x-axis label. Default: t.
`timeScale`	A multiplicative-factor which will be used to scale the x-axis labels. For example, if time-points correspond to 10-minute periods, then setting `timeScale = 1/6` will make the x-axis labels correspond to hours. Default: `timeScale = 1`.
`myPal`	Vector of colors that will be used to produce the plot with all time-series overlayed. If the distinct values of the inferred numbers of change-points is less than 10, the `Set1` pallete of the `RColorBrewer` library is used. Otherwise, the user has to manually define the colors.
`...`	ignored.

Details

The function will produce a plot with all time-series coloured according to the corresponding number of change-points. In addition, it will generate individual plots per time-series displaying the observed data with boxplots which summarize the posterior distribution of change-points locations, conditionally on the most probable number of change-points.

Author(s)

Panagiotis Papastamoulis

`print.beast.object` *Print function*

Description

This function prints a summary of objects returned by the `beast` function.

Usage

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

Arguments

`x`	An object of class `beast.object`, which is returned by the `beast` function.
`...`	ignored.

Details

The function prints a summary of the most probable number (MAP) of change-points per time-series in the form of a table, as well as a list containing the MAP number of change-points and the corresponding locations (posterior medians) per time-series.

Author(s)

Panagiotis Papastamoulis

`proposeTheta` *Move 2*

Description

Proposes an update of θ according to Metropolis-Hastings move 2.

Usage

```
proposeTheta(thetaOld, tau, alpha, beta)
```

Arguments

`thetaOld`	Current values
`tau`	Parameter c.
`alpha`	null
`beta`	null

Value

`mean`	proposed values.

Author(s)

Panagiotis Papastamoulis

`simMultiIndNormInvGamma` *Prior random numbers*

Description

Generation of mean values according to the prior

Usage

```
simMultiIndNormInvGamma(mu, nu, alpha, beta)
```

Arguments

`mu`	means
`nu`	precision parameter
`alpha`	prior parameter
`beta`	prior parameter

Value

null

Author(s)

Panagiotis Papastamoulis

`simulateFromPrior` *Generate change-points according to the prior*

Description

Generate change-points according to the prior distribution conditionally on a given number of change-points.

Usage

```
simulateFromPrior(nTime, startPoint, L = 3)
```

Arguments

`nTime`	Number of time-points
`startPoint`	At least equal to 2.
`L`	null

Value

`cutPoints`	Change-point locations

Author(s)

Panagiotis Papastamoulis

`singleLocalProposal` *Move 3.b*

Description

Implement Metropolis-Hastings Move 3.b.

Usage

```
singleLocalProposal(cutPoints, nTime, mhSinglePropRange, startPoint)
```

Arguments

`cutPoints`	Current state
`nTime`	Number of time-points
`mhSinglePropRange`	Prior parameter.
`startPoint`	Optional.

Value

`newState`	Candidate state
`propRatio`	Proposal ratio

Author(s)

Panagiotis Papastamoulis

`truncatedPoisson` *Truncated Poisson pdf*

Description

Probability density function of the truncated Poisson distribution.

Usage

```
truncatedPoisson(Lmax = 20, gammaParameter = 1)
```

Arguments

`Lmax`	Max number
`gammaParameter`	Location parameter.

Value

`logPrior`	Log-pdf values

Author(s)

Panagiotis Papastamoulis

`updateNumberOfCutpoints`
Move 1

Description

Update the number of change-points according to Metropolis-Hastings move 1.

Usage

```
updateNumberOfCutpoints(cutPoints, nTime, startPoint, LRange, birthProbs)
```

Arguments

`cutPoints`	Current configuration
`nTime`	Number of time-points
`startPoint`	Optional integer
`LRange`	Range of possible values
`birthProbs`	Birth probabilities

Value

`newState`	Candidate state
`propRatio`	Proposal ratio

Author(s)

Panagiotis Papastamoulis

Index