Package ‘bcpmeta’

February 19, 2015

Type Package

Title Bayesian Multiple Changepoint Detection Using Metadata

Version 1.0

Date 2014-05-15

Author Yingbo Li

Maintainer Yingbo Li <ybli@clemson.edu>

Description A Bayesian approach to detect mean shifts in AR(1) time series while accommodating metadata (if available). In addition, a linear trend component is allowed.

Depends R (>= 2.14.0)

Imports mvtnorm

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-21 17:24:25

R topics documented:

bcpmeta-package *Bayesian Multiple Changepoint Detection Using Metadata*

Description

Package for a Bayesian multiple changepoint detection method to detect mean shifts in AR(1) time series. It accomodates metadta (if available) by letting metadata times have higher prior probabilities to be changepoints. The changepoint configuration with the highest posterior probability is the optimal model. Metropolis-Hastings algorithm is utilized for quick stochastic search of a potentially huge model space. This method is ideal for annual series, since it allows a linear trend component, but not yet monthly cycles.

Details

Package:	bcpmeta
Type:	Package
Version:	1.0
Date:	2014-05-15
License:	GPL(>= 2)

The most important functions of this package:
`bcpmeta.model`: find the optimal changepoint configuration,
`bcpmeta.parameters`: given a configuration, estimate model parameters.

Author(s)

Yingbo Li

Maintainer: Yingbo Li <ybli@clemson.edu>

References

Li, Y. and Lund, R. (2014) Bayesian Mulitple Changepoint Detection Using Metadata. (submitted)

bcpmeta.model *Identify the Optimal Changepoint Configuration*

Description

Implement a MCMC algorithm to quick search for the optimal changepoint configuration that has the largest posterior probability.

Usage

```
bcpmeta.model(X, meta, iter = 10000, thin = 10, trend = TRUE,
              EB = TRUE, mu0 = NULL, nu0 = 5, a1 = 1, a2 = 1,
              b1 = 19, b2 = 3, phi.lower = -0.99, phi.upper = 0.99,
              start.eta = NULL, track.time = TRUE, show.summary = TRUE,
              start.year = 1, meta.year = FALSE)
```

Arguments

`X`	a numerical vector. Observed time series.
`meta`	metadata. Either a vector of 0-1 indicators of the same length as `X`, or a numerical vector of the time indice of the metadata times.
`iter`	total number of iterations of MCMC.
`thin`	thinning; save one iteration in every `thin` number of iterations.
`trend`	logical indicating whether to allow the linear trend component.
`EB`	logical indicating whether to use the empirical Bayes method for *sigma^2* and *phi*.
`mu0`	prior mean of regime-wise means *mu_j*. If `NULL`, set to the default value `mean(X)`.
`nu0`	constant factor in prior variance of regim-wise means *mu_j*.
`a1`	the first parameter in the Beta-Binomial prior of non-metadata times.
`a2`	the first parameter in the Beta-Binomial prior of metadata times.
`b1`	the second parameter in the Beta-Binomial prior of non-metadata times.
`b2`	the second parameter in the Beta-Binomial prior of metadata times.
`phi.lower`	lower bound of the range of *phi*
`phi.upper`	upper bound of the range of *phi*
`start.eta`	initial value of the changepoint configuration *eta* for the MCMC. If `NULL`, generated randomly.
`track.time`	logical indicating whether to show process time.
`show.summary`	logical indicating whether to show the top 5 configurations.
`start.year`	year index of the first time point in the series.
`meta.year`	logical indicating whether `meta` is indexed in year, if it consists of the locations of the metadata times (instead of 0-1 indicators).

Details

A Metropolis-Hastings algorithm with interwine of two transitions, a component-wise updating and a simple random swapping. See `references` for details.

Value

`Eta` a `(iter/thin+1)` * `length(X)` matrix. Each row is a changepoint configuration visited by MCMC, in the format of a vector 0-1 indicators.

`map200` a `200 * (length(X) + 3)` matrix. The best 200 changepoint configurations, listed in desecending order. Each row is a vector of 0-1 indicators, followed by `lml` (log likelihood up to a constant), `phi.eb` (if `EB == TRUE`, the empirical Bayes estimate of *phi* under that configuration) and `lpost` (log posterior up to a constant).

`X` observed time series, same as the input value.

`meta` metadata, same as the input value.

`input.parameters` input parameters. Use command `names` to check its components.

Author(s)

Yingbo Li

Maintainer: Yingbo Li <ybli@clemson.edu>

References

Li, Y. and Lund, R. (2014) Bayesian Mulitple Changepoint Detection Using Metadata. (submitted)

See Also

Function `marginal.plot` uses the output of this function as input.

Examples

```
## Create a time series of length 200 with three mean shifts at 50, 100, 150.
data = simgen(2, 1);
X = data$X[1, ];  ## time series
meta = data$meta;  ## locations of metadata times

## For illustration purpose, number of MCMC iteration is set to a small value.
results = bcpmeta.model(X, meta = meta, iter = 1e3, trend = FALSE);
```

`bcpmeta.parameters` *Estimate Model Parameters under a Given Configuration*

Description

Given a changepoint configuration, use Gibbs sampler (or Metropolis-Hastings algorithm within Gibbs) to find posterior mean estimates of model parameters.

Usage

```
bcpmeta.parameters(X, meta, eta, iter = 10000, thin = 10, trend = TRUE,
                   EB = TRUE, mu0 = NULL, nu0 = 5, phi.lower = -0.99,
                   phi.upper = 0.99, sd.xi = 0.1, start.phi = NULL,
                   burnin = 0.2, track.time = TRUE, show.summary = TRUE,
                   start.year = 1, meta.year = FALSE, eta.year = FALSE)
```

Arguments

`X`	a numerical vector. Observed time series.
`meta`	metadata. Either a vector of 0-1 indicators of the same length as `X`, or a numerical vector of the time indice of the metadata times.
`eta`	the changepoint configuration. Either a vector of 0-1 indicators of the same length as `X`, or a numerical vector of the time indice of the changepoint times.
`iter`	total number of iterations of MCMC.
`thin`	thinning; save one iteration in every `thin` number of iterations.
`trend`	logical indicating whether to allow the linear trend component.
`EB`	logical indicating whether to use the empirical Bayes method for *sigma^2* and *phi*.
`mu0`	prior mean of regime-wise means *mu_j*. If `NULL`, set to the default value `mean(X)`.
`nu0`	constant factor in prior variance of regim-wise means *mu_j*.
`phi.lower`	lower bound of the range of *phi*
`phi.upper`	upper bound of the range of *phi*
`sd.xi`	standard deviation of the jump proposal of *log(phi)* in Metropolis-Hastings updating when the fully Bayes method is used.
`start.phi`	initial value *phi* for the MCMC when the fully Bayes method is used. If `NULL`, generated randomly.
`burnin`	the ratio of burnin length compared with the total length of MCMC. All posterior mean estimates are calculated without burnin periods.
`track.time`	logical indicating whether to show process time.
`show.summary`	logical indicating whether to show the estimates of parameters.
`start.year`	year index of the first time point in the series.
`meta.year`	logical indicating whether `meta` is indexed in year, if it consists of the locations of the metadata times (instead of 0-1 indicators).
`eta.year`	logical indicating whether `eta` is indexed in year, if it consists of the locations of the metadata times (instead of 0-1 indicators).

Details

Conditional on the given changepoint configuration *eta*, the posterior mean estimates of regime-wise mean *mu* and trend *alphla* (if `trend == TRUE`) is obtained via Gibbs sampler. If `EB == TRUE`, empirical Bayes estimates of *sigma2* and *phi* are given; otherwise, fully Bayes estimates of them are obtained via Gibbs sampler and Metropolis-Hastings algorithm, under Jeffreys prior and uniform prior respectively.

Value

- `Phi` — the empirical Bayes estimate of *phi* if `EB == TRUE`; or a vector of length `(iter/thin)`, the MCMC samples of *phi* if `EB == FALSE`.
- `Sigmasq` — the empirical Bayes estimate of *sigma2* if `EB == TRUE`; or a vector of length `(iter/thin)`, the MCMC samples of *sigma2* if `EB == FALSE`.
- `Alpha` — a vector of length `(iter/thin)`, the MCMC samples of *alpha* if `trend == TRUE`; or `0` if `trend == FALSE`.
- `Mu` — a `(iter/thin) * (sum(eta)+1)` matrix. Each row is a MCMC sample of *mu*.
- `phi.est` — the empirical Bayes estimate of *phi* if `EB == TRUE`; or the posterior mean if `EB == FALSE`.
- `sigmasq.est` — the empirical Bayes estimate of *sigma2* if `EB == TRUE`; or the posterior mean if `EB == FALSE`.
- `alpha.est` — posterior mean estimate of *alpha*
- `mu.est` — a vector of length `sum(eta)+1`, posterior mean estimate of *mu*
- `X` — observed time series, same as the input value.
- `meta` — metadata, same as the input value.
- `input.parameters` — input parameters. Use command `names` to check its components.
- `change.phi` — ratio of accepting a new phi in the MCMC chain, if `EB == FALSE`.

Author(s)

Yingbo Li

Maintainer: Yingbo Li <ybli@clemson.edu>

References

Li, Y. and Lund, R. (2014) Bayesian Mulitple Changepoint Detection Using Metadata. (submitted)

See Also

Function `cp.plot` uses the output of this function as input.

Examples

```
## Create a time series of length 200 with three mean shifts at 50, 100, 150.
data = simgen(2, 1);
X = data$X[1, ];  ## time series
meta = data$meta;  ## locations of metadata times

## Parameter estimation in the configuration where changepoints are time 50 and 99.
results = bcpmeta.parameters(X, meta = meta, eta = c(50, 99), trend = FALSE);
```

cp.plot *Plot a Changepoint Configuration*

Description

Plot regimes-means (dashed line) against observational series (solid line).

Usage

```
cp.plot(results.parameter, meta.loc = NULL, cex = 1, file.name = NULL, ...)
```

Arguments

`results.parameter`
: output of function `bcpmeta.parameters`, etimates of parameters in a certain changepoint configuration.

`meta.loc`
: the y-coordinate of the x-axis, for the purpose of mark crosses on the x-axis to indicate metadata locations; optional.

`cex`
: width (size) of lines and lables.

`file.name`
: optional; if specified, then the plot is saved to a .ps file under this file name.

`...`
: Arguments to be passed to methods, such as graphical parameters (see `par`).

Details

Metadata times are marked as crosses on the x-axis, if argument `meta.loc` is not `NULL`.

Author(s)

Yingbo Li

Maintainer: Yingbo Li <ybli@clemson.edu>

References

Li, Y. and Lund, R. (2014) Bayesian Mulitple Changepoint Detection Using Metadata. (submitted)

See Also

Function `bcpmeta.parameters`

Examples

```
## Create a time series of length 200 with three mean shifts at 50, 100, 150.
data = simgen(2, 1);
X = data$X[1, ];  ## time series
meta = data$meta;  ## locations of metadata times

## Parameter estimation in the configuration where changepoints are time 50 and 99.
```

```
results = bcpmeta.parameters(X, meta = meta, eta = c(50, 99), trend = FALSE);

## Plot
cp.plot(results, meta.loc = -0.42, cex = 1.5);
```

`marginal.plot` *Plot Posterior Marginal Inclusion Probabilities*

Description

For each time point in the time series, the posterior probability of it being a changepoint time is computed using MCMC method and is plotted as height of the bar here.

Usage

```
marginal.plot(results.mcmc, meta.loc = NULL, cex = 1, burnin = 0.2, file.name = NULL, ...)
```

Arguments

`results.mcmc`	output of function `bcpmeta.model`, record of configurations that are visited by the MCMC.
`meta.loc`	the y-coordinate of the x-axis, for the purpose of mark crosses on the x-axis to indicate metadata locations; optional.
`cex`	width (size) of lines and lables.
`burnin`	the ratio of burnin length compared with the total length of MCMC. Estimates of posterior inclusion probabilites are calculated without burnin periods.
`file.name`	optional; if specified, then the plot is saved to a .ps file under this file name.
`...`	Arguments to be passed to methods, such as graphical parameters (see `par`).

Details

Metadata times are marked as crosses on the x-axis, if argument `meta.loc` is not `NULL`.

Author(s)

Yingbo Li

Maintainer: Yingbo Li <ybli@clemson.edu>

References

Li, Y. and Lund, R. (2014) Bayesian Mulitple Changepoint Detection Using Metadata. (submitted)

See Also

Function `bcpmeta.model`

Examples

```
## Create a time series of length 200 with three mean shifts at 50, 100, 150.
data = simgen(2, 1);
X = data$X[1, ];  ## time series
meta = data$meta;  ## locations of metadata times

## For illustration purpose, number of MCMC iteration is set to a small value.
results = bcpmeta.model(X, meta = meta, iter = 1e3, trend = FALSE);

marginal.plot(results, xlab = 'time', ylab = 'probability');
```

---

simgen *Generate Simulation Data*

Description

Generate independent time series to apply changepoint detection methods on, according to the rules described in the reference, Section 5, Sceario 1 - 3.

Usage

```
simgen(scenario, N = 1000)
```

Arguments

`scenario`	scenario type. Can be chosen from 1, 2, or 3.
`N`	number of independent series to generate.

Value

`X`	a `N * 200` matrix, each row is a simulated time series.
`meta`	locations of metadata times.
`scenario`	scenario type. Can be chosen from 1, 2, or 3; the same as input.
`N`	number of independent series to generate; the same as input.

Author(s)

Yingbo Li

Maintainer: Yingbo Li <ybli@clemson.edu>

References

Li, Y. and Lund, R. (2014) Bayesian Mulitple Changepoint Detection Using Metadata. (submitted)

See Also

`bcpmeta.model`, `bcpmeta.parameters`

Examples

```
## Create a time series of length 200 with three mean shifts at 50, 100, 150.
data = simgen(2, 1);
X = data$X[1, ];  ## time series
meta = data$meta;  ## locations of metadata times
```

Index