Package 'SKFCPD'

February 18, 2024

Type Package

Title Fast Online Changepoint Detection for Temporally Correlated Data

Version 0.2.4

Date 2024-02-15

Maintainer Hanmo Li <hanmo@pstat.ucsb.edu>

Author Hanmo Li [aut, cre],
Yuedong Wang [aut],
Mengyang Gu [aut]

Description Sequential Kalman filter for scalable online changepoint detection by temporally correlated data. It enables fast single and multiple change points with missing values. See the reference: Hanmo Li, Yuedong Wang, Mengyang Gu (2023), <arXiv:2310.18611>.

License GPL (>= 3)

Depends R (>= 3.5.0), methods (>= 4.2.2), rlang (>= 1.0.6), ggplot2 (>= 3.4.0), ggpubr (>= 0.5.0), reshape2 (>= 1.4.4), FastGaSP (>= 0.5.2)

Imports Rcpp (>= 1.0.9)

LinkingTo Rcpp, RcppEigen

NeedsCompilation yes

Encoding UTF-8

Repository CRAN

Date/Publication 2024-02-17 23:30:12 UTC

R topics documented:

SKFCPD-package *Dynamic Linear Model for Online Changepoint Detection*

Description

The 'SKFCPD' package provides estimation of changepoint locations using the Dynamic Linear Model (DLM) within the Bayesian Online Changepoint Detection (BOCPD) framework. The efficient computation is achieved through implementation of the Sequential Kalman filter. The range parameter and noise-to-signal ratio are estimated from training samples via a Gaussian process model. This package is capable of handling multidimensional data with temporal correlations and random missing patterns.

Details

The DESCRIPTION file:

Package:	SKFCPD
Type:	Package
Title:	Fast Online Changepoint Detection for Temporally Correlated Data
Version:	0.2.4
Date:	2024-02-15
Authors@R:	c(person(given="Hanmo",family="Li",role=c("aut", "cre"), email="hanmo@pstat.ucsb.edu"), person(giv
Maintainer:	Hanmo Li <hanmo@pstat.ucsb.edu>
Author:	Hanmo Li [aut, cre], Yuedong Wang [aut], Mengyang Gu [aut]
Description:	Sequential Kalman filter for scalable online changepoint detection by temporally correlated data. It enab
License:	GPL (>= 3)
Depends:	R (>= 3.5.0), methods (>= 4.2.2), rlang (>= 1.0.6), ggplot2 (>= 3.4.0), ggpubr (>= 0.5.0), reshape2 (>=
Imports:	Rcpp (>= 1.0.9)
LinkingTo:	Rcpp, RcppEigen
NeedsCompilation:	yes
Encoding:	UTF-8
Packaged:	2024-02-15 11:15:56 UTC; lihan
Archs:	x64

Index of help topics:

Implements a fast online changepoint detection algorithm using dynamic linear model based on Sequential Kalman filter. It's for temporally correlated data and accepts multi-dimensional datasets

with missing values.

Author(s)

Hanmo Li [aut, cre], Yuedong Wang [aut], Mengyang Gu [aut]

Maintainer: Hanmo Li <hanmo@pstat.ucsb.edu>

References

Li, Hanmo, Yuedong Wang, and Mengyang Gu. *Sequential Kalman filter for fast online change-point detection in longitudinal health records.* arXiv preprint arXiv:2310.18611 (2023).

Fearnhead, Paul, and Zhen Liu. *On-line inference for multiple changepoint problems.* Journal of the Royal Statistical Society Series B: Statistical Methodology 69, no. 4 (2007): 589-605.

Adams, Ryan Prescott, and David JC MacKay. *Bayesian online changepoint detection.* arXiv preprint arXiv:0710.3742 (2007).

Hartikainen, Jouni, and Simo Sarkka. *Kalman filtering and smoothing solutions to temporal Gaussian process regression models.* In 2010 IEEE international workshop on machine learning for signal processing, pp. 379-384. IEEE, 2010.

Gu, Mengyang, and Yanxun Xu. *Fast nonseparable Gaussian stochastic process with application to methylation level interpolation.* Journal of Computational and Graphical Statistics 29, no. 2 (2020): 250-260.

Gu, Mengyang, and Weining Shen. *Generalized probabilistic principal component analysis of correlated data.* The Journal of Machine Learning Research 21, no. 1 (2020): 428-468.

Gu, Mengyang, Xiaojing Wang, and James O. Berger. *Robust Gaussian stochastic process emulation.* The Annals of Statistics 46, no. 6A (2018): 3038-3066.

See Also

SKFCPD

Examples

```
library(SKFCPD)

#-------------------------------------------------------------------------------
# Example: fast online changepoint detection with DEPENDENT data.
#
# Data generation: Data follows a multidimensional Gaussian process with Matern 2.5 kernel.
#-------------------------------------------------------------------------------
# Data Generation
set.seed(1)

n_obs = 150
n_dim = 2
seg_len = c(70, 30, 20,30)
mean_each_seg = c(0,1,-1,0)

x_mat=matrix(1:n_obs)
y_mat=matrix(NA, nrow=n_obs, ncol=n_dim)
```

```
gamma = rep(5, n_dim) # range parameter of the covariance matrix

# compute the matern 2.5 kernel
construct_cor_matrix = function(input, gamma){
  n = length(input)
  R0=abs(outer(input,(input),'-'))
  matrix_one = matrix(1, n, n)
  const = sqrt(5) * R0 / gamma
  Sigma = (matrix_one + const + const^2/3) * (exp(-const))
  return(Sigma)
}

for(j in 1:n_dim){
  y_each_dim = c()
  for(i in 1:length(seg_len)){
    nobs_per_seg = seg_len[i]
    Sigma = construct_cor_matrix(1:nobs_per_seg, gamma[j])
    L=t(chol(Sigma))
    theta=rep(mean_each_seg[i],nobs_per_seg)+L%*%rnorm(nobs_per_seg)
    y_each_dim = c(y_each_dim, theta+0.1*rnorm(nobs_per_seg))
  }
  y_mat[,j] = y_each_dim
}

## Detect changepoints by SKFCPD
Online_CPD_1 = SKFCPD(design = x_mat,
                      response = y_mat,
                      train_prop = 1/3)

## visulize the results
plot_SKFCPD(Online_CPD_1)
```

`Estimate_GP_params` *Estimate parameters from fast computation of GaSP model*

Description

Getting the estimated parameters from fast computation of the Gaussian stochastic process (GaSP) model with the Matern kernel function with a noise.

Usage

```
Estimate_GP_params(input, output, kernel_type='matern_5_2')
```

Arguments

`input`	a vector with dimension num_obs x 1 for the sorted input locations.
`output`	a vector with dimension n x 1 for the observations at the sorted input locations.

`kernel_type` a `character` to specify the type of kernel to use. The current version supports kernel_type to be "matern_5_2" or "exp", meaning that the matern kernel with roughness parameter being 2.5 or 0.5 (power exponent kernel), respectively.

Value

`Estimate_GP_params` returns an S4 object of class `Estimated_GP_params` with estimated parameters including

`beta`	the inverse range parameter, i.e. beta=1/gamma
`eta`	the noise-to-signal ratio
`sigma_2`	the variance parameter

Author(s)

Hanmo Li [aut, cre], Yuedong Wang [aut], Mengyang Gu [aut]

Maintainer: Hanmo Li <hanmo@pstat.ucsb.edu>

References

Hartikainen, Jouni, and Simo Sarkka. *Kalman filtering and smoothing solutions to temporal Gaussian process regression models.* In 2010 IEEE international workshop on machine learning for signal processing, pp. 379-384. IEEE, 2010.

Gu, Mengyang, and Yanxun Xu. *Fast nonseparable Gaussian stochastic process with application to methylation level interpolation.* Journal of Computational and Graphical Statistics 29, no. 2 (2020): 250-260.

Gu, Mengyang, and Weining Shen. *Generalized probabilistic principal component analysis of correlated data.* The Journal of Machine Learning Research 21, no. 1 (2020): 428-468.

Gu, Mengyang, Xiaojing Wang, and James O. Berger. *Robust Gaussian stochastic process emulation.* The Annals of Statistics 46, no. 6A (2018): 3038-3066.

Examples

```
library(SKFCPD)

#--------------------------------------------------------------------------------
# simple example with noise
#--------------------------------------------------------------------------------

y_R<-function(x){
  cos(2*pi*x)
}
###let's test for 100 observations
set.seed(1)
num_obs=100
input=runif(num_obs)
output=y_R(input)+rnorm(num_obs,mean=0,sd=1)
```

```
## run Estimate_GP_params to get estimated parameters
params_est = Estimate_GP_params(input, output)
print(params_est@beta) ## inverse of range parameter
print(params_est@eta) ## noise-to-signal ratio
print(params_est@sigma_2) ## variance
```

`plot_SKFCPD` *Plot for SKFCPD model*

Description

Function to make plots on SKFCPD models after the SKFCPD model has been constructed.

Usage

```
plot_SKFCPD(x, type = "cp")
```

Arguments

`x`	an object of class `SKFCPD`.
`type`	A character specifying the type of plot. `cp` plots the data with estimated changepoints marked in red crossings. `run_length_posterior` plots the matrix of run length posterior distribution.

Value

Two plots: (1) plot of data with the red dashed lines mark the estimated changepoint locations, and (2) plot of the run length posterior distribution matrix. For multidimensional data, only the first dimension is plotted.

Author(s)

Hanmo Li [aut, cre], Yuedong Wang [aut], Mengyang Gu [aut]

Maintainer: Hanmo Li <hanmo@pstat.ucsb.edu>

References

Li, Hanmo, Yuedong Wang, and Mengyang Gu. *Sequential Kalman filter for fast online changepoint detection in longitudinal health records.* arXiv preprint arXiv:2310.18611 (2023).

Examples

```
library(SKFCPD)

#------------------------------------------------------------------------------
# Example: fast online changepoint detection with DEPENDENT data.
#
# Data generation: Data follows a multidimensional Gaussian process with Matern 2.5 kernel.
```

```
#------------------------------------------------------------------------------
# Data Generation
set.seed(1)

n_obs = 150
n_dim = 2
seg_len = c(70, 30, 20,30)
mean_each_seg = c(0,1,-1,0)

x_mat=matrix(1:n_obs)
y_mat=matrix(NA, nrow=n_obs, ncol=n_dim)

gamma = rep(5, n_dim) # range parameter of the covariance matrix

# compute the matern 2.5 kernel
construct_cor_matrix = function(input, gamma){
  n = length(input)
  R0=abs(outer(input,(input),'-'))
  matrix_one = matrix(1, n, n)
  const = sqrt(5) * R0 / gamma
  Sigma = (matrix_one + const + const^2/3) * (exp(-const))
  return(Sigma)
}

for(j in 1:n_dim){
  y_each_dim = c()
  for(i in 1:length(seg_len)){
    nobs_per_seg = seg_len[i]
    Sigma = construct_cor_matrix(1:nobs_per_seg, gamma[j])
    L=t(chol(Sigma))
    theta=rep(mean_each_seg[i],nobs_per_seg)+L%*%rnorm(nobs_per_seg)
    y_each_dim = c(y_each_dim, theta+0.1*rnorm(nobs_per_seg))
  }
  y_mat[,j] = y_each_dim
}

## Detect changepoints by SKFCPD
Online_CPD_1 = SKFCPD(design = x_mat,
                      response = y_mat,
                      train_prop = 1/3)

## visulize the results
plot_SKFCPD(Online_CPD_1)
```

SKFCPD *Getting the results of the SKFCPD model*

Description

Estimating changepoint locations using the Dynamic Linear Model (DLM) within the Bayesian Online Changepoint Detection (BOCPD) framework. The efficient computation is achieved through

implementation of the Kalman filter. The range parameter and noise-to-signal ratio are estimated from training samples via a Gaussian process model. This function is capable of handling multidimensional data with temporal correlations and random missing patterns.

Usage

```
SKFCPD(design = NULL, response = NULL, FCPD = NULL,
init_params = list(gamma = 1, sigma_2 = 1, eta = 1),
train_prop = NULL, kernel_type = "matern_5_2",
hazard_vec=100, print_info = TRUE, truncate_at_prev_cp = FALSE)
```

Arguments

design	A vector with the length of n. The design of the experiment.
response	A matrix with dimension n x q. The observations.
FCPD	An object of the class SKFCPD computed in the previous run of the algorithm.
init_params	A list with estimated range parameter gamma, noise-to-signal parameter eta and variance parameter sigma_2. The default values are gamma=1, eta=1, and sigma_2=1.
train_prop	A numerical value between 0 and 1. The propotation of training samples for parameter estimation. When train_prop=NULL, we skip the training process and specify the parameter values in the argument init_params.
kernel_type	A character specifying the type of kernels of the input. matern_5_2 are Matern correlation with roughness parameter 5/2. exp is power exponential correlation with roughness parameter alpha=2. The default choice is matern_5_2.
hazard_vec	Either a constant or a vector with the length of n. The hazard vector in the SKFCPD method. hazard_vec = 1/hazard_const is the prior probability that a changepoint occur at any time points. The default value of hazard_vec is 100.
print_info	This setting prints out updates on the progress of the algorithm if set to TRUE.
truncate_at_prev_cp	If TRUE, truncate the run length at the most recently detected changepoint. The default value of truncate_at_prev_cp is FALSE.

Value

SKFCPD returns a S4 object of class SKFCPD (see SKFCPD-class).

Author(s)

Hanmo Li [aut, cre], Yuedong Wang [aut], Mengyang Gu [aut]

Maintainer: Hanmo Li <hanmo@pstat.ucsb.edu>

References

Li, Hanmo, Yuedong Wang, and Mengyang Gu. *Sequential Kalman filter for fast online change-point detection in longitudinal health records.* arXiv preprint arXiv:2310.18611 (2023).

Fearnhead, Paul, and Zhen Liu. *On-line inference for multiple changepoint problems.* Journal of the Royal Statistical Society Series B: Statistical Methodology 69, no. 4 (2007): 589-605.

Adams, Ryan Prescott, and David JC MacKay. *Bayesian online changepoint detection.* arXiv preprint arXiv:0710.3742 (2007).

Hartikainen, Jouni, and Simo Sarkka. *Kalman filtering and smoothing solutions to temporal Gaussian process regression models.* In 2010 IEEE international workshop on machine learning for signal processing, pp. 379-384. IEEE, 2010.

Examples

```
library(SKFCPD)

#-------------------------------------------------------------------------------
# Example: fast online changepoint detection with DEPENDENT data.
#
# Data generation: Data follows a multidimensional Gaussian process with Matern 2.5 kernel.
#-------------------------------------------------------------------------------
# Data Generation
set.seed(1)

n_obs = 150
n_dim = 2
seg_len = c(70, 30, 20,30)
mean_each_seg = c(0,1,-1,0)

x_mat=matrix(1:n_obs)
y_mat=matrix(NA, nrow=n_obs, ncol=n_dim)

gamma = rep(5, n_dim) # range parameter of the covariance matrix

# compute the matern 2.5 kernel
construct_cor_matrix = function(input, gamma){
  n = length(input)
  R0=abs(outer(input,(input),'-'))
  matrix_one = matrix(1, n, n)
  const = sqrt(5) * R0 / gamma
  Sigma = (matrix_one + const + const^2/3) * (exp(-const))
  return(Sigma)
}

for(j in 1:n_dim){
  y_each_dim = c()
  for(i in 1:length(seg_len)){
    nobs_per_seg = seg_len[i]
    Sigma = construct_cor_matrix(1:nobs_per_seg, gamma[j])
    L=t(chol(Sigma))
    theta=rep(mean_each_seg[i],nobs_per_seg)+L%*%rnorm(nobs_per_seg)
    y_each_dim = c(y_each_dim, theta+0.1*rnorm(nobs_per_seg))
  }
  y_mat[,j] = y_each_dim
}
```

```
## Detect changepoints by SKFCPD
Online_CPD_1 = SKFCPD(design = x_mat,
                      response = y_mat,
                      train_prop = 1/3)

## visulize the results
plot_SKFCPD(Online_CPD_1)
```

SKFCPD-class *Class* "SKFCPD"

Description

S4 class for SKFCPD where the range parameter and noise-to-signal parameters are estimated from the training samples.

Objects from the Class

Objects of this class are created and initialized with the function SKFCPD that computes the calculations needed for setting up the analysis.

Slots

design: Object of class "matrix" with dimension n x p. The design of the experiment.

response: Object of class "matrix" with dimension n x q. The observations.

test_start: Object of class "numeric". The starting index of test period.

kernel_type: Object of class "character" to specify the type of kernel to use.

gamma: Object of class "vector" with dimension q x 1. The range parameters.

eta: Object of class "vector" with dimension q x 1. The noise-to-signal ratio.

sigma_2: Object of class "vector" with dimension q x 1. The variance parameters.

hazard_vec: Object of class "numeric". The n x 1 hazard vector in the FastCPD method.

KF_params_list: Object of class "list". The list of Kalman filter parameters from the previous run of the algorithm.

prev_L_params_list: Object of class "list". The list of parameters for calculating the quadratic form of the inverse covariance matrix from the previous run of the algorithm.

run_length_posterior_mat: Object of class "matrix" with dimension n x n. The posterior distribution of the run length.

run_length_joint_mat: Object of class "matrix" with dimension n x n. The joint distribution of the run length and the observations.

log_pred_dist_mat: Object of class "matrix" with dimension n x n. The logrithm of the predictive distribution of observations.

cp: Object of class "vector" with length m. The location of estimated changepoints.

Author(s)

Hanmo Li [aut, cre], Yuedong Wang [aut], Mengyang Gu [aut]

Maintainer: Hanmo Li <hanmo@pstat.ucsb.edu>

References

Li, Hanmo, Yuedong Wang, and Mengyang Gu. *Sequential Kalman filter for fast online change-point detection in longitudinal health records.* arXiv preprint arXiv:2310.18611 (2023).

Fearnhead, Paul, and Zhen Liu. *On-line inference for multiple changepoint problems.* Journal of the Royal Statistical Society Series B: Statistical Methodology 69, no. 4 (2007): 589-605.

Adams, Ryan Prescott, and David JC MacKay. *Bayesian online changepoint detection.* arXiv preprint arXiv:0710.3742 (2007).

Hartikainen, Jouni, and Simo Sarkka. *Kalman filtering and smoothing solutions to temporal Gaussian process regression models.* In 2010 IEEE international workshop on machine learning for signal processing, pp. 379-384. IEEE, 2010.

See Also

SKFCPD for more details about how to create a SKFCPD object.

Index