Package 'DynamicGP'

April 18, 2018

Title Local Gaussian Process Model for Large-Scale Dynamic Computer Experiments

Version 1.0-2

Description Fits localized GP model for dynamic computer experiments via singular value decomposition of the response matrix Y for large N (the number of observations) using the algorithm proposed by Zhang et al. (2018) <arXiv:1611.09488>. The current version only supports 64-bit version of R.

License GPL (>= 2)

Date 2018-04-17

Maintainer Ru Zhang <heavenmarshal@gmail.com>

Depends R (>= 2.14)

Imports lhs, laGP, parallel

NeedsCompilation yes

Author Ru Zhang [aut, cre],
Chunfang Devon Lin [aut],
Pritam Ranjan [aut],
Robert B Gramacy [ctb],
Nicolas Devillard [ctb],
Jorge Nocedal [ctb],
Jose Luis Morales [ctb],
Ciyou Zhu [ctb],
Richard Byrd [ctb],
Peihuang Lu-Chen [ctb],
University of Chicago [cph],
University of California [cph]

Repository CRAN

Date/Publication 2018-04-18 09:45:47 UTC

R topics documented:

`DynamicGP-package` *GP Models for Large-Scale Dynamic Computer Experiments*

Description

This R package provides three functions for emulating dynamic computer experiments. The function `svdGP` fits full SVD-based GP model which is computationally demanding for large-scale dyanmic computer experiments. As is well known, the time complexity of fitting a GP model is $O(N^3)$ where N is the number of training/design points. Since fitting a common GP model for really large N would be computationally burdensome, we fit local SVD-based GP models on a sequentially selected small neighborhood set for every test inputs. The function `knnsvdGP` fits K-nearest neighbor SVD-based GP models which selects neighborhood sets based on the Euclidean distance with repect to the test points. The function `lasvdGP` fits local approximate SVD-based GP model using the new algorithm proposed by Zhang et al. (2018).

The lasvdGP is an extension of the local approximate GP (laGP) model developed by Gramacy and Lee (2015) for the emulation of large-scale scalar valued computer experiments. The neighborhood selection and SVD-based GP model fitting algorithm is suitable for parallelization. We use the R package "parallel" for this task. The parallelization can achieve nearly linear speed since the procedure on each test point is independent and identical.

Author(s)

Ru Zhang <heavenmarshal@gmail.com>,

C. Devon Lin <devon.lin@queensu.ca>,

Pritam Ranjan <pritam.ranjan@gmail.com>

References

Gramacy, R. B. and Apley, D. W. (2015) *Local Gaussian process approximation for large computer experiments*, Journal of Computational and Graphical Statistics 24(2), 561-578.

Zhang, R., Lin, C. D. and Ranjan, P. (2018) *Local Gaussian Process Model for Large-scale Dynamic Computer Experiments*, arXiv:1611.09488.

knnsvdGP *K-nearest neighbor SVD-Based GP model*

Description

Fits a K-nearest neighbour SVD-based GP model on a test set `X0`, training set `design` and response matrix `resp`. The local neighbourhood sets consist of `nn` points which are selected by the Euclidean distance with respect to the test points. See Zhang et al. (2018) for details.

Usage

```
knnsvdGP(design, resp, X0=design, nn=20, nsvd = nn, frac = .9,
         gstart = 0.0001, nstarts = 5,centralize=FALSE, maxit=100, verb=0,
         nthread = 4, clutype="PSOCK")
```

Arguments

`design`	An N by d matrix of N training/design inputs.
`resp`	An L by N response matrix of `design`, where L is the length of the time series outputs, N is the number of design points.
`X0`	An M by d matrix of M test inputs. The localized SVD-based GP models will be fitted on every point (row) of `X0`. The default value of `X0` is `design`.
`nn`	The number of neighborhood points selected by the Euclidean distance. the default value is 20.
`nsvd`	The number of design points closest to the test points on whose response matrix to perform the initial singular value decomposition. The default value is `nn`.
`frac`	The threshold in the cumulative percentage criterion to select the number of SVD bases. The default value is 0.9.
`gstart`	The starting number and upper bound of for estimating the nugget parameter. If `gstart = sqrt(.Machine$double.eps)`, the nugget will be fixed at `sqrt(.Machine$double.eps)`, since the it is the lower bound of the nugget term. The default value is 0.0001.
`nstarts`	The number of starting points used in the numerical maximization of the posterior density function. The larger `nstarts` will typically lead to more accurate prediction but longer computational time. The default value is 5.
`centralize`	If `centralize=TRUE` the response matrix will be centralized (subtract the mean) before the start of the algorithm. The mean will be added to the predictive mean at the finish of the algorithm. The default value is `FALSE`.
`maxit`	Maximum number of iterations in the numerical optimization algorithm for maximizing the posterior density function. The default value is 100.
`verb`	A nonnegative integer indicates the level of printing on the screen. If `verb=0` the function is executed in silence. The default value is 0.
`nthread`	The number of threads (processes) used in parallel execution of this function. `nthread=1` implies no parallelization. The default value is 4.

clutype — The type of cluster in the R package "parallel" to perform parallelization. The default value is "PSOCK". Required only if nthread>1.

Value

pmean — An L by M matrix of posterior predicted mean for the response at the test set X0.

ps2 — An L by M matrix of posterior predicted variance for the response at the test set X0.

Author(s)

Ru Zhang <heavenmarshal@gmail.com>,

C. Devon Lin <devon.lin@queensu.ca>,

Pritam Ranjan <pritam.ranjan@gmail.com>

References

Zhang, R., Lin, C. D. and Ranjan, P. (2018) *Local Gaussian Process Model for Large-scale Dynamic Computer Experiments*, arXiv:1611.09488.

See Also

lasvdGP, svdGP.

Examples

```
library("lhs")
forretal <- function(x,t,shift=1)
{
    par1 <- x[1]*6+4
    par2 <- x[2]*16+4
    par3 <- x[3]*6+1
    t <- t+shift
    y <- (par1*t-2)^2*sin(par2*t-par3)
}
timepoints <- seq(0,1,len=200)
design <- lhs::randomLHS(100,3)
test <- lhs::randomLHS(20,3)
## evaluate the response matrix on the design matrix
resp <- apply(design,1,forretal,timepoints)

nn <- 20
gs <- sqrt(.Machine$double.eps)
## knnsvdGP with mutiple (5) start points for GP model estimation
## It use the R package "parallel" for parallelization
retknnmsp <- knnsvdGP(design,resp,test,nn,frac=.95,gstart=gs,
                      centralize=TRUE,nstarts=5,nthread=2,clutype="PSOCK")

## knnsvdGP with single start point for GP model estimation
## It does not use parallel computation
```

```
retknnss <- knnsvdGP(design,resp,test,nn,frac=.95,gstart=gs,
                     centralize=TRUE,nstarts=1,nthread=1)
```

---

lasvdGP *Local Approximate SVD-Based GP Models*

Description

Fits a local approximate SVD-based GP model on a test set `X0`, training/design set `design` and response matrix `resp`. The local neighborhood sets consist of `nn` out of which `n0` points are selected by the Euclidean distance with respect to the test points. The remaining `nn-n0` neighborhood points are selected sequentially by a greedy algorithm proposed by Zhang et al. (2018).

Usage

```
lasvdGP(design, resp, X0=design, n0=10, nn=20,
        nfea = min(1000,nrow(design)),
        nsvd = nn, nadd = 1, frac = .9, gstart = 0.0001,
        resvdThres = min(5, nn-n0), every = min(5,nn-n0),
        nstarts = 5,centralize=FALSE, maxit=100, verb=0,
        nthread = 4, clutype="PSOCK")
```

Arguments

`design`	An N by d matrix of N training/design inputs.
`resp`	An L by N response matrix of `design`, where L is the length of the time series outputs, N is the number of design points.
`X0`	An M by d matrix of M test inputs. The localized SVD-based GP models will be fitted on every point (row) of `X0`. The default value of `X0` is `design`.
`n0`	The number of points in the initial neighborhood set. The initial neighborhood set is selected by the Euclidean distance. The default value is 10.
`nn`	The total number of neighborhood points. The `nn-n0` points are selected sequentially by the proposed algorithm. The default value is 20.
`nfea`	The number of feasible points within which to select the neighborhood points. This function will only consider the `nfea` design points closest to the test point in terms of Euclidean distance when selecting neighborhood points. The default value is the minimum of N and 1000.
`nsvd`	The number of design points closest to the test points on whose response matrix to perform the initial singular value decomposition. The default value is `nn`.
`nadd`	The number of neighborhood points selected at one iteration. The default value is 1.
`frac`	The threshold in the cumulative percentage criterion to select the number of SVD bases. The default value is 0.9.

`gstart` The starting number and upper bound of for estimating the nugget parameter. If `gstart = sqrt(.Machine$double.eps)`, the nugget will be fixed at `sqrt(.Machine$double.eps)`, since the it is the lower bound of the nugget term. The default value is 0.0001.

`resvdThres` The threshold to re-perform SVD. After every `resvdThres` points have been included into the neighborhood set, the SVD of the response matrix will be re-performed and the SVD-based GP model will be refitted. The default value is the minimum of `nn-n0` and 5.

`every` The threshold to refit GP models without re-perform SVD. After every `every` points have been included into the neighborhood set, the GP models will be refitted. But the SVD will not be re-performed. It is suggested `every <= resvdThres`. The default value is the minimum of `nn-n0` and 5.

`nstarts` The number of starting points used in the numerical maximization of the posterior density function. The larger `nstarts` will typically lead to more accurate prediction but longer computational time. The default value is 5.

`centralize` If `centralize=TRUE` the response matrix will be centralized (subtract the mean) before the start of the algorithm. The mean will be added to the predictive mean at the finish of the algorithm. The default value is `FALSE`.

`maxit` Maximum number of iterations in the numerical optimization algorithm for maximizing the posterior density function. The default value is 100.

`verb` A nonnegative integer indicates the level of printing on the screen. If `verb=0` the function is executed in silence. The default value is 0.

`nthread` The number of threads (processes) used in parallel execution of this function. `nthread=1` implies no parallelization. The default value is 4.

`clutype` The type of cluster in the R package "parallel" to perform parallelization. The default value is "PSOCK". Required only if `nthread>1`.

Value

`pmean` An L by M matrix of posterior predicted mean for the response at the test set `X0`.

`ps2` An L by M matrix of posterior predicted variance for the response at the test set `X0`.

Author(s)

Ru Zhang <heavenmarshal@gmail.com>,

C. Devon Lin <devon.lin@queensu.ca>,

Pritam Ranjan <pritam.ranjan@gmail.com>

References

Zhang, R., Lin, C.D. and Ranjan, P. (2018) *Local Gaussian Process Model for Large-scale Dynamic Computer Experiments*, arXiv:1611.09488.

See Also

knnsvdGP, svdGP.

Examples

```
library("lhs")
forretal <- function(x,t,shift=1)
{
    par1 <- x[1]*6+4
    par2 <- x[2]*16+4
    par3 <- x[3]*6+1
    t <- t+shift
    y <- (par1*t-2)^2*sin(par2*t-par3)
}
timepoints <- seq(0,1,len=200)
design <- lhs::randomLHS(100,3)
test <- lhs::randomLHS(20,3)
## evaluate the response matrix on the design matrix
resp <- apply(design,1,forretal,timepoints)

n0 <- 15
nn <- 20
gs <- sqrt(.Machine$double.eps)

## lasvdGP with mutiple (5) start points for GP model estimation,
## It use the R package "parallel" for parallelization
retlamsp <- lasvdGP(design,resp,test,n0,nn,frac=.95,gstart=gs,
                    centralize=TRUE,nstarts=5,nthread=2,clutype="PSOCK")

## lasvdGP with single start point for GP model estimation,
## It does not use parallel computation
retlass <- lasvdGP(design,resp,test,n0,nn,frac=.95,gstart=gs,
                   centralize=TRUE,nstarts=1,nthread=1)
```

---

svdGP *Full SVD-Based GP Models*

Description

This function fits a full SVD-based GP model with test set `X0`, design set `design` and response matrix `resp`.

Usage

```
svdGP(design,resp,X0=design,nstarts=5,d=NULL,gstart=0.0001,
      frac=.9,centralize=FALSE,nthread=4,clutype="PSOCK")
```

Arguments

design	An N by d matrix of N training/design inputs.
resp	An L by N response matrix of design, where L is the length of the time series outputs, N is the number of design points.
X0	An M by d matrix of M test inputs. The default value of X0 is design.
nstarts	The number of starting points used in the numerical maximization of the posterior density function. The larger nstarts will typically lead to more accurate prediction but longer computational time. The default value is 5.
d	the start value of lengthscale parameter of GP models, a length q vector of positive values. If d=NULL, the start value will be selected automatically. The default value is NULL.
gstart	The starting number and upper bound of for estimating the nugget parameter. If gstart = sqrt(.Machine$double.eps), the nugget will be fixed at sqrt(.Machine$double.eps), since the it is the lower bound of the nugget term. The default value is 0.0001.
frac	The threshold in the cumulative percentage criterion to select the number of SVD bases. The default value is 0.9.
centralize	If centralize=TRUE the response matrix will be centralized (subtract the mean) before the start of the algorithm. The mean will be added to the predictive mean at the finish of the algorithm. The default value is FALSE.
nthread	The number of threads (processes) used in parallel execution of this function. nthread=1 implies no parallelization. The default value is 4.
clutype	The type of cluster in the R package "parallel" to perform parallelization. The default value is "PSOCK". Required only if nthread>1.

Value

pmean	An L by M matrix of posterior predicted mean for the response at the test set X0.
ps2	An L by M matrix of posterior predicted variance for the response at the test set X0.

Author(s)

Ru Zhang <heavenmarshal@gmail.com>,

C. Devon Lin <devon.lin@queensu.ca>,

Pritam Ranjan <pritam.ranjan@gmail.com>

See Also

knnsvdGP, lasvdGP.

Examples

```
library("lhs")
forretal <- function(x,t,shift=1)
{
    par1 <- x[1]*6+4
    par2 <- x[2]*16+4
    par3 <- x[3]*6+1
    t <- t+shift
    y <- (par1*t-2)^2*sin(par2*t-par3)
}
timepoints <- seq(0,1,len=200)
design <- lhs::randomLHS(50,3)
test <- lhs::randomLHS(50,3)
## evaluate the response matrix on the design matrix
resp <- apply(design,1,forretal,timepoints)

## fit full SVD-based GP model
ret <- svdGP(design,resp,test,frac=.95,
             centralize=TRUE,nthread=2)
```

Index