Package ‘CGP’

February 19, 2015

Type Package

Title Composite Gaussian process models

Version 2.0-2

Date 2014-09-20

Author Shan Ba and V. Roshan Joseph

Maintainer Shan Ba <shanbatr@gmail.com>

Description Fit composite Gaussian process (CGP) models as described in Ba and Joseph (2012) “Composite Gaussian Process Models for Emulating Expensive Functions”, Annals of Applied Statistics. The CGP model is capable of approximating complex surfaces that are not second-order stationary. Important functions in this package are CGP, print.CGP, summary.CGP, predict.CGP and plotCGP.

License LGPL-2.1

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-21 07:43:39

R topics documented:

CGP-package ......................................................... 2
CGP ................................................................. 3
CGP.default ......................................................... 5
CGPEst ............................................................ 6
plotCGP ............................................................. 8
predict.CGP ......................................................... 9
print.CGP .......................................................... 11
summary.CGP ....................................................... 12

Index 14
Description

Build nonstationary surrogate models for emulating computationally-expensive computer simulations (computer models).

Details

Package: CGP
Type: Package
Version: 2.0-2
Date: 2014-09-20
License: LGPL-2.1

This package contains functions for fitting the composite Gaussian process (CGP) model, which consists of two Gaussian processes (GPs). The first GP captures the smooth global trend and the second one models local details. The model also incorporates a flexible variance model, which makes it more capable of approximating surfaces with varying volatility. It can be used as an emulator (surrogate model) for approximating computationally expensive functions that are not second-order stationary. When the underlying surface is stationary, the fitted CGP model should degenerate to a standard (stationary) GP model ($\lambda \approx 0$).

The package implements maximum likelihood method to estimate model parameters and also provides predictions (with 5% and 95% prediction quantiles) for unobserved input locations. Leave-one-out cross validation diagnostic plot is also supported.

**Gaussian correlation functions**

$$g(h) = \exp\left(-\sum_{j=1}^{p} \theta_j h_j^2\right), \quad l(h) = \exp\left(-\sum_{j=1}^{p} \alpha_j h_j^2\right)$$

(with unknown parameters $\theta$ and $\alpha$) are used to describe the correlations in the global and local processes respectively.

For a complete list of functions, please use `help(package="CGP")`. Important functions are `CGP`, `print.CGP`, `summary.CGP`, `predict.CGP` and `plotCGP`.

Author(s)

Shan Ba and V. Roshan Joseph
Maintainer: Shan Ba <shanbatr@gmail.com>

References

CGP

Composite Gaussian process models

Description

Estimate parameters in the composite Gaussian process (CGP) model using maximum likelihood methods. Calculate the root mean squared (leave-one-out) cross validation error for diagnosis, and export intermediate values to facilitate predict.CGP function.

Usage

CGP(x, yobs, ...)

Arguments

x The design matrix
yobs The vector of response values, corresponding to the rows of x
... Optional arguments for the CGPEst function, including lower bound for the nugget parameter, number of random starts, etc. See CGPEst for details.

Details

This function calls the CGPEst function for fitting a composite Gaussian process (CGP) model based on the given design matrix x and the observed responses yobs. The fitted model consists of a smooth GP to capture the global trend and a local GP which is augmented with a flexible variance model to capture the change of local volatilities. For p input variables, such two GPs involve 2p + 2 unknown parameters in total. As demonstrated in Ba and Joseph (2012), by assuming $\alpha_j = \theta_j + \kappa$ for $j = 1, \ldots, p$, fitting the CGP model only requires estimating $p + 3$ unknown parameters, which is comparable to fitting a stationary GP model ($p$ unknown parameters).

Value

This function fits the CGP model and returns an object of class “CGP”. Function predict.CGP can be further used for making new predictions and function summary.CGP can be used to print a summary of the “CGP” object.

An object of class “CGP” is a list containing at least the following components:

- lambda Estimated nugget value ($\lambda$)
- theta Vector of estimated correlation parameters ($\theta$) in the global GP
- alpha Vector of estimated correlation parameters ($\alpha$) in the local GP
- bandwidth Estimated bandwidth parameter ($b$) in the variance model
- rmscv Root mean squared (leave-one-out) cross validation error
- yp_jackknife Vector of Jackknife (leave-one-out) predicted values
- mu Estimated mean value ($\mu$) for global trend
- tau2 Estimated variance ($\tau^2$) for global trend
beststart  Best starting value found for optimizing the log-likelihood

Author(s)
Shan Ba <shanbatr@gmail.com> and V. Roshan Joseph <roshan@isye.gatech.edu>

References

See Also
predict.CGP, print.CGP, summary.CGP, CGPEst

Examples

```r
x1<-c(0,.02,.075,.08,.14,.15,.155,.156,.18,.22,.29,.32,.36,
      .37,.42,.5,.57,.63,.72,.785,.8,.84,.925,1)
x2<-c(.29,.02,.12,.58,.38,.87,.01,.12,.22,.08,.34,.185,.64,
      .02,.93,.15,.42,.71,1,0,.21,.5,.785,.21)
X<-cbind(x1,x2)
yobs<-sin(1/(x1*x.7+0.3)*(x2*0.7+0.3))

# Fit the CGP model
# Increase the lower bound for nugget to 0.01 (Optional)
mod<-CGP(X,yobs,nugget_l=0.01)
summary(mod)

mod$objval
#-27.4537
mod$lambda
#0.6210284
mod$theta
#6.065497 8.093402
mod$alpha
#143.1770 145.2849
mod$bandwidth
#1
mod$rmscv
#0.5714969
```
Description

Default method for the “CGP” class. This is an intermediate function, not intended for direct use.

Usage

## Default S3 method:
CGP(x, yobs, ...)

Arguments

- **x**
  - The design matrix
- **yobs**
  - The vector of response values, corresponding to the rows of x
- **...**
  - Optional arguments to be passed to `CGPEst` function

Details

This is an intermediate function which calls `CGPEst` for parameter estimation and call to the results. It sets the class of returned object to “CGP”. It is not intended for direct use.

Value

This function creates an object of class “CGP”.

Author(s)

Shan Ba <shanbatr@gmail.com> and V. Roshan Joseph <roshan@isye.gatech.edu>

References


See Also

`CGP`
CGPEst

Estimate composite Gaussian process models

Description

Estimate parameters in the composite Gaussian process (CGP) model using maximum likelihood methods. Calculate the root mean squared (leave-one-out) cross validation error for diagnosis, and export intermediate values to facilitate predictCGP. This function is usually called by the generic function CGP.

Usage

CGPEst(xL yobsL nugget_l = PNPPQL num_starts = UL
theta_l = nullL alpha_l = nullL kappa_u = nullL)

Arguments

x
The design matrix

yobs
The vector of response values, corresponding to the rows of x

nugget_l
Optional, default is “0.001”. The lower bound for the nugget value (λ in the paper)

num_starts
Optional, default is “5”. Number of random starts for optimizing the likelihood function

theta_l
Optional, default is “0.0001”. The lower bound for all correlation parameters in the global GP (θ in the paper)

alpha_l
Optional. The lower bound for all correlation parameters in the local GP (α in the paper). It is also the upper bound for all correlation parameters in the global GP (the θ). Default is log(100)*mean(1/dist(Stand_X)^2), where Stand_X<-apply(X,2,function(x) (x-min(x))/max(x-min(x))). Please refer to Ba and Joseph (2012) for details

kappa_u
Optional. The upper bound for κ, where we define α_j = θ_j + κ for j = 1,...,p. Default value is log(10^6)*mean(1/dist(Stand_X)^2), \ where Stand_X<-apply(X,2,function(x) (x-min(x))/max(x-min(x)))

Details

This function is to be called by the generic function CGP. It estimates parameters in the composite Gaussian process (CGP) model using maximum likelihood methods, calculates the root mean squared (leave-one-out) cross validation error to measure goodness-of-fit, and also computes intermediate values to facilitate predictCGP.
Value
This function returns a list containing at least the following components:

- `var_names`: Vector of input variable names
- `lambda`: Estimated nugget value (\( \lambda \))
- `theta`: Vector of estimated correlation parameters (\( \theta \)) in the global GP
- `alpha`: Vector of estimated correlation parameters (\( \alpha \)) in the local GP
- `bandwidth`: Estimated bandwidth parameter (\( b \)) in the variance model
- `rmscv`: Root mean squared (leave-one-out) cross validation error
- `yp_jackknife`: Vector of Jackknife (leave-one-out) predicted values
- `mu`: Estimated mean value (\( \mu \)) for global trend
- `tau2`: Estimated variance (\( \tau^2 \)) for global trend
- `beststart`: Best starting value found for optimizing the log-likelihood
- `objval`: Optimal objective value for the negative log-likelihood (up to a constant)
- `sig_matrix`: Diagonal matrix containing standardized local variances at each of the design points
- `sf`: Standardization constant for rescaling the local variance model
- `res2`: Vector of squared residuals from the estimated global trend
- `invQ`: Matrix of \(( G + \lambda \Sigma^{1/2} L \Sigma^{1/2})^{-1} \)
- `temp_matrix`: Matrix of \(( G + \lambda \Sigma^{1/2} L \Sigma^{1/2})^{-1} (y - \hat{\mu} 1) \)

Author(s)
Shan Ba <shanbatr@gmail.com> and V. Roshan Joseph <roshan@isye.gatech.edu>

References

See Also
- CGP

Examples

```r
#Training data
X<-c(0.775, 0.83, 0.85, 1.05, 1.272, 1.335, 1.365, 1.45, 1.639, 1.675, 1.88, 1.975, 2.06, 2.09, 2.18, 2.27, 2.3, 2.36, 2.38, 2.39)
yobs<-sin(10*pi*X)/(2*X)+(X-1)^4

#Fit the CGP model
#mod<-CGPEst(X,yobs)
```
Description

Draw jackknife (leave-one-out) actual by predicted plot to measure goodness-of-fit.

Usage

plotCGP(object)

Arguments

object An object of class "CGP"

Details

Draw the actual observed values on the y-axis and the jackknife (leave-one-out) predicted values on the x-axis. The goodness-of-fit can be measured by how well the points lie along the 45 degree diagonal line.

Value

This function draws the jackknife (leave-one-out) actual by predicted plot.

Author(s)

Shan Ba <shanbatr@gmail.com> and V. Roshan Joseph <roshan@isye.gatech.edu>

References

predict.CGP

See Also
cgp

Examples

```r
x1=-c(0, .02, .075, .08, .14, .155, .156, .18, .22, .29, .32, .36, .37, .42, .5, .57, .63, .72, .785, .8, .84, .925, 1)
x2=-c(.29, .02, .12, .58, .38, .87, .01, .12, .22, .08, .34, .185, .64, .02, .93, .15, .42, .71, 1, .0, .21, .5, .785, .21)
X<-cbind(x1, x2)
yobs<-x1^2 + x2^2

#The CGP model
#mod<-cgp(X, yobs, nugget_l=0.001)
#plotCgp(mod)
```

### predict.CGP

**Predict from the composite Gaussian process model**

#### Description

Compute predictions from the composite Gaussian process (CGP) model. 95% prediction intervals can also be calculated.

#### Usage

```r
predict(object, newdata = NULL, PI = FALSE, ...)
```

#### Arguments

- `object` An object of class "CGP"
- `newdata` Optional. The matrix of predictive input locations, where each row of `newdata` corresponds to one predictive location
- `PI` If TRUE, 95% prediction intervals are also calculated. Default is FALSE
- `...` For compatibility with generic method `predict`

#### Details

Given an object of "CGP" class, this function predicts responses at unobserved `newdata` locations. If the `PI` is set to be TRUE, 95% predictions intervals are also computed.

If `newdata` is equal to the design matrix of the object, predictions from the CGP model will be identical to the `yobs` component of the object and the width of the prediction intervals will be shrunk to zero. This is due to the interpolating property of the predictor.
Value

The function returns a list containing the following components:

- **Yp**: Vector of predictive values at newdata locations (Yp=gp+lp)
- **gp**: Vector of predictive values at newdata locations from the global process
- **lp**: Vector of predictive values at newdata locations from the local process
- **v**: Vector of predictive standardized local volatilities at newdata locations
- **Y_low**: If PI=TRUE, vector of 5% predictive quantiles at newdata locations
- **Y_up**: If PI=TRUE, vector of 95% predictive quantiles at newdata locations

Author(s)

Shan Ba <shanbatr@gmail.com> and V. Roshan Joseph <roshan@isye.gatech.edu>

References


See Also

- `cgp`, `print.CGP`, `summary.CGP`

Examples

```r
### A simulated example from Gramacy and Lee (2012) "Cases for the nugget

#Training data
X<-c(0.775, 0.83, 0.85, 1.05, 1.272, 1.335, 1.365, 1.45, 1.639, 1.675,
1.88, 1.975, 2.06, 2.09, 2.18, 2.27, 2.3, 2.36, 2.38, 2.39)
yobs<-sin(10*pi*X)/(2*X)+(X-1)^4

#Testing data
UU<-seq(from=0.7,to=2.4,by=0.001)
y_true<-sin(10*pi*UU)/(2*UU)+(UU-1)^4

plot(UU,y_true,type="l",xlab="x",ylab="y")
points(X,yobs,col="red")

#Fit the CGP model
mod<-CGP(X,yobs)
summary(mod)

mod$objval
#-40.17315
mod$lambda
#0.01877432
mod$theta
#2.43932
```
print.CGP

mod$alpha
# 578.8898
mod$bandwidth
# 1
mod$rmse cv
# 0.3055192

# Predict for the testing data 'UU'
modpred<-predict(mod,UU)

# Plot the fitted CGP model
# Red: final predictor; Blue: global trend
lines(UU,modpred$yp,col="red",lty=3,lwd=2)
lines(UU,modpred$gp,col="blue",lty=5,lwd=1.8)

print.CGP  

---

**CGP model summary information**

**Description**

Print a brief summary of a “CGP” object.

**Usage**

```r
## S3 method for class 'CGP'
print(x, ...)
```

**Arguments**

- `x`: An object of class "CGP"
- `...`: For compatibility with generic method `print`

**Details**

This function prints a brief summary of a “CGP” object.

**Value**

This function prints the results of:

- `lambda`: Estimated nugget value ($\lambda$)
- `theta`: Estimated correlation parameters ($\theta$) in the global GP
- `alpha`: Estimated correlation parameters ($\alpha$) in the local GP
- `bandwidth`: Estimated bandwidth parameter ($b$) in the variance model

**Author(s)**

Shan Ba <shanbatr@gmail.com> and V. Roshan Joseph <roshan@isye.gatech.edu>
References


See Also

CGP, summary.CGP, predict.CGP

Examples

```r
x1<-c(0,.02,.075,.08,.14,.155,.156,.18,.22,.29,.32,.36,.37,.42,.5,.57,.63,.72,.785,.8,.84,.925,1)
x2<-c(.29,.02,.12,.58,.38,.87,.01,.12,.22,.08,.34,.185,.64,.02,.93,.15,.42,.71,1,0,.21,.5,.785,.21)
X<-cbind(x1,x2)
yobs<-sin(1/(x1*0.7+0.3)*(x2*0.7+0.3)))

#Fit the CGP model
#mod<-CGP(X,yobs)
#summary(mod)
```

### Description

Print a summary of a “CGP” object.

#### Usage

```r
## S3 method for class 'CGP'
summary(object, ...)
```

#### Arguments

- `object` An object of class "CGP"
- `...` For compatibility with generic method summary

#### Details

This function prints a summary of a “CGP” object.
### Value

This function prints the results of:

- **lambda**: Estimated nugget value ($\lambda$)
- **theta**: Estimated correlation parameters ($\theta$) in the global GP
- **alpha**: Estimated correlation parameters ($\alpha$) in the local GP
- **bandwidth**: Estimated bandwidth parameter ($b$) in the variance model
- **rmsecv**: Root mean squared (leave-one-out) cross validation error
- **mu**: Estimated mean value ($\mu$) for global trend
- **tau2**: Estimated variance ($\tau^2$) for global trend
- **beststart**: Best starting value found for optimizing the log-likelihood
- **objval**: Optimal objective value for the negative log-likelihood (up to a constant)

### Author(s)

Shan Ba <shanbatr@gmail.com> and V. Roshan Joseph <roshan@isye.gatech.edu>

### References


### See Also

CGP, print.CGP, predict.CGP

### Examples

```r
x1 <- c(0, 0.02, 0.075, 0.08, 0.14, 0.15, 0.155, 0.156, 0.18, 0.22, 0.29, 0.32, 0.36, 0.37, 0.42, 0.5, 0.57, 0.63, 0.72, 0.785, 0.8, 0.84, 0.925, 1)
x2 <- c(0.29, 0.02, 0.12, 0.58, 0.38, 0.87, 0.01, 0.12, 0.22, 0.08, 0.34, 1.85, 0.64, 0.02, 0.93, 0.15, 0.42, 0.71, 1, 0.21, 0.5, 0.785, 0.21)
X <- cbind(x1, x2)
yobs <- sin(1/(x1^0.7+0.3)*(x2^0.7+0.3)))

# Fit the CGP model
# mod <- CGP(X, yobs)
# summary(mod)
```
Index

*Topic package
  CGP-package, 2

CGP, 2, 3, 5–7, 9, 10, 12, 13
CGP-package, 2
CGP.default, 5
CGPEst, 3–5, 6

plotCGP, 2, 8
predict.CGPG, 2, 4, 9, 12, 13
print.CGPG, 2, 4, 10, 11, 13

summary.CGPG, 2, 4, 10, 12, 12