Package: GPM (via r-universe)

November 5, 2024

Type Package

Title Gaussian Process Modeling of Multi-Response and Possibly Noisy Datasets

Version 3.0.1

Date 2019-03-21

Author Ramin Bostanabad, Tucker Kearney, Siyo Tao, Daniel Apley, and Wei Chen (IDEAL)

Maintainer Ramin Bostanabad <bostanabad@u.northwestern.edu>

Description Provides a general and efficient tool for fitting a response surface to a dataset via Gaussian processes. The dataset can have multiple responses and be noisy (with stationary variance). The fitted GP model can predict the gradient as well. The package is based on the work of Bostanabad, R., Kearney, T., Tao, S. Y., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. International Journal for Numerical Methods in Engineering, 114, 501-516.

License GPL-2

LazyData FALSE

Encoding UTF-8

Imports Rcpp (>= 0.12.19), lhs(>= 0.14), randtoolbox(>= 1.17), lattice(>= 0.20-34), pracma(>= 2.1.8), foreach(>= 1.4.4), doParallel(>= 1.0.14), parallel(>= 3.5), iterators(>= 1.0.10)

Suggests RcppArmadillo

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 3.5), stats (>= 3.5)

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-03-21 15:33:29 UTC

Contents

Auxil — *An auxiliary function used in calculating the negative log-likelehood and its gradient*

Description

Calculates some auxiliary paramters to obtain the negative log-likelehood and its gradient.

Usage

```
Auxil(Omega, X, Y, CorrType, MinEig, Fn, n, dy)
```

Arguments

`Omega`	The vector storing all the hyperparameters of the correlation function. The length of `Omega` depends on the `CorrType`. See `reference 1`.
`X`	Matrix containing the training (aka design or input) data points. The rows and columns of `X` denote individual observation settings and input dimension, respectively.
`Y`	Matrix containing the output (aka response) data points. The rows and columns of `Y` denote individual observation responses and output dimension, respectively.
`CorrType`	The correlation function of the GP model. Choices include `'G'` (default), `'PE'`, `'LBG'`, and `'LB'`. See `Fit` and the `references`.
`MinEig`	The smallest eigen value that the correlation matrix is allowed to have, which in return determines the appraopriate nugget that should be added to the correlation matrix.
`Fn`	A matrix of `1`'s with `nrow(X)` rows and `1` column. See `reference 1`.
`n`	Number of observations, `nrow(X)`.
`dy`	Number of responses, `ncol(Y)`.

Details

Since `Auxil` is shared between `NLogL` and `NLogL_G` during optimization, ideally it should be run only once (e.g., via memoisation). Such an implementation is left for future editions.

Value

ALL A list containing the following components (based on CorrType, some other parameters are also stored in ALL):

- R The correlation matrix whose smallest eigen value is >= MinEig.
- L Cholesky decomposition of R.
- Raw_MinEig The smallest eigen value of R before adding Nug_opt.
- Nug_opt The added nugger to R.
- B

References

1. Bostanabad, R., Kearney, T., Tao, S., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. Int J Numer Meth Eng, 114, 501-516.
2. Plumlee, M. & Apley, D. W. (2017) Lifted Brownian kriging models. Technometrics, 59, 165-177.

See Also

Fit to see how a GP model can be fitted to a training dataset.
Predict to use the fitted GP model for prediction.
Draw to plot the response via the fitted model.

Examples

```
# see the examples in the fitting function.
```

CorrMat *Two Functions for Constructing the Correlation Matrix in* GPM *Package*

Description

The CorrMat_Sym() function builds the auto-correlation matrix corresponding to dataset X while the CorrMat_Vec() function builds the correlation matrix between datasets X1 and X2.

Usage

```
CorrMat_Sym(X, CorrType, Omega)
CorrMat_Vec(X1, X2, CorrType, Omega)
```

Arguments

`X, X1, X2` Matrices containing the numeric data points. The rows and columns of both `X1` and `X2` denote individual observation settings and dimension, respectively.

`CorrType` The correlation function of the GP model. Choices include `'G'` (default), `'PE'`, `'LBG'`, and `'LB'`. See the `references` for the details.

`Omega` The vector storing all the scale (aka roughness) parameters of the correlation function. The length of `Omega` depends on the `CorrType`. See `reference 1`.

Value

R The Correlation matrix with size `nrow(X1)`-by-`nrow(X2)`. See here.

Note

This function is **NOT** exported once the GPM package is loaded.

References

1. Bostanabad, R., Kearney, T., Tao, S. Y., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. International Journal for Numerical Methods in Engineering, 114, 501-516.
2. Plumlee, M. & Apley, D. W. (2017) Lifted Brownian kriging models. Technometrics, 59, 165-177.

See Also

`Fit` to see how a GP model can be fitted to a training dataset.
`Predict` to use the fitted GP model for prediction.
`Draw` to plot the response via the fitted model.

Examples

```
# see the examples in \code{\link[GPM]{Fit}}
```

`Draw` *The Plotting Function of* `GPM` *Package*

Description

Plots the predicted response along with the assocaited uncertainty via the GP model fitted by `Fit`. Accepts multi-input and multi-output models. See `Arguments` for more details on the options.

Usage

```
Draw(Model, Plot_wrt, LB = NULL, UB = NULL, Values = NULL,
  Response_ID = NULL, res = 15, X1Label = NULL, X2Label = NULL,
  YLabel = NULL, Title = NULL, PI95 = NULL)
```

Arguments

`Model` The GP model fitted by `Fit`.

`Plot_wrt` A binary vector of length `p` where `p` is the dimension of the inputs in `Model`. A maximum (minimum) of `2` (`1`) elements can be `1`. The elemenets set to `1`, would correspond to the plotting axes.

`LB, UB` Vectors of length `sum(Plot_wrt)` indicating the lower and upper bounds used for plotting. The first (second) element corresponds to the first (second) non-zero element of `Plot_wrt`.

`Values` A vector of length `p-sum(Plot_wrt)`. The values are assigned to the variables NOT used in plotting and correspond to the zeros in `Plot_wrt`.

`Response_ID` A positive integer indicating the response that should be plotted if `Model` is multi-response.

`res` A positive integer indicating the number of points used in plotting. Higher values will result in smoother plots.

`X1Label` A string for the label of axis `1`.

`X2Label` A string for the label of axis `2`, if plotting a surface.

`YLabel` A string for the label of the response axis.

`Title` A string for the title of the plot.

`PI95` Flag (a scalar) indicating whether the `95%` prediction interval should be plotted. Set it to a non-zero value to turn the flag "on".

References

1. Bostanabad, R., Kearney, T., Tao, S., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. Int J Numer Meth Eng, 114, 501-516.

2. Plumlee, M. & Apley, D. W. (2017) Lifted Brownian kriging models. Technometrics, 59, 165-177.

See Also

`Fit` to see how a GP model can be fitted to a training dataset.
`Predict` to use the fitted GP model for prediction.

Examples

```
# See the examples in the fitting function.
```

Fit *The Fitting Function of* GPM *Package*

Description

Fits a Gaussian process (GP) to a set of simulation data as described in reference 1. Both the inputs and outputs can be multi-dimensional. The outputs can be noisy in which case it is assumed that the noise is stationary (i.e., its variance is not a function of x).

Usage

```
Fit(X, Y, CorrType = 'G', Eps = 10^(seq(-1, -12)), AnaGr = NULL, Nopt = 5,TraceIt = 0,
MaxIter = 100, Seed = 1, LowerBound = NULL, UpperBound = NULL,
StopFlag = 1, Progress = 0, DoParallel = 0, Ncores = NULL)
```

Arguments

X — Matrix containing the training (aka design or input) data points. The rows and columns of X denote individual observation settings and input dimension, respectively.

Y — Matrix containing the output (aka response) data points. The rows and columns of Y denote individual observation responses and output dimension, respectively.

CorrType — The type of the correlation function of the GP model. Choices include 'G' (default), 'PE', 'LBG', and 'LB'. See the references for the details. For smooth (or analytic) functions, choose either 'G' or 'LBG'. Fitting is faster if 'G' is chosen.

Eps — A vector containing the smallest eigen value(s) that the correlation matrix is allowed to have. The elements of Eps must be in [0, 1] and sorted in a descending order.

AnaGr — Flag indicating whether the gradient of the log-likelihood should be taken analytically (!= 0) or numerically (== 0). For now, only available when CorrType == 'G' or CorrType == 'PE'. If AnaGr != 0, the fitted model will generally be more accurate.

Nopt — The number of times the log-likelihood function is optimized when Eps[1] is used to constraint the smallest eigen value that the correlation matrix is allowed to have. Higher Nopt will increase fitting time as well as the chances of finding the global optimum. If nrow(X) is large (i.e., large training datasets), Nopt can be small.Analyzing the optimization results for Eps[1] and when Progress != 0 will determine if Nopt has been large enough.

TraceIt — Non-negative integer. If positive, tracing information on the progress of the optimization is **printed**. There are six levels of tracing (see optim) and higher values will produce more tracing information.

MaxIter — Maximum number of iterations allowed for each optimization (see optim).

Seed — An integer for the random number generator. Use this to make the results reproducible.

LowerBound, UpperBound
: To estimate the scale (aka roughness) parameters of the correlation function, the feasible range should be defined. LowerBound and UpperBound are vectors determining, resepectively, the lower and upper bounds. Their length depends on the parametric form of the correlation function (see reference 1 for the details).

StopFlag
: Flag indicating whether the optimization must be stopped if the negative log-likelihood increases with decreasing Eps[i].

Progress
: Flag indicating if the fitting process should be summarized. Set it to !=0 to turn it on.

DoParallel
: If != 0, optimizations will be done in parallel.

Ncores
: Number of cores to use if DoParallel != 0. The default is the maximum number of physical cores.

Value

Model A list containing the following components:

- CovFunc A list containing the type and estimated parameters of the correlation function.
- Data A list storing the original (but scaled) data.
- Details A list of some parameters (used in prediction) as well as some values reporting the total run-time (cost) and the added nugget (Nug_opt) for satisfying the constraint on the smallest eigen value of the correlation matrix.
- OptimHist The optimization history.
- Setting The default/provided settings for running the code.

References

1. Bostanabad, R., Kearney, T., Tao, S., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. Int J Numer Meth Eng, 114, 501-516.
2. Plumlee, M. & Apley, D. W. (2017) Lifted Brownian kriging models. Technometrics, 59, 165-177.

See Also

optim for the details on L-BFGS-B algorithm used in optimization.
Predict to use the fitted GP model for prediction.
Draw to plot the response via the fitted model.

Examples

```
# 1D example: Fit a model (with default settings) and evaluate the performance
# by computing the root mean squared error (RMSE) in prediction.
library(lhs)
X <- 5*maximinLHS(15, 1)
Y <- 2*sin(2*X) + log(X+1)
M <- Fit(X, Y)
XF <- matrix(seq(0, 5, length.out = 100), 100, 1)
```

```
YF <- Predict(XF, M)
RMSE <- sqrt(mean((YF$YF - (2*sin(2*XF) + log(XF+1)))^2))

## Not run:
# 1D example: Fit a model, evaluate the performance, and plot the response
# along with 95% prediction interval
X <- 10*maximinLHS(10, 1) - 5
Y <- X*cos(X)
M <- Fit(X, Y)
XF <- matrix(seq(-5, 5, length.out = 500), 500, 1)
YF <- Predict(XF, M)
RMSE <- sqrt(mean((YF$YF - (XF*cos(XF)))^2))
Draw(M, 1, res = 20)

# 2D example: Fit a model, evaluate the performance, and plot the response
# surface along with 95% prediction interval
X <- 2*maximinLHS(10, 2) - 1
Y <- X[, 1]^2 + X[, 2]^2
M <- Fit(X, Y, CorrType = "PE")
XF <- 2*maximinLHS(100, 2) - 1
YF <- Predict(XF, M)
RMSE <- sqrt(mean((YF$YF - (XF[, 1]^2 + XF[, 2]^2))^2))
library(lattice)
Draw(M, c(1, 1), res = 15, PI95=1)

# 2D example: Plot the previous model wrt X1 in the [-2, 2]
# interval with X2=1
Draw(M, c(1, 0), LB = -2, UB = 2, res = 15, PI95=1)

# 3D example: Compare the performance of Gaussian ("G") and lifted Browninan
# with Gamma=1 ("LBG")
X <- 2*maximinLHS(50, 3) - 1
Y <- cos(X[, 1]^2) + 2*sin(X[, 2]^2) + X[, 3]^2
M_G <- Fit(X, Y)
M_LBG <- Fit(X, Y, CorrType = "LBG")
XF <- 2*maximinLHS(500, 3) - 1
YF_G <- Predict(XF, M_G)
YF_LBG <- Predict(XF, M_LBG)
RMSE_G <- sqrt(mean((YF_G$YF - (cos(XF[, 1]^2) + 2*sin(XF[, 2]^2) + XF[, 3]^2))^2))
RMSE_LBG <- sqrt(mean((YF_LBG$YF - (cos(XF[, 1]^2) + 2*sin(XF[, 2]^2) + XF[, 3]^2))^2))

# 3D example: Draw the response in 2D using the M_G model when X3=0
Draw(M_G, c(1, 1, 0), PI95 = 0, Values = 0, X1Label = 'Input 1', X2Label = 'Input 2')

# 3D example: 2D response
X <- 2*maximinLHS(50, 3) - 1
Y <- cbind(cos(X[, 1]^2) + 2*sin(X[, 2]^2) + X[, 3]^2, rowSums(X))
M <- Fit(X, Y)
Draw(M, c(0, 1, 1), Response_ID = 2, Values = 0.5)

# 2D example with noise
X <- 2*maximinLHS(100, 2) - 1
Y <- X[, 1]^2 + X[, 2]^2 + matrix(rnorm(nrow(X), 0, .5), nrow(X), 1)
```

```
M <- Fit(X, Y)
# Estimating the noise variance (should be close to 0.5^2)
M$Details$Nug_opt*M$CovFunc$Parameters$Sigma2*M$Data$Yrange^2

## End(Not run)
```

`MatrixAlgebra` *A Set of Functions for Doing Some Calculations on Matrices in* `GPM` *Package*

Description

These functions perform some matrix algebra to calculate the log-likelihood function.

Usage

```
Eigen(A)
CppSolve(A, B)
LowerChol(A)
```

Arguments

`A`	Numeric, symmetric, and positive definite matrix.
`B`	Numeric matrix or vector.

Value

`Eigen(A)`) returns the smallest eigen value of A. `CppSolve(A, B)` solves for `X` in `AX=B`. `LowerChol(A)` return the lower triangular Cholesky decomposition of `A`.

Note

These functions are **NOT** exported once the GPM package is loaded.

See Also

`Fit` to see how a GP model can be fitted to a training dataset.
`Predict` to use the fitted GP model for prediction.
`Draw` to plot the response via the fitted model.

Examples

```
# see the examples in \code{\link[GPM]{Fit}}
```

NLogL *The Function for calculating the Negative Log-Likelehood in* GPM *Package*

Description

Calculates the negative log-likelihood (excluding all the constant terms) as described in `reference 1`.

Usage

```
NLogL(Omega, X, Y, CorrType, MinEig, Fn, n, dy)
```

Arguments

`Omega` The vector storing all the hyperparameters of the correlation function. The length of `Omega` depends on the `CorrType`. See `reference 1`.

`X` Matrix containing the training (aka design or input) data points. The rows and columns of `X` denote individual observation settings and input dimension, respectively.

`Y` Matrix containing the output (aka response) data points. The rows and columns of `Y` denote individual observation responses and output dimension, respectively.

`CorrType` The correlation function of the GP model. Choices include `'G'` (default), `'PE'`, `'LBG'`, and `'LB'`. See `Fit` and the `references`.

`MinEig` The smallest eigen value that the correlation matrix is allowed to have, which in return determines the appraopriate nugget that should be added to the correlation matrix.

`Fn` A matrix of `1`'s with `nrow(X)` rows and `1` column. See `reference 1`.

`n` Number of observations, `nrow(X)`.

`dy` Number of responses, `ncol(Y)`.

Details

`Fit` calls this function with *scaled* `X` and `Y`. That is, when the user fits a GP model by calling `Fit(X, Y)`, `X` and `Y` are mapped to the `[0, 1]` region and then passed to this function.

Value

nlogl The negative log-likelihood (excluding all the constant terms). See the `references`.

References

1. Bostanabad, R., Kearney, T., Tao, S., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. Int J Numer Meth Eng, 114, 501-516.
2. Plumlee, M. & Apley, D. W. (2017) Lifted Brownian kriging models. Technometrics, 59, 165-177.

See Also

Fit to see how a GP model can be fitted to a training dataset.
Predict to use the fitted GP model for prediction.
Draw to plot the response via the fitted model.

Examples

```
# see the examples in the fitting function.
```

NLogL_G *The Function for calculating the gradient of Negative Log-Likelihood in* GPM *Package*

Description

Calculates the gradient of negative log-likelihood wrt Omega.

Usage

```
NLogL_G(Omega, X, Y, CorrType, MinEig, Fn, n, dy)
```

Arguments

Omega	The vector storing all the hyperparameters of the correlation function. The length of Omega depends on the CorrType. See reference 1.
X	Matrix containing the training (aka design or input) data points. The rows and columns of X denote individual observation settings and input dimension, respectively.
Y	Matrix containing the output (aka response) data points. The rows and columns of Y denote individual observation responses and output dimension, respectively.
CorrType	The correlation function of the GP model. Choices include 'G' (default), 'PE', 'LBG', and 'LB'. See Fit and the references.
MinEig	The smallest eigen value that the correlation matrix is allowed to have, which in return determines the appraopriate nugget that should be added to the correlation matrix.
Fn	A matrix of 1's with nrow(X) rows and 1 column. See reference 1.
n	Number of observations, nrow(X).
dy	Number of responses, ncol(Y).

Details

This function is used in Fit if AnaGr != 0.

Value

NLogL_G The gradient of negative log-likelihood wrt Omega. See the references.

References

1. Bostanabad, R., Kearney, T., Tao, S., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. Int J Numer Meth Eng, 114, 501-516.
2. Plumlee, M. & Apley, D. W. (2017) Lifted Brownian kriging models. Technometrics, 59, 165-177.

See Also

`Fit` to see how a GP model can be fitted to a training dataset.
`Predict` to use the fitted GP model for prediction.
`Draw` to plot the response via the fitted model.

Examples

```
# see the examples in the fitting function.
```

`Predict` *The Prediction Function of* `GPM` *Package*

Description

Predicts the reponse(s), associated prediction uncertainties, and gradient(s) of the GP model fitted by `Fit`.

Usage

```
Predict(XF, Model, MSE_on = 0, YgF_on = 0, grad_dim = rep(1, ncol(XF)))
```

Arguments

`XF` Matrix containing the locations (settings) where the predictions are desired. The rows and columns of `XF` denote individual observation settings and input dimension, respectively.

`Model` The GP model fitted by `Fit`.

`MSE_on` Flag (a scalar) indicating whether the uncertainty (i.e., mean squared error `MSE`) associated with prediction of the response(s) should be calculated. Set to a nonzero value to calculate `MSE`.

`YgF_on` Flag (a scalar) indicating whether the gradient(s) of the response(s) are desired. Set to a non-zero value to calculate the gradient(s). See `note` below.

`grad_dim` A binary vector of length `ncol(XF)`. The gradient of the response(s) will be calculated along the dimensions where the corresponding element of `grad_dim` is `1`. `grad_dim` is ignored if `YgF_on == 0`.

Value

Output A list containing the following components:

- `YF` A matrix with `n` rows (the number of prediction points) and `dy` columns (the number of responses).
- `MSE` A matrix with `n` rows and `dy` columns where each element represents the prediction uncertainty (i.e., the expected value of the squared difference between the prediction and the true response) associated with the corresponding element in `YF`.
- `YgF` An array of size `n` by `sum{grad_dim}` by `dx`.

Note

1. The gradient(s) can be calculated if `CorrType='G'` or `CorrType='LBG'`. If `CorrType='PE'` or `CorrType='LB'`, the gradient(s) can only be calculated if `Power = 2` and `Gamma = 1`, respectively.
2. For efficiency, make sure the inputs are vecotrized and then passed to `Predict`. Avoid passing inputs individually in a `for` loop.

References

1. Bostanabad, R., Kearney, T., Tao, S., Apley, D. W. & Chen, W. (2018) Leveraging the nugget parameter for efficient Gaussian process modeling. Int J Numer Meth Eng, 114, 501-516.
2. Plumlee, M. & Apley, D. W. (2017) Lifted Brownian kriging models. Technometrics, 59, 165-177.

See Also

`Fit` to see how a GP model can be fitted to a training dataset.
`Draw` to plot the response via the fitted model.

Examples

```
# See the examples in the fitting function.
```

Index