

Package ‘EzGP’

May 7, 2026

Title Easy-to-Interpret Gaussian Process Models for Computer Experiments

Version 0.1.0

Description Fit model for datasets with easy-to-interpret Gaussian process modeling, predict responses for new inputs. The input variables of the datasets can be quantitative, qualitative/categorical or mixed. The output variable of the datasets is a scalar (quantitative). The optimization of the likelihood function can be chosen by the users (see the documentation of `EzGP_fit()`). The modeling method is published in “EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors” by Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng (2022) <[doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462)>.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Depends R (>= 4.2.0), stats (>= 4.2.0)

Imports methods (>= 4.2.0), nloptr (>= 2.0.3)

Suggests testthat (>= 3.0.0)

NeedsCompilation no

Author Jiayi Li [cre, aut],
Qian Xiao [aut],
Abhyuday Mandal [aut],
C. Devon Lin [aut],
Xinwei Deng [aut]

Maintainer Jiayi Li <jiayili0123@outlook.com>

Repository CRAN

Date/Publication 2023-07-06 18:40:08 UTC

Contents

cov_m	2
EEzGP_fit	3
EEzGP_predict	5
EzGP_data	6
EzGP_fit	7
EzGP_predict	9
LEzGP_data	10
LEzGP_fit	11
LLF_gradients	13
Index	15

cov_m	<i>The Function for Constructing the Covariance Matrix in EzGP Package</i>
-------	--

Description

Builds the covariance matrix for the given dataset according to different models.

Usage

```
cov_m(X, p, q, m, n, parv, tau = 0, models = 0)
```

Arguments

X	Matrix or data frame containing the inputs of training data. Each row represents the input setting of a data point and the columns are values of quantitative variables and qualitative variables.
p	Number of quantitative factors in the given dataset X.
q	Number of qualitative factors in the given dataset X.
m	A vector containing numbers of levels in qualitative factors.
n	Number of training data points
parv	Parameters in the EzGP/EEzGP model
tau	Nugget if needed. The default nugget is 0, otherwise it has to be a non-negative real value.
models	Model indicator that indicates which model the covariance matrix is built for. 0 for EzGP model, 1 for EEzGP model. The default setting is 0.

Details

[EzGP_fit](#), [EzGP_predict](#), [EEzGP_fit](#), [EEzGP_predict](#), [LEzGP_fit](#), and [LLF_gradients](#) will call this function.

Value

The covariance matrix for the given dataset.

Note

This function is used inside other functions in this package and is **NOT** exported once the EzGP package is loaded.

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

See Also

[EzGP_fit](#) to see how an EzGP model can be fitted to a training dataset.
[EzGP_predict](#) to use the fitted EzGP model for prediction.
[EEzGP_fit](#) to see how an EEzGP model can be fitted to a training dataset.
[EEzGP_predict](#) to use the fitted EEzGP model for prediction.
[LEzGP_fit](#) to see how a LEzGP model can be fitted to a training dataset.

Examples

```
# see the examples in the documentation of the function EzGP_fit.
```

EEzGP_fit

The Fitting Function of EEzGP Model

Description

Fits an Efficient Easy-to-Interpret Gaussian process (EEzGP) model to a dataset as described in reference 1. The input variables are mixed (with both quantitative and qualitative inputs). The output variable is quantitative and scalar.

Usage

```
EEzGP_fit(  
  X,  
  Y,  
  p,  
  q,  
  m,  
  tau = 0,  
  lb = "T",  
  ub = "T",
```

```

x0 = "T",
xtol_rel = 1e-05,
maxeval = 100,
algorithm = "NLOPT_LD_LBFGS"
)

```

Arguments

X	Matrix or data frame containing the inputs of training data. Each row represents the input setting of a data point and the columns are values of quantitative variables and qualitative variables.
Y	Vector containing the outputs of training data points.
p	Number of quantitative factors in the given dataset X.
q	Number of qualitative factors in the given dataset X.
m	A vector containing numbers of levels in qualitative factors.
tau	Nugget if needed. The default nugget is 0, otherwise it has to be a non-negative real value.
lb	Vector with lower bounds of the parameter estimation. "T" for applying the default setting of lb (a vector of length number of parameters whose elements are all 0.1), otherwise one must provide a vector with the length being the number of parameters.
ub	Vector with upper bounds of the parameter estimation. "T" for applying the default setting of ub (a vector of length number of parameters whose first q+1 elements are 100 and the rest number of parameters - q - 1 elements are 10), otherwise one must provide a vector with length of the number of parameters.
x0	Vector with starting values for the optimization. "T" for applying the default setting of x0 (a vector made by $(lb + ub)/2$), otherwise one must provide a vector with the length being the number of parameters.
xtol_rel	Stopping criterion for relative change reached.
maxeval	Termination condition by specifying a maximum number of function.
algorithm	Optimization algorithm. See NLOpt Algorithms for more available algorithms.

Value

A model of class "EzGP model" list of the following items:

- param A list containing the estimated parameters
- data A list containing the fitted dataset and the information for fitting

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

See Also

[EEzGP_predict](#) to use the fitted EEzGP model for prediction.

Examples

```
# Example with 3 quantitative and 3 qualitative variables (dataset included in the package):
#   Fit an EEzGP model (with default settings), and then perform the prediction.
p = 3
q = 3
m=c(3,3,3)
tau = 0
X = EzGP_data[1:25, 1:(p+q)]
Y = EzGP_data[1:25, p+q+1]
X_new = EzGP_data[26:30, 1:(p+q)]
# EEzGP Model and Prediction
model <- EEzGP_fit(X, Y, p, q, m)
pred <- EEzGP_predict(X_new, model, MSE_on = 1)
result <- LLF_gradients(X, Y, p, q, m, model$param, tau = 0, models = 1)
# Results showing
model
pred
result
```

 EEzGP_predict

The Prediction Function of EEzGP Model

Description

Predicts the output of the EEzGP model fitted by [EEzGP_fit](#).

Usage

```
EEzGP_predict(X_new, model, MSE_on = 0)
```

Arguments

X_new	Matrix or vector containing the input(s) where the predictions are to be made. Each row is an input vector.
model	The EEzGP model fitted by EEzGP_fit .
MSE_on	A scalar indicating whether the uncertainty (i.e., mean squared error MSE) is calculated. Set to a non-zero value to calculate MSE.

Value

A prediction list containing the following components:

- \hat{Y} A vector containing the prediction values
- MSE A vector containing the prediction uncertainty (i.e., the covariance or covariance matrix for the output(s) at prediction location(s))

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

See Also

[EEzGP_fit](#) to fit EEzGP model for the datasets.

Examples

```
# This function is used in a similar way as the use of EzGP_predict.  
# See the examples in the documentation of the function EEzGP_fit.
```

EzGP_data

Dataset for the example in function 'EzGP_fit'

Description

Data are sampled from the modified math function based on Example 4.1 in the paper listed in references. There are 3 quantitative factors and 3 qualitative factors each having 3 levels. In this dataset, there are 1296 data points. For the simplicity of illustration, we take the first 81 rows as training data points, and the last 1215 rows as testing data points.

Usage

```
data(EzGP_data)
```

Format

A named list containing training data and testing data:

"x1" 1st quantitative factor

"x2" 2nd quantitative factor

"x3" 3rd quantitative factor

"z1" 1st qualitative factor, which has 3 levels

"z2" 2nd qualitative factor, which has 3 levels

"z3" 3rd qualitative factor, which has 3 levels

"y" Response vector

Source

The dataset can be generated with the code at the end of this description file.

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

Examples

```
data(EzGP_data)
#Number of quantitative factors
p = 3
#Number of qualitative factors
q = 3
#Vector containing numbers of levels in qualitative factors
m=c(3,3,3)
# Nugget
tau = 0

X = EzGP_data[1:81, 1:(p+q)]
Y = EzGP_data[1:81, p+q+1]
X_new = EzGP_data[82:1296, 1:(p+q)]
```

EzGP_fit

The Fitting Function of EzGP Model

Description

Fits an Easy-to-Interpret Gaussian process (EzGP) model to a dataset as described in reference 1. The input variables are mixed (with both quantitative and qualitative inputs) The output variable is quantitative and scalar.

Usage

```
EzGP_fit(
  X,
  Y,
  p,
  q,
  m,
  tau = 0,
  lb = "T",
  ub = "T",
  x0 = "T",
  xtol_rel = 1e-05,
  maxeval = 100,
  algorithm = "NLOPT_LD_LBFGS"
)
```

Arguments

X	Matrix or data frame containing the inputs of training data. Each row represents the input setting of a data point and the columns are values of quantitative variables and qualitative variables.
Y	Vector containing the outputs of training data points.
p	Number of quantitative factors in the given dataset X.
q	Number of qualitative factors in the given dataset X.
m	A vector containing numbers of levels in the qualitative factors.
tau	Nugget if needed. The default nugget is 0, otherwise it has to be a non-negative real value.
lb	Vector with lower bounds of the parameter estimation. "T" for applying the default setting of lb (a vector of length number of parameters whose elements are all 0.1), otherwise one must provide a vector with length of the number of parameters.
ub	Vector with upper bounds of the parameter estimation. "T" for applying the default setting of ub (a vector of length number of parameters whose first q+1 elements are 100 and the rest number of parameters - q - 1 elements are 10), otherwise one must provide a vector with the length being the number of parameters.
x0	Vector with starting values for the optimization. "T" for applying the default setting of x0 (a vector made by $(lb + ub)/2$), otherwise one must provide a vector with the length being the number of parameters.
xtol_rel	Stopping criterion for relative change reached.
maxeval	Termination condition by specifying a maximum number of function.
algorithm	Optimization algorithm. See NLOpt Algorithms for more available algorithms.

Value

A model of class "EzGP model" list of the following items:

- param A list containing the estimated parameters
- data A list containing the dataset and the information for fitting

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

See Also

[EzGP_predict](#) to use the fitted EzGP model for prediction.

Examples

```

# Example with 3 quantitative and 3 qualitative variables (dataset included in the package):
#   Fit an EzGP model (with default settings), and then perform the prediction.
#   This example may run for a while.
p = 3
q = 3
m=c(3,3,3)
tau = 0
X = EzGP_data[1:15, 1:(p+q)]
Y = EzGP_data[1:15, p+q+1]
X_new = EzGP_data[16:20, 1:(p+q)]
# EzGP Model and Prediction
model <- EzGP_fit(X, Y, p, q, m)
pred <- EzGP_predict(X_new, model, MSE_on = 1)
result <- LLF_gradients(X, Y, p, q, m, model$param)
# Results showing
model
pred
result

```

EzGP_predict

*The Prediction Function of EzGP Model***Description**

Predicts the output of the EzGP model fitted by [EzGP_fit](#).

Usage

```
EzGP_predict(X_new, model, MSE_on = 0)
```

Arguments

X_new	Matrix or vector containing the input(s) where the predictions are to be made. Each row is an input vector.
model	The EzGP model fitted by EzGP_fit .
MSE_on	A scalar indicating whether the uncertainty (i.e., mean squared error MSE) is calculated. Set to a non-zero value to calculate MSE.

Value

A prediction list containing the following components:

- `Y_hat` A vector containing the prediction values
- `MSE` A vector containing the prediction uncertainty (i.e., the covariance or covariance matrix for the output(s) at prediction location(s))

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

See Also

[EzGP_fit](#) to fit EzGP model for the datasets.

Examples

```
# see the examples in the documentation of the function EzGP_fit.
```

LEzGP_data

Dataset for the example in function 'LEzGP_fit'

Description

Data are sampled from the modified math function based on Example 4.2 and Example 4.3 in the paper listed in references. There are 9 quantitative factors and 9 qualitative factors each having 3 levels. In this dataset, there are 8250 data points. For the simplicity of illustration, we take the first 8150 rows as training data points, and the last 100 rows as testing data points.

Usage

```
data(LEzGP_data)
```

Format

A named list containing training data and testing data:

"**x1-x9**" 1st quantitative factor to the 9th quantitative factor

"**z1-z9**" 1st qualitative factor to the 9th qualitative factor, which all have 3 levels

"**ry**" Response vector

Source

The dataset can be generated with the code at the end of this description file.

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

Examples

```

data(LEzGP_data)
#Number of quantitative factors
p = 9
#Number of qualitative factors
q = 9
#Vector containing numbers of levels in qualitative factors
m=rep(3,9)
# Nugget
tau = 0

X = LEzGP_data[1:8150, 1:(p+q)]
Y = LEzGP_data[1:8150, p+q+1]
X_new = LEzGP_data[8151:8250, 1:(p+q)]

```

LEzGP_fit

*The Fitting Function of LEzGP Model***Description**

Fits a Localized Easy-to-Interpret Gaussian process (LEzGP) model to a dataset as described in reference 1. The input variables are mixed (with both quantitative and qualitative inputs) The output variable is quantitative and scalar.

Usage

```

LEzGP_fit(
  X,
  Y,
  p,
  q,
  m,
  tar_z,
  ns,
  models = 1,
  tau = 0,
  lb = "T",
  ub = "T",
  x0 = "T",
  xtol_rel = 1e-05,
  maxeval = 100,
  algorithm = "NLOPT_LD_LBFGS"
)

```

Arguments

X Matrix or data frame containing the inputs of training data. Each row represents the input setting of a data point and the columns are values of quantitative variables and qualitative variables.

Y	Vector containing the outputs of training data points.
p	Number of quantitative factors in the given dataset X.
q	Number of qualitative factors in the given dataset X.
m	A vector containing numbers of levels in qualitative factors.
tar_z	A vector containing the qualitative part of the chosen target input (described in reference 1).
ns	The chosen tuning parameter (described in reference 1)
models	The model for fitting the selected proper subset of the dataset X. 0 for EzGP model, 1 for EEzGP model.
tau	Nugget if needed. The default nugget is 0, otherwise it has to be a non-negative real value.
lb	Vector with lower bounds of the parameter estimation. "T" for applying the default setting of lb (a vector of length number of parameters whose elements are all 0.1), otherwise one must provide a vector with the length being the number of parameters.
ub	Vector with upper bounds of the parameter estimation. "T" for applying the default setting of ub (a vector of length number of parameters whose first q+1 elements are 100 and the rest number of parameters - q - 1 elements are 10), otherwise one must provide a vector with length of the number of parameters.
x0	Vector with starting values for the optimization. "T" for applying the default setting of x0 (a vector made by (lb + ub)/2), otherwise one must provide a vector with length being the number of parameters.
xtol_rel	Stopping criterion for relative change reached.
maxeval	Termination condition by specifying a maximum number of function.
algorithm	Optimization algorithm. See NLOpt Algorithms for more available algorithms.

Value

A model of class "LEzGP model" list of the following items:

- param A list containing the estimated parameters
- data A list containing the fitted dataset and the information for fitting

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

See Also

[EzGP_predict](#) to use the fitted EzGP model for prediction if your LEzGP model is fitted based on the EzGP model.

[EEzGP_predict](#) to use the fitted EEzGP model for prediction if your LEzGP model is fitted based on the EEzGP model.

Examples

```

# Example with 9 quantitative and 9 qualitative variables (dataset included in the package):
#   Fit a LEzGP model based on the EEzGP/EzGP model(with default settings), and then
#   perform the prediction.
p = 9
q = 9
m=rep(3,9)
tau = 0
X = LEzGP_data[1:60, 1:(p+q)]
Y = LEzGP_data[1:60, p+q+1]
X_new = LEzGP_data[61:70, 1:(p+q)]
tar_z = X_new[1, (p+1):(p+q)]
ns = 7
# LEzGP Model Based on EEzGP Model
model <- LEzGP_fit(X, Y, p, q, m, tar_z, ns)
y_hat <- EEzGP_predict(X_new, model)
# Results showing
model
y_hat

```

LLF_gradients	<i>The Log-likelihood Function and The Analytical Gradients in EzGP Package</i>
---------------	---

Description

Calculates the log-likelihood function value and the analytical gradients as described in reference 1.

Usage

```
LLF_gradients(X, Y, p, q, m, parv, tau = 0, models = 0)
```

Arguments

X	Matrix or data frame containing the inputs of training data. Each row represents the input setting of a data point and the columns are values of quantitative variables and qualitative variables.
Y	Vector containing the outputs of training data points.
p	Number of quantitative factors in the given dataset X.
q	Number of qualitative factors in the given dataset X.
m	A vector containing numbers of levels in qualitative factors.
parv	Parameters in the EzGP/EEzGP model.
tau	Nugget if needed. The default nugget is 0, otherwise it has to be a non-negative real value.
models	Model indicator that indicates which model the likelihoods and analytical gradients are applied to. 0 for EzGP model, 1 for EEzGP model.

Value

A list of the following items:

- `objective` The log-likelihood function value.
- `gradient` The analytical gradients.

References

1. "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors", Qian Xiao, Abhyuday Mandal, C. Devon Lin, and Xinwei Deng ([doi:10.1137/19M1288462](https://doi.org/10.1137/19M1288462))

See Also

[EzGP_fit](#) to see how an EzGP model can be fitted to a training dataset.
[EzGP_predict](#) to use the fitted EzGP model for prediction.
[EEzGP_fit](#) to see how an EEzGP model can be fitted to a training dataset.
[EEzGP_predict](#) to use the fitted EEzGP model for prediction.
[LEzGP_fit](#) to see how a LEzGP model can be fitted to a training dataset.

Examples

```
# see the examples in the documentation of the function EzGP_fit.
```

Index

* **dataset**

EzGP_data, [6](#)

LEzGP_data, [10](#)

* **example**

EzGP_data, [6](#)

LEzGP_data, [10](#)

cov_m, [2](#)

EEzGP_fit, [2](#), [3](#), [3](#), [5](#), [6](#), [14](#)

EEzGP_predict, [2](#), [3](#), [5](#), [5](#), [12](#), [14](#)

EzGP_data, [6](#)

EzGP_fit, [2](#), [3](#), [7](#), [9](#), [10](#), [14](#)

EzGP_predict, [2](#), [3](#), [8](#), [9](#), [12](#), [14](#)

LEzGP_data, [10](#)

LEzGP_fit, [2](#), [3](#), [11](#), [14](#)

LLF_gradients, [2](#), [13](#)