Package 'gpr'

February 20, 2015

Version 1.1

Date 2013-08-27

Title A Minimalistic package to apply Gaussian Process in R

License GPL-3

Author Afshin Sadeghi

Maintainer ORPHANED

Depends R (>= 2.13)

Description This package provides a minimalistic functionality necessary to apply Gaussian Process in R. They provide a selection of functionalities of GPML Matlab library.

NeedsCompilation no

Repository CRAN

Date/Publication 2014-02-07 14:10:18

X-CRAN-Original-Maintainer Afshin Sadeghi <sadeghi.afshin@gmail.com>

X-CRAN-Comment Orphaned on 2014-02-07 as the maintainer could not maintain it officially any more.

R topics documented:

`covNoise` *Independent noise function.*

Description

Independent covariance function, ie "white noise", with specified variance.

This covariance function is parameterized as: k(x^p,x^q) = s2 * solve(delta(p,q)) , in which s2 is the noise variance and solve(delta(p,q)) is a Kronecker delta function where is 1 if p==q and its zero otherwise. hyperparameter and is defined by: loghyper = [log(sqrt(s2))]

Usage

```
covNoise(loghyper= NULL ,  x = NULL , z = NULL,  testset.covariances= FALSE)
```

Arguments

`loghyper`	loghyper is hyperparameter vector variable.
`x`	Input parameter array to define the function over.
`z`	Index number of loghyper vector.
`testset.covariances`	Logic value to decide to compute testset covariances or not.

Value

If `z` is not null and `testset.covariances` is TRUE this function calculates test set covariances and if its FALSE the function computes derivative matrix. When covNoise is called without parameters is reports the minimum number of parameters other than loghyper which it can accept. The output of this function is a list consisting variables A and B. B will include testset covariances calculation when `testset.covariances` is TRUE.

Author(s)

Afshin Sadeghi

References

Carl Edward Rasmussen and Christopher K. I. Williams.Gaussian Processes for Machine Learning. *MIT Press*, 2006. ISBN 0-262-18253-X. Carl Edward Rasmussen & Hannes Nickisch. gpml(GAUSSIAN PROCESS REGRESSION AND CLASSIFICATION Toolbox) Matlab Library.

Examples

```
params= covNoise()
params
```

covSEiso *Squared Exponential covariance function with isotropic distance measure.*

Description

This covariance function is defined as: k(x^p,x^q) = sf2 * exp(- t(x^p - x^q)*inv(P)*(x^p - x^q)/2) , in which where the P matrix is ell^2 times the unit matrix and sf2 is the signal variance. The hyperparameter vector is loghyper = [log(ell) ,log(sqrt(sf2))]

Usage

```
covSEiso(loghyper = NULL , x = NULL , z =  NULL , testset.covariances= FALSE)
```

Arguments

`loghyper`	`loghyper` is hyperparameter vector variable.
`x`	`x` is Input parameter array to apply the function over.
`z`	`z` is Index number of loghyper vector.
`testset.covariances`	`testset.covariances` is a logic value to decide to compute testset covariances or not.

Value

If `z` is not null and `testset.covariances` is TRUE this function calculates test set covariances and if its FALSE the function computes derivative matrix. When covNoise is called without parameters is reports the minimum number of parameters other than loghyper which it can accept. The output of this function is a list consisting variables A and B. B will include testset covariances calculation when `testset.covariances` is TRUE.

Author(s)

Afshin Sadeghi

References

Carl Edward Rasmussen and Christopher K. I. Williams.Gaussian Processes for Machine Learning. *MIT Press*, 2006. ISBN 0-262-18253-X. Carl Edward Rasmussen & Hannes Nickisch. gpml(GAUSSIAN PROCESS REGRESSION AND CLASSIFICATION Toolbox) Matlab Library.

Examples

```
params= covSEiso()
params
```

covSum *Sum of covariance functions*

Description

Provides sum functionality of two covariance functions .

Usage

```
covSum(covfuncsum , logtheta  = NULL,  x  = NULL, z = NULL, testset.covariances= FALSE)
```

Arguments

covfuncsum	covfuncsum is a string variable which is consist of two covariance function names seperated by a "," .
logtheta	logtheta is hyperparameter vector variable.
x	Input parameter to define the function over
z	Index number of logtheta vector
testset.covariances	Logic value to decide to compute testset covariances or not.

Value

If z is not null and testset.covariances is TRUE this function calculates test set covariances and if its FALSE the function computes derivative matrix. When covNoise is called without parameters is reports the minimum number of parameters other than logtheta which it can accept. The output of this function is a list consisting variables A and B. B will include testset covariances calculation when testset.covariances is TRUE.

Author(s)

Afshin Sadeghi

References

Carl Edward Rasmussen and Christopher K. I. Williams.Gaussian Processes for Machine Learning. *MIT Press*, 2006. ISBN 0-262-18253-X. Carl Edward Rasmussen & Hannes Nickisch. gpml(GAUSSIAN PROCESS REGRESSION AND CLASSIFICATION Toolbox) Matlab Library.

Examples

```
params= covSum("covSEiso,covNoise")
params
```

gpr *Gaussian process regression function.*

Description

This function delivers two applications: 1. Training : When no test data are given, the function returns minus the log likelihood and its partial derivatives with respect to the hyperparameters. Training method is to fit the hyperparameters.

2.Prediction: If test data are given, then marginal Gaussian predictions are computed, and function returns mean and variance describing the Gaussian predictions. In cases where the covariance function has noise contributions, the variance returned in S2 variable is for noisy test targets. To have variance of the noise-free latent function, the noise variance should be substracted .

Usage

```
gpr(logtheta, covfunc.gpr, x, y, xstar = NULL,  partial.derivatives = FALSE)
```

Arguments

`logtheta`	`logtheta` is a vector that specifies hyperparameter variable.
`covfunc.gpr`	`covfunc.gpr` is a string value containing name of a covariance function to apply gpr on it.
`x`	`x` is Input parameter array and it is passed to `covfunc.gpr` function in gpr application. Its a n by D array
`y`	`y` is an other input parameter of `covfunc.gpr` function in gpr application. It is usually a vector that defines target value of function covfunc.gpr and the input array x and has n rows.
`xstar`	`xstar` is input array of test data. It has the same dimentions of `x`.
`partial.derivatives`	Logical value that when it is TRUE and the test data is given gpr executes prediction process.

Details

There are only `covSum`, `covNoise` and `covSEiso` covariance functions available in this package to be used with gaussion process. More ever this is an open source script with line by line description so that you can learn the code and write your own covariance functions to call gpr on them. If you like to contribute and add your own functions to this package please contact the package maintainer. Please refere to references to get info about the other consistent covariance functions.

Value

When `partial.derivatives` is FALSE returns a list containing minus the log likelihood and its vector of partial derivatives . When `partial.derivatives` is TRUE returns a list that consists of two objects. The first is a mean vector which same number of rows of `x` and `y` and the second object variance vector that has same number of rows of `x` and `y`. They describe the result Gaussian prediction.

Author(s)

Afshin Sadeghi

References

1. Carl Edward Rasmussen and Christopher K. I. Williams.Gaussian Processes for Machine Learning. *MIT Press*, 2006. ISBN 0-262-18253-X. 2. Carl Edward Rasmussen & Hannes Nickisch. gpml(GAUSSIAN PROCESS REGRESSION AND CLASSIFICATION Toolbox) Matlab Library.

`minimize` *Minimize function for differentiable multivariate functions.*

Description

Performs mimization on a differentiable multivariate function and returns a function value and a vector of partial derivatives the input function. The Polack-Ribiere flavour of conjugate gradients is used to compute search directions, and a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria is used together with the slope ratio method for guessing initial step sizes.

Details from gpml: Please consider that minimize execution path computes a new search direction during extrapolation using conjugate gradients (Polack-Ribiere flavour), or reverts to steepest if there was a problem in the previous line-search. Returns the best value so far, if two consecutive line-searches fail, or whenever it run out of function evaluations or line-searches. During extrapolation, the "f" function may fail either with an error or returning Nan or Inf, and minimize should handle this gracefully. If minimize stopped within a few iterations, it could be an indication that the function values and derivatives are not consistent (ie, there may be a bug in the implementation of your "f" function).

Usage

```
minimize(X, f, .length, covfunc, x, y)
```

Arguments

`X`	Starting point is given by array `X` .
`f`	`f` is a string value containing the function name which is supposed to apply minimize on it.
`.length`	`.length` defines how long minimize procedure must be executed.
`covfunc`	is string value that gives the name of covariance function which is passed on to the function `f`.
`x`	Input parameter which is passed on to the function `f`.
`y`	An other input parameter like `x` which is passed on to the function. It usually defines target value function covfunc and the input array `x`. `f`.

Value

if .length is positive, it defines the maximum number of line searches, if negative its absolute gives the maximum allowed number of function evaluations. .length can have a second optional component, which indicates the reduction in function value to be expected in the first line-search. its default value is 1.0 .

The function returns when either its length is up, or if no further progress can be made due to a (local) minimum or due to numerical problems. The function returns a list consisting of the found solution `X`, a vector of function values `fX` indicating the progress made and `i` the number of iterations.

Author(s)

Afshin Sadeghi

References

Carl Edward Rasmussen and Christopher K. I. Williams.*Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 0-262-18253-X. Carl Edward Rasmussen & Hannes Nickisch. gpml(GAUSSIAN PROCESS REGRESSION AND CLASSIFICATION Toolbox) Matlab Library.

Examples

```
loghyper= array(c(-1,-1,-1), dim=c(3,1))
covfunc ="covSum,covSEiso,covNoise"
x= array(c(1,1,0,0), dim=c(2,2))
y= array(c(1,0), dim=c(2,1))
loghyper = minimize(loghyper, 'gpr', 10, covfunc, x, y)
loghyper
```

Index