

computeOptimCriteria

Object function to compute the loss of a fully initialized and tuned regression model. Computes the mean std. log. loss of angles MSLLA or radius MSLLR as function evaluation value for bayesopt. Perform noise adjustment in cycles in bayesopt.

Syntax

```
MSLL = computeOptimCriteria(OptVar, Mdl, TestDS, SLL, verbose)
```

Description

MSLL = **computeOptimCriteria**(**OptVar**, **Mdl**, **TestDS**, **SLL**, **verbose**)

Input Arguments

OptVar optimization variable. Noise level passed by bayesopt algorithm.

Mdl model struct.

TestDS loaded test data by infront processed sensor array simulation.

SLL indicates which loss is used for MSLL. SLLA for angle and SLLR for radius.

verbose activates prompt for true or 1. Vice versa for false or 0.

Output Arguments

MSLL mean standardized logarithmic loss. Function evaluation value for optimGPR

Requirements

- Other m-files required: None
- Subfunctions: tuneKernel, lossDS, mean
- MAT-files required: None

See Also

- [optimGPR](#)
- [tuneKernel](#)
- [lossDS](#)
- [mean](#)

Created on March 05. 2021 by Tobias Wulf. Copyright Tobias Wulf 2021.

```
function MSLL = computeOptimCriteria(OptVar, Mdl, TestDS, SLL, verbose)

    % push current variance value into GPR
    Mdl.s2n = OptVar.s2n;

    % tune kernel with new noise variance
    Mdl = tuneKernel(Mdl, verbose);

    % get loss on dataset for angular prediction
    switch SLL
        case 'SLLA'
            [~, SLL] = lossDS(Mdl, TestDS);
```

```
        case 'SLLR'  
            [~, ~, SLL] = lossDS(Mdl, TestDS);  
  
        otherwise  
            error('Unknown SLL %s.', SLL);  
        end  
  
        % return mean loss to evaluate optimization run  
        MSSL = mean(SLL);  
    end
```