computeStdLogLoss

Compute SLL loss between test targets and predictive mean dependend on predictive variance plus used variance for noisy covariance matrix as variance of normal distriubtion over predictive means s2 = fcov + s2n. The pred functions returns the standard deviation s so sqrt of the variance.

Syntax

```
[SLL, SE, s2] = computeStdLogLoss(y, fmean, s)
```

Description

[SLL, SE, s2] = computeStdLogLoss(y, fmean, s) compute standardized logaritmic loss by squared error of ideal test data predicted data and standard deviation of predicted data.

Input Argurments

y column vector of ideal data to compare with.

fmean column vector of predictive mean.

s column vector of standard deviation related to predictive mean.

Output Argurments

SLL column vector of standardized logarithmic loss.

SE squared error between **y** and **fmean**.

s2 variance column vector.

Requirements

- Other m-files required: None
- Subfunctions: log
- MAT-files required: None

See Also

- predFrame
- predDS
- lossDS

Created on February 15. 2021 by Tobias Wulf. Copyright Tobias Wulf 2021.

```
function [SLL, SE, s2] = computeStdLogLoss(y, fmean, s)
    arguments
        % validate inputs as real column vectors of same length
        y (:,1) double {mustBeReal, mustBeVector}
        fmean (:,1) double {mustBeReal, mustBeVector, mustBeEqualSize(y, fmean)}
        s (:,1) double {mustBeReal, mustBeVector, mustBeEqualSize(y, s)}
    end
    % squared error
    SE = (y - fmean).^2;

    % s as standard deviation of nomal destributed fmean so square it for
    % variance.
```

```
    s2 = s.^2;

    % logaritmic error
    SLL = 0.5 * (log(2 * pi * s2) + SE ./ s2);
end

% Custom validation functions
function mustBeEqualSize(a, b)
    if ~isequal(length(a), length(b))
        eid = 'Size:notEqual';
        msg = 'Vectors must be the same length.';
        throwAsCaller(MException(eid,msg))
    end
end
```

Published with MATLAB® R2020b