QFC

Quadratic Fractional Covariance function. Computes covariance matrix K. Works with raw matrix data. Precise solution.

Syntax

```
K = QFC(Ax, Bx, Ay, By, theta)
```

Description

K = QFC(Ax, Bx, Ay, By, theta) computes quadratic distances bewtween data points and parametrize it with height and length scales. Computes distance with quadratic Frobenius Norm.

Input Argurments

Ax matrix of cosine simulation components.

Bx matrix of cosine simulation components.

Ay matrix of sine simulation components.

By matrix of sine simulation components.

theta vector of kernel parameters.

Output Argurments

K noise free covarianc matrix.

Requirements

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

See Also

- initQFC
- meanPolyQFC

Created on November 06. 2019 by Klaus Jünemann. Copyright Klaus Jünemann 2019.

```
function K = QFC(Ax, Bx, Ay, By, theta)
    arguments
        % validate data as real matrices of same size in 1st and 2nd dimension
        Ax (:,:,:) double {mustBeReal}
        Bx (:,:,:) double {mustBeReal, mustBeFitSize(Ax,Bx)}
        Ay (:,:,:) double {mustBeReal, mustBeFitSize(Ax,Ay)}
        By (:,:,:) double {mustBeReal, mustBeFitSize(Ax,By)}
        % validate kernel parameters as 1x2 vector
        theta (1,2) double {mustBeReal}
    end

    % get number of observations for each dataset, cosine and sine matrices have
    % equal sizes just extract size from one
    [~, ~, M] = size(Ax);
    [~, ~, N] = size(Bx);
```

```
    % expand covariance parameters, variance and lengthscale
    c2 = 2 * theta(2)^2; % 2*sl^2
    c1 = theta(1) * c2;   % s2f * c

    % allocate memory for K
    K = zeros(M, N);

    % loop through observation points and compute the covariance for each
    % observation against another
    for m = 1:M
        for n = 1:N
            % get distance between m-th and n-th observation
            distCos = Ax(:,:,m) - Bx(:,:,n);
            distSin = Ay(:,:,m) - By(:,:,n);

            % compute quadratic frobenius norm distance as separated
            % distances of cosine and sine, norm of vector fields
            r2 = sum(distCos .^ 2 , 'all') + sum(distSin .^ 2 , 'all');

            % engage lengthscale and variance on distance
            K(m,n) = c1 / (c2 + r2);
        end
    end
end

function mustBeFitSize(A, B)
    % Test for equal size
    if ~isequal(size(A,1,2), size(B,1,2))
        eid = 'Size:notEqual';
        msg = 'Sizes of  are not fitting.';
        throwAsCaller(MException(eid,msg))
    end
end
```

Published with MATLAB® R2020b