initQFCAPX

Attaches QFCAPX kernel to model struct. Depending on mean options attach zero mean functions and sets all related kernel parameters and dependencies to zero. If mean is polynom fitting, attaches meanPolyQFCAPX as basis function to build polynom matrix H and sets a none zero mean function. Computes dataset inputs vectors or scalars. Kernel works on vector data.

Syntax

```
Mdl = initQFCAPX(Mdl)
```

Description

Mdl = initQFCAPX(Mdl) loads approximated quadratic fraction covariance function and basis function depending on mean in**Mdl** struct. Sets input function to Frobenius Norm. Reprocess training matrix data to vector data.

Input Argurments

Mdl struct with model parameter and training data.

Output Argurments

Mdl struct with attached kernel functionality

Requirements

- Other m-files required: None
- Subfunctions: QFCAPX, meanPolyQFCAPX, frobeniusNorm
- MAT-files required: None

See Also

- initKernel
- meanPolyQFCAPX
- QFCAPX
- frobeniusNorm


```
function Mdl = initQFCAPX(Mdl)

    % set QFC kernel function
    Mdl.kernelFun = @QFCAPX;

    % set input transformation function to apply adjustments to
    % covariance function, norm input matrice with frobenius norm to scalar or
    % vectors.
    Mdl.inputFun = @(X) frobeniusNorm(X, false);

    % transform traning data to vectors
    Ncos = zeros(Mdl.N, 1);
    Nsin = zeros(Mdl.N, 1);
    for n = 1:Mdl.N
        Ncos(n) = Mdl.inputFun(Mdl.Xcos(:,:,n));
        Nsin(n) = Mdl.inputFun(Mdl.Xsin(:,:,n));
    end

    % update training data with norm vectors
```

```
    Mdl.Xcos = Ncos;
    Mdl.Xsin = Nsin;

    % set mean function to compute cosine and sine H matrix
    switch Mdl.mean
        % zero mean m(x) = 0
        case 'zero'
            % set polyDegree to -1 for no polynom indication
            Mdl.polyDegree = -1;

            % set basis function
            Mdl.basisFun = @(X) 0;

        % mean by polynom m(x) = H' * beta
        case 'poly'
            % set basis function produces a (polyDeg+1)xN H matrix
            Mdl.basisFun = @(X) meanPolyQFCAPX(X, Mdl.polyDegree);

        % end mean select QFC kernel
        otherwise
            error('Unknown mean function %.', Mdl.mean);
    end
end
```

Published with MATLAB® R2020b