covafill

v0.2.1

Generated by Doxygen 1.8.9.1

Thu Feb 18 2016 11:52:08

Contents

1 Main Page

covafill is a C++ template library for local polynomial regression of covariates in state-space models. The covafill library is based on the `Eigen` library for linear algebra, and includes several modules:

- The Core module which provides the base functionality for local polynomial regression
- The Tree module which provides a search tree approximation to local polynomial regression
- The Interpolate module which provides classes for cubic interpolation in 1-3 dimensions
- The JAGS module, which provides a module for using covafill with `JAGS`
- The TMB module which provides functionality to use covafill with `TMB`.

The Core module

The Core module provides the class covafill for local polynomial regression.

Local polynomial regression

For simplicity, consider the univariate model

$$y_i = g(x_i) + \epsilon_i$$

where $g : \mathbb{R} \mapsto \mathbb{R}$ is a smooth function and $\epsilon_i \sim N(0, \sigma^2)$. To do local polynomial regression of g at x_0, we do a taylor expansion of order p,

$$g(x) \approx g(x_0) + g^{(1)}(x_0)(x - x_0) + \frac{1}{2!}g^{(2)}(x_0)(x - x_0)^2 + \cdots + \frac{1}{p!}g^{(p)}(x_0)(x - x_0)^p$$

Substituting into the original model,

$$y_i = g(x_0) + g^{(1)}(x_0)(x - x_0) + \frac{1}{2!}g^{(2)}(x_0)(x - x_0)^2 + \cdots + \frac{1}{p!}g^{(p)}(x_0)(x - x_0)^p + \epsilon_i$$

we obtain a linear model with coefficients $\theta = (g(x_0), g^{(1)}(x_0), g^{(2)}(x_0), \ldots, g^{(p)}(x_0))^T$, obervations $\mathbf{Y} = (y_1, y_2, \ldots, y_n)^T$, and the design matrix

$$\mathbf{X} = \begin{pmatrix} 1 & (x_1 - x_0) & \frac{1}{2!}(x_1 - x_0)^2 & \cdots & \frac{1}{p!}(x_1 - x_0)^p \\ 1 & (x_2 - x_0) & \frac{1}{2!}(x_2 - x_0)^2 & \cdots & \frac{1}{p!}(x_2 - x_0)^p \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & (x_n - x_0) & \frac{1}{2!}(x_n - x_0)^2 & \cdots & \frac{1}{p!}(x_n - x_0)^p \end{pmatrix}$$

As we are interested in a local estimate, observations are weighed by their distance to x_0. The weights form the diagonal matrix $\mathbf{W}$ with

$$w_{ii} = \det(H^{-1})\left(1 - \|H^{-1} \cdot (x_i - x_0)\|^2\right) \vee 0$$

Now the estimates are obtained by

$$\hat{\theta} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{Y}$$

giving both the estimated function value at x_0 and estimates of the first p derivatives.

The Interpolate module

Cubic interpolation

The Tree module

Search Tree

Approximation to local polynomial regression

The JAGS and TMB modules

The JAGS and TMB modules...

JAGS example

```
model {
      cf <- covafill(x,obsC,obs,h,2.0)
      sigma ~ dunif(0,100)
      tau <- pow(sigma, -2)
      for(i in 1:N) {
            y[i] ~ dnorm(cf[i],tau)
        }
}
```

TMB example

```
#include <TMB.hpp>
#include <covafill/TMB>

template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_MATRIX(obs);
  DATA_MATRIX(coord);
  DATA_VECTOR(covObs);
  DATA_INTEGER(p);
  DATA_VECTOR(h);

  PARAMETER(logObsSd);
  PARAMETER(logObsTSd);
  PARAMETER(logStatSd);
  PARAMETER_MATRIX(x);

  Type nll = 0.0;
  covafill<Type> cf(coord,covObs,h,p);

  // Contribution from states
  for(int i = 1; i < x.cols(); ++i){
    nll -= dnorm(x(0,i), x(0,i-1), exp(logStatSd),true);
    nll -= dnorm(x(1,i), x(1,i-1), exp(logStatSd),true);
  }
  // contribution from observations
  for(int i = 0; i < obs.cols(); ++i){
    nll -= dnorm(obs(0,i), x(0,i), exp(logObsSd),true);
    nll -= dnorm(obs(1,i), x(1,i), exp(logObsSd),true);
    vector<Type> tmp = x.col(i);
    Type val = evalFill((CppAD::vector<Type>)tmp, cf)[0];
    nll -= dnorm(obs(2,i), val, exp(logObsTSd),true);
  }


  return nll;
}
```

2 Module Index

2.1 Modules

Here is a list of all modules:

3 Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

4 Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

5 Module Documentation

5.1 Core module

Classes

- class covafill< scalartype_ >

  Class to do local polynomial regression.

5.1.1 Detailed Description

The Core module of covafill provides a class for local polynomial regression.

```
#include <covafill/Core>
```

5.2 Interpolation module

Classes

- class bicubicInterpolation< scalartype_ >

  Class for bi-cubic interpolation of local polynomial regression on a square.
- class cubicInterpolation< scalartype_ >

  Class for cubic interpolation in dimension 1-3 of local polynomial regression on a square.
- class ncubicInterpolation< scalartype_ >

  Class for n-cubic interpolation (n = 1,2,3) of local polynomial regression on a square. The class should not be used as anything but a common parent for the dimension specific interpolation classes.
- class tricubicInterpolation< scalartype_ >

  Class for tri-cubic interpolation of local polynomial regression on a square.
- class unicubicInterpolation< scalartype_ >

  Class for cubic interpolation of local polynomial regression on a square.

5.2.1 Detailed Description

The Interpolate module of covafill provides classes for cubic interpolation in 1-3 dimensions. The class serves as a common wrapper for the dimension specific interpolation classes.

```
#include <covafill/Interpolate>
```

5.3 JAGS module

Classes

- class jags::covafillJAGS::covafillModule

  Class that defines a JAGS Module for local polynomial regression.
- class jags::covafillJAGS::covafillJAGS

  Class that defines the covafill function for local polynomial regression to be used in a JAGS model.

5.3.1 Detailed Description

The JAGS module defines a JAGS module to use the function covafill for local polynomial regression in a JAGS model.

```
#include <covafill/JAGS>
```

5.4 TMB module

Classes

- class atomicEvalFill< Type >

  CppAD atomic class to use estimated derivatives in automatic differentiation. See CppAD::atomic_base for further documentation.
- class atomicEvalTree< Type >

  CppAD atomic class to use estimated derivatives in automatic differentiation. See CppAD::atomic_base for further documentation.

Functions

- CppAD::vector< double > evalFill (CppAD::vector< double > tx, const covafill< double > &cf) CSKIP(

  Evaluates a covafill object, cf, at the coordinates tx.
- template<class Type >
  CppAD::vector< Type > evalFill (CppAD::vector< Type > tx, const covafill< AD< Type > > &cf)
- template<class Type >
  CppAD::vector< AD< Type > > evalFill (CppAD::vector< AD< Type > > tx, covafill< AD< Type > > cf)
- CppAD::vector< double > evalTree (CppAD::vector< double > tx, const covatree< double > &ct) CSKIP(

  Evaluates a covatree object, ct, at the coordinates tx.
- template<class Type >
  CppAD::vector< Type > evalTree (CppAD::vector< Type > tx, const covatree< AD< Type > > &ct)
- template<class Type >
  CppAD::vector< AD< Type > > evalTree (CppAD::vector< AD< Type > > tx, covatree< AD< Type > > ct)

5.4.1 Detailed Description

The TMB module of covafill provides functions to evaluate a covafill or covatree object from a TMB model such that the estimated gradients are used in the automatic differentiation.

```
#include <covafill/TMB>
```

5.4.2 Function Documentation

5.4.2.1 template< class Type > CppAD::vector<Type> evalFill (CppAD::vector< Type > *tx*, const covafill< AD< Type > > & *cf*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

5.4.2.2 template< class Type > CppAD::vector<AD<Type > > evalFill (CppAD::vector< AD< Type > > *tx*, covafill< AD< Type > > *cf*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

5.4.2.3 template< class Type > CppAD::vector<Type> evalTree (CppAD::vector< Type > *tx*, const covatree< AD< Type > > & *ct*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

5.4.2.4 template< class Type > CppAD::vector< AD< Type > > evalTree (CppAD::vector< AD< Type > > *tx*, covatree< AD< Type > > *ct*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

5.5 Tree module

Classes

- class covanode< scalartype_ >

  Class that defines nodes of a covatree.
- class covatree< scalartype_ >

  Class that defines a covatree for search tree approximated local polynomial regression.

5.5.1 Detailed Description

The Tree module of covafill provides a class for search tree approximated local polynomial regression.

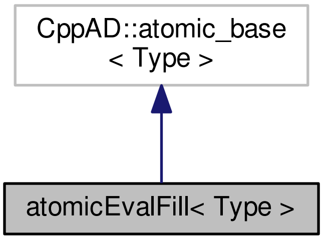
```
#include <covafill/Interpolate>
```

6 Class Documentation

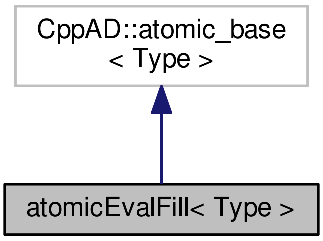
6.1 atomicEvalFill< Type > Class Template Reference

CppAD atomic class to use estimated derivatives in automatic differentiation. See CppAD::atomic_base for further documentation.

Inheritance diagram for atomicEvalFill< Type >:



Collaboration diagram for atomicEvalFill< Type >:



Public Member Functions

- atomicEvalFill (const char ∗name, covafill< AD< Type > > cf_)

  Constructs class to evaluate atomic function.

6.1.1 Detailed Description

template<class Type>class atomicEvalFill< Type >

CppAD atomic class to use estimated derivatives in automatic differentiation. See CppAD::atomic_base for further documentation.

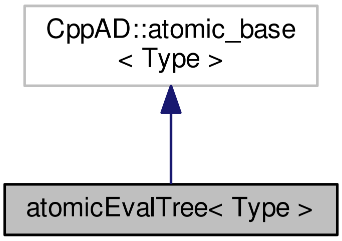
The documentation for this class was generated from the following file:

- atomic.hpp

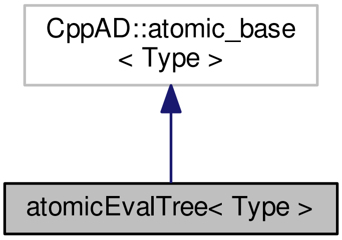
6.2 atomicEvalTree< Type > Class Template Reference

CppAD atomic class to use estimated derivatives in automatic differentiation. See CppAD::atomic_base for further documentation.

Inheritance diagram for atomicEvalTree< Type >:



Collaboration diagram for atomicEvalTree< Type >:



Public Member Functions

- atomicEvalTree (const char ∗name, covatree< AD< Type > > ct_)

  Constructs class to evaluate atomic function.

6.2.1 Detailed Description

template< class Type > class atomicEvalTree< Type >

CppAD atomic class to use estimated derivatives in automatic differentiation. See CppAD::atomic_base for further documentation.

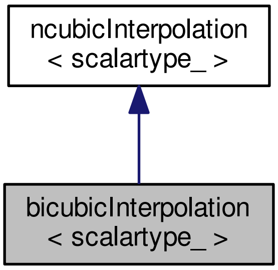
The documentation for this class was generated from the following file:

- atomic_Tree.hpp

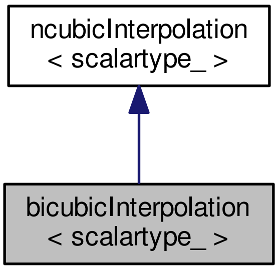
6.3 bicubicInterpolation< scalartype_ > Class Template Reference

Class for bi-cubic interpolation of local polynomial regression on a square.

Inheritance diagram for bicubicInterpolation< scalartype_ >:



Collaboration diagram for bicubicInterpolation< scalartype_ >:



Public Member Functions

- bicubicInterpolation (covafill< scalartype > ∗cf, vectortype minCoord, vectortype maxCoord)

  Constructs a bicubicInterpolation class from a covafill class cf, an boundaries of the interpolation square defined by the minimum coordinates, minCoord, and maximum coordinates, maxCoord, in each dimension, e.g., minCoord = (0,0) and maxCoord = (1,1).

- virtual vectortype operator() (vectortype newcoord)

  Calculates the interpolation prediction at newcoord.

Additional Inherited Members

6.3.1 Detailed Description

template$<$typename scalartype_$>$class bicubicInterpolation$<$ scalartype_ $>$

Class for bi-cubic interpolation of local polynomial regression on a square.

The documentation for this class was generated from the following file:

- bicubicInterpolation.hpp

6.4 covafill$<$ scalartype_ $>$ Class Template Reference

Class to do local polynomial regression.

Public Member Functions

- covafill (const covafill$<$ scalartype_ $>$ &x)

  Constructs a covafill class from another covafill class x.
- covafill (matrixtype coordinates_, vectortype observations_)

  Constructs a covafill class with coordinates matrix coordinates_, observation vector obervations, bandwiths 1, and polynomial degree 2.
- covafill (matrixtype coordinates_, vectortype observations_, scalartype h_, int p_)

  Constructs a covafill class with coordinates matrix coordinates_, observation vector obervations, bandwiths h_, and polynomial degree p_.
- covafill (matrixtype coordinates_, vectortype observations_, vectortype h_, int p_)

  Constructs a covafill class with coordinates matrix coordinates_, observation vector obervations, bandwiths h_, and polynomial degree p_.
- int getDim () const
- void setH (scalartype h_)
- void setH (vectortype h_)
- vectortype operator() (vectortype x0, bool returnAll=false) const

  Calculates the local polynomial regression estimate at x0. If returnAll is false, then only the function and first derivative estimates are returned. Otherwise all estimates are returned.
- vectortype operator() (vectortype x0, scalartype excludeRadius, bool returnAll=false) const

  Calculates the local polynomial regression estimate at x0. All observations with coordinates x such that $||x-x_0|| > r$, where r is exludeRadius. If returnAll is false, then only the function and first derivative estimates are returned. Otherwise all estimates are returned.
- covafill$<$ scalartype $>$ & operator= (const covafill$<$ scalartype $>$ &rhs)

  Assignment operator for covafill.

Public Attributes

- matrixtype coordinates
- vectortype observations
- int p
- vectortype h

6.4.1 Detailed Description

template$<$typename scalartype_$>$class covafill$<$ scalartype_ $>$

Class to do local polynomial regression.

6.4.2 Member Function Documentation

6.4.2.1 template< typename scalartype_ > int covafill< scalartype_ >::getDim ( ) const

Returns the covariate dimension.

6.4.2.2 template< typename scalartype_ > void covafill< scalartype_ >::setH (scalartype *h_*)

Sets all bandwiths to h_.

Referenced by covafill< scalartype_ >::covafill().

6.4.2.3 template< typename scalartype_ > void covafill< scalartype_ >::setH (vectortype *h_*)

Sets the bandwiths from a vector. The length of h_ must match the covariate dimension.

6.4.3 Member Data Documentation

6.4.3.1 template<typename scalartype_> matrixtype covafill< scalartype_ >::coordinates

Coordinates/covariates of input.

Referenced by covatree< scalartype_ >::covatree(), and covafill< scalartype_ >::operator=().

6.4.3.2 template<typename scalartype_> vectortype covafill< scalartype_ >::h

Vector of (positive) bandwiths - one for each covariate.

Referenced by covafill< scalartype_ >::operator=().

6.4.3.3 template<typename scalartype_> vectortype covafill< scalartype_ >::observations

Input observations.

Referenced by covafill< scalartype_ >::operator=().

6.4.3.4 template<typename scalartype_> int covafill< scalartype_ >::p

Polynomial degree.

Referenced by covafill< scalartype_ >::operator=().

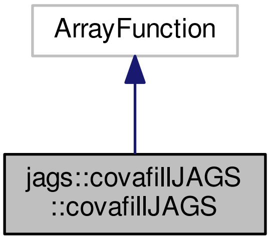
The documentation for this class was generated from the following files:

- covafill.hpp
- covafill_constructors.hpp
- covafill_operators.hpp
- covafill_privateFunctions.hpp
- covafill_publicFunctions.hpp

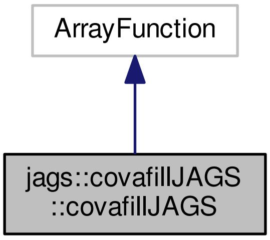
6.5 jags::covafillJAGS::covafillJAGS Class Reference

Class that defines the covafill function for local polynomial regression to be used in a JAGS model.

Inheritance diagram for jags::covafillJAGS::covafillJAGS:



Collaboration diagram for jags::covafillJAGS::covafillJAGS:



Public Member Functions

- covafillJAGS ()

  Default constructor.
- void evaluate (double ∗value, std::vector< double const ∗ > const &args, std::vector< std::vector< unsigned int > > const &dims) const

  Evaluates a covafill object.
- std::vector< unsigned int > dim (std::vector< std::vector< unsigned int > > const &dims, std::vector< double const ∗ > const &values) const

  Returns dimension of result.
- bool checkParameterDim (std::vector< std::vector< unsigned int > > const &dims) const

  Function to check parameter dimensions. Currently returns true for any input dimension.

6.5.1 Detailed Description

Class that defines the covafill function for local polynomial regression to be used in a JAGS model.

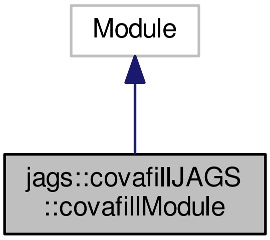
The documentation for this class was generated from the following files:

- covafillJAGS.hpp
- covafillJAGS.cpp

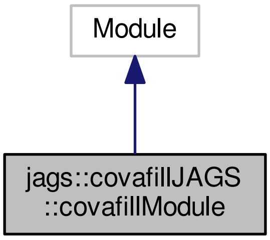
6.6 jags::covafillJAGS::covafillModule Class Reference

Class that defines a JAGS Module for local polynomial regression.

Inheritance diagram for jags::covafillJAGS::covafillModule:



Collaboration diagram for jags::covafillJAGS::covafillModule:



6.6.1 Detailed Description

Class that defines a JAGS Module for local polynomial regression.

The documentation for this class was generated from the following file:

- covafill.cpp

6.7 covanode$<$ scalartype_ $>$ Class Template Reference

Class that defines nodes of a covatree.

Public Member Functions

- covanode (matrixtype coordSplit, scalartype minSplitSize_, covafill$<$ scalartype $>$ $*$cf, vectortype minCoords, vectortype maxCoords)

Constructs a node in a covatree.
- int getDim ()

  Get coordinate dimension.
- vectortype operator() (vectortype newcoord)

  Returns the interpolated value at newcoord of the local polynomial regressions at the corners of the boundary box.

6.7.1 Detailed Description

template<typename scalartype_>class covanode< scalartype_ >

Class that defines nodes of a covatree.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 template<typename scalartype_ > covanode< scalartype_ >::covanode (matrixtype *coordSplit*, scalartype *minSplitSize_*, covafill< scalartype > ∗ *cf*, vectortype *minCoords*, vectortype *maxCoords*)

Constructs a node in a covatree.

Parameters

coordSplit	The remaining coordinates in the split at which we are now creating a note
*minSplitSize_*	The minimum number of coordinates at which the node will create a subtree.
cf	A covafill object for local polynomial regression at the corners of the boundary box.
minCoords	Minimum coordinates of the boundary box corners, e.g., (0,0) in two dimensions.
maxCoords	Maximum coordinates of the boundary box corners, e.g., (1,1) in two dimensions.

The documentation for this class was generated from the following files:

- covanode.hpp
- covanode_constructors.hpp
- covanode_operators.hpp

6.8 covatree< scalartype_ > Class Template Reference

Class that defines a covatree for search tree approximated local polynomial regression.

Public Member Functions

- covatree (scalartype minSplitSize_, covafill< scalartype > ∗cf)

  Constructs a tree from a covafill object cf with minimum number of coordinates at which a sub tree will be created minSplitSize_.
- int getDim ()

  Get coordinate dimension.
- vectortype operator() (vectortype newcoord) const

  Returns the interpolated value at newcoord of the local polynomial regressions at the corners of the boundary box.

6.8.1 Detailed Description

template<typename scalartype_>class covatree< scalartype_ >

Class that defines a covatree for search tree approximated local polynomial regression.

The documentation for this class was generated from the following files:

- covatree.hpp
- covatree_constructors.hpp

6.9 cubicInterpolation< scalartype_ > Class Template Reference

Class for cubic interpolation in dimension 1-3 of local polynomial regression on a square.

Public Member Functions

- cubicInterpolation (covafill< scalartype > ∗cf, vectortype minCoords, vectortype maxCoords)

  Constructs a bicubicInterpolation class from a covafill class cf, an boundaries of the interpolation square defined by the minimum coordinates, minCoord, and maximum coordinates, maxCoord, in each dimension, e.g., minCoord = (0,0) and maxCoord = (1,1).

- vectortype operator() (vectortype newcoord)

  Returns the interpolation prediction at newcoord.

6.9.1 Detailed Description

template<typename scalartype_>class cubicInterpolation< scalartype_ >

Class for cubic interpolation in dimension 1-3 of local polynomial regression on a square.

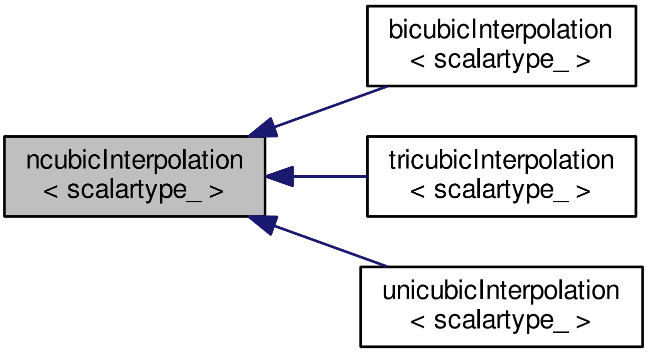
The documentation for this class was generated from the following file:

- cubicInterpolation.hpp

6.10 ncubicInterpolation< scalartype_ > Class Template Reference

Class for n-cubic interpolation (n = 1,2,3) of local polynomial regression on a square. The class should not be used as anything but a common parent for the dimension specific interpolation classes.

Inheritance diagram for ncubicInterpolation< scalartype_ >:



Public Member Functions

- ncubicInterpolation (covafill< scalartype > *cf, vectortype minCoord_, vectortype maxCoord_)

  Constructs a n-cubicInterpolation class from a covafill class cf, an boundaries of the interpolation square defined by the minimum coordinates, minCoord, and maximum coordinates, maxCoord, in each dimension, e.g., minCoord = (0,0) and maxCoord = (1,1).
- virtual vectortype operator() (vectortype newcoord)=0

  Calculates the interpolation prediction at newcoord.

Protected Member Functions

- ncubicInterpolation (vectortype minCoord_, vectortype maxCoord_)

  Constructor from coordinates. Should in general not be called.

Protected Attributes

- int dim
- vectortype minCoord
- vectortype maxCoord

6.10.1 Detailed Description

template<typename scalartype_>class ncubicInterpolation< scalartype_ >

Class for n-cubic interpolation (n = 1,2,3) of local polynomial regression on a square. The class should not be used as anything but a common parent for the dimension specific interpolation classes.

6.10.2 Member Data Documentation

6.10.2.1 template<typename scalartype_> int ncubicInterpolation< scalartype_ >::dim [protected]

Dimension of coordinates, i.e., the n in n-cubic.

6.10.2.2 template<typename scalartype_> vectortype ncubicInterpolation< scalartype_ >::maxCoord [protected]

maximum coordinates of boundary box.

6.10.2.3 template<typename scalartype_> vectortype ncubicInterpolation< scalartype_ >::minCoord [protected]

Minimum coordinates of boundary box.

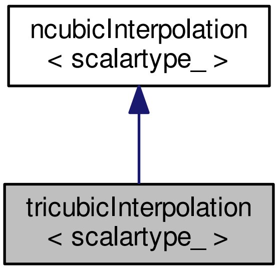
The documentation for this class was generated from the following file:

- ncubicInterpolation.hpp

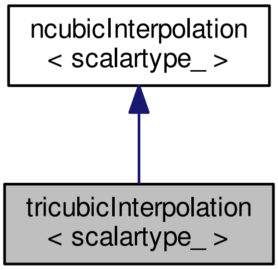
6.11 tricubicInterpolation< scalartype_ > Class Template Reference

Class for tri-cubic interpolation of local polynomial regression on a square.

Inheritance diagram for tricubicInterpolation< scalartype_ >:



Collaboration diagram for tricubicInterpolation< scalartype_ >:



Public Member Functions

- tricubicInterpolation (covafill< scalartype > ∗cf, vectortype minCoord, vectortype maxCoord)

  Constructs a tricubicInterpolation class from a covafill class cf, an boundaries of the interpolation square defined by the minimum coordinates, minCoord, and maximum coordinates, maxCoord, in each dimension, e.g., minCoord = (0,0,0) and maxCoord = (1,1,1).
- virtual vectortype operator() (vectortype newcoord)

  Calculates the interpolation prediction at newcoord.

Additional Inherited Members

6.11.1 Detailed Description

template<typename scalartype_>class tricubicInterpolation< scalartype_ >

Class for tri-cubic interpolation of local polynomial regression on a square.

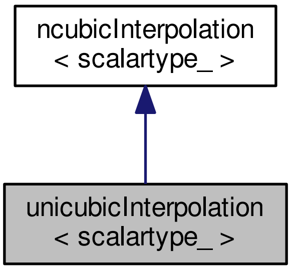
The documentation for this class was generated from the following file:

- tricubicInterpolation.hpp

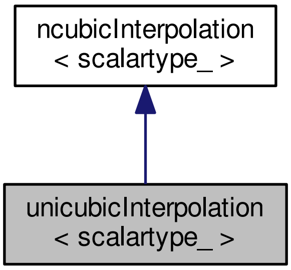
6.12 unicubicInterpolation$<$ scalartype_ $>$ Class Template Reference

Class for cubic interpolation of local polynomial regression on a square.

Inheritance diagram for unicubicInterpolation$<$ scalartype_ $>$:



Collaboration diagram for unicubicInterpolation$<$ scalartype_ $>$:



Public Member Functions

- unicubicInterpolation (covafill$<$ scalartype $>$ $*$cf, vectortype minCoord, vectortype maxCoord)

  Constructs a unicubicInterpolation class from a covafill class cf, an boundaries of the interpolation square defined by the minimum coordinates, minCoord, and maximum coordinates, maxCoord, in each dimension, e.g., minCoord = 0 and maxCoord = 1.

- virtual vectortype operator() (vectortype newcoord)

  Calculates the interpolation prediction at newcoord.

Additional Inherited Members

6.12.1 Detailed Description

template<typename scalartype_>class unicubicInterpolation< scalartype_ >

Class for cubic interpolation of local polynomial regression on a square.

The documentation for this class was generated from the following file:

- unicubicInterpolation.hpp

Index