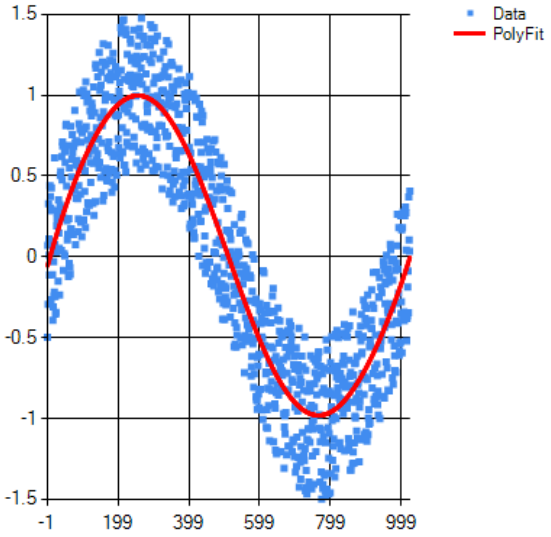
Polynomial Fitting in C++ using Boost

October 7, 2013 by vilipetek

A while back I needed to implement an algorithm utilizing an equivalent function of polyfit/polyval in LabView. The original algorithm was created in LabView for research purposes and since we were in the process of creating a commercial version of the device, the algorithm implementation in C++ needed to mirror the LabView implementation. I found that there is no simple and neat implementation of polyfit() polyval() in C++ so through a bit of digging, research, and implementation I came up with my own functions.

Polynomial fitting function returns coefficients of a polynomial representing a given data. MathWorks has a decent explanation of the function. Below is a graph showing 1000 randomly varying points and the polynomial curve in

red generated from the given points.



I used boost::ublas since it provides simple matrix class and fast LU decomposition. If you are unfamiliar with boost you may find more information on Boost C++ site. If you wish to install Boost libraries on windows, I recommend downloading the recompiled install package from BoostPro site.

PolyFitExample was writen in C++/CLI since windows forms provides a simple graph component. You may access the source by clicking the link.

```
template<typename T>
std::vector<T> polyfit( const std::vector<T>& oX,
    const std::vector<T>& oY, int nDegree )
{
    using namespace boost::numeric::ublas;

    if ( oX.size() != oY.size() )
        throw std::invalid_argument( "X and Y vector sizes do not match" );

    // more intuative this way
    nDegree++;

    size_t nCount =  oX.size();
    matrix<T> oXMatrix( nCount, nDegree );
    matrix<T> oYMatrix( nCount, 1 );

    // copy y matrix
    for ( size_t i = 0; i < nCount; i++ )
    {
        oYMatrix(i, 0) = oY[i];
    }

    // create the X matrix
    for ( size_t nRow = 0; nRow < nCount; nRow++ )
    {
        T nVal = 1.0f;
        for ( int nCol = 0; nCol < nDegree; nCol++ )
```

```
        {
            oXMatrix(nRow, nCol) = nVal;
            nVal *= oX[nRow];
        }
    }

    // transpose X matrix
    matrix<T> oXtMatrix( trans(oXMatrix) );
    // multiply transposed X matrix with X matrix
    matrix<T> oXtXMatrix( prec_prod(oXtMatrix, oXMatrix) );
    // multiply transposed X matrix with Y matrix
    matrix<T> oXtYMatrix( prec_prod(oXtMatrix, oYMatrix) );

    // lu decomposition
    permutation_matrix<int> pert(oXtXMatrix.size1());
    const std::size_t singular = lu_factorize(oXtXMatrix, pert);
    // must be singular
    BOOST_ASSERT( singular == 0 );

    // backsubstitution
    lu_substitute(oXtXMatrix, pert, oXtYMatrix);

    // copy the result to coeff
    return std::vector<T>( oXtYMatrix.data().begin(), oXtYMatrix.data().end()
}
```

```
template<typename T>
std::vector<T> polyval( const std::vector<T>& oCoeff,
    const std::vector<T>& oX )
{
    size_t nCount =  oX.size();
    size_t nDegree = oCoeff.size();
    std::vector<T>  oY( nCount );

    for ( size_t i = 0; i < nCount; i++ )
    {
        T nY = 0;
        T nXT = 1;
        T nX = oX[i];
        for ( size_t j = 0; j < nDegree; j++ )
        {
            // multiply current x by a coefficient
            nY += oCoeff[j] * nXT;
            // power up the X
            nXT *= nX;
        }
        oY[i] = nY;
    }

    return oY;
}
```

- C++, Math
- PictureBox Control Zoom
- Polynomial fitting in C++ (not using Boost)

One thought on “Polynomial Fitting in C++ using Boost”

Pingback: Polynomial fitting in C++ (not using Boost) | Vili Petek
Comments are closed.

Search …

Recent Posts

Multilayer Perceptron
Multiclass Perceptron in C#
Binary Perceptron in C#
Polynomial Fitting in C#
WPF Window Size, Position, and Style Persistence

Categories

AI
C Sharp
C++
Data Manipulation
DDC/CI
GUI
Imaging
Java
Math
RVO
Uncategorized
VESA
WPF

Copyright © 2017 · GeneratePress · WordPress