|  |
| --- |
|  |
| SeesharpTools.JXI.  Mathematics.Curvefitting |
| Library Design |
|  |
|  |
|  |

|  |
| --- |
|  |

**文档版本**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订说明 | 修订人 |
| 2017/08/04 | 0.1 | 初版，SeesharpTool.JXI.Mathematics.Curvefitting | 谢晓姣 |
| 2017/09/01 | 0.2 | 添加B样条拟合，加入权重 | 谢晓姣 |

**目录**

[1. CurveFitting 1](#_Toc2595)

[1.1. 概要 1](#_Toc14626)

[1.1.1. 功能概述 1](#_Toc16014)

[1.1.2. 线性拟合 1](#_Toc23297)

[1.1.3. 指数拟合 2](#_Toc5044)

[1.1.4. 多项式拟合 2](#_Toc17132)

[1.1.5. B样条拟合 3](#_Toc24317)

[1.2. C#静态类库 1](#_Toc11403)

## 

# CurveFitting

## 概要

### 功能概述

曲线拟合包括线性拟合、指数拟合、多项式拟合、样条拟合：

### 线性拟合

假设输入的数据，（*i=*1,2,...,*n*）大体满足线性关系，可近似采用线性方程：

 公式 1.1

拟合，的函数关系。利用最小二乘法，即选择合适的参数*a*，*b*，使得误差平方和：

 公式 1.2

达到最小值。根据通常求最小值方法，要使式（1.2）极小，*a*，*b*必须满足下列方程组：

 公式 1.3

或者：

 公式 1.4

w为权重矩阵，用户没有提供时为全1的对角矩阵，求解线性方程组得到a、b。

### 指数拟合

假设输入的数据，（*i=*1,2,...,*n*）大体满足线性关系，可近似采用指数方程： 公式 1.6

拟合，的函数关系，*a*>0。

对式（1.5）等式两边取对数：

 公式 1.7

讲指数关系转化为线性关系，用线性拟合的方式求得参数*a*,*b*的值。权重矩阵为：

或者用户自己给的权重矩阵。

### 多项式拟合

假设输入的数据，（*i=*1,2,...,*n*）用多项式：

 公式 1.8

拟合*m*个系数，使得误差平方和：

 公式 1.9

达到最小值：

 公式 1.10

或：

 公式 1.11

令，，则：

 公式 1.12

可将方程化为：

 公式 1.13

或：

 公式 1.14

解出*ai*就可以得到*m*阶多项式拟合的参数。

### B样条拟合

在平面上有共n个点，现作一条函数使其经过这n个点。设集合是关于点的角标的[集合](https://baike.baidu.com/item/%E9%9B%86%E5%90%88/2908117" \t "https://baike.baidu.com/item/%E6%8B%89%E6%A0%BC%E6%9C%97%E6%97%A5%E6%8F%92%E5%80%BC%E6%B3%95/_blank)，，作n个多项式。对于任意，都有使得:

 公式 1.15

是n-1次[多项式](https://baike.baidu.com/item/%E5%A4%9A%E9%A1%B9%E5%BC%8F" \t "https://baike.baidu.com/item/%E6%8B%89%E6%A0%BC%E6%9C%97%E6%97%A5%E6%8F%92%E5%80%BC%E6%B3%95/_blank)，且满足并且。最后可得

 公式 1.16

形如上式的插值多项式称为拉格朗日（*Lagrange*）插值多项式。

例如：当n=4时，上面的公式可简化为：

 公式 1.17

这是一个过4个点的唯一的三次多项式。

## C#静态类库

|  |  |
| --- | --- |
| 方法定义 | 说明 |
| public static void LinearFit(...) | 线性拟合，提供多个重载1。 |
| public static void ExponentialFit(...) | 指数拟合，提供多个重载2 |
| public static void PolynomialFit(...) | 多项式拟合，提供多个重载3。 |
| public static void BSplineFit(...) | 样条拟合，提供多个重载4。 |

注1：LinearFit多态方法一览：

void LinearFit(double[] x, double[] y, double interestX, out double fittedY,double[] weight=null)

void LinearFit(double[] x, double[] y, double interestX, out double fittedY, out double slope, out double intercept,double[] weight=null)

void LinearFit(double[] x, double[] y, double[] interestX, ref double[] fittedY,double[] weight=null)

void LinearFit(double[] x, double[] y, double[] interestX, ref double[] fittedY, out double slope, out double intercept,double[] weight=null)

注2：ExponentialFit多态方法一览：

void ExponentialFit(double[] x, double[] y, double interestX, out double fittedY,double[] weight=null)

void ExponentialFit(double[] x, double[] y, double interestX, out double fittedY, out double amplitude, out double damping, out double offset,double[] weight=null)

void ExponentialFit(double[] x, double[] y, double[] interestX, ref double[] fittedY,double[] weight=null)

ExponentialFit(double[] x, double[] y, double[] interestX, ref double[] fittedY, out double amplitude, out double damping, out double offset,double[] weight=null)

注3：PolynomialFit多态方法一览：

PolynomialFit(double[] x, double[] y, int order, double interestX, out double fittedY,double[] weight=null)

PolynomialFit(double[] x, double[] y, int order, double interestX, out double fittedY, ref double[] coefficients,double[] weight=null)

PolynomialFit(double[] x, double[] y, int order, double[] interestX, ref double[] fittedY,double[] weight=null)

PolynomialFit(double[] x, double[] y, int order, double[] interestX, ref double[] fittedY, ref double[] coefficients,double[] weight=null)

注4：BSplineFit多态方法一览：

BSplineFit(double[] x, double[] y, int degree, double interestX, out double fittedY,double[] weight=null)

BSplineFit(double[] x, double[] y, int degree, double[] interestX, ref double[] fittedY,double[] weight=null)

degree控制曲线的误差以及平滑度