# 第7章 多元回归法及其在智能传感器系统中的应用

本章内容：

* 多元回归法
* 多元回归方程
* 多元回归法在传感器温度补偿中的应用
* 多元回归法及其性能指标计算的Python实现

上一章已经对智能传感器的相关技术进行了初步的探讨。本章将重点介绍多元回归法及其在智能传感器系统中的应用，主要包括多元回归法、多元回归方程及其在传感器温度补偿中的应用。此外，本章还简要介绍了多元回归法及其性能指标计算的Python实现。通过对这些技术的学习，我们可以更好地理解和掌握多元回归方程的原理和应用，并在实际应用中更好地运用这些技术，从而提高传感器的测量精度和稳定性。

# §7.1 多元回归分析法与多元回归方程

多元回归分析法建立逆模型的核心思想是：欲消除个干扰量对主传感器测量目标参量的影响，就要设置个监测干扰量的辅助传感器，以建立更完备的逆模型。这个逆模型是一个元常系数高阶回归方程，其中模型的阶数由允许的误差范围来决定。

当需要消除一个干扰量（即）时，需要建立一个由两个传感器组成的智能传感器系统，其逆模型为二元回归方程，以进行两个传感器的数据融合。类似地，当需要消除两个干扰量（即）时，需要建立一个由三个传感器组成的智能传感器系统，其逆模型为三元回归方程，以进行三个传感器的数据融合。

## 7.1.1 二元回归法

两个传感器可以测量并获取两个参量的信息。有多种算法可以融合这两个信息，其中一种是曲面拟合算法，也就是二元回归分析法。接下来，将以压力传感器为例来说明这种数据融合算法。

### 7.1.1.1 二元回归分析法基本原理

假如压力传感器输出是电压，并且对温度有交叉灵敏度。如果只使用传统的一维标定实验来获取压力传感器的输入（压力）和输出（电压)特性曲线，然后用它来计算被测压力值，将会产生较大的误差。因为被测量不是输出值的一元函数。为了更准确地测量压力，需要用另一个温度传感器的输出电压来表示温度，然后用和的二元函数来描述压力，即：

由二维坐标（,）决定的位于一个平面上，可以用二次曲面拟合方程，也就是二元回归方程来描述，即：

同理有：

忽略高阶无穷小，可得：

|  |  |  |
| --- | --- | --- |
|  |  | （7.1） |

式中为常系数；为回归方程式的项数，根据允许的误差来选取。

忽略高阶无穷小，可得：

|  |  |  |
| --- | --- | --- |
|  |  | （7.2） |

式中为常系数；

如果式（7.1）和式（7.2）中的常系数均已知，那么这两个式子就是求取被测量P或T更完备的逆模型，从而可以消除交叉敏感。为了得到这些常系数，需要首先进行二维标定实验，并利用最小二乘法原理来确定使均方误差最小的常系数。

### 7.1.1.2 实验标定

在压力传感器的量程范围内确定个压力标定点。同样，在工作温度范围内确定个温度标定点。于是由压力与温度标准值发生器产生的在各个标定点的标准输入值为：

：，，，，

：，，，，

对于每个标定点，都有一个标准输入值及其对应的输出值和。于是，就可以在个不同温度状态下对压力传感器进行静态标定。总共可以得到个标定点。每个标定点都有4个标定值，也称为一个样本对，包括两个传感器的输入量（即主测量压力和辅参量温度）以及它们相应的输出量（分别为和）。根据这些标定数据可以获得不同温度状态的输入输出特性，即特性簇，如图7-1（a）所示。同时，还可以获得对应于不同压力状态的温度传感器的输入输出特性，即特性簇，如图7-1（b）所示。

|  |  |
| --- | --- |
| （a）特性簇 | （b）特性簇 |

图7-1 压力传感器输入输出特性

### 7.1.1.3 二元回归方程待定常数的确定

为确定式（7.1）和式（7.2）所表征的二元回归方程式的常系数，通常采用最小二乘法，求取满足均方误差最小的常系数值。系数和的求法相同。下面以为例说明求取步骤。

（1）第个标定点的压力数据计算值（以下简称计算值）为，，根据式（7.1）可得：

（2）第个标定点的压力标定值与计算值，之间的误差为，对应的方差为，则有：

|  |  |  |
| --- | --- | --- |
|  |  | （7.3） |

式中，，，，，。

（3）所有标定点压力标定值与计算值之差的平方和为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.4） |

式中，为标定点总数，当压力标定点数，温度标定点数时，。

由式（7.4）可见，是常系数，，，，，的多元函数。

（4）要找到回归方程待定系数的最小二乘最优解，可以通过求解多元函数的极值来实现。对函数求关于的偏导数并令其等于0可得，，,。展开该式可得以下六个方程：

|  |  |  |
| --- | --- | --- |
|  |  | （7.5） |

由式（7.5）可得：

|  |  |  |
| --- | --- | --- |
|  |  | （7.6） |

式中，，2 ，，；，1，2，，。

由线性代数知识，可将式（7.6）写成矩阵形式：

|  |  |  |
| --- | --- | --- |
|  |  | （7.7） |

式中：—待求常系数矩阵，;

—维系数矩阵，;

—由标定压力值构成的维列矩阵。

于是，回归方程待定常系数，，，，，的最小二乘最优解的求解式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.8） |

这就是求解最小二乘问题的方程组法。

## 7.1.2 三元回归分析法

### 7.1.2.1 单一功能（只测一个目标参量）的三传感器数据融合

通过监测两个非目标参数，也就是两个干扰量，可以有效地消除这些干扰量对测量结果的影响，从而提高单功能传感器对目标参数的测量精度。为了实现这一目标，可以选择任何能够测量这两个干扰参数的传感器，只需将它们放置在同一测量场中，使其受到与测量目标参数的传感器相同强度的干扰。

仍以压阻式压力传感器为例，其输出不但受到工作环境温度的影响，而且受到电源供电电流的影响。为了消除这两个参量的影响，需要对和分别进行监测，建立如图7-2所示的三传感器数据融合的智能传感器系统。

  
图7-2 三传感器数据融合智能传感器系统框图

对图7-2所示的三传感器数据融合智能传感器系统进行三维标定实验，确立三元回归方程：

式中：—规定的被测参量压力；

—压力传感器输入压力为P时的输出电压值；

—监测工作环境温度用温度传感器的输出电压值；

—监测供电电流用传感器的输出电压值；

—可忽略的高阶（大于二阶）无穷小量。

忽略高阶无穷小量，可得：

|  |  |  |
| --- | --- | --- |
|  |  | （7.9） |

利用三维标定实验得到的标定数据，就可以根据均方误差最小原则确定式（7.9）中的常系数。这个三元回归方程就可以用来建立图7-2所示的三传感器数据融合智能传感器系统，以抑制两个干扰量的交叉敏感，提高原传感器系统对温度漂移和电源波动的稳定性。

### 7.1.2.2 三功能（测量三个参量）的传感器数据融合

以实现测量压力（差）、静压和温度三个参量的三功能传感器ST-3000智能变送器为例。检测三个参量的三个传感器相互之间存在交叉灵敏度，每个传感器进行刻度转换的逆模型都应是三元回归方程，即：

上述三个方程都如式（7.9）所示，共有个未知待定常数，需要由三维标定实验数据来确定。

为简化处理，可以先进行降维处理。假设对静压的测量精度要求不高，可将其作为一元函数来对待，即：

假如静压主要影响压力（差）的零点输出，则产生的干扰量可以用表示。与静压输出的关系可以由阶多项式方程描述：

|  |  |  |
| --- | --- | --- |
|  |  | （7.10） |

式中：，，，，—可由标定实验确定的待定常系数。

选定个不同静压值（，2，3，，），测定相应压力（差）的零点（，2，3，，），即：

利用上述标定数据，就可以根据均方误差最小原则确定式（7.10）中的常系数。

在测量过程中，首先需要对与三个输入量、、相应的三个输出量、、进行采样。然后根据采样值代入式（7.10）计算，最后与采样值做减法，得：

|  |  |  |
| --- | --- | --- |
|  |  | （7.11） |

式中：—消除了零点干扰量后的压力（差）输出值

据此被测压力（差）值降为二元函数，即：

|  |  |  |
| --- | --- | --- |
|  |  | （7.12） |

静压对温度输出基本上没有影响，故被测温度可以由二元函数表示，即：

|  |  |  |
| --- | --- | --- |
|  |  | （7.13） |

降元后，与的数据可以通过二传感器数据融合技术进行处理，即可以采用式（7.1）和式（7.2）来进行数据融合处理。

# §7.2 基于二元回归分析法实现压力传感器的温度补偿

本节拟设计一个温度自补偿的智能传感器系统，以实现压阻式压力传感器的温度补偿。该智能传感器的原理框图如图7-3所示。

  
图7-3 温度自补偿智能传感器系统原理框图

由图7-3可见，数据融合模块接收来自两个传感器的输入：一个是需要进行温度补偿的主传感器，即压力传感器；另一个是用于监测温度干扰的辅助传感器。对一个干扰量的回归模型分析，就是将式（7.1）作为逆模型进行两传感器数据融合。具体设计步骤如下。

## 7.2.1二维标定实验

主传感器是JCY-101型压阻式压力传感器，其输入量为压力，输出量为电压，由于主传感器的输出会受到环境温度的影响，因此需要使用另一个辅助传感器来测量温度，其输出量为电压。

为了进行标定，采用压力标准值发生器为主传感器提供标准输入值，并采用恒温箱为辅助传感器提供标准输入值。在主传感器的量程内，选取个不同的压力值作为标定点。这些压力值分别为：

（）：0，1.0，2.0，3.0，4.0，5.0，，2，，；

同时，在主传感器的工作温度范围内（），选取个不同的温度值作为标定点。这些温度值分别为：

（）：21.5，28.0，34.0，44.0，50.0，70.0，，2，，。

主传感器和辅助传感器被放置在同一个恒温箱中，并逐一调整恒温箱的温度，使其达到每一个温度标定点。为保证实验数据的准确性，首先在每个温度环境下等待15分钟，使传感器达到稳定状态。然后，给主传感器依次施加每一个压力标定点。最后依次记录主传感器和辅助传感器的输出电压和，以及施加的压力。本实验设置了组温度值和组压力值，共采集个标定数据点，如表7-1所示。

表7-1 JCY-101型压力传感器的二维实验标定数据

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| 21.5 |  | -13.84 | 10.69 | 28.88 | 47.05 | 65.19 | 83.36 |
|  | 27.64 | 26.95 | 26.43 | 25.92 | 25.45 | 24.94 |
| 28.0 |  | -13.49 | 9.32 | 26.34 | 43.12 | 59.99 | 76.82 |
|  | 34.41 | 33.93 | 33.47 | 32.93 | 32.47 | 31.91 |
| 34.0 |  | -10.80 | 7.54 | 24.84 | 42.05 | 59.25 | 76.38 |
|  | 37.76 | 36.92 | 36.44 | 35.97 | 35.39 | 35.09 |
| 44.0 |  | -9.72 | 6.56 | 23.87 | 41.21 | 58.58 | 75.87 |
|  | 54.88 | 53.97 | 52.87 | 52.41 | 51.93 | 51.55 |
| 50.0 |  | -8.62 | 4.86 | 21.84 | 38.70 | 56.32 | 73.75 |
|  | 65.77 | 64.79 | 63.84 | 62.91 | 61.99 | 61.06 |
| 70.0 |  | -7.72 | 3.72 | 21.25 | 38.60 | 55.56 | 73.28 |
|  | 86.12 | 84.94 | 83.78 | 82.65 | 81.55 | 80.45 |

## 7.2.2 数据处理

### 7.2.2.1 计算常系数并建立二元回归方程

利用表7-1的实验标定数据和（）可以构成式（7.7）中的矩阵和矩阵。利用这两个矩阵求解方程组（7.8），即可得各常系数的值为：

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

将计算而得的常系数值，代入式（7.1）中，就可以确定用于消除交叉敏感的逆模型。利用这个逆模型编写的程序可以实现基于回归分析法的压力传感器的温度补偿。

### 7.2.2.2 融合计算结果

将测量值代入由上述常系数值确立的逆模型，可以计算得到目标参数P。这个参数，即融合结果，如表7-2所示。

表7-2 不同温度条件下压力标定值与融合处理输出的压力计算值（MPa）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 温度  目标参量  压力 | 21.5 | 28.0 | 34.0 | 44.0 | 50.0 | 70.0 | 平均值 |
| 1 | 标定值 | 0.00 | | | | | | |
| 计算值 | -0.19 | -0.14 | 0.02 | 0.11 | 0.16 | 0.12 | 0.03 |
| 偏差 | -0.19 | -0.14 | 0.02 | 0.11 | 0.16 | 0.12 | 0.03 |
| 2 | 标定值 | 1.00 | | | | | | |
| 计算值 | 1.12 | 1.10 | 1.02 | 1.03 | 0.95 | 0.82 | 1.02 |
| 偏差 | 0.12 | 0.10 | 0.02 | 0.03 | -0.05 | -0.18 | 0.02 |
| 3 | 标定值 | 2.00 | | | | | | |
| 计算值 | 2.11 | 2.03 | 1.98 | 2.02 | 1.94 | 1.90 | 2.00 |
| 偏差 | 0.11 | 0.03 | -0.02 | 0.02 | -0.06 | -0.1 | 0.00 |
| 4 | 标定值 | 3.00 | | | | | | |
| 计算值 | 3.10 | 2.96 | 2.94 | 3.03 | 2.94 | 2.97 | 3.00 |
| 偏差 | 0.10 | -0.04 | -0.06 | 0.03 | -0.06 | -0. 03 | 0.00 |
| 5 | 标定值 | 4.00 | | | | | | |
| 计算值 | 4.10 | 3.90 | 3.90 | 4.04 | 3.98 | 4.02 | 4.00 |
| 偏差 | 0.10 | -0.10 | -0.10 | 0.04 | -0.02 | 0.02 | 0.00 |
| 6 | 标定值 | 5.00 | | | | | | |
| 计算值 | 5.10 | 4.85 | 4.87 | 5.05 | 5.02 | 5.12 | 5.00 |
| 偏差 | 0.10 | -0.15 | -0.13 | 0.05 | 0.02 | 0.12 | 0.00 |

## 7.2.3 数据融合处理后JCY-101型压力传感器性能的综合评价

在评估数据融合效果时，主要关注的指标包括温度影响系数、线性度和误差系数。通过比较融合处理前后的温度影响系数，可以评估温度稳定性的改善程度。同样，通过比较融合处理前后的线性度指标，可以评估静态性能的改善程度。误差系数则用于评估压力拟合值和标定值之间的差异。

### 7.2.3.1 温度影响系数

温度影响系数是评估传感器性能随温度变化情况的重要指标，其主要包括零位温度系数和灵敏度温度系数。

1. 零位温度系数

零位温度系数表示零位值 随温度漂移的速度，在数值上等于温度改变，零位值的最大改变量与量程之比。

在融合处理之前，零位温度系数的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.14） |

在融合处理之后，零位温度系数的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.15） |

式中：—工作温度变化范围，；

—压力传感器满量程输出值；

—压力传感器满量程输入值；

—工作温度变化范围内，压力传感器零点漂移最大值；

—逆模型融合计算在范围内的零点压力最大偏差。

由表 7-1 所列实验标定数据及表 7-2 融合处理后数据分别可知：

于是可得

融合处理前：

融合处理后：

2. 灵敏度温度系数

灵敏度温度系数表示灵敏度随温度漂移的速度，在数值上等于温度改变时，灵敏度的相对改变量。

在融合处理之前，灵敏度温度系数的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.16） |

在融合处理之后，灵敏度温度系数的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.17） |

式中：、—同一输入压力作用下，工作温度为、时，压力传感器的输入值；

、U—同一输入压力作用下，工作温度为、时，压力传感器的输出值；

—工作温度变化范围，。

由表7-1所列标定数据可知；压力传感器的输出电压信号随工作温度升高而减小，在满量程压力值（)输入时，输出电压值随温度变化有最大改变量为；且，则计算可得融合前灵敏度温度系数为：

由表7-2融合处理后数据可知，在温度范围内，融合计算值不存在随温度变化单调上升或下降的规律，而是围绕期望值（压力标定值）随机偏离，在满量程5.0 MPa时，两个温度点融合计算压力值的最大偏差== 5.104-5.123=-0.019MPa，则计算可得融合后灵敏度温度系数为：

可见，由以上二元二阶六项多项式表征的逆模型进行处理融合，处理前后的数据表明：零位温度系数由提高到；灵敏度温度系数由提高到。故传感器的温度稳定性得到了较大的改善。如果需要进一步提高融合效果，可以考虑增加逆模型多项式的项数（从而减小误差项的数值），或者尝试使用其他的融合算法。

### 7.2.3.2 线性度

1. 融合处理前

使用标定的静态特性计算最小二乘法线性度，其拟合直线方程为:

由上述直线方程计算得到的压力拟合值、标定值及其拟合偏差列入表7-4中。

表7-4融合处理前拟合值、标定值及其拟合偏差

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 标定值 | 0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| 拟合值 | -0.151 | 1.131 | 2.081 | 3.301 | 3.979 | 4.929 |
| 拟合偏差 | -0.151 | 0.131 | 0.081 | 0.031 | -0.021 | -0.071 |

单位：;

量程：。

根据表7-4中的数据，可以得到最大的拟合偏差。因此，最小二乘法的非线性误差为：

2. 融合后

为了简化，使用理论线性度进行评估，其方程为：

式中：，。

由上述理论直线方程计算得到的压力拟合值、标定值及其拟合偏差列入表7-5中。

表7-5融合处理后拟合值、标定值及其拟合偏差

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 标定值 | 0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| 拟合值 | 0.038 | 1.056 | 2.030 | 3.01 | 4.003 | 4.998 |
| 拟合偏差 | 0.038 | 0.056 | 0.030 | 0.01 | 0.003 | -0.002 |

单位：;

量程：。

根据表7-5的数据，可以得到最大的拟合偏差。因此，最小二乘法的非线性误差为：

融合处理使得非线性引用误差从3％降低到1％，这显著提高了JCY-101型压力传感器的线性度。

### 7.2.3.3 误差系数

误差系数是一组评估拟合值和标定值差异的重要指标。常用的误差系数包括均方误差、均方根误差、平均绝对误差、误差平方和、平均绝对百分比误差、决定系数和校正决定系数。

1. 均方误差

均方误差（Mean Squared Error, MSE）是拟合值与标定值差值平方的平均值。它衡量了拟合值与标定值之间的偏离程度。MSE的值越小，说明预测模型对实验数据的拟合更精确。其计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.18） |

式中：—标定值；

—拟合值；

—样本数量。

表7-2中的拟合偏差就是拟合值与标定值的差值，即，则式（7.18）可改写为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.19） |

在表7-2中，样本数量，则根据表中数据可求得。

2. 均方根误差

均方根误差（Root Mean Squared Error, RMSE）是均方误差的平方根。RMSE的值越小，说明预测模型对实验数据的拟合更精确。其计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.20） |

式中：—标定值；

—拟合值；

—样本数量。

表7-2中的拟合偏差就是拟合值与标定值的差值，即，则式（7.20）可改写为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.21） |

在表7-2中，样本数量，根据表中数据可求得，则。

3. 平均绝对误差

平均绝对误差（Mean Absolute Error, MAE）是拟合值与标定值差值绝对值的平均数。MAE的值越小，说明预测模型对实验数据的拟合更精确。其计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.22） |

式中：—标定值；

—拟合值；

—样本数量。

表7-2中的拟合偏差就是拟合值与标定值的差值，即，则式（7.22）可改写为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.23） |

在表7-2中，样本数量，则根据表中数据可求得。

4. 误差平方和

误差平方和（Sum of Squared Errors, SSE）是拟合值与标定值差值的平方和。SSE的值越小，说明预测模型对实验数据的拟合更精确。其计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.24） |

式中：—标定值；

—拟合值；

—样本数量。

表7-2中的拟合偏差就是拟合值与标定值的差值，即，则式（7.24）可改写为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.25） |

在表7-2中，样本数量，则根据表中数据可求得。

5. 平均绝对百分比误差

平均绝对百分比误差（Mean Absolute Percentage Error, MAPE）是拟合值与标定值差值与标定值的比值的绝对值的平均数，以百分比形式表示。MAPE的值越小，说明预测模型对实验数据的拟合更精确。其计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.26） |

式中：—标定值；

—拟合值；

—样本数量。

表7-2中的拟合偏差就是拟合值与标定值的差值，即，则式（7.26）可改写为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.27） |

在表7-2中，样本数量。然而，由于表中部分样本的标定值为零，无法计算平均绝对百分比误差（MAPE）。因此，对于这种特定样本，无法使用MAPE作为误差系数。

6. 决定系数

决定系数（Coefficient of Determination, R²）是一种用于衡量模型对样本数据拟合程度的统计量。它的值介于0和1之间，越接近1，说明模型对样本数据的拟合程度越好。其计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.28） |

式中：—标定值；

—拟合值；

—标定值的平均值；

—样本数量。

在表7-2中，样本数量，则根据表中数据可求得。

7. 校正决定系数

校正决定系数（Adjusted Coefficient of Determination, ）是对决定系数的修正，以考虑模型中自变量的数量。当模型中的自变量数量增加时，即使这些变量对因变量的影响不大，决定系数R²也可能会增加。因此，使用校正决定系数来调整自变量数量的影响，从而更准确地评估模型的性能。校正决定系数的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （7.29） |

式中：—标定值；

—拟合值；

—样本数量；

—自变量的维数。

在表7-2中，样本数量，自变量数量，则根据表中数据可求得。

计算结果表明，误差系数（包括均方误差、均方根误差、平均绝对误差和误差平方和）的值相对较小，决定系数和校正决定系数的值接近1。这表明模型对样本数据的拟合程度非常好，能够满足JCY-101型压力传感器温度补偿的需要。

# §7.3 多元回归法及其性能指标计算的Python实现

本节将首先介绍多元回归法的python实现，主要是通过scikit-learn库中的PolynomialFeatures和LinearRegression类实现多元回归。接着，我们将介绍如何利用Python实现性能指标的计算，主要包括温度影响系数和误差系数。

## 7.3.1 多元回归法的python实现

多元回归法的Python实现，依赖于scikit-learn库中的预处理PolynomialFeatures类和线性回归LinearRegression类。在开始使用这两个模块之前，必须先安装scikit-learn库并在程序中引用它们。scikit-learn库的安装在前面的章节中已经介绍过，本节着重介绍这两个类的引用。

PolynomialFeatures类的引用命令：

from sklearn.preprocessing import PolynomialFeatures

LinearRegression类的引用命令：

from sklearn.linear\_model import LinearRegression

PolynomialFeatures类用于生成由所有多项式组合构成的新特征矩阵，从而可以使用线性回归方法来实现非线性的多元回归，其类定义为:

class sklearn.preprocessing.PolynomialFeatures(degree=2, \*, interaction\_only=False, include\_bias=True, order='C')

该类的参数包括：

degree：多项式特征的程度，也即多元回归方程中自变量的最高阶次，默认值=2；

interaction\_only：如果为true，只能交叉相乘，默认False；

include\_bias：默认True, 此时特征矩阵包含元素1，多元回归方程中包含常数项，否则不包含常数项。

order：在密集情况下输出数组的顺序。“ F”阶的计算速度更快，但可能会减慢后续的估计量。

该类的属性包括：

powers\_：指数输出

n\_input\_features\_：输入特征的总数；

n\_output\_features：多项式输出特征的总数。

该类主要包括以下方法：

fit(X, y=None)：计算输入数据的所有多项式特征的组合。

transform(X)：根据fit方法的计算结果，将输入数据转换为多项式特征。

fit\_transform(X, y=None, \*\*fit\_params)：结合fit和transform两个操作，首先对输入数据进行fit操作，然后再对其进行transform操作。

使用该类生成特征矩阵的实例代码如下：

#创建一个PolynomialFeatures实例，其中degree根据允许的误差来选取

quadratic\_featurizer = PolynomialFeatures(degree=2)

#拟合拟合数据，然后对其进行转换，x是初始数据，xx是转换后的数据

xx = quadratic\_featurizer.fit\_transform(x)

LinearRegression类用于执行最小二乘线性回归并生成多元回归方程的系数，其类定义为：

class sklearn.linear\_model.LinearRegression(\*, fit\_intercept=True, normalize=False, copy\_X=True, n\_jobs=None)

该类的参数包括：

fit\_intercept：是否计算此模型的截距，默认True，计算截距；

normalize：是否归一化处理，默认False，不进行归一化处理；

copy\_X：是否复制X，默认True，复制X；

n\_job：用于计算的核心数。

该类的属性包括：

coef\_：线性回归问题的估计系数；

rank\_：矩阵X的秩；

singular\_：X的奇异值；

intercept\_：线性模型中的截距项，如果设置fit\_intercept = False，则截距为0.0。

该类主要包括以下方法：

fit(X, y[, sample\_weight])：拟合线性模型。

predict(X)：使用线性模型进行预测。

score(X, y[, sample\_weight])：返回决定系数的预测。

使用LinearRegression实现线性回归的实例代码如下：

#创建一个LinearRegression实例：

lr= LinearRegression()

#拟合线性模型，其中xxPolynomialFeatures类生成的新特征矩阵

lr.fit(xx, y)

#提取多元回归方程的系数和截距

coef =lr. coef\_ #系数

intercept =lr. intercept\_ #截距

#预测，将待预测的变量以数组形式输入，在此以x1示例

y\_pred=lr.predict(quadratic\_featurizer.fit\_transform(np.array(x1)))

## 7.3.2 性能指标计算的python实现

在评估多元回归模型的性能时，主要关注的指标包括温度系数、线性度和误差系数。前面的章节已经详细地介绍了如何用Python实现线性度的计算。本章将重点介绍如何用Python实现温度系数和误差系数的计算。

### 7.3.2.1 温度系数计算的Python实现

本节主要内容是使用Python来计算温度系数。以压力传感器为例，其中，目标参量是压力（在程序中用P表示），对应的输出是输出电压（在程序中用U\_P表示）。干扰参量是（在程序中用T表示），通过温度传感器进行检测，其输出是温度传感器的输出电压（在程序中用U\_T表示）。这种方法具有通用性，只需在程序中将P和U\_P替换为目标参数传感器的输入和输出，将T和U\_T替换为用于检测干扰参数的传感器的输入和输出即可。

在使用Python计算温度系数时，需要经过一系列的计算步骤。在这些步骤中，经常需要求取最大值和最小值。求取最大值和最小值主要有两种方法：一种是使用Python的内置函数max()和min()，另一种是使用pandas库的DataFrame或Series对象的.max()和.min()方法。选择哪种方法主要取决于数据的类型。如果数据是DataFrame或Series，那么应该使用.max()和.min()方法。如果数据是列表或元组，那么应该使用max()和min()函数。例如，如果将压力传感器的输出电压U\_P保存为一个数组，可以使用max(U\_P)来求取压力传感器输出电压的最大值。如果将干扰参量温度保存为一个DataFrame类型T，那么需要使用.max()和.min()方法，即T.max()和T.min()来求取最高温度和最低温度。此外，还会使用abs()函数来计算绝对值，where()函数来找到满足特定条件的索引。

接下来，详细介绍每个计算步骤和对应的Python代码实现：

1. 计算工作温度变化范围。这可以通过计算最高温度和最低温度之差来实现。

delta\_T = T.max() - T.min()

2. 计算压力传感器满量程输出值和输入值。这可以通过找到压力传感器的最大输出值和最大输入值来实现。

U\_FS = max(U\_P)

P\_FS = max(P)

3. 计算工作温度变化范围内的压力传感器零点漂移最大值和零点压力最大偏差。这可以通过在工作温度变化范围内，找到压力传感器零点漂移的最大值和零点压力的最大偏差来实现。

#求取标定压力的列索引

zero\_pressure\_column = df.columns[df.columns == 0]

#求取标定压力时，压力传感器的输出电压

U\_P\_at\_zero\_P = df[df.iloc[:, 1] == 'U\_P'].loc[:, zero\_pressure\_column].values.flatten()

#求取最大偏差

delta\_U\_0m=abs(min(U\_P\_at\_zero\_P) - max(U\_P\_at\_zero\_P))

4. 计算逆模型融合计算在工作温度变化范围内的零点压力最大偏差。可以通过在工作温度变化范围内，找到逆模型融合计算的零点压力的最大偏差来实现。在这里，y\_pred是逆模型融合计算的预测压力值。

#求取标定压力等于0的索引

zero\_pressure\_indices = np.where(P == 0)

#求取压力为0时的预测压力值

P\_pred\_at\_zero\_P = y\_pred[zero\_pressure\_indices]

#预测值与给定值0的差值的最大值

delta\_P\_0m = max(abs(P\_pred\_at\_zero\_P - 0))

5. 计算满量程时，压力传感器在最低温度和最高温度的输出电压。可以通过找到满量程压力对应的索引并获取相应的压力传感器输出值，然后获取所有的唯一温度值，并找到最小和最大温度对应的索引，来实现求取最小和最大温度下的压力传感器输出值。

#找到满量程压力对应的索引

full\_scale\_pressure\_indices = np.where(P == P\_FS)

#获取满量程压力对应的压力传感器输出值

U\_P\_at\_full\_scale\_P = U\_P[full\_scale\_pressure\_indices]

#获取所有的唯一温度值

unique\_temperatures = np.unique(T)

#找到最小温度对应的索引

min\_temperature\_index = np.argmin(unique\_temperatures)

#获取最小温度下的压力传感器输出值

U\_P\_at\_min\_P=U\_P\_at\_full\_scale\_P[min\_temperature\_index]

#找到最大温度对应的索引

max\_temperature\_index = np.argmax(unique\_temperatures)

#获取最大温度下的压力传感器输出值

U\_P\_at\_max\_P=U\_P\_at\_full\_scale\_P[max\_temperature\_index]

6. 计算满量程时，在最低温度和最高温度的压力预测值。可以通过找到满量程压力对应的索引并获取相应的压力预测值，然后获取所有的唯一温度值，并找到最小和最大温度对应的索引，来实现求取最小和最大温度下的压力预测值。

#获取满量程压力对应的预测压力值

P\_pred=y\_pred[full\_scale\_pressure\_indices]

#获取最小温度下的预测压力值

P\_pred\_at\_min\_P=P\_pred[min\_temperature\_index]

#获取最大温度下的预测压力值

P\_pred\_at\_max\_P=P\_pred[max\_temperature\_index]

7. 计算温度系数。利用上面获取的数据，根据式（7.14）、（7.15）、（7.16）和（7.17），编程计算温度系数。

#计算零位温度系数

alpha\_0\_before = delta\_U\_0m / U\_FS \* 1 / delta\_T

alpha\_0\_after = delta\_P\_0m / P\_FS \* 1 / delta\_T

#计算灵敏度温度系数

alpha\_s\_before = (U\_P\_at\_min\_P - U\_P\_at\_max\_P) / (U\_P\_at\_min\_P \* delta\_T)

alpha\_s\_after = (P\_pred\_at\_min\_P - P\_pred\_at\_max\_P) / (P\_pred\_at\_min\_P \* delta\_T)

### 7.3.2.2 误差系数计算的Python实现

利用Python编程计算误差系数的方法主要有两大类：使用内置库函数和自行编程计算。第一类适用于有内置的评估函数计算对应误差系数的情形，例如：计算均方误差、平均绝对误差和决定系数。第二类一般没有内置的评估函数直接计算，需要手动编程来计算相应的误差系数，例如：均方根误差、误差平方和、平均绝对百分比误差和校正决定系数。

对于第一类，主要通过scikit-learn库的metrics子库中的函数实现，例如，使用mean\_squared\_error()来计算均方误差，mean\_absolute\_error()来计算平均绝对误差，以及r2\_score()来计算决定系数。前面已经介绍过scikit-learn库的安装，这里主要介绍评估函数的引用，其引用命令如下：

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

这三个评估函数的常用参数均是标定值y和预测值y\_pred，则误差系数的示例代码如下：

#计算均方误差(MSE)

mse = mean\_squared\_error(y\_true, y\_pred)

#计算平均绝对误差(MAE)

mae = mean\_absolute\_error(y\_true, y\_pred)

#计算决定系数(R^2)

r2 = r2\_score(y\_true, y\_pred)

对于第二类，我们通常需要根据计算公式进行编程，其实例代码如下：

#计算均方根误差(RMSE)

rmse = np.sqrt(mse)

print("均方根误差: ", rmse)

#计算误差平方和(SSE)

sse = np.sum((y\_true - y\_pred) \*\* 2)

print("误差平方和: ", sse)

#计算平均绝对百分比误差(MAPE)

mape = np.mean(np.abs((y\_true - y\_pred) / y\_true)) \* 100

print("平均绝对百分比误差: ", mape)

# 计算校正决定系数

n = x.shape[0] # 样本数量

p = X\_poly.shape[1] # 特征数量

adjusted\_r2 = 1 - (1 - r2) \* (n - 1) / (n - p - 1)

根据上面的介绍，压力传感器温度补偿系统中进行回归模型拟合、预测和性能指数计算的Python程序如下：

import pandas as pd

from sklearn.preprocessing import PolynomialFeatures

from sklearn.linear\_model import LinearRegression

import numpy as np

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

# 加载Excel文件

df = pd.read\_excel('dyhgsj.xlsx')

# 从DataFrame中提取U\_P、U\_T、T和P值

U\_P = df[df.iloc[:, 1] == 'U\_P'].iloc[:, 2:].values.flatten()

U\_T = df[df.iloc[:, 1] == 'U\_T'].iloc[:, 2:].values.flatten()

T = df[df.iloc[:, 1] == 'U\_P'].iloc[:, 0].values.repeat(df.columns[2:].shape[0])

P = np.tile(df.columns[2:].astype(float), df[df.iloc[:, 1] == 'U\_P'].shape[0])

# 将U\_P、U\_T和T组合成回归的输入特征

X = np.column\_stack((U\_P, U\_T))

# 创建degree=2的多项式特征

poly\_features = PolynomialFeatures(degree=2, include\_bias=False)

X\_poly = poly\_features.fit\_transform(X)

# 创建并训练模型

model = LinearRegression()

model.fit(X\_poly, P)

# 打印模型参数

print("截距: ", model.intercept\_)

print("系数: ", model.coef\_)

# 预测结果

y\_pred = model.predict(X\_poly)

# 计算工作温度变化范围

delta\_T = T.max() - T.min()

# 计算压力传感器满量程输出值和输入值

U\_FS = max(U\_P)

P\_FS = max(P)

# 计算工作温度变化ΔT范围内，压力传感器零点漂移最大值和零点压力最大偏差

zero\_pressure\_column = df.columns[df.columns == 0]

U\_P\_at\_zero\_P = df[df.iloc[:, 1] == 'U\_P'].loc[:, zero\_pressure\_column].values.flatten()

delta\_U\_0m=abs(min(U\_P\_at\_zero\_P) - max(U\_P\_at\_zero\_P))

# 计算逆模型融合计算在ΔT范围内的零点压力最大偏差

zero\_pressure\_indices = np.where(P == min(P))

P\_pred\_at\_zero\_P = y\_pred[zero\_pressure\_indices] # 压力为0时的预测压力值

delta\_P\_0m = max(abs(P\_pred\_at\_zero\_P - 0)) # 预测值与给定值0的差值的最大值

# 计算零位温度系数

alpha\_0\_before = delta\_U\_0m / U\_FS \* 1 / delta\_T

alpha\_0\_after = delta\_P\_0m / P\_FS \* 1 / delta\_T

# 找到满量程压力对应的索引

full\_scale\_pressure\_indices = np.where(P == P\_FS)

# 获取满量程压力对应的压力传感器输出值

U\_P\_at\_full\_scale\_P = U\_P[full\_scale\_pressure\_indices]

# 获取所有的唯一温度值

unique\_temperatures = np.unique(T)

# 找到最小温度对应的索引

min\_temperature\_index = np.argmin(unique\_temperatures)

# 获取最小温度下的压力传感器输出值

U\_P\_at\_min\_P=U\_P\_at\_full\_scale\_P[min\_temperature\_index]

# 找到最大温度对应的索引

max\_temperature\_index = np.argmax(unique\_temperatures)

# 获取最大温度下的压力传感器输出值

U\_P\_at\_max\_P=U\_P\_at\_full\_scale\_P[max\_temperature\_index]

# 获取满量程压力对应的预测压力值

P\_pred=y\_pred[full\_scale\_pressure\_indices]

# 获取最小温度下的预测压力值

P\_pred\_at\_min\_P=P\_pred[min\_temperature\_index]

# 获取最大温度下的预测压力值

P\_pred\_at\_max\_P=P\_pred[max\_temperature\_index]

# 计算灵敏度温度系数

alpha\_s\_before = (U\_P\_at\_min\_P - U\_P\_at\_max\_P) / (U\_P\_at\_min\_P \* delta\_T)

alpha\_s\_after = (P\_pred\_at\_min\_P - P\_pred\_at\_max\_P) / (P\_pred\_at\_min\_P \* delta\_T)

print("零位温度系数（融合处理之前）: ", alpha\_0\_before)

print("零位温度系数（融合处理之后）: ", alpha\_0\_after)

print("灵敏度温度系数（融合处理之前）: ", alpha\_s\_before)

print("灵敏度温度系数（融合处理之后）: ", alpha\_s\_after)

# 计算均方误差

mse = mean\_squared\_error(P, y\_pred)

print("均方误差: ", mse)

# 计算均方根误差

rmse = np.sqrt(mse)

print("均方根误差: ", rmse)

# 计算平均绝对误差

mae = mean\_absolute\_error(P, y\_pred)

print("平均绝对误差: ", mae)

# 计算误差平方和

sse = np.sum((P - y\_pred) \*\* 2)

print("误差平方和: ", sse)

# 计算平均绝对百分比误差

mape = np.mean(np.abs((P - y\_pred) / P)) \* 100

print("平均绝对百分比误差: ", mape)

# 计算决定系数

r2 = r2\_score(P, y\_pred)

print("决定系数: ", r2)

# 计算校正决定系数

n = P.shape[0] # 样本数量

p = X\_poly.shape[1] # 特征数量

adjusted\_r2 = 1 - (1 - r2) \* (n - 1) / (n - p - 1)

print("校正决定系数: ", adjusted\_r2)

执行上面的代码，可以得到以下结果：

|  |
| --- |
| 截距: 0.2786935362741745  系数: [5.03828723e-02 1.28222327e-02 1.69783360e-05 1.22306071e-04 -1.06769140e-04]  零位温度系数（融合处理之前）: 0.0015137424065536141  零位温度系数（融合处理之后）: 0.0007806260907334728  灵敏度温度系数（融合处理之前）: 0.0024932227872647757  灵敏度温度系数（融合处理之后）: -7.772234172944678e-05  均方误差: 0.008738506804008978  均方根误差: 0.09347998076598528  平均绝对误差: 0.07953333297866069  误差平方和: 0.3145862449443232  平均绝对百分比误差: inf  决定系数: 0.9970039405243398  校正决定系数: 0.9965045972783965 |

当部分样本的标定值为零时，平均绝对百分比误差的结果会是无穷大（inf）。这些结果与上一节手工计算的结果完全一致。这证实了我们的Python程序是正确的。

# 习题7

1. 什么是多元回归分析法的核心思想？

2. 什么是最小二乘法？它在多元回归分析中有什么作用？

3. 什么是回归方程的项数？它由什么决定？

4. 什么是交叉灵敏度？它会对传感器的测量结果产生什么影响？

5. 什么是均方误差最小原则？它在传感器数据融合中有什么作用？

6. 什么是温度影响系数？它包括哪两个方面？

7. 什么是均方误差？它的计算公式是什么？

8. 什么是决定系数？它有什么意义？