哈爾濱工業大學

人工智能数学基础实验报告

题　　目　数据拟合

学　　院　计算机科学与技术

专　　业　人工智能

学　　号

学　　生

任 课 教 师　刘绍辉

哈尔滨工业大学计算机科学与技术学院

2023.5

实验一：数据拟合

1 实验内容

使用 ransac 方法和最小二乘方法来拟合曲线和直线

1.1 直线拟合

根据直线方程 $ax+by+c=0$, 产生 (x_i, y_i) 随机点，然后增加随机噪声成为 (x_i+n_i, y_i+m_i), 根据这些点，拟合直线 $ax+by+c=0$ 中的参数

如果有一系列平行直线 $ax+by+c_1=0, ax+by+c_2=0, ax+by+c_3=0$, 然后对直线上的点添加类似的噪声，拟合这些平行直线

1.2 曲线拟合

自己设计曲线方程，例如圆方程，椭圆方程，然后添加适当的噪声（例如高斯噪声），然后分别采用 ransac 和最小二乘方法进行拟合

如果添加一些外点，拟合效果如何？是否有方法改进！

2 算法简介及实现细节

2.1 RANSAC

ransac算法是一种基于随机采样的迭代算法，用于从一组包含离群值或错误数据的观测数据中估计出数学模型的参数。相比最小二乘方法，它能够剔除不合格的数据，因此对于有部分错误数据的数据样本，能够更快更准地给出辨识结果。ransac算法的一个典型应用是图像中的直线拟合问题。给定一组图像中的点，我们想要找到一条直线来表示这些点。但是这些点可能包含一些噪声或者不属于直线的点，如果使用最小二乘方法，就会受到这些点的影响，导致拟合效果不好。而使用ransac算法，就可以剔除这些外点，只用内点来拟合直线，得到更好的结果

ransac算法的基本流程如下：

1.给定以下参数：

data：一组观测数据

model：拟合模型（例如线性、二次曲线等）

n：用于拟合的最小数据组数

k：算法规定的最大迭代次数

t：数据和模型匹配程度的阈值，在t范围内即为内点，在范围外即为外点

d：表示模型合适的最小数据组数

2.返回以下结果：

bestFit：一组最匹配的模型参数

3. 迭代过程：

初始化迭代次数和最佳误差

随机选择n个数据作为样本，根据样本计算模型参数

将所有数据带入模型，计算与模型的距离，统计距离小于t的内点数

如果内点数大于d，则认为找到了一个好的模型，用所有内点重新拟合模型，计算误差

如果误差小于最佳误差，则更新最佳误差和最佳拟合参数

增加迭代次数，直到达到k或者满足收敛条件

2.2 最小二乘法

数据拟合的最小二乘算法是一种数学回归分析方法，用于根据一组数据点确定最佳拟合曲线或最佳拟合直线。这种方法的目标是使数据点与拟合曲线之间的偏差（残差）的平方和最小。偏差是观测值与拟合值之间的垂直距离。最小二乘算法可以用于线性或非线性模型，其中线性模型具有形如$y = mx + b$的方程，非线性模型具有更复杂的方程。

最小二乘算法的实现细节取决于所使用的模型和编程语言。一般来说，实现步骤如下：

1. 定义一个模型方程，包含一个或多个参数。
2. 计算每个数据点与模型方程之间的偏差，并将其平方。
3. 对所有数据点求偏差平方和，并将其作为目标函数。
4. 使用优化算法（如梯度下降、牛顿法等）寻找使目标函数最小化的参数值。
5. 评估拟合结果的优劣，例如使用R平方、均方误差等指标。

3 实验设置及结果分析（包括实验数据集）

3.1 数据集生成

本文利用了 NumPy 库来生成数据集。以此提供了高效的多维数组对象和相关的数学函数。

NumPy 的核心是 ndarray 对象，它是一个多维数组，可以存储同质的数据。ndarray 对象支持基本的数学运算，如加、减、乘、除、求幂等。NumPy 还提供了许多数学函数，如指数函数、对数函数、统计函数等。这些函数可以对 ndarray 对象进行操作，也可以对标量进行操作。

首先，使用 np.linspace 函数生成均匀分布的 X 轴数据，并通过函数表达式生成对应的 Y 轴数据

```
data_x = np.linspace(t_min, t_max, n_samplers)
data_y = func(self.data_x)
```

接着，使用 np.random.normal 函数生成正态分布的随机数，将这些随机数加到数据中，以获得含有高斯噪声的数据。

```
data_x += np.random.normal(scale=s_noise, size=n_samplers)
data_y += np.random.normal(scale=s_noise, size=n_samplers)
```

数据集的末端也添加了一些外点。这些外点与数据完全无关。

```
data_x[:n_outliers] = 3 + 0.5 * np.random.normal(size
    =(n_outliers, 1))
data_y[:n_outliers] = -3 + 5 * np.random.normal(size=
    n_outliers)
```

3.2 直线拟合

在实验中，我们使用了 scikit-learn 库来拟合数据。scikit-learn ，也叫sklearn，是一个基于 python 的开源机器学习工具包，它依赖于 python 的数值计算库，如 NumPy、SciPy 和 Matplotlib，实现了高效的算法应用。它包含了几乎所有主流的机器学习算法。

我们使用了 scikit-learn 库中的 LinearRegression 和 RANSACRegressor 类来实现 RANSAC 和最小二乘算法模型。我们还使用了 scikit-learn 库中的 mean_squared_error 来计算均方误差，以分析各组数据的拟合情况。

在本次实验中，我们使用了 y = x+4+offset 的直线方程，并在 offset 分别为 0、5 和 10 时进行了拟合。每组数据集包含 100 个数据点，其中有 10 个外点。拟合结果如下：

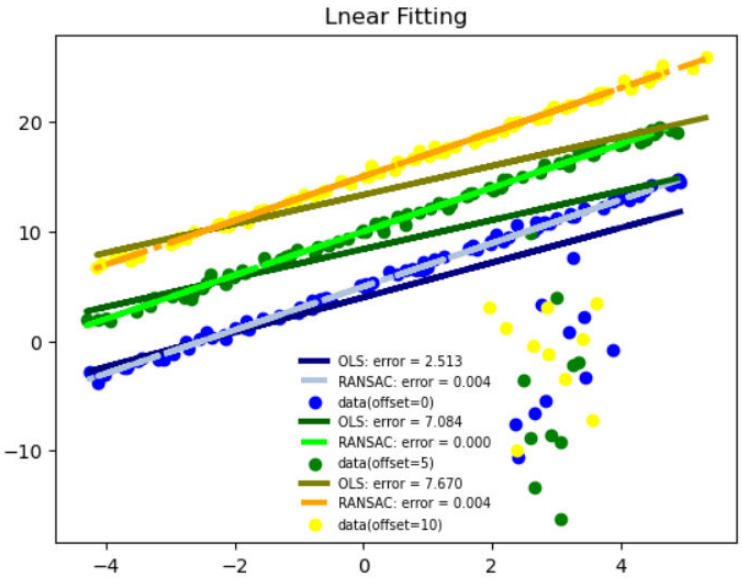

图 1: 直线拟合结果

3. 3 曲线拟合

为了更好地拟合曲线，实验在原有的线性模型基础上引入了 PolynomialFeatures 模型。PolynomialFeatures 类是一个数据预处理器，其能够生成一个新的特征矩阵，包含原始特征的所有多项式组合。例如，如果原始特征是$[x_1, x_2]$，那么二次多项式特征就是 $[1, x_1, x_2, x^2, x_1x2, x^2]$。这样就可以将线性回归模型扩展为多项式回归模型，从而提高模型的拟合能力。

为了将模型与 RANSAC 模型和最小二乘模型结合起来，实验使用了 make_pipeline 函数。make_pipeline 函数是一个快捷方式，其能够根据给定的估计器构造一个 Pipeline 对象。Pipeline 对象能够将多个转换器和一个最终的估计器串联起来，形成一个复合的估计器。这样可以简化数据预处理和模型训练的流程，避免重复的代码和参数。

在本次实验中，采用了 $y = \sin(x) + 0.4\sin(0.5x)$ 的曲线方程，并构建了一个包含 90 个内点和 10 个外点，共 100 个点的数据集。拟合结果如下：

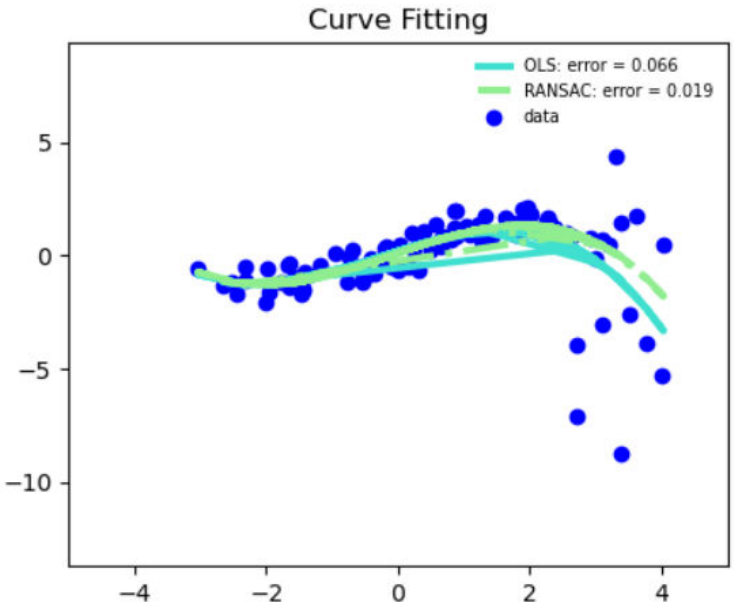

图 2: 曲线拟合结果

4 结论

RANSAC算法和最小二乘法是两种常用的模型拟合方法，在实验中二者均有不错的表现。

最小二乘法假设所有的数据点都符合高斯分布，因此对于有较大噪声或无效点的数据不太适用，容易受到离群点的影响。

RANSAC算法则是一种迭代的方法，它从数据中随机抽取一部分点作为内点，用这些点来估计模型参数，然后用剩余的点来验证模型的好坏，重复这个过程直到找到最优的模型。RANSAC算法可以有效地排除离群点，适用于有较多噪声或无效点的数据。

二者的改进方法有：

最小二乘法可以通过加权、正则化、稀疏等方式来提高拟合效果和稳定性，也可以结合其他优化算法如梯度下降、牛顿法等来求解。

RANSAC算法可以通过调整迭代次数、内点阈值、置信度等参数来提高拟合效果和效率，也可以结合其他拟合方法如LM算法、DLT算法等来求解。另外，还有一些基于RANSAC的变种算法，如

MSAC、MLESAC、PROSAC等，它们对RANSAC进行了一些改进和优化。

参考文献

[1] https://scikit-learn.org/stable/auto_examples/linear_model/plot_robust_fit.html#sphx-glr-auto-examples-linear-model-plot-robust-fit-py.

[2] https://blog.csdn.net/zhoucoolqi/article/details/105497572.

[3] https://blog.csdn.net/MoreAction_/article/details/106443383.

[4] https://juejin.cn/post/7068904750296596511.