

RANSAC 算法实验报告

■■■■^{*}, ■■■■[†]

东南大学 ■■■■, 南京 210096

【摘要】 本次实验中主要对随机抽样一致 (RANSAC) 算法进行了复现并对结果进行了可视化呈现。RANSAC 算法是一种迭代算法, 用于从一组包含异常值的观察数据中估计数学模型的参数, 并降低异常值对估计值的影响。因此, 该算法也可用于异常值检测。RANSAC 是一种非确定性算法, 它仅以一定的概率产生合理的结果, 并且该概率随着允许的迭代次数增加而增加。该算法于 1981 年由 SRI International 的 Fischler 和 Bolles 首次发布。他们使用 RANSAC 来解决位置确定问题 (LDP), 其目标是确定一组已知位置的 landmark 在空间中投射到图像上的点。

【关键词】 RANSAC 算法

1 实验环境

本次实验中算法运行环境硬件环境与软件环境如下:

- Intel i7-10875H
- DDR 4 32GB Memory
- x86_64
- Intel Geforce RTX2060
- Windows 10
- Python 3.7
 - Numpy 1.21.1
 - Scipy 1.7.1
 - Matplotlib 3.4.3

2 实验内容

本次实验内容主要包括了如下内容:

- 构建数据集

- 复现 RANSAC 算法
- 对运行时间与算法精度进行评估

其中数据集包括一定比例的内点和噪声点, 噪声来源可以为高斯分布或者均匀分布。

2.1 数据集构建

在本次实验实现中, 目标曲线固定为

$$x - y = 0 \quad (1)$$

为了方便直观对比不同噪声和参数设置下目标函数和学习到的函数之间的差距。

数据集构建算法如1所示。

需要注意在该算法中, 对数据集进行随机打乱并不是有必要的一步, 由于 RANSAC 算法本身需要进行随机抽样的特点, 打乱与否对结果并无显著影响。

2.2 RANSAC 算法复现

RANSAC 算法原理如算法2所示。

2.3 RANSAC 算法评估

对 RANSAC 算法的评估标准主要包括计算时间, 迭代轮数, 以及模型拟合参数与真实参数的差距。

3 实验结果与分析

本节主要包括两部分, 第一部分为算法计算时间和迭代轮数的统计分析; 第二部分为算法拟

实验时间: 2021 年 10 月 31 日

报告时间: 2021 年 10 月 31 日

[†] 指导教师

*学号: ■■■■

*E-mail: ■■■■

Input: 噪声类型 NoiseType; 高斯分布参数 μ, σ ; 数据集大小 n ; 外点比重 r

Result: 点集 p

```

for  $i = 0, 1, 2, \dots, n - 1$  do
     $p_{xi} \leftarrow \text{random.uniform}(0, 10)$ ;
     $p_{yi} \leftarrow p_{xi}$ ;
end
if NoiseType == Uniform then
    for  $i = \lceil n \times (1 - r) \rceil, \lceil n \times (1 - r) \rceil + 1, \lceil n \times (1 - r) \rceil + 2, \dots, n - 1$  do
         $p_{xi} \leftarrow \text{random.uniform}(0, 10)$ ;
    end
end
else if NoiseType == Gaussian then
    for  $i = \lceil n \times (1 - r) \rceil, \lceil n \times (1 - r) \rceil + 1, \lceil n \times (1 - r) \rceil + 2, \dots, n - 1$  do
         $p_{xi} \leftarrow \text{random.gaussian}(\mu, \sigma)$ ;
         $p_{yi} \leftarrow \text{random.gaussian}(\mu, \sigma)$ ;
    end
end
else
    Raise NotImplementedError;
end
random.shuffle( $p$ );

```

Algorithm 1: 数据集产生算法

合结果的可视化分析。

3.1 计算时间与迭代轮数评估

计算时间与迭代轮数评估部分中，噪声来源固定为 uniform，threshold 固定为 0.7，噪点率固定为 0.2，采样数固定为 4。

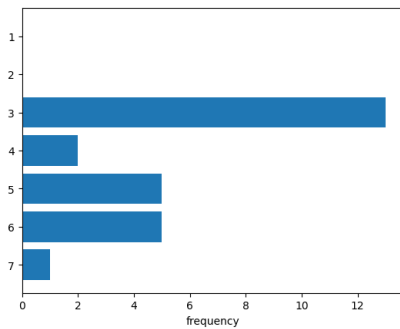


图 1 迭代次数统计频率（共计 26 次实验）

平均运行时间 ($10^{-3} s$)
3.01

表 1 50 次随机实验平均运行时间

Input: 阈值 t ; 目标内点量 g ; 概率 p

Data: 数据集 D

Result: 拟合直线参数 m

```

best_inliers  $\leftarrow 0$ ;
iteration_count  $\leftarrow 0$ ;
inlier_set  $\leftarrow \emptyset$ ;
max_iteration  $\leftarrow \infty$ ;
 $i \leftarrow 0$ ;
while  $i < \text{max\_iteration}$  do
     $i \leftarrow i + 1$ ;
     $ic \leftarrow 0$ ;
    if inlier_set ==  $\emptyset$  then
         $p_r \leftarrow \text{random select } r \text{ points from } p$ ;
         $p \leftarrow p - p_r$ ;
        inlier_set  $\leftarrow p_r$ ;
    end
     $m \leftarrow \text{estimated model paramter from inlier set}$ ;
     $p_i \leftarrow \text{all inlier points with } t$ ;
    update max_iteration with  $p$ ;
    if  $p_i = \emptyset$  then
         $p \leftarrow p + p_r$ ;
        inlier_set  $\leftarrow \emptyset$ ;
    end
    else
        inlier_set  $\leftarrow \text{inlierset} + p_i$ ;
    end
    if  $\|p_r\| > \text{best\_inliers}$  then
        best_inliers  $\leftarrow \|p_r\|$ ;
        best_model  $\leftarrow m$ ;
    end
    if best_inliers  $> g$  then
        break;
    end
end

```

Algorithm 2: RANSAC 算法

从上图可以看出，RANSAC 算法的迭代轮数分布总体和二项分布比较相近。

3.2 模型拟合效果评估

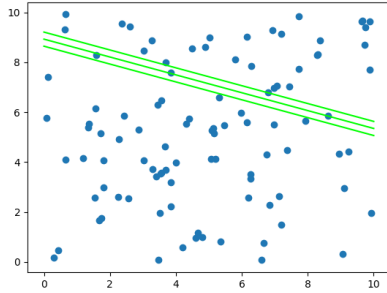


图 2 实验参数：总点数为 100，噪点率为 0.8，噪声类型为 Uniform，阈值为 0.5，目标内点数为 10

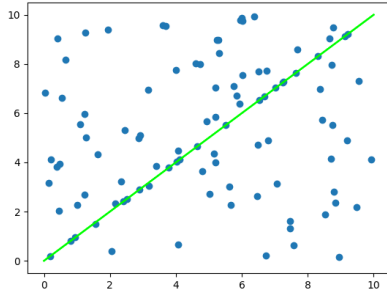


图 3 实验参数：总点数为 100，噪点率为 0.1，噪声类型为 Uniform，阈值为 0.05，目标内点数为 30

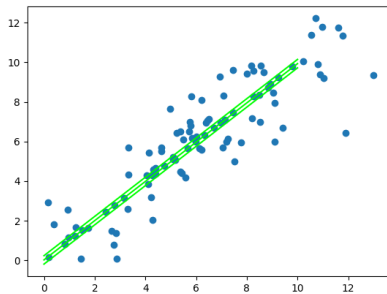


图 4 实验参数：总点数为 100，噪点率为 0.4，噪声类型为 Gaussain，阈值为 0.1，目标内点数为 20

从图中可以看出在噪声点数量并不是非常大的时候，RANSAC 算法能够很好的拟合出目标函数曲线并指示出全部的异常点。然而当信噪比过低或者参数不恰当的时候，RANSAC 算法就会无法得到足够的信息用于拟合目标函数或者提前终

止。该结果表示 RANSAC 算法对噪声的处理能力是存在一定局限的，并且其仍然对先验知识有一定要求，需要人工寻找合理的参数。

在实际应用中可以考虑增加实验轮数来提高 RANSAC 算法找到正确模型的概率。同时当噪声影响比较大的时候应当改变拟合方法或者尝试进行异常排查。

4 结论

在本次实验中对 RANSAC 算法的复现以及对结果可视化表明 RANSAC 算法作为一种工程上的基于概率的模型参数估计算法，其在正常情况下能够有效的过滤外点并且最终以一定概率给出最优解。但 RANSAC 算法的参数较多，并且部分参数依赖经验。同时 RANSAC 算法难以推广到同时多个模型的拟合，存在一定的局限性。

附录

本次实验中所用全部代码可在压缩包内于本报告同目录下找到，总共包括两个文件共 174 行代码，分别为 `main.py` 和 `utils.py`。