

基于WiFi无线信号的室内定位研究

崔宇骁 学号：21009102088

西安电子科技大学
通信工程学院
指导教师：王杰令

2025 年 5 月 26 日

目录

- ① 研究背景与意义
- ② 研究内容与方法
- ③ 基于接收信号强度定位的基本原理
- ④ 基于接收信号强度的指纹定位
- ⑤ 仿真环境建模
- ⑥ 最小二乘法及其改进
- ⑦ KNN回归算法及其优化
- ⑧ 其他机器学习方法进行定位
- ⑨ 总结与展望

研究背景与意义

在移动互联网和物联网技术快速发展的背景下，位置服务已成为现代社会不可或缺的基础设施。虽然全球卫星导航系统（GNSS）在室外环境中表现出色，但其在室内环境中的应用受到严重限制，主要由于信号衰减和多径效应等问题。

而随着WiFi在世界各地的普及，其覆盖面越来越广泛，WLAN技术凭借其广泛部署的基础设施、较低的实施成本和良好的可扩展性，在室内定位领域占据重要地位。其接入点（Access Point, AP）或基站定期发送的信标信号中所含的接收信号强度（Received Signal Strength, RSS）为室内定位技术（Indoor Positioning Technology）提供了坚实的应用基础。

研究内容与方法

本文以接收信号强度为特征进行室内定位，首先使用MATLAB建立仿真环境，部署信号接入点，构建位置指纹库（Fingerprint Database）并进行数据处理。首先使用最小二乘法（Least Squares Method）定位进行定位，并根据最小二乘法的实现原理及其存在的问题，提出栅格梯度下降搜索法（Grid Gradient Descent Search Algorithm），在保证一定的定位精度的同时降低矩阵运算的阶数，在低精度定位需求场景中显著减少计算量。然后使用KNN回归进行定位并通过卡尔曼滤波（Kalman Filter）和粒子滤波（Particle Filter）对KNN回归结果进行优化以提高定位精度。最后引入梯度提升（Gradient Boosting）、随机森林（Random Forest）机器学习算法进行对比。

基于接收信号强度定位的基本原理

接收信号强度是一种易获取的物理层参数，表示接收端收到的信号功率，其数值受距离、墙体遮挡、多径反射、设备摆放角度等多种因素影响。尽管RSS具有较大的波动性，但它仍是当前WiFi定位中应用最为广泛、易于获取的数据类型。

对数距离路径损耗模型（Log-Distance Path Loss Model）是描述无线信号在传播环境中功率衰减的经典经验模型，广泛应用于室内外环境下的信号强度预测与定位技术。其核心思想是：信号功率随传播距离的增加呈对数衰减规律，并结合环境因素引入随机噪声项以反映实际传播的不确定性。

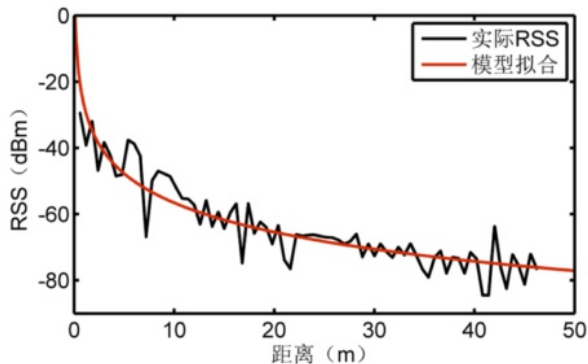
基于接收信号强度定位的基本原理

其基本模型如下：

$$RSS = RSS_0 - 10n \log_{10}\left(\frac{d}{d_0}\right) \quad (1)$$

式中 RSS 为接收信号强度； RSS_0 表示发射端与接收端在参考距离时的信号强度，通常以dBm（分贝毫瓦）为单位；在室内环境中，参考距离取1m； n 为路径损耗指数，用来描述信号衰减随距离变化的剧烈程度，在室内环境（例如办公室，走廊）中典型取值为 $n = 2.5 \sim 4$ ，本研究取值使用 $n = 2.5$ 。

此传播模型能够完成接收信号强度与距离的转换，在仿真环境中的采样点处获取接收信号强度，并根据对数距离路径损耗模型估算出该接收端距离发射端接入点的距离。



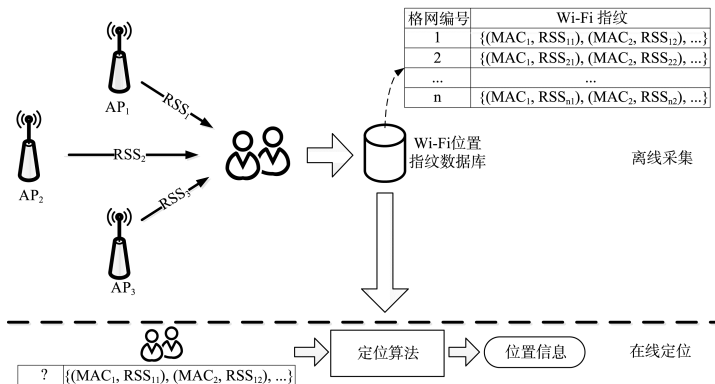
图中的黑线是一组在室内环境中测量的实际数据，红线是对数距离损耗模型的拟合结果，可以看出模型可以反映总体趋势，但和真实室内环境下的情况还是有较大区别，黑线的波动是实际的信号传播环境造成的。

位置指纹

位置指纹定位技术（Location Fingerprinting）通过建立物理空间与信号特征的空间映射关系实现目标定位。其核心思想是将特定位置对应的唯一信号特征定义为“指纹”，并通过模式匹配算法将实时采集的特征与离线指纹数据库进行比对。指纹可以是多种类型的，任何位置独特的（对区分位置有帮助的）特征都可以用来作为一个指纹。比如某个位置上通信信号的多径结构、某个位置上是否能检测到接入点或基站、某个位置上检测到的来自基站信号的接收信号强度、某个位置上通信时信号的往返时间或延迟，这些都能作为一个位置指纹。

接收信号强度作为位置指纹

假设有一个固定的信号发射源，由前述基本原理可知在距离其不同距离的位置上的平均RSS的衰减和距离的对数成正比，RSS可以被用来计算移动设备与AP（或基站）之间的距离，该距离可以用来做移动设备的三边角测量从而定位，而实际环境的影响会造成RSS误差较大，从而使得定位误差较大。然而，如果一个移动设备能接收到来自多个发射源的信号，或者固定的多个基站都能感知到同一个移动设备，那么可以使用来自多个发射源或者多个接收器的RSS组成一个RSS向量，作为和位置相联系的指纹。使用来自放置的多个接入点（Access Point, AP）的接收信号强度组成一个信号强度向量，作为和位置相联系的指纹。



上图虚线以上的离线阶段为获取接收信号强度并建立位置指纹库的基本过程，虚线以下的在线定位阶段为基于位置指纹数据库的定位过程。

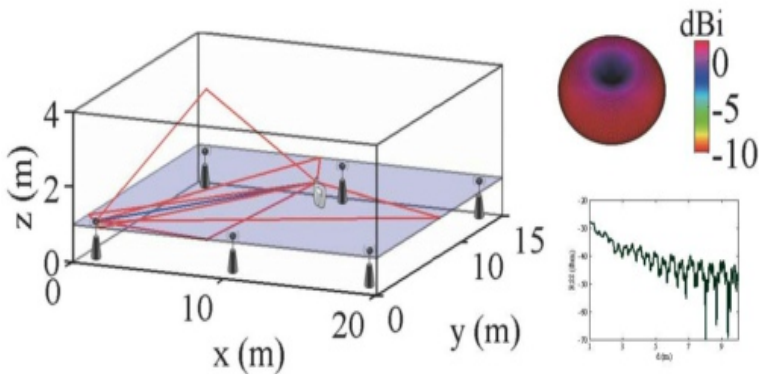
仿真环境建模

在水平面上以指定间隔0.01m生成等距网格点，形成覆盖全区域的密集采样矩阵。每个网格点记录三维坐标信息，并作为接收设备的位置进行信号强度计算。对于每个网格点，计算其与AP之间的直线距离，基于自由空间传播模型估算信号强度。直射路径的信号衰减与距离成反比，同时考虑信号传播时延引起的相位偏移；引入镜像法模拟墙面反射效应，针对六个平面（前、后、左、右、天花板、地面）分别生成虚拟镜像AP，计算反射路径。通过几何关系计算实际AP到镜像点、再至接收点的总传播距离。根据菲涅尔反射系数计算信号衰减，该系数由墙面材料属性（如混凝土介电常数）和入射角度共同决定。

仿真环境建模

记录每条反射路径的传播时延，转化为相位变化后进行复数叠加。针对天花板和地面的垂直反射路径，额外引入天线方向图衰减因子，修正因天线辐射方向性导致的信号强度变化，并考虑电磁波垂直极化特性对反射系数的影响。将直射路径与六条主要反射路径的电场强度进行复数叠加，模拟多径干涉效应。总电场强度反映了信号在传播过程中的衰减、相位偏移及多径叠加的综合结果。将合成电场强度转换为接收信号强度（RSS），单位为dBm。计算中计入发射与接收天线的增益，最终生成每个网格点对应的RSS值。输出结果为二维矩阵，每行包含网格点坐标及其对应的来自各个AP的RSS测量值。生成的指纹库以矩阵形式存储。

仿真环境建模



上图是一个假定的空旷的房间，有一个发射源（AP）在房间左边的角落，一个接收器在房间的中央，接收器收到这个AP的信号中包含了来自1条直射路径与6条（墙壁）反射路径的信号（多次反射的影响很小，可以忽略），上图右下角的曲线是某条直线上RSS随距离的衰减，通过这种射线跟踪技术可以降低对数模型带来的误差

指纹库的数据处理

此前以0.01m为采样间隔的指纹库是一个高精度的指纹库，但该高精度指纹库数据量庞大，在具体情景实现时会导致内存占用过高，数据点冗余等问题。因此应该对该高精度指纹库进行精简，本文采用两种方法对原高精度指纹库进行抽样，分别为均匀采样和随机采样。均匀采样的采样间隔设置为0.1m，即每隔0.1m在高精度指纹库中取样一个点。随机采样从所有的网格中随机进行挑选，并且保证采样的点数与均匀采样的点数相同。由此实现了指纹库的精简，后续从指纹库中对数据进行采集时选用均匀采样指纹库和随机采样指纹库这两个精简指纹库，称之为离线数据。

模拟在线数据的采集

为了模拟实际定位过程中在线数据的采集：编写一个生成随机轨迹的函数，首先设置一个随机起始点，并设置随机步长（确保在允许范围内）和随机相位来模拟该点转向运动，同时根据房间尺寸设计边界处理来确保轨迹不会超出边界，设置一个步长来具体确定该轨迹中点的个数。根据这一个随机轨迹的点的坐标，在离线指纹库中根据划分的网格，来寻找该坐标点最接近的网格，确保该坐标映射到最近的网格点，然后获得该网格在离线指纹库中对应的RSS。这就完成了模拟在线数据的采集，保存坐标（供后续定位结果对比使用）和对应的RSS，将这个指纹库称为在线数据。

最小二乘法

最小二乘法的基本思想是使得误差最小化最小二乘定位通过最小化误差的平方和来寻找数据的最佳匹配，从而达到优化结果的目的。根据信号传播模型，我们把在每一个点收集到的对应AP的RSS值转换为相应的距离对应AP的距离： (d_1, d_2, \dots, d_n) 假设我们的待定位节点坐标为 (x, y) ，我们设定的信号接入点AP坐标为 (x_i, y_i) ， $i = 1, 2, \dots, n$ 待定位节点和AP之间的距离可以由以下方程组进行表示：

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ \dots \\ (x - x_n)^2 + (y - y_n)^2 = d_n^2 \end{cases} \quad (2)$$

最小二乘法

用第 i 个方程减去第 $i+1$ 个方程, $i = 1, 2, n-1$ 以此来消去二次方幂次参数 $x^2 + y^2$ 得到以下方程组:

$$\begin{cases} 2(x_2 - x_1)x + 2(y_2 - y_1)y = d_1^2 - d_2^2 - (x_1^2 + y_1^2) + (x_2^2 + y_2^2) \\ 2(x_3 - x_2)x + 2(y_3 - y_2)y = d_2^2 - d_3^2 - (x_2^2 + y_2^2) + (x_3^2 + y_3^2) \\ \dots \\ 2(x_n - x_{n-1})x + 2(y_n - y_{n-1})y = d_{n-1}^2 - d_n^2 - (x_{n-1}^2 + y_{n-1}^2) + (x_n^2 + y_n^2) \end{cases} \quad (3)$$

以一个矩阵形式来表示这个方程组, 记为

$$\mathbf{AX} = \mathbf{B} \quad (4)$$

最小二乘法

其中矩阵方程的各个参数分别为：

$$\mathbf{A} = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) \\ \dots & \dots \\ 2(x_n - x_{n-1}) & 2(y_n - y_{n-1}) \end{bmatrix} \quad (5)$$

$$\mathbf{B} = \begin{bmatrix} d_1^2 - d_2^2 - (x_1^2 + y_1^2) + (x_2^2 + y_2^2) \\ d_2^2 - d_3^2 - (x_2^2 + y_2^2) + (x_3^2 + y_3^2) \\ \dots \\ d_{n-1}^2 - d_n^2 - (x_{n-1}^2 + y_{n-1}^2) + (x_n^2 + y_n^2) \end{bmatrix} \quad (6)$$

$$\mathbf{X} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (7)$$

最小二乘法

由于在 n 个方程中，待定位节点坐标 \mathbf{X} 不会满足所有方程，因此设置一个误差向量为 $\varepsilon = \mathbf{AX} - \mathbf{B}$ ，取误差向量的平方和则有：

$$\mathbf{E} = \|\varepsilon\|^2 = \varepsilon^T \varepsilon = (\mathbf{AX} - \mathbf{B})^T (\mathbf{AX} - \mathbf{B}) \quad (8)$$

若要使得误差最小，则要使得 \mathbf{E} 最小，对上式对 \mathbf{X} 进行求导，令导数为 0，可以得到：

$$\frac{d\mathbf{E}}{d\mathbf{X}} = 2\mathbf{A}^T (\mathbf{AX} - \mathbf{B}) - 2\mathbf{A}^T \mathbf{B} = 0 \quad (9)$$

求解上述方程可以得到待定位节点的估计坐标 \mathbf{X} ：

$$\mathbf{X} = \mathbf{A}^T \mathbf{A}^{-1} (\mathbf{A}^T \mathbf{B}) \quad (10)$$

最小二乘法定位结果

上图为使用最小二乘法进行室内定位的轨迹对比图以及定位误差的分布，接入点放置在室内环境的6个角落且间隔相同。其中图左示的蓝色实线为随机生成的真实轨迹，紫色的虚线为所有接入点通过最小二乘法预测得出的预测轨迹，星号符号代表该室内区域环境中放置的接入点位置，右示的柱状图为定位误差的频数分布。

最小二乘法噪声敏感性分析

上图为在添加了RSS噪声之后的最小二乘法平均定位误差，横轴为噪声标准差的取值，纵轴为该噪声标准差下对应的平均定位误差。由图可见，最小二乘法在噪声的影响下虽然整体误差趋势趋于上升，但误差表现较为不稳定，误差存在局部的小范围波动现象。在经过多次的仿真之后，得出最终的平均定位误差在6.5m左右

最小二乘法的改进：栅格梯度下降搜索法

基本思路

根据 RSS 数据进行区域选择，然后仅在选定区域内，利用该区域内的AP使用最小二乘法 (Least Squares, LS) 进行位置估计，这样就降低了矩阵的阶数，最终根据区域边界对结果进行约束处理，提高精度并降低计算量。

RSS 区域选择 (MSE)

- 分成4个大区域 (2×2)，通过 RSS均方误差 (MSE)进行选择
- RSS MSE 公式：

$$\text{MSE}_i = \frac{1}{N} \sum_{j=1}^N (R_{ij}^{\text{mean}} - R_j)^2$$

- R_j : 当前测点 RSS, R_{ij}^{mean} : 区域*i*中第*j*个AP平均RSS

矩阵构造

RSS转换为距离

$$d_i = 10^{\frac{RSS_i - P_0}{-10 \cdot n}}, \quad P_0 = -40 \text{ dBm}, \quad n = 2.5$$

A, B 矩阵构造 ($i = 2, 3, \dots, N$)

$$A_{i-1} = [2(x_i - x_{i-1}), 2(y_i - y_{i-1})]$$

$$B_{i-1} = d_{i-1}^2 - d_i^2 - x_{i-1}^2 - y_{i-1}^2 + x_i^2 + y_i^2$$

LS 解法

$$\hat{\mathbf{x}} = A^T A^{-1} A^T B$$

边界修正与结果输出

边界约束函数

- 根据选择的**区域编号** (1-4), 定义对应 $(x_{min}, x_{max}, y_{min}, y_{max})$ 边界
- 保证估计位置 \hat{x} 在区域内:

$$\hat{x} = \min(\max(\hat{x}, x_{min}), x_{max})$$

$$\hat{y} = \min(\max(\hat{y}, y_{min}), y_{max})$$

最终输出

- **estimated_position**: 修正后的目标估计位置 (\hat{x}, \hat{y})
- **selected_region**: 选中的大区域编号 (1 ~ 4)

结果分析

可以看到平均定位误差相比于原最小二乘法运算误差提高了许多。但其中的计算量由六阶矩阵乘法降低为四阶矩阵乘法，即从216次标量乘法降低为64次标量乘法，降低了70.37%。尽管精度未有提升，但其计算量的下降在某些对计算资源和效率要求比较严苛的系统环境中仍然有一定的应用价值，并且该算法的基本思想具有一定的可扩展性。

KNN回归算法数学模型

输入与输出

- 输入空间： $\mathcal{R} \subseteq \mathbb{R}^n$ (n 维RSS信号强度)
- 输出空间： $\mathcal{Y} \subseteq \mathbb{R}^2$ (二维坐标位置)
- 训练集： $\{(\mathbf{R}_i, (x_i, y_i))\}_{i=1}^N$ ，其中 \mathbf{R}_i 为RSS向量， (x_i, y_i) 为真实位置

算法公式

对于测试样本 \mathbf{r}_{test} ：

$$(\hat{x}, \hat{y}) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(\mathbf{r}_{\text{test}})} (x_i, y_i) \quad (11)$$

其中：

- $\mathcal{N}_k(\mathbf{r}_{\text{test}})$ 是 \mathbf{r}_{test} 的 k 近邻集合(以欧氏距离度量)

算法实现细节

Algorithm 1 KNN回归预测流程

输入: 训练集 $\{(\mathbf{R}_i, (x_i, y_i))\}_{i=1}^N$, 测试点 \mathbf{r}_{test}

输出: 预测位置 (\hat{x}, \hat{y})

- 1: 计算距离: $d_i = \|\mathbf{R}_i - \mathbf{r}_{\text{test}}\|_2, \forall i \in [1, N]$
 - 2: 选择距离最小的 k 个邻居 $\mathcal{N}_k(\mathbf{r}_{\text{test}})$
 - 3: 预测位置: $(\hat{x}, \hat{y}) = \frac{1}{k} \sum_{i \in \mathcal{N}_k} (x_i, y_i)$
-

定位精度计算:

$$\text{Accuracy} = \frac{1}{M} \sum_{j=1}^M \|(\hat{x}, \hat{y})_j - (x_j, y_j)^{\text{true}}\|_2 \quad (12)$$

结果分析

由图可以看到使用KNN算法的定位误差要低于最小二乘算法，且由此前KNN回归算法的基本原理可知：该算法不需要预先得知接入点的位置，因此比最小二乘法有更加广泛的引用前景。

由图中曲线可以看出此算法定位误差显著低于最小二乘算法，且噪声条件下的稳定性较高。

数据预处理与标准化

标准化变换

对RSS信号 $\mathbf{R} \in \mathbb{R}^n$ 进行标准化:

$$\mathbf{R}^{\text{scaled}} = \frac{\mathbf{R} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \quad (13)$$

其中:

- $\boldsymbol{\mu} = \text{mean}(\text{offline_rss})$ (训练集均值向量)
- $\boldsymbol{\sigma} = \text{std}(\text{offline_rss})$ (训练集标准差向量)

训练阶段

- 计算统计量:

$$\mu = \frac{1}{N} \sum_{i=1}^N \mathbf{R}_i$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{R}_i - \mu)^2}$$

- 转换训练数据:

$$\mathbf{R}_{\text{train}} = \text{scaler}(\text{offline_rss})$$

测试阶段

- 使用训练集的 μ, σ 转换:

$$\mathbf{R}_{\text{test}} = \frac{\text{rss} - \mu}{\sigma}$$

- 保持位置坐标不变:

$$(x, y)_{\text{train}} = \text{offline_location}$$

$$(x, y)_{\text{test}} = \text{trace}$$

交叉验证选择最优 k 值

5折交叉验证误差

对于每个候选超参数 k ，使用5折交叉验证（将数据集分成 5 份，每次用 4 份训练，1 份验证）评估模型误差：

$$CV(k) = \frac{1}{5} \sum_{j=1}^5 MSE^{(j)}(k)$$

其中，第 j 折的均方误差定义为：

$$MSE^{(j)}(k) = \frac{1}{n_j} \sum_{i=1}^{n_j} \left\| (\hat{x}, \hat{y})_i^{(j)}(k) - (x_i, y_i)^{(j)} \right\|_2^2$$

最优参数选择

最终选取使交叉验证误差最小的 $k^* = \arg \min_k CV(k)$ ：

交叉验证分数（默认评分指标）

R^2 决定系数

在每折交叉验证中，使用 R^2 分数衡量预测位置的拟合程度：

$$R^2 = 1 - \frac{\sum_{i=1}^{n_j} \|(x, y)_i^{(j)} - (x_i, y_i)^{(j)}\|_2^2}{\sum_{i=1}^{n_j} \|(x_i, y_i)^{(j)} - \overline{(x, y)}^{(j)}\|_2^2}$$

其中：

- $(x_i, y_i)^{(j)}$ 为第 j 折验证集中的真实位置
- $(x, y)_i^{(j)}$ 为预测位置
- $\overline{(x, y)}^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} (x_i, y_i)^{(j)}$ 为该验证集位置均值

取平均得到每个 k 值对应的交叉验证分数：

$$\text{mean_test_score}(k) = \frac{1}{K} \sum_{j=1}^K R_j^2(k)$$

实验结果可视化

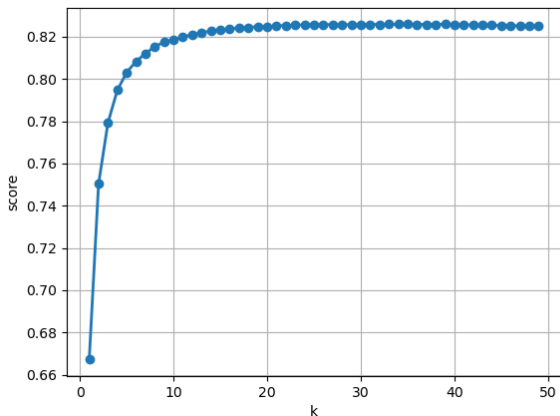


图: 不同 k 值下的交叉验证得分变化 (R^2)

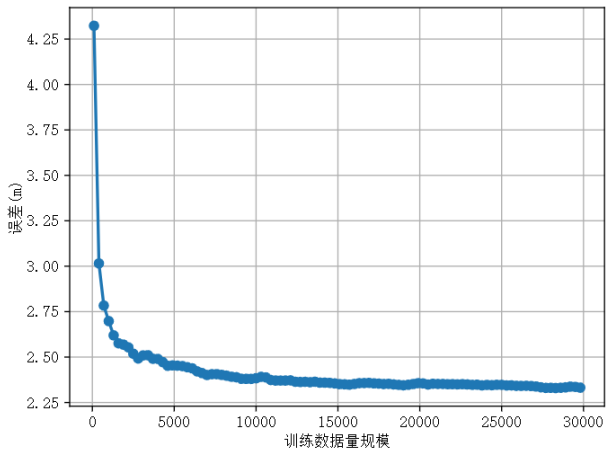


图: 训练数据量与精确度的关系

KNN与卡尔曼滤波组合定位

系统建模

设目标状态为位置与速度的组合向量：

$$\mathbf{x}_t = [x_t, v_x^t, y_t, v_y^t]^\top$$

每个时刻观测值为位置：

$$\mathbf{z}_t = [x_t, y_t]^\top$$

系统模型如下：

- 状态转移模型（匀速运动）： $\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t$
- 观测模型（只能观测位置）：

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

状态转移矩阵与观测矩阵

状态转移矩阵 \mathbf{F} 和观测矩阵 \mathbf{H} 的参数如下: ($\Delta t = 1$)

矩阵定义

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

滤波过程与参数设置

噪声建模

- 状态过程噪声协方差矩阵（简化为对角矩阵）：

$$\mathbf{Q} = \sigma_q^2 \cdot \mathbf{I}_4 \quad (\sigma_q^2 = 0.01)$$

- 观测噪声协方差矩阵（估计观测误差为4米）：

$$\mathbf{R} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

滤波步骤（预测-更新）

每个时刻执行以下递推：

$$\begin{aligned}\text{预测: } \hat{\mathbf{x}}_t^- &= \mathbf{F}\hat{\mathbf{x}}_{t-1} \\ \mathbf{P}_t^- &= \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^\top + \mathbf{Q}\end{aligned}$$

$$\begin{aligned}\text{更新: } \mathbf{K}_t &= \mathbf{P}_t^- \mathbf{H}^\top (\mathbf{H}\mathbf{P}_t^- \mathbf{H}^\top + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_t^-) \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^-\end{aligned}$$

卡尔曼滤波流程

Algorithm 2 卡尔曼滤波定位流程

输入：初始状态 \mathbf{x}_0 , 状态协方差矩阵 \mathbf{P}_0 , 系统参数 $\mathbf{F}, \mathbf{H}, \mathbf{Q}, \mathbf{R}$

- 1: **for** 每一个时刻 t **do**
 - 2: 预测状态和协方差: $\hat{\mathbf{x}}_t^- = \mathbf{F}\hat{\mathbf{x}}_{t-1}, \quad \mathbf{P}_t^- = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^\top + \mathbf{Q}$
 - 3: 计算卡尔曼增益: $\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}^\top (\mathbf{H}\mathbf{P}_t^- \mathbf{H}^\top + \mathbf{R})^{-1}$
 - 4: 更新估计状态: $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_t^-)$
 - 5: 更新协方差矩阵: $\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^-$
 - 6: **end for**
-

粒子滤波原理简介

粒子滤波（Particle Filter）是一种基于蒙特卡洛方法的递推贝叶斯估计算法，适用于非线性、非高斯系统。

基本思想

利用一组带权重的粒子来近似系统的后验分布，依次经历以下步骤：

- **预测：**根据运动模型预测粒子位置
- **更新：**根据观测更新粒子权重
- **重采样：**根据有效粒子数判断是否重采样
- **估计：**使用加权平均估计目标状态

粒子状态与预测模型

每个粒子的状态为 $\mathbf{x} = [x, v_x, y, v_y]^\top$ ，包含位置和速度。

预测公式

在时刻 t ，粒子根据如下运动模型进行预测：

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \omega, \quad (14)$$

其中 ω 为系统噪声。

在本模型中，运动为：

$$x_t = x_{t-1} + v_{t-1} \cos(\theta)$$

$$y_t = y_{t-1} + v_{t-1} \sin(\theta)$$

$$\theta_t = \theta_{t-1} + \mathcal{N}(0, \sigma_\theta^2)$$

$$v_t = v_{t-1} + \mathcal{N}(0, \sigma_v^2)$$

观测更新与权重计算

观测值 \mathbf{z} 为当前时刻的 KNN 估计位置，更新粒子权重如下：

观测模型（高斯）

假设粒子的观测误差服从二维高斯分布，则粒子 i 的权重更新为：

$$w_i \propto p(\mathbf{z} | \mathbf{x}_i) = \mathcal{N}(\|\mathbf{z} - \mathbf{x}_i^{\text{pos}}\|; 0, \sigma_d^2) \quad (15)$$

更新后归一化：

$$w_i \leftarrow \frac{w_i}{\sum_j w_j} \quad (16)$$

重采样与状态估计

重采样判断

使用有效粒子数 N_{eff} 判断是否重采样:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N w_i^2} \quad (17)$$

若 $N_{\text{eff}} < \frac{N}{2}$, 则进行重采样。

最终估计位置

粒子的加权平均作为最终估计:

$$\hat{\mathbf{x}} = \sum_{i=1}^N w_i \mathbf{x}_i \quad (18)$$

粒子滤波伪代码

Algorithm 3 粒子滤波流程

输入: 粒子集合 $\{\mathbf{x}_i\}_{i=1}^N$, 权重 $\{w_i\}$, 观测位置 \mathbf{z}

输出: 状态估计 $\hat{\mathbf{x}}$

- 1: 粒子状态预测: $\mathbf{x}_i \leftarrow f(\mathbf{x}_i) + \text{noise}$
 - 2: 权重更新: $w_i \leftarrow w_i \cdot p(\mathbf{z} \mid \mathbf{x}_i)$
 - 3: 权重归一化
 - 4: **if** $N_{\text{eff}} < N/2$ **then**
 - 5: 重采样
 - 6: **end if**
 - 7: 估计位置: $\hat{\mathbf{x}} = \sum w_i \mathbf{x}_i$
-

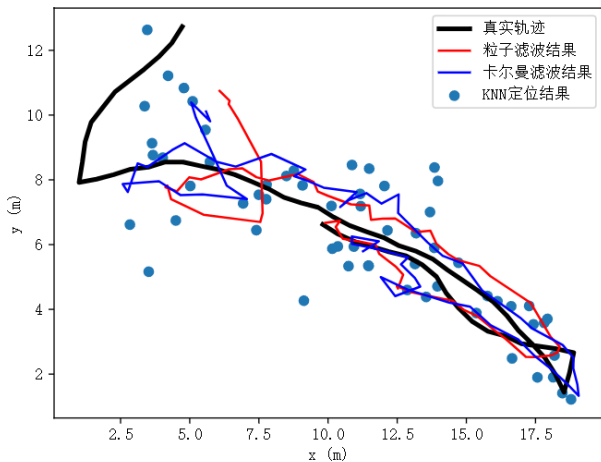


图: 不同预测方式的定位效果对比

随机森林回归简介

基本原理

随机森林（Random Forest）是一种集成学习方法，通过构建多个决策树并对其结果进行集成来提高预测精度。

在定位任务中，输入为RSS向量，输出为二维坐标，使用回归模型完成映射：

$$f : \mathbf{R} \in \mathbb{R}^n \rightarrow (x, y) \in \mathbb{R}^2 \quad (19)$$

- 对每一棵树，使用训练集的有放回抽样（Bootstrap采样）生成数据子集；
- 在每个划分点随机选择特征子集进行最佳切分；
- 最终输出多个树的预测结果均值：

$$\hat{\mathbf{y}} = \frac{1}{T} \sum_{t=1}^T f_t(\mathbf{R}) \quad (20)$$

其中 T 为总树数， f_t 为第 t 棵树的回归函数。

随机森林回归预测流程

Algorithm 4 随机森林回归预测流程

输入: 训练集 $(\mathbf{R}_i, (x_i, y_i))_{i=1}^N$, 测试集RSS \mathbf{R}_{test}

输出: 预测位置 (\hat{x}, \hat{y})

- 1: 对每一棵树 t , 使用Bootstrap采样构造数据集 D_t
- 2: 使用 D_t 训练回归树 f_t
- 3: 对于 \mathbf{R}_{test} , 计算每棵树的预测 $\hat{y}_t = f_t(\mathbf{R}_{\text{test}})$
- 4: 聚合所有树的结果: $\hat{y} = \frac{1}{T} \sum_{t=1}^T \hat{y}_t$

定位误差计算:

$$\text{Accuracy} = \frac{1}{M} \sum_{j=1}^M \|(x, \hat{y})_j - (x_j, y_j)^{\text{true}}\|_2 \quad (21)$$

梯度提升算法

问题建模

给定离线指纹库数据集：

$$\mathcal{D}_{\text{offline}} = \{(\mathbf{r}_i, \mathbf{p}_i)\}_{i=1}^N$$

其中， $\mathbf{r}_i \in \mathbb{R}^M$ 表示第 i 个位置的RSS向量， $\mathbf{p}_i = [x_i, y_i]^T \in \mathbb{R}^2$ 为其对应的真实位置坐标。

回归模型的目标是学习一个函数 $f: \mathbb{R}^M \rightarrow \mathbb{R}^2$ ，使得对于任意在线RSS观测 \mathbf{r} ，可预测其位置：

$$\hat{\mathbf{p}} = f(\mathbf{r}) = [\hat{x}, \hat{y}]^T$$

梯度提升预测流程

输入

离线RSS指纹数据集: $\{\mathbf{r}_i, \mathbf{p}_i\}_{i=1}^N$, 在线RSS观测: \mathbf{r}

训练阶段

对于每个坐标维度 $j \in \{x, y\}$, 训练回归器:

$$f_j(\cdot) \leftarrow \text{GBR}(\{\mathbf{r}_i, p_{i,j}\}_{i=1}^N)$$

预测阶段

对于在线RSS观测 \mathbf{r} , 输出预测位置:

$$\hat{\mathbf{p}} = [f_x(\mathbf{r}), f_y(\mathbf{r})]^T$$

结果说明

经过在Jupyter Notebook调用scikit-learn库运行随机森林回归算法，定位误差约为2.29m，运行时间约为29s。经过在Jupyter Notebook调用scikit-learn库运行梯度提升回归算法，定位误差约为2.22m，运行时间约为43s。

总结

本研究立足于室内环境下的定位这一问题，基于接收信号强度（RSS）这一基本信号特征与位置指纹法，构建了一个室内仿真环境。通过使用最小二乘算法、KNN回归算法、随机森林算法、梯度提升算法进行定位，并对KNN回归算法进行了卡尔曼滤波与粒子滤波来探寻如何提高室内定位精度和稳定性。

首先研究基于最小二乘算法的室内定位，这一经典算法以线性矩阵乘法为基础，根据信号接收强度和接入点（AP）的坐标来估算距离，并进行最优化的矩阵运算，经实验后得出该定位算法定位误差在6.5m左右。在经典最小二乘算法的基础上，以降低计算量为需求，提出了一种基于区域划分的栅格梯度下降搜索法，经实验，该新算法能够将计算量降低约70.37%，但误差上升到了12.53m，在一些低误差需求且计算资源有限的环境中有应用的前景，但仍需进行一定的误差控制以及计算方法的优化。

总结

之后本研究采用了KNN回归定位算法、并对KNN回归定位算法的结果进行了卡尔曼滤波和粒子滤波，计算得出：卡尔曼滤波将KNN回归的定位误差由2.3m降低到了1.7m左右，粒子滤波将定位误差降低到了1.8m左右。这为将数据滤波算法应用到机器学习中的回归算法提供了一定的应用价值。最后采用了随机森林回归、梯度提升回归算法进行室内定位，随机森林回归的定位误差约为2.29m，梯度提升回归算法的定位误差约为2.22m，但运行时间较长。

存在的不足

本文研究仍然存在一些不足：

本研究基于软件环境仿真，缺乏真实的复杂室内环境下的大规模验证；

在滤波算法中，卡尔曼滤波的各个矩阵参数较为简化，并未进行自适应以及交叉验证的处理，粒子滤波中的运动模型参数也较为简化，有进一步优化的空间；

本文提出的栅格梯度下降搜索法的误差仍然过高，仍需要进一步的自适应优化处理在降低计算量的初衷之上来降低其误差。

展望

在真实复杂的多种室内环境（个人房间、图书馆、办公室）下采集数据来适应定位的现实需求。

深入研究卡尔曼滤波和粒子滤波的特征参数寻求优化滤波效果。

从最小二乘法的矩阵构建以及矩阵特征的方向来寻求降低计算量的方法，并保证一定的定位精度。

如果有一天

a