

参赛密码 _____

(由组委会填写)

“华为杯”第十三届全国研究生
数学建模竞赛

学 校 东南大学

参赛队号 10286250

1. 李华繁

队员姓名 2. 贡欢

3. 贾程浩



题 目 基于无线通信基站的室内三维定位问题

摘 要：

本文主要研究基于无线通信基站的方式进行三维空间定位的相关问题。首先，通过对无线信号传播方式的分析，引入 NLOS 误差和测量误差，建立误差测量模型，并引入尺度因子，对基本的基于误差的模型进行修正，将定位问题归结为以最小化修正的误差平方和为目标的无约束多维非线性优化模型（问题一）。核心算法采用 Nelder-Mead 单纯形搜索算法，并引入随机扰动求解模型。在定位精度方面，单个坐标定位误差 5m 内的累积频率达到 0.9，三维坐标误差 20m 内的累积频率达到 0.8；在定位速度方面，外层循环 50 次时，定位一次耗时低于 0.75s（Intel Core(TM) i5-4590 CPU @3.30GHz）。

为确定可以实现近似最优定位的最少基站数（问题二），对近似最优进行了形式化描述，建立以最小化基站数为目标，以“近似最优”条件为约束的概率模型。基于蒙特卡洛思想，从统计意义上求解，发现参与定位的基站数小于 4 时，定位误差较大，随着基站数的增加，在 [6, 9] 的范围内，定位精度迅速提升，随后趋于稳定，取最少的实现近似最优定位的基站数为 7。

为了计算出运动终端的运动轨迹（问题三），首先调用所设计的定位算法计算同一运动终端在多个位置上的坐标，发现位置坐标散点图呈现出明显的集中曲线状分布，用二次曲线对散点图进行拟合，基于最小二乘法估计曲线参数，从而得到移动终端在二维平面上的移动轨迹。

基于统计学习方法尝试解决终端能否被定位的判别问题（问题四），将问题转化为二分类问题，结合对测量误差的分析，选择均值、方差、小于 200 的 TOA 数量、小于 250 的 TOA 数量、大于 300 的 TOA 数量和大于 380 的 TOA 数量共 6 个特征，采用朴素贝叶斯法，将所提供 samplecase 数据集作为训练集和测试集学习一个贝叶斯分类器，测试显示的分类正确率接近 80%，并用分类器对待判别数据进行判别。

最后，简要讨论了网络中平均连接度数与定位精度之间的关系。将平均定位误差作为精度的衡量指标，并对其做去量纲处理，以平均连接度数为自变量，以定位精度为因变量，采用类似于问题二的方法，计算它们之间的关系散点图。用指数函数拟合散点图，对关系进行定量刻画，所得到的拟合曲线为 $\omega = 42.12 \exp(-0.9064\lambda) + 1.079$ (ω 为去量纲化的定位误差)。

关键字： 无线基站 三维定位 无约束非线性优化 贝叶斯判别 曲线拟合

1. 问题描述

室内定位问题具有重要的商用价值，在工业界和学术界都是一个研究热点。本文关注基于运营商无线通信基站的方式对终端设备进行定位，并尝试解决的问题是：基于基站的测量信息，建立数学模型并设计算法计算终端在三维空间中的位置坐标。具体地，有如下四个方面的问题：

一、利用 10 组 LOS 或 NLOS 传播环境下从手持终端到基站的 TOA 测量数据和所有基站的三维坐标，计算出所有终端的三维坐标。

二、利用 10 组 TOA 测量数据和所有基站的三维坐标，设计算法，用尽可能少的基站数目实现近似最优的三维定位精度。

三、利用 5 组对移动终端采集到的 TOA 数据，设计算法计算出终端的运动轨迹。

四、由于基站的通信半径是有限的，所以只有在基站通信半径覆盖范围内的终端才有可能测到自身到基站的距离。而一个终端只有获得它与足够数目的基站之间的距离测量值，才能完成定位。如果假设每个基站的通信半径为 200 米（超过范围的测量数据无效），现需要根据给定的 5 组测量信息数据集，设计算法寻找出可以被基站定位的所有终端，并进一步分析终端的平均连接度数 λ 与定位精度之间的关系。其中 λ 定义为： $\lambda = \frac{\text{所有可以被定位终端到基站之间的连接数}}{\text{终端数}}$ 。

2. 符号说明

符号	含义
M	基站数
N	终端数
$\mathbf{x} = (x, y, z)^T$	待定位终端坐标
\mathbf{x}_i	第 i 个基站坐标
c	光速
r_i	根据 TOA 测量值计算的距离
β_i	修正模型的尺度因子
CDF	累积频率分布直方图
ERR	定位误差
ω	系统的定位精度（无量纲）

3. 问题分析

本节，对四个问题进行简要的分析，在下一节，我们将讨论对问题的具体建模。

3.1 对输入数据的分析

提供的基本数据是 TOA 测量值，除了问题三提供的数据之外，一个文件对应于一个场景下的测量数据，对于每一个终端，都有 20 到 60 个基站的一次测量值。一方面，注意到很多文献中讨论定位问题时，考虑的都是少量基站的定位，研究者往往假定测量值是准确并且 NLOS 情况比较少或者容易判别出这种情形 (乃至重构 LOS 数据) 的，这都是比较理想的假设，而实际场景很复杂，存在各种误差，对我们所要研究的室内定位问题，更不能对 NLOS 情况抱有乐观的假设；另一方面，这里并没有一个终端与一个基站之间的连续信号的测量值，所以应重点考虑从多基站的测量结果入手来设计求解终端位置的算法。

3.2 建模方向

问题一

问题一是我们要研究的关键问题，也是解决后续所有问题的基础。讨论无线定位的文献资料非常多，有关于 GPS 定位的，也有基于 WIFI 定位的，以及我们关注的基站定位；定位的方法也有很多，主要有：基于 TOA (Time of Arrival, 到达时间)、TDOA (Time Difference of Arrival, 到达时间差)、AOA (Angle of Arrival, 到达角度)、信号强度、载波相位，以及将上述多种参量相结合的融合定位方法等。在基于 TOA/TDOA 的定位方式下，讨论得最多、理论也最成熟的建模方法是利用几何关系的建模（球面相交、双曲面相交等），经典求解算法主要有基于最小二乘估计的 Chan 氏算法、泰勒展开法和基于模型线性化和高斯假设的扩展卡尔曼滤波法 [3]。现有的许多文献和学位论文主要是对这几种方法的改进和结合，近些年，随着统计学习方法的发展，一些新的机器学习算法也被应用到定位计算中，比如粒子滤波算法、神经网络等。从最优化理论的角度看，定位问题最后归结为一个非线性优化问题。一方面，我们可以基于经典的方法进行改进；另一方面，也可以尝试统计学习算法以及智能算法等。

建模时，一个重点需要考虑的因素是误差分析。误差来源主要来源于两个方面，其一是测量误差，主要来源于测量仪器，如终端和基站的时钟不同步以及噪声的影响；其二是信号传播特性带来的误差，它不随测量仪器的精度变化而改变，主要来源有多径效应、非视距传播 (NLOS)、多址干扰等。当然，考虑主要因素而忽略次要因素也是必要的，考虑过多的影响因素将会使模型变得复杂而难以求解。

问题二

问题二要求设计算法，使用尽可能少的参与定位的基站数目来实现近似最优的三维定位精度。显然，为了顺利解决该问题，问题一中所建立的模型和求解算法至关重要。当然，首要的问题是如何定义“近似最优”这个模糊的概念，可以认为它是一种状态，在该状态下，再增加基站数也不会显著增加定位精度；也可以认为它是对于定位误差（其定义见公式 (2)）而言的一个阈值，定位误差低于这个阈值，就可以认为是“近似最优”的，比如这个阈值取 70m。

我们有两种思路：一种思路是从优化模型的角度入手，建立以最小化基站数为目标、以近似最优的三维定位精度为约束的优化模型。难点是模型的求解，虽然目标函数很简单，但是可以预见约束非常复杂。而且，决策变量似乎也难以明确。所以可以尝试从另一种思路入手，即基于统计规律确定最少的基站数。依据提供了终端真实坐标的验证数据集，我们对于每个场景下的多个终端（或者所有终端），按照一定步长或者从较小的值开始依次增加参与定位的基站数，并运行问题一中设计的定位算法，根据得到的估计坐标与真实坐标计算出定位误差，然后统计出定位误差随着参与定位基站数量增加的变化趋势。一个理想的情况当然是，随着基站数量的增加，定位误差会

在某个基站数或某一段基站数范围内突变，即由较高的定位误差迅速降低至较低的水平，这样就便于我们从统计意义上确定最少的基站数；而不太理想的情况是，随着基站数的增加，定位误差的降低比较平缓，这就可能有两个原因：其一是客观原因，即在基于基站进行室内定位的方式下，近似精确定位的基站数本来就是很多，这个原因的可能性不大，除非测试数据存在较大的误差以至于难以使用；第二个原因就是所采用的定位算法有问题，需要回到这一环节考虑修正模型或设计求解精度更高的算法。

问题三

问题三需要根据给定的对运动终端的 TOA 测量数据，设计算法计算出终端的运动轨迹。对于该问题的解决思路，有几个需要注意的地方。

其一，是对所提供数据的理解。要注意到问题 3 的 TOA 测量数据矩阵的含义与其他几个问题有所不同。在其他几问中，TOA 矩阵的含义是每一行都对应一个不同的终端，每一行的数据都是与不同基站之间的测量结果，且并不要求一定是同时测量的（虽然是否同时测量并不影响前面问题的分析）；而第三问中，一个文件中的 TOA 矩阵记录的始终是同一个终端在移动过程中位于多个位置上的测量值，矩阵的每一行也是终端在某个位置上与多个基站之间的测量结果，可以认为是一次采样，且一行的数据应该是同时测量的。注意所提到的“采样”的含义也不同于我们通常所说的连续采样，这里一行的数据并没有时间上的先后关系，即并不是某一个特定基站与该终端连续多次测量的结果，而是一组基站同时对该终端一次测量的结果。这意味着我们在问题一中设计的定位算法在本问题中依然适用。

其二，是对所要求呈现的结果的理解。由于问题三的 TOA 矩阵中一行的数据并没有时序关系，虽然相邻两行的数据可能存在时序关系，但是我们并不清楚相邻两行的采样时间间隔是多少，所以无法估计出运动终端在移动过程中的速度。因此，我们认为问题三是需要解决的是静态轨迹拟合问题。即，首先根据 TOA 测量数据，运行所设计的定位算法，对每一次采样估计一个位置。之后用所有估计的位置画散点图，由于数据是对同一个终端运动过程中多次采样的，散点图应该呈现出某种连续的曲线形态，但由于定位误差的原因，散点图会呈现出一定的粗细和分布，所以须要对其进行曲线拟合，以得到轨迹方程。

最后，就是注意到问题三的三个场景下的数据均是在二维平面，所以降低了定位和拟合的难度。

问题四

问题四的描述中，告诉了我们一个事实：基站的通信半径都是有限的，对于某个基站，终端只有在其通信半径覆盖范围内的 TOA 测量值才是有效的，若超过范围，虽然有测量数据，但是无效。现在问题中假设每个基站的通信半径覆盖范围是 $200m$ 。

我们要解决的问题就是设计算法寻找出给定数据集中可以被基站定位的所有终端。

现在要判断一个终端能否被定位，需要满足的条件是：终端在足够数目的基站通信半径覆盖范围内。我们顺着条件往下分析，首先，“足够数目”到底是多少数目？它和我们在第二问中所求的“可以实现近似最优定位的最少基站数”有什么关系？另一个问题是，如何尽可能准确地判断终端是否在基站的通信半径覆盖范围内？这似乎是个不可解的问题。所以我们可以尝试换种思路，比如在统计意义上对能否定位进行判断，利用所提供的验证数据集，我们尝试分离出能够被定位和不能被定位的终端（当然前提是对之前所述的“足够数目”有一个估计），获取它们 TOA 数据的统计特性比如分布特性，然后在问题四的数据集中，根据终端 TOA 数据的统计特性，结合之前所获得的先验信息，对该终端能否被定位做出判断。但是这种方式过于粗略，且主观因素较多。所以我们可以尝试机器学习，在 TOA 数据中挖掘一些特征，并用一些常用的机器学习算法建立一个模型（可以用 sample_case001-005 来训练与测试模型），然后用模型对问题所要求处理的数据进行判断。

最后，问题四还有一个更进一步的问题：讨论“连接度数” λ 与定位精度的关系。首先要理解“连接度数”和“定位精度”这两个物理量的含义。连接度数被定义为：所有可以被定位终端到基站之间的连接数与网络中所有终端数量的比值。它是一个系统层面的概念，反映的是整个网络中每个终端的平均连接数，可以认为反映的是系统内平均每个终端能够获得的连接资源。可以这样形象地考虑：如果平摊到每个终端上的连接数多了，也就意味着平均每个终端获得的通信资源多了（通信更畅通了），自然地，定位精度也就提高了。当然，这里的“定位精度”应从系统层面上来理解。我们可以从对这两个物理量进行量化入手（连接度数已经有了量化方式），如果难以直接分析它们之间的内在联系，可以通过数值实验来发现关系，比如将它们的散点图画出来，进行曲线拟合等。

4. 数学模型与求解

4.1 定位精度的评价模型

在讨论定位模型之前，我们先简要介绍一下对定位精度的评价，以便于我们考察所建立的模型和求解算法的性能。

通常采用的指标有：均方误差 (MSE)、均方根误差 (RMSE)、克拉美罗下界 (CRLB)、定位误差与累积分布函数。

运行算法，对目标进行 N 次位置估计，设第 i 次估计的位置坐标为 (x_i, y_i, z_i) ， N 次估计的平均值为 $(\bar{x}, \bar{y}, \bar{z})$ ，则均方误差定义为：

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N [(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2] \quad (1)$$

均方根误差是 MSE 的开方，即 $\text{RMSE} = \sqrt{\text{MSE}}$ 。

CRLB 在信号检测与估计中有广泛地应用，主要是用来确定一个无偏估计的下界。将算法得出的 MSE/RMSE 与 CRLB 比较，就可以对算法的性能与最佳性能的差距有一个定量的认识。这里不再详细罗列 CRLB 的推导。

均方误差与 CRLB 在理论分析中用得较多，而工程上采用得更多的是用定位误差与累积分布函数作为衡量定位精度的指标。定位误差即估计值与真实值的偏差：

$$\text{ERR} = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2 + (\hat{z} - z)^2} \quad (2)$$

其中 $(\hat{x}, \hat{y}, \hat{z})$ 表示估计值， (x, y, z) 表示真实值。定位误差累积分布函数指定位误差在某个精度门限以下的概率。

本文中，我们主要采用定位误差，并结合误差直方图与累积分布函数来验证定位算法的精度。

4.2 NLOS 误差抑制模型

NLOS 问题是目前基站定位系统提高定位精度必须要解决的关键问题，即便提高系统接收机的 TOA 测量精度，也无法避免 NLOS 误差，它仅与电波的传播环境有关，而与基站网络的类型无关。

通常，有两种处理 NLOS 误差的角度。一种是先测量数据进行预处理，消除 NLOS 误差或重构 LOS 值，再用普通的定位算法求解；另一种是把 NLOS 误差带入模型中，通过一些处理手段和算法降低其影响。前面已经提到，除了第三问，提供的数据中，一个终端和一个基站之间只有一个测量数据，所以试图直接消除 NLOS 误差的思路有难度，所以，我们考虑直接将误差项引入模型中，并通过技术手段降低其影响。

基本模型

设参与定位的基站 (BS) 数为 M ，将基于 TOA 的测量过程建模为：

$$r_i = d_i + L_i + \nu_i, \quad i = 1, 2, \dots, M \quad (3)$$

其中， $r_i = \text{TOA}(i) \cdot c$ 是第 i 个基站到终端 (MS) 的测量距离， d_i 是 LOS 的传播距离， L_i 是由于 NLOS 引起的误差， ν_i 是由其他因素如系统的测量噪声、时钟同步误差等导致的误差累积项。设待定位终端的坐标是 $\mathbf{x} = (x, y, z)^T$ ，第 i 个基站的坐标是 $\mathbf{x}_i = (x_i, y_i, z_i)^T$ 。

定义终端对第 i 个基站的定位残量为

$$e_i(\mathbf{x}) = r_i - \|\mathbf{x} - \mathbf{x}_i\| \quad (4)$$

其中 $\|\cdot\|$ 是范数的运算符号，这里我们只采用二维范数，因此 $\|\mathbf{x} - \mathbf{x}_i\|$ 表示坐标 \mathbf{x} 与 \mathbf{x}_i

的欧氏距离。在基于误差最小的估计方法中，通常采用的决策模型框架是：

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \sum_{i=1}^M \rho(e_i(\mathbf{x})) \quad (5)$$

其中 $\rho(\cdot)$ 是关于残量的一个最优准则函数，在加权最小二乘法中，采用的准则函数就是 $\rho(e_i(\mathbf{x})) = \sum_{i=1}^M (\alpha_i e_i(\mathbf{x}))^2$ ，其中 α_i 是权重，反映基站重要程度。

修正的模型

如果直接采用上述基本模型，如加权最小二乘估计，一个须要仔细考虑的方面是如何衡量基站的权重，可以认为距离终端越近、信号强度越大（可以通过测量数据进行粗略的定性估计）的基站具有较大的权重，但是在室内环境下，加上 NLOS 的复杂情况，距离终端越近的基站的测量数据也不一定准确，甚至可能由于距离近，反而受测量误差的影响更大。所以直接尝试赋予权值是个困难的过程。另外，也可以考虑将 M 个权重变量作为自变量带入模型中，通过算法去估计一组参数（比如，可以采用机器学习中经典的 SMO 序列最小最优化算法），但是由于多引入了 M 个变量，而基站的数量又比较多（不小于 20 个），这就增加了模型的求解难度和算法收敛速度，而且并不能很好地解决 NLOS 干扰较强的情形。为了更好地处理 NLOS 传播误差，我们从另一个角度入手：尝试对前述残量函数进行修正 [1]。

在存在 NLOS 传播的情形中，一个事实是 NLOS 误差会导致测量距离比真实直线距离大得多。即如下不等式成立：

$$e_i(\mathbf{x}) = r_i - \|\mathbf{x} - \mathbf{x}_i\| \geq 0 \quad (6)$$

引入松弛变量（尺度因子） β_i ，将上述不等式变为等式：

$$\beta_i r_i - \|\mathbf{x} - \mathbf{x}_i\| = 0 \quad (7)$$

β_i 满足 $0 \leq \beta_i \leq 1$ 。而且， β_i 存在一个下界，以一个简化的二维示意图为例来进行说明：如图1所示，三个基站参与定位，由于存在 NLOS 传播，导致三圆并不交于一点（三维空间中则是三个球面），而是交出一个区域，终端的位置就在这个区域中。设 η_i 是 β_i 的下界，则在图中， $\eta_1 = |O_1 I_1|/r_1$ ， $\eta_2 = |O_2 I_2|/r_2$ ， $\eta_3 = |O_3 I_3|/r_3$ 。于是， β_i 的范围是：

$$\eta_i \leq \beta_i \leq 1, \quad i = 1, 2, \dots, M \quad (8)$$

现在我们分析 LOS 情形，使得上面的讨论能够处理到该情况。当传播是 LOS 时， r_i 接近真实直线距离，使得 $r_i - \|\mathbf{x} - \mathbf{x}_i\|$ 接近于 0，但是由于可能存在的时钟同步误差，该式子可能出现小 0 的情况，这时为了使等式 (7) 成立， β 就会大于 1，但是不会大得很多，因为相比 NLOS 误差，测量误差要小得多，我们用符号 δ_i 表示这个不大的

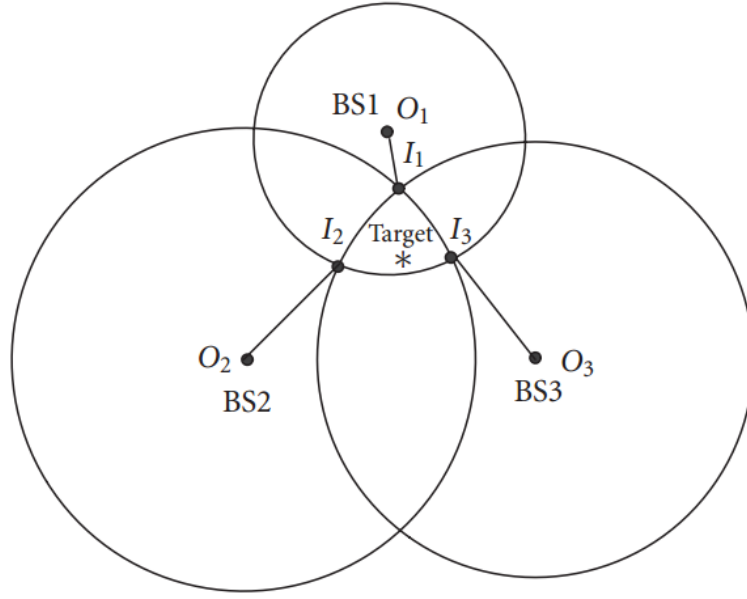


图 1 β_i 下界存在性示意图.

量，于是，将 β_i 的范围进一步修正为：

$$\eta_i \leq \beta_i \leq 1 + \delta_i, \quad i = 1, 2, \dots, M \quad (9)$$

同时，将残量函数修正为：

$$e'_i(\mathbf{x}) = \beta_i r_i - \|\mathbf{x} - \mathbf{x}_i\| \quad (10)$$

如果对 β_i 都取得比较精准，那么 e'_i 会非常接近于 0。于是，我们须要最小化的目标函数即为修正后残量函数的平方和：

$$g(\mathbf{x}) = \sum_{i=1}^M (e'_i(\mathbf{x}))^2 \quad (11)$$

决策即：

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \sum_{i=1}^M (e'_i(\mathbf{x}))^2 \quad (12)$$

综上，将修正后的定位模型整理如下：

$$\begin{aligned} & \text{minimize} \quad g(\mathbf{x}) = \sum_{i=1}^M (\beta_i r_i - \|\mathbf{x} - \mathbf{x}_i\|)^2 \\ & \text{subject to} \quad \eta_i \leq \beta_i \leq 1 + \delta_i, \quad i = 1, 2, \dots, M \end{aligned} \quad (13)$$

其中， η_i 是 β_i 的下界， δ_i 是一个不大的量。

求解算法

根据前面的分析，三维定位问题归结为对多维非线性函数的极值求解问题。但是(13)的求解并不容易。直观上来看，如果取到了精确的 β_i 和位置坐标 \mathbf{x} ， $g(\mathbf{x})$ 将等于0，但是，对 β_i 的精确估计并不容易。我们尝试过将 β_i 作为自变量直接求解模型，试过二次规划算法（比如序列二次规划），以及一些基于偏导的算法（如拟牛顿法），发现它有非常多的局部极值，有时这些局部极值与真实位置相差很大；而且在基于偏导的算法中，由于要计算多维函数的偏导，使得算法的收敛速度比较慢，求得的结果也并不理想（即便不将 β_i 作为自变量也是如此）。

另外，我们也尝试过文献[1]提出的数值迭代搜索算法，实验发现求解精度比较低，而且对初始位置的设置很敏感，搜索步长的设置也会对结果产生较大影响。虽然在文献中，研究者指出该算法的性能比较好（均方根误差低），但是在我们所获得的实测数据中的表现并不理想。分析其原因，主要应该在于实际环境的复杂，即实测数据是室内环境，数据的测量受到各种干扰的影响，而且实测数据的特性往往比研究者的仿真数据也要复杂得多。

但是，我们可以采用文献[1]中算法对 β_i 的处理技巧，以及算法大框架，而替换其中核心的迭代搜索模块，尝试用其他高效的搜索算法求解上述多维非线性优化问题。我们设计的算法如 Algorithm 1所示。

我们的算法核心搜索模块采用 Nelder-Mead 单纯型搜索算法 [2]。这是我们经过许多尝试而最终选择的算法，该算法的主要优点是：（1）不依赖于梯度，可以处理不连续的高维函数；（2）收敛速度快。在我们的计算机上（处理器：Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz，操作系统：64 位 Windows 10 操作系统），Matlab 中运行一次 Nelder-Mead 单纯型搜索算法仅需要约 15ms。另外，基于蒙特卡洛思想，我们在核心算法的外层加上一个大循环，通过多次运行，同时对 β_i 进行扰动，以增强全局寻优能力。

试验发现外层循环 50 次即可达到不错的精度，这样对终端进行一次定位的时间只要 0.75s。另外，我们在试验中也发现，如果外层循环次数（蒙特卡洛投点次数）足够大（比如取 200-500），那么算法的求解精度会非常高，对 x 坐标和 y 坐标的误差会在 2m 以内， z 坐标的误差稍大，但也在 4m 以内。但是这样一次定位的时间就会比较长，是秒量级的。所以，要在循环次数与求解精度之间做一个权衡。我们建议循环次数取 100。

求解结果与分析

实验环境为 Matlab R2015a。求解程序见附录附录 A。首先，我们根据现有的数据集 sample_case001-sample_case005 进行实验以验证算法的性能。

如前所述，我们采用定位误差 (ERR)，并结合误差直方图与累积分布函数来验证算

Algorithm 1 基于 Nelder-Mead 单纯型搜索的定位算法

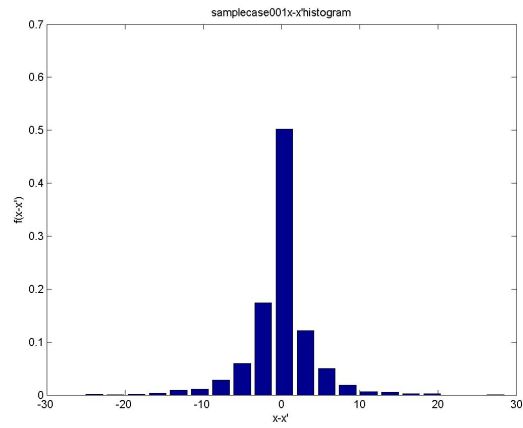
```
1: 设置运行次数  $T$ ，默认取 100。初始化循环计数器  $k \leftarrow 1$ ，初始化当前找到的目标函数最小值  $\text{minObj} \leftarrow \infty$ ，待定位终端坐标记为  $\mathbf{x}$ 。  
2: while  $k \leq T$  do  
3:   设置  $\eta_i, i = 1, 2, \dots, M$ ，默认取值 0.5。  
4:   建立一个随机变量  $\epsilon$ ，其作用是扰动因子，它的取值比较小，且服从某种分布，默认服从标准正态分布。  
5:   按右式生成  $\beta_i$ :  $\beta_i = \eta_i + (1 - \eta_i)\epsilon, i = 1, 2, \dots, M$   
6:   使用 Nelder-Mead 单纯型法搜索  $g(\mathbf{x}) = \sum_{i=1}^M (e'_i(\mathbf{x}))^2$  的极值点，初始解设为所有基站的几何重心，即  $(\frac{1}{M} \sum_{i=1}^M x_i, \frac{1}{M} \sum_{i=1}^M y_i, \frac{1}{M} \sum_{i=1}^M z_i)^T$ 。记算法返回的解为  $\mathbf{x}_{\text{Nelder-Mead}}$ ，对应的目标函数值为  $\text{objVal}$ 。  
7:   if  $\text{objVal} < \text{minObj}$  then  
8:     更新:  $\text{minObj} \leftarrow \text{objVal}, \mathbf{x} \leftarrow \mathbf{x}_{\text{Nelder-Mead}}$ 。  
9:   end if  
10:   $k \leftarrow k + 1$ 。  
11: end while  
12: return  $\mathbf{x}, \text{minObj}$ 
```

法的定位精度。分别对 5 个场景下的数据集运行算法，对每个终端位置进行定位，之后根据定位结果和真实位置计算每个坐标下的误差以及均方根误差，然后做分布直方图和累积分布图（CDF）。

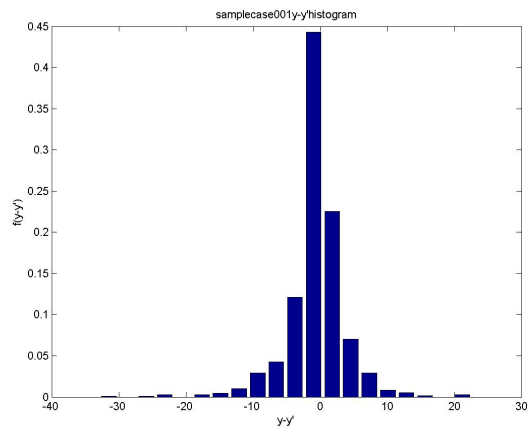
限于篇幅，这里只给出 sample_case001 的统计结果，其他四个场景的统计结果请见附件文件夹。首先是直方图，我们分别统计了 x, y, z 的误差、二维坐标 (x, y) 的定位误差以及三位坐标 (x, y, z) 的定位误差。

如图2所示，从单个坐标误差直方图可以直观看出，无论是哪个坐标，对绝大部分终端的误差都分布在 0 两侧，大体呈现出正态分布，而且方差较小，绝对误差超过 10m 的比率低于 0.05。另外，从图3 的定位误差直方图可见（由于定位误差都为正值，所以直方图只分布在正半轴），无论是只考虑二维 (x, y) 的定位误差还是三维空间的 (x, y, z) 定位误差，误差超过 40m 的比率也都低于 0.05，且呈现半正态分布的形态。一个值得注意的点是三维坐标的定位误差直方图没有二维坐标的定位误差直方图集中，这也体现了三维空间定位的难度比二维平面定位要高。要总的来说，我们对各个坐标的估计接近无偏估计。

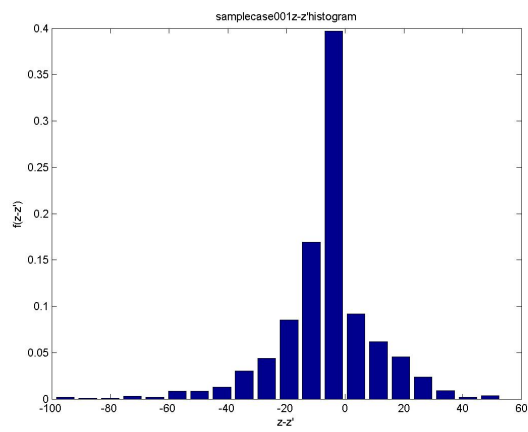
接下来是累积分布图，单个坐标误差和定位误差的 CDF 分别如图4和图5所示。CDF 更进一步地反映了误差的分布情况。对单个坐标而言， x 和 y 的绝对误差在 5m 以内的累积频率都达到 0.9， z 则相对低一些，5m 内绝对误差的累积频率约为 0.7。说明



(a)

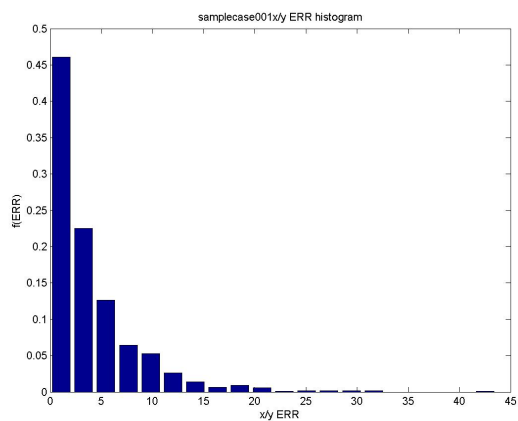


(b)

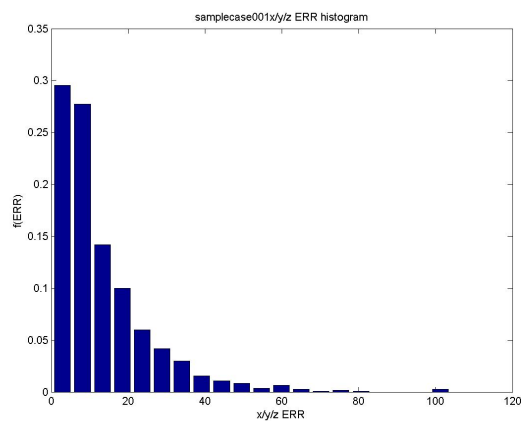


(c)

图 2 sample_case001 坐标误差直方图: (a) x 坐标; (b) y 坐标; (c) z 坐标.

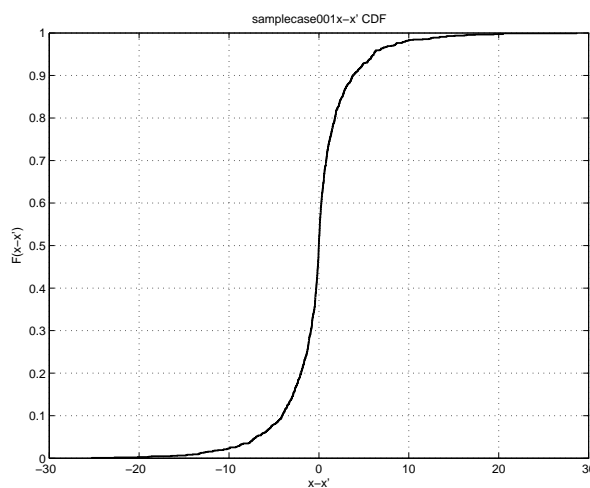


(a)

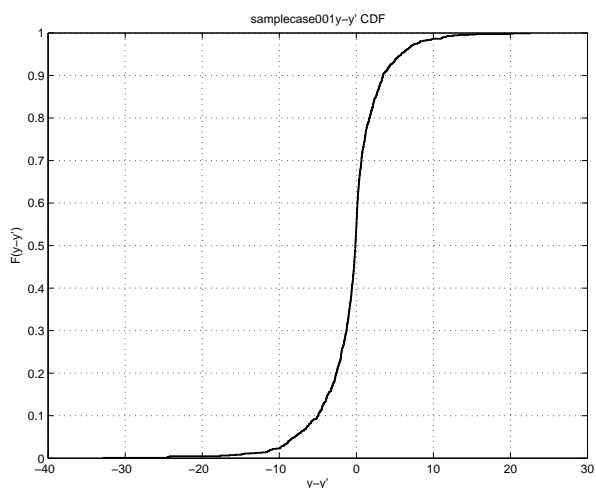


(b)

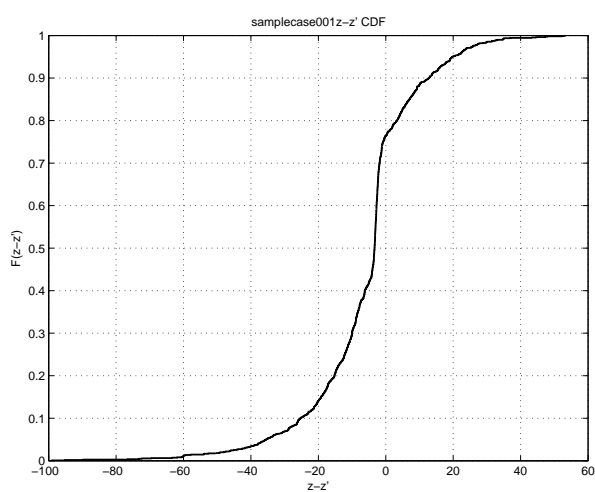
图3 sample_case001 定位误差直方图: (a) (x, y) ; (b) (x, y, z) .



(a)



(b)



(c)

图4 sample_case001 坐标误差 CDF: (a) x 坐标; (b) y 坐标; (c) z 坐标.

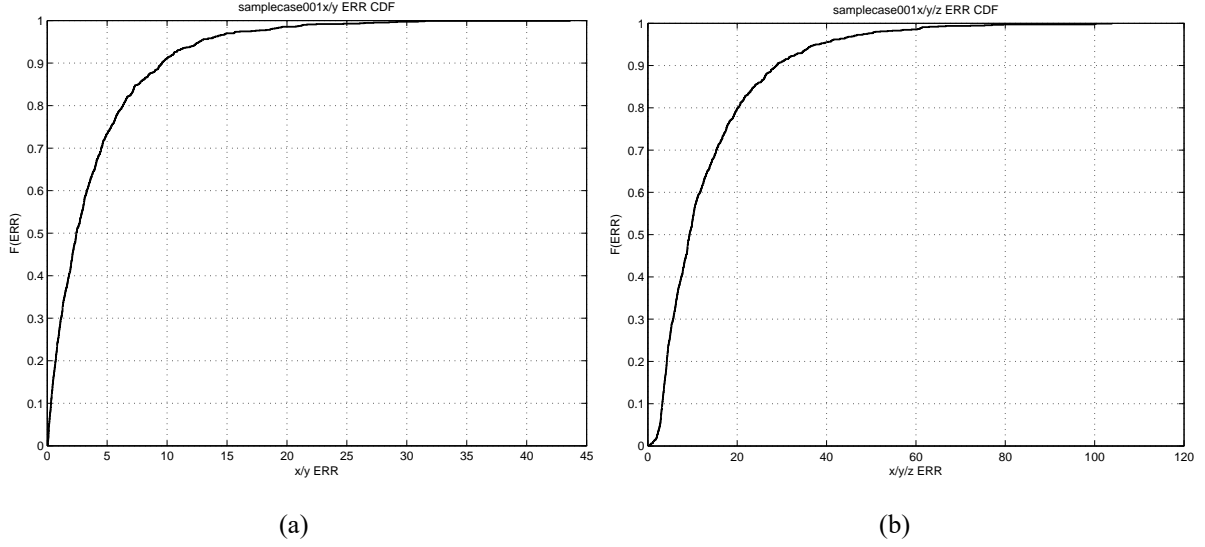


图5 sample_case001 定位误差 CDF: (a) (x, y) ; (b) (x, y, z) .

对 z 坐标的定位难度较大，这可能是由于复杂的环境状况导致求解的误差比其他坐标大。另外，二维定位误差 10m 以内的累积频率达到了 0.9，三维定位误差则在 0.6 左右，同样反映了三维定位的难度。总地来看，定位误差的 CDF 曲线随着误差的增加先快速增加，而后缓慢增加，直到 1，体现了我们求解算法的有效性和较高的可靠性。

其他四个场景的结果与该场景相似，这体现了所设计的定位算法的稳定性。

最后，我们对问题一所要求的 10 组测试场景运行定位算法，求得的结果请参见附带的.txt 文件（output_case_001.txt-output_case_010.txt）。

4.3 最少定位基站的概率模型

要确立可以实现近似最优定位的最少基站数量，一个首要问题是对“近似最优”这个模糊概念的定义，我们将它作如下的形式化定义。

定义 1 (近似最优) 近似最优，是一种状态，该状态下有 $m(m \geq 3)$ 个参与定位的基站，平均定位误差小于某个阈值 ERR_{thr} ，且再增加 m ，定位误差不再发生显著变化。

说明：上面采用的阈值符号，“ERR”是定位误差的符号，其定义见公式 (2)，下标“thr”是英文单词 threshold 的简写。结合问题一中算法的求解结果，我们定义 $ERR_{thr} \leq 70$ （单位：米）。是否发生“显著变化”可以利用假设检验来判别，也可以根据定位误差的一些统计量如方差是否小于某个水平来判别，具体操作方法要依据实验结果而定。

数学模型

在“问题分析”中，我们提到，可以尝试建立以最小化基站数为目标、以近似最优的三维定位精度为约束的优化模型，现在我们给出该模型的形式化描述。

目标函数记为基站数 m 。约束要求实现近似最优的定位精度，依据前面我们对

“近似最优”的定义，有两个层面的约束条件：一是精度，而是稳定性。对于精度的约束即：

$$E[\text{ERR}(m)] < \text{ERR}_{thr} \quad (14)$$

其中 $E[\text{ERR}(m)]$ 的含义是“在参与定位的基站数为 m 时定位误差的期望”， $\text{ERR}_{thr} \leq 70$ 。对稳定性的约束，如果从方差角度来刻画，描述如下：

$$D[\text{ERR}(\tau)|\tau > m] < \xi \quad (15)$$

其中， $D[\text{ERR}(\tau)|\tau > m]$ 的含义是“当基站数大于 m 时定位误差的条件方差”， ξ 是某个可以接受的方差阈值。

综上，我们的模型是：

$$\begin{aligned} & \text{minimize} \quad m \\ & \text{subject to} \quad E[\text{ERR}(m)] < \text{ERR}_{thr} \\ & \quad \quad \quad D[\text{ERR}(\tau)|\tau > m] < \xi \end{aligned} \quad (16)$$

求解方法与结果分析

显然，上面所建立的模型是一个概率模型，无法直接采用经典的优化算法进行求解。我们采用统计手段进行求解。

一种简单自然的思路是，对一个文件的数据，遍历所有的终端，对每个终端，再枚举所有的基站数，运行定位算法，统计出每个基站数下定位误差的平均值，最后对所有终端每个基站数下的定位误差平均值取平均，以基站数为自变量，以定位误差平均值为因变量，连接所有的数据点，就能得到定位误差随着基站数变化而变化的关系曲线图。但是这种方法不可取。因为对一个基站数，尝试枚举所有基站组合，是阶乘量级的，举个例子，一共有 20 个基站，以参与定位的基站数为 10 为例，就有 $C_{20}^{10} = 184756$ 种，即使一次定位速度达 $0.1s$ ，也要约 5.13 小时，这显然是无法接受的，何况还有更多的基站数（最多有 60 个），以及还要遍历所有的终端（至少 1000 个）。

基于上面的考量，我们必须采用其他的搜索方式。这里，我们采用系统抽样的方法对上面操作的计算复杂性进行优化。主要有两点：一、不遍历所有的终端，虽然终端数并不很多，但是在统计意义上，遍历所有终端数没有必要，也会成倍增加算法耗时。所以我们拟采取从所有 1000 多个终端中随机抽样一定数量的终端（比如，100 个），对这些终端进行考察；二、不遍历所有基站组合。基本的思路还是基站数逐渐增加，但是对每个基站数（设为 m ），不采取枚举所有的组合然后求平均的方法，而是在所有基站中随机取 m 个（或者就取前 m 个）进行一次定位计算。这样可以大大提高计算速度。

看起来上述方法的随机性很大，但是经过我们的试验发现，结果具有很好的稳定性。本质上，我们的方法也属于蒙特卡洛方法。我们在 5 个验证数据集上执行以上算

Algorithm 2 系统抽样算法

- 1: 设置抽样次数num。
 - 2: 随机抽选一个终端。
 - 3: 根据总基站数，按合适步长增加参与定位的基站数 m ，对每个 m ，调用问题一的定位算法求解终端位置，并计算定位误差。
 - 4: 执行第 2-3 步num次，得到所有抽样终端在若干基站数下的定位误差。对每个基站数，计算所有抽样终端定位误差的平均值，作为对该基站数下定位误差的估计。
-

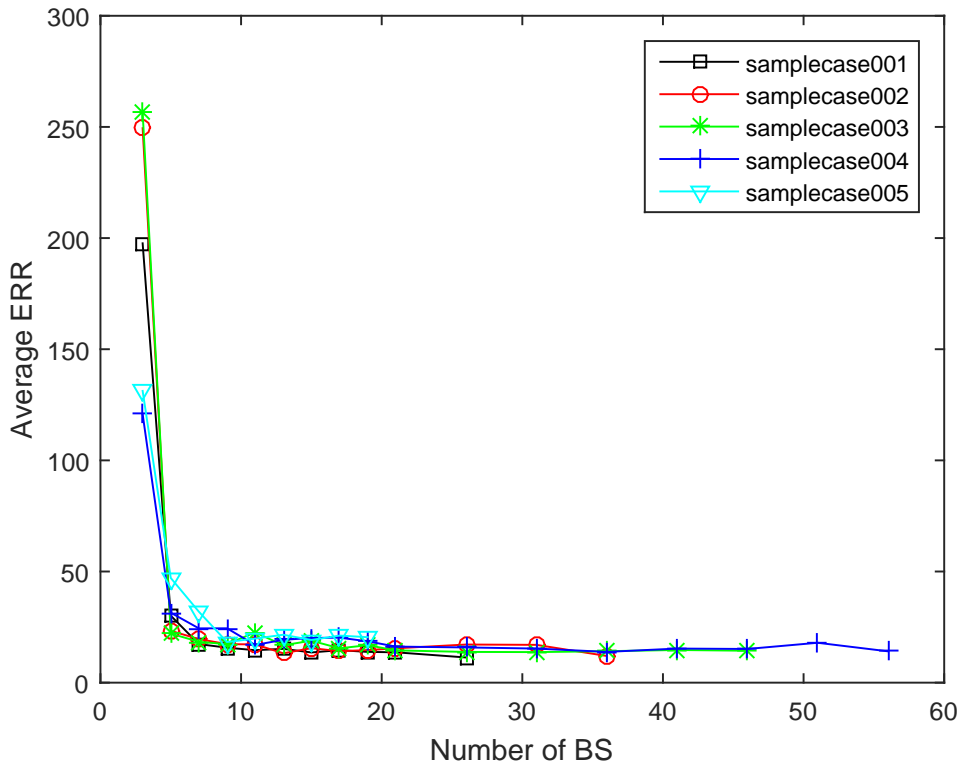


图 6 平均定位误差随参与定位基站数变化曲线

法，将得到的平均定位误差随基站数增加而变化的曲线画在一张图里，如图6所示（求解程序见附录）。

前面我们提到，理想的情况是，随着基站数量的增加，定位误差会在某个基站数或某一段基站数范围内突变，从较大的定位误差迅速降低至较低的误差。而现在图6所呈现的结果正是我们所期望的理想情况。虽然由于抽样数较少，且每一个抽样终端下对应的一个基站数的定位误差只是随机地计算了一次，使得画出的曲线并不光滑，但是整体走势非常清晰明了。而且 5 个不同场景下的曲线呈现出及其相似的形态，即：当参与定位的基站数小于 4 时，定位误差很大；而当基站数超过 5 后，在从 6 增加到 10 的过程中，定位精度迅速提升，且之后随着基站数的增加，精度的提升变得不再明显。这样的结果说明了客观上确实存在一个较小的基站数，使得定位精度接近最优，

表 1 拟合的方程和相关统计量

Case	拟合函数	R-square	RMSE
Case021	$y = 0.003977x^2 - 0.5036x + 4.014$	0.9995	3.4224
Case022	$y = 0.003989x^2 - 0.5011x + 4.734$	0.9998	3.1926
Case023	$y = 0.003994x^2 - 0.4999x + 9.391$	0.9999	2.2339
Case024	$y = 0.00397x^2 - 0.5047x + 12.77$	0.9993	5.5323
Case025	$y = 0.003995x^2 - 0.5009x + 9.411$	0.9996	4.8511

而且不随场景的变化而有剧烈变化。

由于定位精度迅速提升的过程对应的基站数大约在 $[6, 9]$ 的范围，我们不妨取这个最小的能实现近似最优定位的基站数为 7。

4.4 运动轨迹重建模型

在关于第三问的“问题分析”中，我们提到的思路是：首先根据第三问对应的 TOA 测量数据，运行我们的定位算法对每一次采样（TOA 矩阵每一行）计算出一个估计位置。之后根据所有估计的位置坐标画散点图，观察散点图的形态，并选择合适的曲线模型进行拟合，得到轨迹方程。

对所提供的 5 个文件分别调用定位算法进行定位，并保存定位结果。将结果画成散点图的形式，通过观察，我们发现所有散点图都有明显的轨迹特征，且趋势较为平缓，有一定的曲率，因此我们拟采用二次曲线模型，用线性最小二乘法进行拟合。

4.5 基于线性最小二乘法的二次曲线拟合

设我们的定位算法对一个场景下所有采样的定位坐标依次为 $(x_i, y_i), i = 1, 2, \dots, N$ ，我们要拟合的曲线模型为 $y = \phi(x)$ 。一条曲线拟合得“好坏”的评价标准通常有如下几个指标：偏差绝对值 $\sum_{i=1}^N |\phi(x_i) - y_i|$ ，偏差绝对值的最大值 $\max_i \sum_{i=1}^N |\phi(x_i) - y_i|$ ，以及偏差平方和 $\sum_{i=1}^N |\phi(x_i) - y_i|^2$ 。都是通过使上面三式最小化来实现曲线拟合的。

我们采用偏差平方和作为要最小化的准则函数，选择二次函数 $y = ax^2 + bx + c$ 作为模型，其中 a, b, c 是我们要估计的参数，建立如下的无约束模型：

$$\underset{a,b,c}{\text{minimize}} \quad \sum_{i=1}^N |(ax_i^2 + bx_i + c) - y_i|^2 \quad (17)$$

最小二乘法在曲线拟合中是非常常用的一种方法，关于其原理这里不再详述，可以用最小二乘法作为上述模型的求解算法。我们直接给出求解结果，如图7所示。

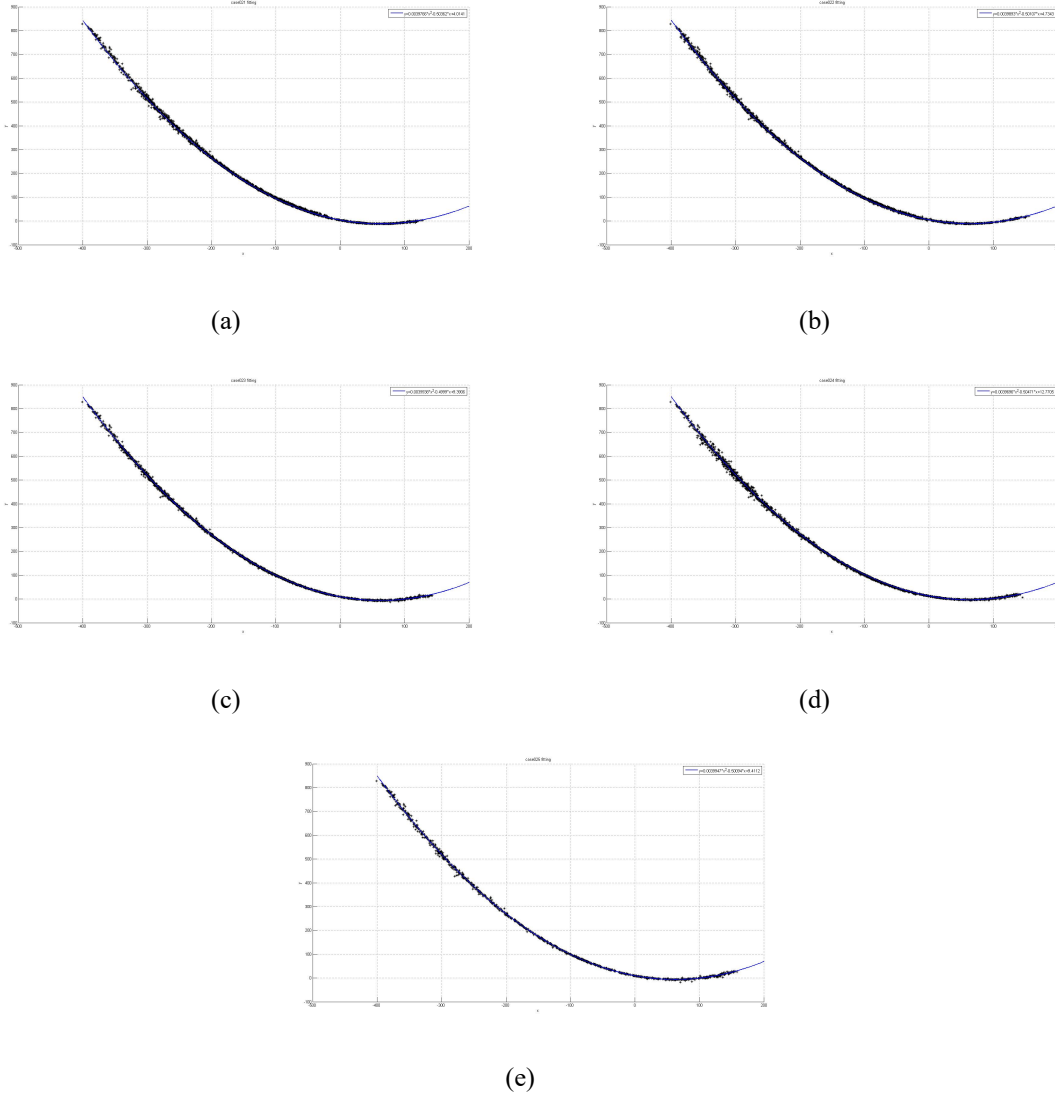


图 7 坐标误差直方图: (a) caseinput021; (b) caseinput022; (c) caseinput023; (d) caseinput024; (e) caseinput025.

五个场景下移动终端的拟合轨迹方程以及相应的统计量如表1所示，可见，R-square 都不小于 0.99，说明我们的拟合效果非常好。

4.6 基于统计学习方法的定位判别模型

TOA 特征提取

如“问题分析”中所述，我们拟采取建立统计机器学习模型的手段解决终端定位的判别问题。首先，需要提取合适的特征。由于我们唯一的获取特征的渠道就是对应于一个终端的若干个 TOA 数据，所以应尝试从 TOA 数据中挖掘特征。我们的基本思想是采用一些基本统计量如均值、方差，再加上和通信半径有关的量，比如小于或者大于某个阈值的 TOA 数据的数量。为了选择合适的阈值，我们作如下估算：

回顾一下第一问中我们建立的测量模型 $r_i = d_i + L_i + \nu_i$ ，其中， d_i 是 LOS 距离， L_i 是 NLOS 误差， ν_i 是测量误差等其他误差。如果终端位于通信半径内，那么现在有： $0 \leq d_i \leq 200, 0 \leq L_i \leq 120 + \delta, -60 \leq \nu_i \leq 60$ 。其中， $L_i \leq 120 + \delta$ 的理由是“由于 NLOS 导致的时延最高可能超过 400ns”， δ 是一个相比 120 较小的数； $-60 \leq \nu_i \leq 60$ 的理由是“由于时钟不同步引起的误差范围在 200ns 以内”。于是有 $-60 \leq r_i \leq 380 + \delta$ 。实测的 r_i 没有赋值，所以 380 是一个需要重点考量的大小，如果 r_i 的测量值大于了 380，我们有理由认为该测量数据是无效的可能性很大。所以我们选择一行 TOA 向量中“大于 380 的 TOA 数据点的数量”作为一个特征。另外，我们对 sample 数据集进行了一些简单的统计分析，选择“小于 250 的 TOA 数据点的数量”，“大于 300 的 TOA 数据点的数量”作为特征，再加上“小于 200 的 TOA 数据点的数量”。

综上，我们选择的 TOA 特征是：均值、方差、小于 200 的数量、小于 250 的数量、大于 300 的数量、大于 380 的数量。

确定了特征以后，还要选择数据集，包括训练数据集和测试数据集，当然应该采用所提供的 sample_case001-005。但是在尝试机器学习算法之前，还需要解决一个问题，如何确定 sample_case 中所有终端能否被定位？由于我们知道了 sample_case 中每个终端的真实位置，基于 200m 的通信半径假设就能判断所有 TOA 数据是否有效，但是还必须知道“至少几个有效地 TOA 数据才能实现定位”。在问题描述中，说法是“一个终端只有获得它与足够数目的基站之间的距离测量值，才能完成定位”，结合第二问的结果，参与定位的基站数不应小于 4，否则定位误差太大，而当基站数超过 6，就几乎能实现近似最优的定位，鉴于此，我们认为：如果一个终端同时在至少 5 个基站数的通信半径覆盖范围内，那么它可以被定位。于是，我们就能对训练和测试数据集的“ y ”进行赋值，并且令“能够被定位”取 1，“不能被定位”取 0。

基于朴素贝叶斯法的判别模型

本节，先简要介绍朴素贝叶斯法的基本概念及算法步骤，然后将该方法用于我们的数据集。

朴素贝叶斯 (Naïve Bayes) 法是基于贝叶斯定理与特征条件独立假设的分类方法。对于给定的训练数据集，首先基于特征条件独立假设学习输入/输出的联合概率分布；然后基于此模型，对给定的输入 \mathbf{x} ，利用贝叶斯定理求出后验概率最大的输出 y 。朴素贝叶斯分类器是一个典型的概率生成模型，其训练过程就是参数估计的过程，参数估计是基于训练数据集利用最大似然准则进行的 [4]。

约定如下符号：训练集 $D = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)$ ，其中， \mathbf{x}_i 是 n 维样本空间中的实例特征向量， $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})^T$ ， $x_i^{(j)}$ 是第 i 个样本的第 j 个特征，若 $x_i^{(j)}$ 是离散型变量，设 $x_i^{(j)}$ 的取值集合为 $\{a_{j,1}, a_{j,2}, \dots, a_{j,S_j}\}$ ， $a_{j,l}$ 表示第 j 个特征可能取的第 l 个值 ($j = 1, 2, \dots, n, l = 1, 2, \dots, S_j$)； $y_i \in \mathcal{Y} = \{c_1, c_2, \dots, c_L\}$ 是实例 \mathbf{x}_i 的标

记。于是朴素贝叶斯法的学习与分类过程的形式化描述如下：

Algorithm 3 朴素贝叶斯分类

- 1: 输入：训练集 D 和待分类实例特征向量 \mathbf{x} ;
 - 2: 输出：待分类实例 \mathbf{x} 所属的类标记 y .
 - 3: 计算类的先验概率 $P(Y = c_k) = \frac{1}{m} \sum_{i=1}^m I(y_i = c_k)$, $k = 1, 2, \dots, L$ 以及条件概率 $P(X^{(j)} = a_{j,l} | Y = c_k) = \frac{\sum_{i=1}^m I(x_i^{(j)} = a_{j,l}, y_i = c_k)}{\sum_{i=1}^m I(y_i = c_k)}$, $j = 1, 2, \dots, n, l = 1, 2, \dots, S_j, k = 1, 2, \dots, L$;
 - 4: 对于给定的实例 $\mathbf{x} = (x^{(1)}, x^{(2)}, \dots, x^{(n)})^T$, 计算 $P(Y = c_k | \mathbf{X} = \mathbf{x}) = P(Y = c_k) \prod_{j=1}^n P(X^{(j)} = x^{(j)} | Y = c_k)$, $k = 1, 2, \dots, L$;
 - 5: 确定 \mathbf{x} 所属类别 $y = \arg \max_{c_k} P(Y = c_k | \mathbf{X} = \mathbf{x})$, $k = 1, 2, \dots, L$
-

接下来，我们在所使用的数据集上运行朴素贝叶斯算法，以获得贝叶斯分类器。首先要获得特征矩阵 (\mathbf{X})：对 TOA 矩阵的每一行，计算出前面所述的 6 个特征，构成特征向量 (\mathbf{x}_i)；同时计算每一个终端的“标记值” (y 值)，即按是否有不少于 5 个有效的 TOA 数据点判断其能否被定位，如果是，则赋值 1，否则赋值 0。这样，对一个 $N \times M$ 的 TOA 矩阵，我们就能获得一个 $N \times 6$ 的特征矩阵以及 $N \times 1$ 的标记向量，可以用来输入模型。在我们所使用的 Matlab R2015a 实验环境中，提供了朴素贝叶斯算法的调用接口 `fitcnb()`，非常方便我们进行数值实验。

为什么选择朴素贝叶斯法，因为我们通过试验发现，该方法简单易实现，而且有着不错的分类正确率，另外，相比其他复杂的机器学习算法如 SVM，其判别速度也很快，对于一个文件中 1000 个终端的判别时间不到 0.5s。

一般而言，朴素贝叶斯法受训练数据集规模影响比较大，往往训练集规模越大，分类效果也越好。所以，我们选取 sample_case001-005 中的其中 4 个 (sample_case001, sample_case002, sample_case003, sample_case005) 作为训练数据集（共 4600 个样本），剩下一个 (sample_case004) 作为测试数据集（共 1400 个样本）。得到的结果是：对 sample_case004 的分类正确率达到了 79.43%。虽然这个分类正确性并不是很高，但是我们认为它是可以被接受的，说明了分类器的有效性。

最后，我们将获得的模型用于对改问题未分类数据的判别。具体的结果请参见文件包中相应的.txt 文件（结果的截图证明请参见附录??）。一些统计结果如表2所示。

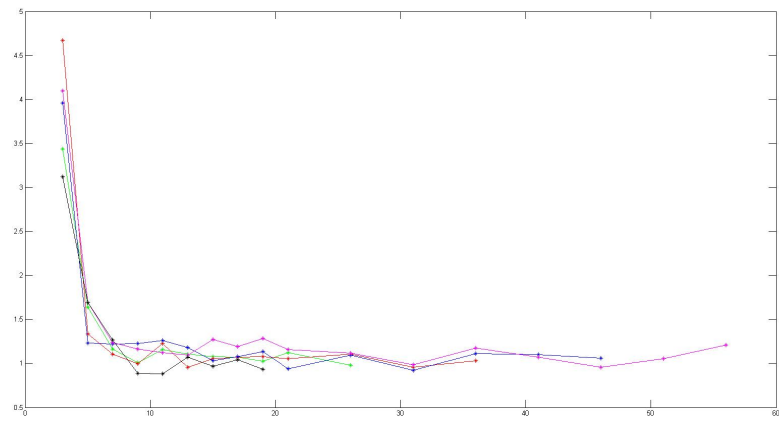
平均连接度数与定位精度之间的定量刻画

作为本节的最后一小节，我们简要讨论对平均连接度数 λ 与定位精度之间关系的定量刻画过程。

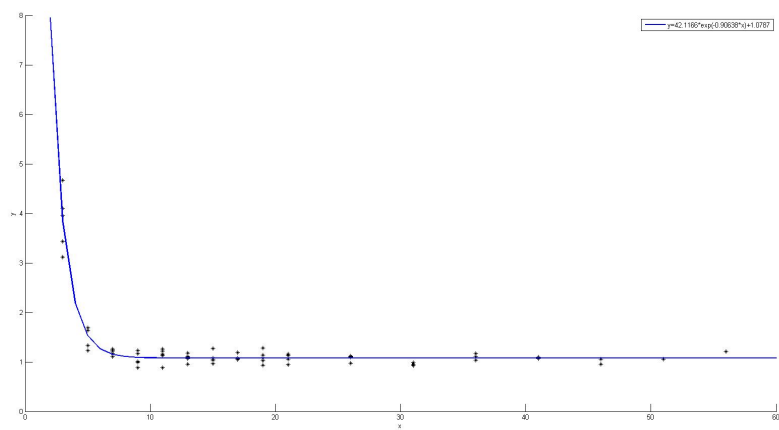
λ 的意义是平均一个终端可以连接到的基站数。在已知终端能够被定位的前提下，可以根据三维空间中坐标计算出终端与所有基站之间的距离，从中选择距离小于等于

表 2 朴素贝叶斯分类器的分类结果统计

Case	共有终端数	能够被定位的终端数
Case026	1100	40
Case027	1200	807
Case028	1300	178
Case029	1400	900
Case030	1000	16



(a)



(b)

图 8 λ 与 ERR'/ERR 关系图: (a) 5 个 sample_case 的折线图; (b) 指数函数拟合曲线.

200m 的所有基站的数量，即为该终端的连接数。因此在终端能够被定位时， λ 是一个已知量。在问题二求解近似最优的三维定位精度过程中，用 ERR（定位误差）衡量定位精度，ERR 越小，表示定位精度越高，这里沿用该方法。设不同基站个数的定位精度是 ERR'，所有基站个数的定位精度是 ERR。因为存在量纲的问题，所以我们使用 $\omega = ERR'/ERR$ 来衡量定位精度，以消除量纲的影响。

使用五组 sample_case 求解 λ 和 ω 的关系，求解过程与问题二的方法大致相同，只是把原来的 ERR 替换为 $\omega = ERR'/ERR$ ，五组 sample_case 的运行结果如图8所示。可见曲线轨迹与问题二中的及其相似，这与我们之前的分析也相吻合：随着平均连接数的增加，定位精度会迅速提升，随后达到系统瓶颈。

为了得到更精确的描述，可以对上面的所有数据点进行曲线拟合，观察曲线形态，可以选择指数函数 $\omega = a \cdot \exp(-b \cdot \lambda) + c$ 作为拟合模型，其中 a, b, c 是待估计的参数。使用最小二乘法拟合，置信水平为 95% 下，得到的参数估计结果是： $a = 42.12$ （置信区间 (18.15, 66.08)）， $b = 0.9064$ （置信区间 (0.7188, 1.094)）， $c = 1.079$ （置信区间 (1.027, 1.131)）；R-square 统计量是 0.9429，RMSE 统计量是 0.1858，这说明拟合的效果是很理想的。于是，我们得到 λ 与定位精度 ω 之间的定量关系为：

$$\omega = 42.12 \exp(-0.9064\lambda) + 1.079. \quad (18)$$

5. 总结

本文分别讨论了四个与基站定位有关的问题，并建立了相应的数学模型。问题一中，同时考虑 NLOS 误差以及测量误差，建立了测量的修正模型，将定位问题归结为无约束多维非线性优化问题，并设计了高效的搜索算法，可以实现快速且精度较高的定位。对问题二建立了概率优化模型，并基于蒙特卡洛思想求解最少的能够实现近似最优定位的基站数。问题三中，基于第一问的定位算法获得二维平面的若干位置，通过对位置散点图的拟合确定了移动终端的运动轨迹。对问题四，通过建立统计学习模型，使用机器学习算法（朴素贝叶斯）对多个场景下的终端进行能否被定位的判别，最后讨论了平均连接度数 λ 与定位精度之间的定量关系。

6. 参考文献

- [1] Tang H, Park Y, Qiu T. NLOS mitigation for TOA location based on a modified deterministic model[J]. Research Letters in Signal Processing, 2008, 2008(1):4.
- [2] Jc. L, Mh. W, Pe. W, et al. Convergence properties of the Nelder-Mead simplex method in low dimensions[J]. Siam Journal on Optimization, 2006, 9(1):112-147.

- [3] 林加涛. MC-DS-CDMA 移动通信系统中移动台定位技术的研究 [D]. 南京理工大学, 2004.
- [4] 李航. 统计学习方法 [M]. 清华大学出版社, 2012.
- [5] 刘林. 非视距环境下的无线定位算法及其性能分析 [D]. 成都: 西南交通大学, 2007.
- [6] 林国军. 蜂窝网无线定位算法及其性能分析 [D][D]. 成都: 西南交通大学, 2008.
- [7] 胡旦. 基于 TDOA 的蜂窝网定位跟踪算法研究 [D]. 电子科技大学, 2013.
- [8] 阚云鹤. 蜂窝网中基于 TDOA 的无线定位算法研究 [D]. 大连: 大连理工大学, 2008.
- [9] 龚福春, 李文元, 聂少涛. 基于 TDOA 的 3G 通信网移动终端定位技术 [C]//2007 北京地区高校研究生学术交流会通信与信息技术会议论文集 (下册). 2008.
- [10] 王昕, 王宗欣, 刘石. 一种考虑非视线传播影响的 TOA 定位算法 [J]. 通信学报, 2001, 22(3): 1-8. 张保峰, 刘同佩, 韩燕, 等. 基于 TOA 的三维空间定位算法研究 [J]. 计算机工程与设计, 2007, 28(14): 3364-3366.

附录 A 问题 1 求解的 Matlab 代码

主文件（脚本文件）：**mainProblem1.m**

说明：这里只是对一个文件进行读写并求解的代码，其他文件类似（当然，可以批处理）

```
% 运行本程序之前要先以矩阵形式导入文件case001_input., 其他文件类似txt
clc
global xBS
[xBS, TOA] = readData(case001input);
[x,minObj] = solveMD_MonteCarlo( TOA );
writeData('~\output_case_001.txt',x) % 注意：代表了你写文件的文件夹地址~
```

函数文件：**readData.m**

%是待输入的矩阵，注意不是的文件，而是导入后的矩阵atxt

```
function[xBS, TOA] = readData(inputData)
    bs_num = inputData(1,1);    %基站数量
    ms_num = inputData(2,1);    %终端数量
    dimension = inputData(3,1); %维度
    xBS = inputData(4:(3+bs_num), 1:dimension);
    TOA = inputData(4+bs_num: 3+bs_num+ms_num, 1:bs_num);
end
```

函数文件：**writeData.m**

%是待输入的路径和文件名，是需要写入的结果pathoutputData

```
function[] = writeData(path,outputData)
    filename = fopen(path, 'w');    %以写模式写文件
    [m, n] = size(outputData);
    for i = 1: m
        for j = 1: n
            fprintf(filename, '%10.4f', outputData(i, j));
        end
        fprintf(filename, '\n');    %用于写完一行之后换行
    end
    fclose(filename);
end
```

函数文件（核心求解算法）：**solveMD_MonteCarlo.m**

注意：以下两个函数都在同一个函数文件中。

```
function [x,minObj] = solveMD_MonteCarlo( TOA )
global N M xBS beta r
M = size(xBS,1);
N = size(TOA,1);
x = zeros(N,3);
for i = 1:N
    r = TOA(i,:) * (3*1e8);
    yita = 0.5 * ones(1,M);
    T = 50; % 循环次数
    tmpx = zeros(T,3);
    minObj = Inf;
    tic
    for k = 1:T
        epsilon = normrnd(0,1);
        beta = yita + (1-yita)*epsilon;
        % 调用无约束多维优化算法
        [x_f,fval] = fminsearch(@gfun,mean(xBS),optimset('MaxFunEvals',1000));
        tmpx(k,:) = x_f;
        if fval < minObj
            minObj = fval;
            x(i,:) = tmpx(k,:);
        end
    end
end
toc
end

end

function [ obj ] = gfun( x )
%目标函数
global M xBS beta r
normVec = sum(abs(repmat(x,M,1)-xBS).^2,2).^(1/2);
obj = sum((beta.*r - normVec').^2);
```

end

附录 B 问题 2 求解的 Matlab 代码

主文件（脚本文件）：**mainProblem2.m**

clc

% 注意：这是对一个文件进行处理

% 说明：之前要将以矩阵形式导入工作区 *sample_case001_ansMatlab*

% 处理完之后请保存 *meanERR*

global xBS_m m

[xBS, TOA] = readData(samplecase005input);

N = **size**(TOA,1); % 终端数

M = **size**(xBS,1); % 基站数

numSample = 100; % 抽样数, 大约运行 40min

tic

selectedOrder = **randperm**(N);

locationDB = cell(numSample,M); % 存放定位结果的元胞数组

ERR = **zeros**(numSample,M); % 存放定位误差的矩阵

step = 2; % 基站数步长

for i = 1:numSample

idMS = selectedOrder(i); % 第次抽样的终端 *iID*

for m = 3:step:M % 依次增加基站数, 第次循环取前个基站 *mm*

xBS_m = xBS(1:m,:); % 取前个基站坐标 *m*

[x,minObj] = solveOneMS(TOA(idMS,:)); % 调用定位算法对一个终端进行定位

ERR(i,m) = **norm**(x-samplecase001ans(idMS,:));

locationDB{i,m} = x; % 存放定位结果

end

end

toc

newERR = ERR(:,3:step:M);

meanERR = **mean**(newERR);

%plot(3:step:M, meanERR, '-o', 'LineWidth',1.25), hold on

```
%xlabel参与定位的基站数(''), ylabel('RMSE')
```

函数文件（定位算法）：**solveOneMS.m**

说明：与solveMD_MonteCarlo.m的功能一样，只是这是对一个终端进行定位的，且参与定位的基站数可调。

```
function [ x, minObj ] = solveOneMS(TOA)
```

```
%solveOneMS 对一行数据进行求解TOA
```

```
%Input:
```

```
% TOA: 一行数据TOA
```

```
global xBS_m m beta r
```

```
r = TOA(1:m) * (3*1e8);
```

```
yita = 0.5 * ones(1,m);
```

```
T = 100; % 迭代次数
```

```
minObj = Inf;
```

```
for k = 1:T
```

```
    epsilon = normrnd(0,1);
```

```
    beta = yita + (1-yita)*epsilon;
```

```
    % 调用无约束多维优化算法
```

```
    [x_tmp,fval_tmp] = fminsearch(@myfun,mean(xBS_m),optimset('MaxFunEvals',1000
```

```
    if fval_tmp < minObj
```

```
        minObj = fval_tmp;
```

```
        x = x_tmp;
```

```
    end
```

```
end
```

```
end
```

```
function [ obj ] = myfun( x )
```

```
global m xBS_m beta r
```

```
normVec = sum(abs(repmat(x,m,1)-xBS_m).^2,2).^(1/2);
```

```
%{
```

```
for i = 1:M
```

```
    normVec(i) = norm(x-xBS(i,:));
```

```
end
```

```
%}
```

```
obj = sum((beta.*r - normVec').^2);
```

```
end
```