



中国研究生创新实践系列大赛
中国光谷·“华为杯”第十九届中国研究生
数学建模竞赛

学 校 北京邮电大学

参赛队号 22100130054

1. 冯天晴

队员姓名 2. 王世杰

3. 刘彦兵

中国研究生创新实践系列大赛

中国光谷·“华为杯”第十九届中国研究生 数学建模竞赛

题 目 基于压缩感知的快速超分辨率雷达定位与追踪算法

摘 要：

随着人工智能时代的来临，自动化领域的研究内容越来越凸显其重要性。其中移动场景下的超分辨率定位问题不仅在自动驾驶、军事等重要领域应用广泛，而且在商业场景、生活家庭中也起到很大的作用。现阶段最常见的移动场景定位产品是调频连续波雷达 FMCW，但由于目前的常用算法分辨率较低，难以满足超分辨率的需求，因此迫切需要设计与开发用于雷达的超分辨率移动场景定位算法。本文在压缩感知理论基础上，建立了一套高效超分辨率的雷达定位与追踪算法。

传统的雷达定位方法在采集大量的雷达信号后难以做到超分辨率定位，压缩感知是完全独立于传统雷达定位方法的一种思路，它利用物体在空间分布的稀疏性来对信号进行重建，所需的数据量远少于传统的方法。但在现有的雷达定位研究中，即使用压缩感知方法也难以实现超分辨率定位，其局限性主要分为两个方面，一方面是在于压缩感知恢复的信号分辨率完全取决于对目标区域的划分，因此为实现超分辨率需要承担平方级别的运算量增长；另一方面是压缩感知难以恢复近似重叠的信号，通常极有可能将重叠信号识别为多个低强度信号。本文致力于解决这两方面的局限性，并就静止物体的定位以及移动物体的追踪两部分展开。

对静止物体的超分辨率快速定位是实现移动场景定位的前提，本文提出时空分块的压缩感知算法，在不损耗数据量的同时大幅度降低了算法运行时间，同时利用密度峰值聚类算法解决压缩感知方法在面对重叠信号时的“欠稀疏”表现，并采用了连续优化的方式对重建信号进行更新以实现超分辨率。

对移动物体的在线追踪算法是解决移动场景超分辨率定位的关键，其最大难点在于设计低复杂度的在线算法。本文将该问题归结至数据同化问题，考虑将前几个时刻得到的信息与当前时刻获取的信息进行同化，以期更快速地对当前目标位置进行重建。为此，提出了局部压缩感知算法，利用前一时时刻求解得到的位置信息，来对当前时刻的压缩感知备选格子点进行范围约束，极大提升了算法效率与分辨率。另外，本文借鉴深度学习中的 Dropout 思想，利用观测数据冗余的特点，在时空分块压缩感知算法随机丢弃一些分块，显著减少了算法运行时间。

针对于题目给出的数据，第 1、2、4 问均可以在静止物体的定位的框架下来考虑，利用设计的算法，解得这三问的物体数均为 2 个。第 1 问无噪声数据，重建得到两个物体的极坐标分别为(7.0048, -0.0026)与(7.0048, 0.0026)。第 2 问噪声数据，重建得到两个物体的极坐标分别为(8.2057, -0.0054)，(8.2058, 0.0054)。第 4 问天线老化数据，重建得到两个物

体的极坐标分别为 $(6.1042, 0.0026)$, $(6.0041, -0.0026)$ 。第3小问考虑的是运动物体的追踪问题,利用本文建立的算法,求得物体数量为2个,初始状态的极坐标分别为 $(6.0042, -0.0079)$ 与 $(6.0042, 0.0079)$,重建平均时间为0.1231每chirp周期。

最后,本文在几个不同噪声强度的测试算例上测试并运行算法,均取得良好的恢复效果,证明了该算法的有效性与鲁棒性。

本文的主要贡献是通过考虑时空分块、Dropout 等做法以解决压缩感知在运算上的问题,引入连续优化、局部压缩感知来解决压缩感知分辨率的问题,基本解决了压缩感知在超分辨率场景下的两大局限性问题,证明了压缩感知有解决超分辨率雷达定位与追踪问题的潜力。

关键词: 超分辨率定位; 压缩感知; 雷达定位

公众号关注：建模忠哥
获取更多资源

目录

1 问题背景及问题重述	5
1.1 问题背景	5
1.2 问题重述	5
1.3 文献综述	6
2 静止物体的超分辨率定位算法	7
2.1 信号的稀疏表示	7
2.2 压缩感知雷达定位的算法	8
2.2.1 信号的重建理论	9
2.2.2 信号的重建算法	10
2.2.3 超分辨率优化	12
2.3 不同场合的求解结果	15
3 运动物体的超分辨率追踪算法	17
3.1 数据同化方法简介	17
3.2 超分辨率目标追踪算法	19
3.2.1 局部压缩感知	19
3.2.2 Dropout	19
3.3 任务求解	20
3.3.1 模拟数据上的结果	20
3.3.2 问题数据的结果	22
4 灵敏度分析	24
4.1 低噪声	24
4.2 中噪声	24
4.3 高噪声	25
5 结论与展望	26
5.1 结论	26

5.2 工作展望	26
参考文献	28
附录	30

公众号关注：建模忠哥
获取更多资源

1 问题背景及问题重述

1.1 问题背景

随着科技的日益发展, 物品定位技术广泛应用在社会的各个方面. 工业中需要智能机器对不同标签的材料进行定位与选取; 物流业需要自动驾驶机器进行不同类型的快递筛选和运送; 而我们的生活中, 也应用到很多这样的技术, 例如对不同的生活用品贴上电路标签, 利用智能机器进行全屋查询, 可以精准定位到这些物体, 将极大地提升人们生活的便利性. 但是这也带来其他的问题, 例如在一些精准度要求较高的领域, 需要在移动场景中精准探测多个物体的位置, 则需要更优化的算法来实现.

现有移动场景定位产品通常采用调频连续波雷达 FMCW, 通过发射线性增长频率的信号波, 以及接收反射回来的信号波来进行定位. 目前该产品大多采用基线算法, 其得到的分辨率较低, 不能满足日益增长的超分辨率定位需求. 现有的压缩感知算法利用了空间物体分布的稀疏性, 可以有效提升分辨率. 杨瑞峰 and 郭晨霞 [20] 针对混合声源定位精度低的问题, 在不牺牲定位精度的情况下提高计算效率. 该研究虽然对定位精度的计算效率进行优化, 但无法处理不规则阵列声源的定位问题. 因此处理这种连续傅里叶字典场景并设计低复杂度算法是一个巨大的挑战, 本研究通过建模以及设计对应算法来解决这个难点, 进一步提高移动场景超分辨率定位的效率.

1.2 问题重述

首先对本研究内容的四个子问题进行初步分析:

- 问题一: 针对提供的无噪声仿真数据, 建立定位模型, 计算出物体相对位置. 该问题实际是超分辨率定位的最基础问题, 在无噪声的情况下, 以超高的分辨率识别物体位置信息是对算法有效性的一个最基本的检验. 并需要以此为基础展开混合噪声数据以及物体运动追踪物体的研究.
- 问题二: 针对提供的高斯噪声仿真数据, 利用一个 chirp 周期内的 IF 信号, 设计超分辨率算法精确定位多个物体. 该问题在问题一的基础上混合了噪声, 而本质内容其实是不变的, 真正可以在实际中使用的算法理应能够处理强噪声、弱噪声的情况.
- 问题三: 设计在线低复杂度算法, 利用一帧中频信号来超分辨率定位, 并且通过数值实验验证算法性能. 相比于前两问, 该问题是一个动态的问题, 也可以认为是超分辨率目标定位追踪, 该问题的最大难点在于如何设计低复杂度的算法快速求解.
- 问题四: 考虑实际场景中由于老化等原因, 天线阵列对于自身的定位也会有误差. 针对提供的仿真数据, 设计提升定位算法的鲁棒性的改进算法. 该问题本质上与问题一以

及问题二并无区别, 仍然属于静止物体识别的问题, 位置信息的误差也可以视为噪声来解决.

综合上述分析, 该研究内容可以分解为静止物体识别与动态物体追踪两个子内容, 而每一个子内容都需要实现高分辨率. 本文以此为思路展开研究.

1.3 文献综述

超分辨率定位任务一直是科学家们研究的热点, 尤其是在移动场景中, 同时有多物体信号叠加的情况下, 如何极大化物体定位精度是目前研究的难题. 有很多学者都进行了多种测试与研究, Haris et al. [10]对一个训练深度神经网络的框架进行分析, 其中超分辨率子网络通过与传统检测损失的权衡, 明确地将检测损失纳入其训练目标, 最终得出端到端的训练过程可以为任何可微检测器训练超分辨率预处理. 该实验虽提高了对象检测器对来自 COCO 和 PASCAL VOC 数据集的低分辨率图像的准确度, 但未深入研究现实条件下多物体的超分辨定位. 在雷达定位的研究领域内, Roos et al. [16]设计了一种优化的稀疏天线阵, 稀疏阵列中缺失的天线单元用压缩感知算法重建. 该算法不需要了解所考虑场景中的目标数量, 但其准确性有待进一步验证. Fortunati et al. [9]研究了基于压缩感知的波束形成器的统计特性, 证明了基于压缩感知的 DOA 估计器能够保证超分辨率特性, 并且表明, 与经典的波束形成器不同, 基于压缩感知的波束形成器具有自适应算法(例如 Capon 和 MUSIC)的一些理想特性.

此外, 在移动场景中, 通过卡尔曼滤波对多物体信号叠加条件下的超分辨定位问题是热点话题. Ullah et al. [17]开发了一个由扩展卡尔曼滤波器(EKF)、无迹卡尔曼滤波器(UKF)和粒子滤波器(PF)三种定位算法组成的框架. 该研究团队通过仿真结果显示, 所提出的定位算法可用于目标跟踪、机器人定位等多种用途, 极大程度提高定位性能, 但是该研究的算法框架仍然无法精准处理移动场景下的物体定位. Feng et al. [8]提出了一种基于扩展卡尔曼滤波器融合的单基站单距离直接定位算法(DPA), 在定位精度要求不高的情况下, 该算法可以大大降低基站部署的复杂度和成本. 为了降低计算复杂度, 这篇文章还提出了基于单基站和三基站的无迹卡尔曼滤波器融合定位算法. 仿真结果表明, 无迹卡尔曼滤波器融合算法能够获得比扩展卡尔曼滤波器融合定位算法更好的定位精度.

从上述内容看到, 国内外已有不少学者进行超分辨率定位与追踪的研究, 包括一些在雷达定位领域的研究, 但大多都是在理想场景下进行模拟, 想要达到真正的超分辨率定位在实际操作中仍然存在很大的困难. 压缩感知理论是独立于传统雷达定位方法的另外一种思路, 其潜力仍然有待深入挖掘.

2 静止物体的超分辨率定位算法

压缩感知是近年来兴起的一种新的信号处理方法, 包括在图像处理等领域都有重要的应用前景. 它能够为解决雷达系统中巨量数据的采集、存储和传输问题提供了一种有效而便捷的手段. 自 Donoho [6]提出压缩感知以来, 该方法被越来越多的应用在不同的领域, Candès et al. [3]指出将压缩感知用在信号恢复领域可以得到前所未有的效果. 自此之后, 陆续有学者运用压缩感知的方法来解决信号领域的各种问题. 同时压缩感知的基本理论与方法也得到了发展. 压缩感知理论突破了传统采样方法的瓶颈, 它可以将信号的采样和压缩过程合并, 达到只要有少量的数据 (远少于传统采样方法), 就可以恢复信号的效果, 这大大降低了信号采样和传输的成本.

2.1 信号的稀疏表示

在信号重建问题上使用压缩感知方法的前提是信号具有稀疏性, 而雷达接收到的测量信号是不具有稀疏性的, 因此 Herman and Strohmer [12]在一维采样问题上考虑了构建正交基矩阵的方法, 基本思想是将研究区域按照“时间 \times 空间”的方式进行布局, 并对该布局进行格点划分, 每一个格点便对应了一个正交基矩阵. 当某一个正交基矩阵得到匹配时, 便认为该格点对应的时空位置是具有信号的. 本文在此理论基础上继续深化研究, 优化传统的做法.

考虑问题研究的区域, 即以原点为中心, 半径 10 米以内、开口向上张开圆心角为 100° 的扇形区域, 对其进行均匀格点划分. 更细的划分将会带来更高的分辨率, 但同时会造成计算时间的上升, 这一点在后文将会体现. 例如我们想要边长为 1m 的格点 (称分辨率为 1m), 则可以在 $[-10, 10] \times [0, 10]$ 的矩形区域上以 1 为步长生成均匀格点, 再丢弃在研究区域之外的格点, 只保留研究区域内的格点. 图 1 给出了一个例子, 其中分辨率取为 1m.

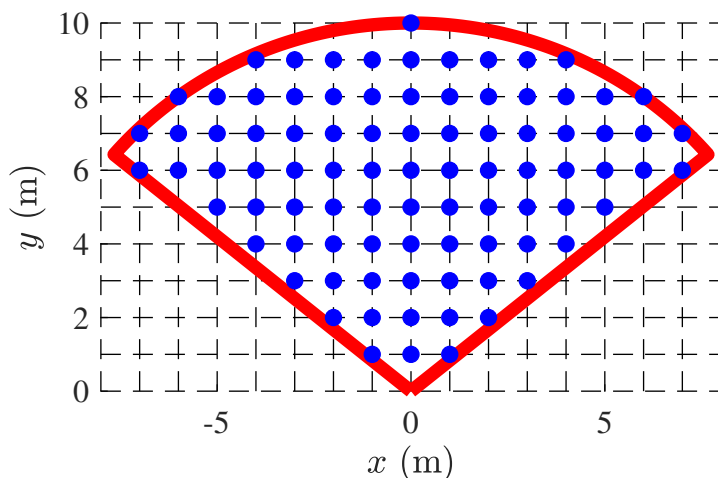


图 1: 对研究区域 (红色线内) 进行分辨率为 1m 的格点划分, 其中蓝色点为生成的格子点.

相比于在极坐标系上进行均匀划分而言, 直角坐标系上的均匀划分将会更加“均匀”, 极坐标系的均匀划分将会导致小的极径处的格子点过于密集, 大的极径处的格子点过于稀疏, 这将对实现超分辨率带来更大的难度. 因此本文的研究均采用直角坐标系的均匀划分.

根据前文所述的划分方式, 现将目标区域划分为 N 个格点, 第 k 个格点上的直角坐标记作 $(x_k, y_k) = (r_k \sin \theta_k, r_k \cos \theta_k)$, $k = 1, 2, \dots, N$, 将所有格点上对应的反射系数拼接成向量, 即 $\alpha_N = (a_1, a_2, \dots, a_N)^\top \in \mathbb{C}^N$, 注意此处的反射系数并非物体真实的反射性, 由于物体在格点上的分布是稀疏的, 因此大部分的格点上的反射系数将会接近于 0, 非 0 反射性的格点便是物体所处的位置.

将一个 chirp 周期划分为 M 个时刻, 天线在每个时刻都会接受到中频信号, 定义 t 时刻的信号感知矩阵 ($t = 0, 1, \dots, M-1$)

$$\mathbf{E}(t) = \left(\exp \left\{ j \left(2\pi \gamma T_S t \frac{R_{n,k}}{c} + 2\pi f_0 \frac{R_{n,k}}{c} \right) \right\} \right)_{n=0, \dots, N_a-1, k=1, \dots, N}, \quad (1)$$

该矩阵的 (i, j) 元素表示在 t 时刻时除去物体反射性 a_k 的中频信号, 用 $z_n(t)$ 表示天线 n 在 t 时刻接收到的中频信号 ($n = 0, 1, \dots, N_a - 1$), 则 t 时刻的所有天线接受到的信号向量与感知矩阵之间有如下线性关系:

$$\mathbf{z}(t) \doteq (z_0(t), z_1(t), \dots, z_{N_a-1}(t))^\top = \mathbf{E}(t) \alpha_N. \quad (2)$$

对 M 个时刻的线性关系按行分块, 得到如下的线性方程组:

$$\begin{pmatrix} \mathbf{z}(0) \\ \vdots \\ \mathbf{z}(M-1) \end{pmatrix} = \begin{pmatrix} \mathbf{E}(0) \\ \vdots \\ \mathbf{E}(M-1) \end{pmatrix} \alpha_N, \quad (3)$$

将上式简记作 $\mathbf{Z} = \mathbf{E} \alpha_N$, 其中 $\mathbf{Z} \in \mathbb{C}^{MN_a}$ 为测量信号, $\mathbf{E} \in \mathbb{C}^{MN_a \times N}$ 为感知矩阵, α_N 为稀疏信号, 在本文的研究问题中, 其物理意义为每个格点上物体的反射系数, 而当格点上没有物体时, 反射系数应该为 0, 这与 α_N 的稀疏假设是吻合的.

根据我们对研究区域做格点划分的细致程度, 尤其是在当前的超分辨率研究背景下, N 可以取作很大的数值, 因此观测信号 \mathbf{Z} 可以视作高维稀疏信号 α_N 在感知矩阵 \mathbf{E} 下的投影, 见图 2.

2.2 压缩感知雷达定位的算法

信号重建是压缩感知理论中的核心内容, 也是其最主要的应用方向. 其目的是将 K 稀疏的信号 α 从测量信号 \mathbf{Z} 中恢复出来. 从数学上来说就是求解线性方程组 3 的问题. 而从图 2 可以看到, 经过投影得到的测量信号 \mathbf{Z} 的维数通常远小于真实信号 α . 这会导致求解线性方程组 3 成为一个 NP 难问题.

$$\begin{pmatrix} \vdots \\ \mathbf{Z} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \mathbf{E} \\ \vdots \end{pmatrix} \begin{pmatrix} * \\ * \\ * \end{pmatrix} \quad \alpha_N$$

图 2: 观测信号 \mathbf{Z} 是稀疏信号 α_N 在感知矩阵下的投影, 通常观测信号的维数远小于真实信号的维度, 但真实信号是稀疏的, 大部分元素均为 0.

但是考虑到真实信号 α 的稀疏特性, 可以不直接求解线性方程而是利用稀疏信号的重建算法进行求解, 假设 α 是 K 稀疏向量, 即 $\|\alpha\|_0 \leq K$, 可以理解为物体的数量是不超过 K 个的, 则为保证 α 可以被准确恢复, Candes and Tao [2] 对感知矩阵 \mathbf{E} 提出了受限等容特性 (Restricted Isometry Property, RIP) 条件, 它是压缩感知可以成功恢复信号的一个充分条件, 其详细描述可见附录 1.

2.2.1 信号的重建理论

由于向量的 0-范数是最直接的稀疏性体现, 因此信号重构的问题可写作优化模型:

$$\hat{\alpha}_N = \arg \min_{\alpha_N} \|\alpha_N\|_0, \quad \text{s.t. } \mathbf{Z} = \mathbf{E}\alpha_N, \quad (4)$$

但 0-范数最小化问题也是一个 NP 难问题, 同时其数值计算也不稳定, 难以直接求解, 压缩感知领域的最常见做法是将 0-范数最小化问题转化为 1-范数最小化问题, 即

$$\hat{\alpha}_N = \arg \min_{\alpha_N} \|\alpha_N\|_1, \quad \text{s.t. } \mathbf{Z} = \mathbf{E}\alpha_N, \quad (5)$$

后者是一个线性规划问题. RIP 理论以及许多数值实践均证明了 1-范数最小化问题的解在很多场合下等价于 0-范数最小化问题的解. 在信号处理领域, [4] 将求解 1-范数最小化问题称作基追踪 (Basis Pursuit) 方法, 同时他们还考虑到实际情中的信号传输过程中都叠加了噪声, 例如本问题的第二个场景就的雷达回波信号中就包含着噪声, 从而将优化问题 5 的约束条件进行放宽, 即不需要严格满足等式约束, 而是对其 2-范数进行限制, 即

$$\hat{\alpha}_N = \arg \min_{\alpha_N} \|\alpha_N\|_1, \quad \text{s.t. } \|\mathbf{Z} - \mathbf{E}\alpha_N\|_2 \leq \delta, \quad (6)$$

其中 δ 体现了噪声的能量大小. 又根据约束优化的极值理论, 求解上述问题等价于求解下面的无约束优化问题:

$$\min_{\alpha_N} \frac{1}{2} \|\mathbf{Z} - \mathbf{E}\alpha_N\|_2^2 + \mu \|\alpha_N\|_1, \quad (7)$$

参数的建议取值是 $\mu = \sigma \sqrt{2 \log(N \cdot N_a)}$, 其中 σ 是噪声的功率 [19]. 数学上将该问题称作 ℓ_1 正则化问题, 并开发了许多的求解算法. 并且它是一个凸优化问题, 现有的凸优化求解软件, 例如 Mosek, Gurobi 等均可求解该问题, 但此类问题的计算量仍然很大.

2.2.2 信号的重建算法

为解决计算复杂性的问题, 本文针对该复数变量 ℓ_1 正则化问题, 建立交替方向乘子法 (ADMM), 该方法发展于 1970 年代, 与许多其他算法密切相关, 如对偶分解、乘子方法、近似点算法等. [18] 建立了考虑负权重的压缩感知问题的 ADMM 算法, 与本文研究内容相似, 但其变量均为实数, 而本文的研究内容涉及的变量均为复数, 附录 2 将介绍解决复数变量压缩感知问题的技巧, 并将其归结至实数变量问题. 此处我们给出解决实数变量的 ℓ_1 正则化问题的 ADMM 算法, 见算法 1, 具体的推导过程可见附录 3, 求解 ℓ_1 正则化问题的 ADMM 算法 Matlab 程序在附录 4 给出.

算法 1: 求解 ℓ_1 正则化问题的 ADMM 算法

Input: 观测信号 $\mathbf{Z} \in \mathbb{R}^L$, 感知矩阵 $\mathbf{E} \in \mathbb{R}^{L \times N}$, 参数 μ , 最大迭代次数 MaxIter.

- 1 初始化 $\lambda^{(0)} \in \mathbb{R}^L$, $\alpha^{(0)} \in \mathbb{R}^N$;
- 2 $\mathbf{s}^{(0)} \leftarrow \mathcal{P}_{[-\mu, \mu]}(\mathbf{E}^\top \lambda^{(0)} + \mu \alpha^{(0)})$;
- 3 **for** $b = 1, 2, \dots, \text{MaxIter}$ **do**
- 4 $\lambda^{(b)} \leftarrow (\mathbf{E}\mathbf{E}^\top + \mu\mathbf{I})^{-1}(\mu(\mathbf{Z} - \mathbf{E}\alpha^{(b-1)}) + \mathbf{E}\mathbf{s}^{(b-1)})$;
- 5 $\mathbf{s}^{(b)} \leftarrow \mathcal{P}_{[-\mu, \mu]}(\mathbf{E}^\top \lambda^{(b)} + \mu \alpha^{(b-1)})$;
- 6 $\alpha^{(b)} \leftarrow \alpha^{(b-1)} + \frac{\lambda^{(b)} - \mathbf{s}^{(b)}}{\mu}$;

Output: 重建信号 $\hat{\alpha}_N \leftarrow \alpha^{(b)}$.

算法中的 \mathcal{P} 为投影算子, $\mathcal{P}_{[-\mu, \mu]}(\mathbf{x})$ 将向量 \mathbf{x} 的超出区间 $[-\mu, \mu]$ 的分量压缩至该区间的端点上.

注意到算法中乘子 $\lambda^{(b)}$ 的更新涉及到求解 $\mathbf{E}\mathbf{E}^\top + \mu\mathbf{I}$ 的逆矩阵, 该矩阵的维度是观测信号向量的长度 L . 在本文的研究背景下, 一般而言有 $L = N_a \times M = 22016$, 即使 ADMM 算法收敛速度很快, 且该矩阵在迭代过程中不发生改变, 我们可以事先求解出逆矩阵, 但求逆运算的复杂度是 $\mathcal{O}(L^3)$, 这一计算代价仍然是不可承受的.

为降低这一计算代价, 我们提出了时空分块的 ADMM 算法, 考虑到求解压缩感知问题并不需要有过多的测量信号, 雷达天线阵列的观测数据相对而言是过于冗余的. 我们在模拟也中发现仅使用 20 % 的信号数据所取得的重建效果与使用所有测量信号数据的重建效果类似, 见图 3.

因此, 基于上述分析, 我们考虑进行时空分块, 即分别在时间与空间上进行分块. 这里

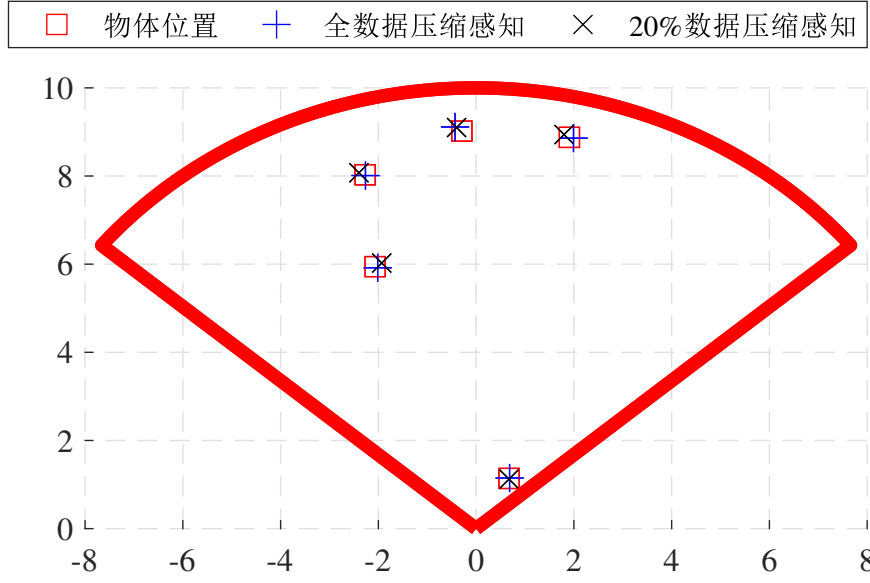


图 3: 在一个模拟数据集上 (物体数 $K = 5$) 用所有数据进行压缩感知定位以及仅用 20% 数据进行压缩感知定位的对比图, 其中红色 ‘□’ 是物体位置, 蓝色 ‘+’ 号是全数据压缩感知的定位, 黑色 ‘×’ 号是 20% 数据的压缩感知定位.

时间分块指的是将一个 chirp 周期内的时刻进行划分, 而空间分块指的是将雷达天线进行分组, 假设我们在时间上分为 3 块, 空间上分为 2 块, 则总计可以将观测的信号数据分为 6 块. 对于每一个块内的数据, 我们分别调用算法 1, 最后将重建的信号进行合并, 且按照一定的阈值舍弃信号, 算法 2 给出了基本描述. 时空分块算法的最大好处便是降低矩阵 $\mathbf{E}\mathbf{E}^T + \mu\mathbf{I}$ 的维度, 从而降低求逆矩阵的运算量.

算法 2: 时空分块的压缩感知算法

Input: 分块数目 B , 分块的观测信号 $\mathbf{Z}_1, \dots, \mathbf{Z}_B$, 相对应的感知矩阵 $\mathbf{E}_1, \dots, \mathbf{E}_B$, 参数 μ , ADMM 算法的最大迭代次数 MaxIter , 信号的丢弃阈值 ρ .

```

1  $\alpha \leftarrow 0_L$ ;
2 for  $\kappa = 1, 2, \dots, B$  do
3   % 在每一个分块上均调用算法 1;
4    $\alpha_\kappa \leftarrow \text{ADMM\_CS}(\mathbf{Z}_\kappa, \mathbf{E}_\kappa, \mu, \text{MaxIter})$ ;
5    $\alpha \leftarrow \alpha + \alpha_\kappa$ ;
6  $\text{maxSignal} \leftarrow \max(\alpha)$ ;
7 for  $l = 1, 2, \dots, L$  do
8   if  $|\alpha[l]| \leq \rho \cdot \text{maxSignal}$  then
9      $\alpha[l] \leftarrow 0$ ;

```

Output: 重建信号 $\hat{\alpha}_N \leftarrow \alpha^{(b)}$.

值得注意的是, 算法 2 不仅仅局限于调用 ADMM 算法, 实际也可以在每一个分块上分别调用 Mosek 等求解器, 这将大幅度提升效率. 此外, 在每一个分块上求解 ℓ_1 正则化问题

是相互独立的, 因此该算法可以很容易并行实现. 该算法在大幅度减少运算时间的同时也没有舍弃任何的观测信号数据, 这也保证了结果的稳定性. 我们将图 3 对应的算例同时也用算法 2 嵌套 ADMM 以及 Mosek 求解器进行求解, 其中时间分块数为 10, 空间分块数为 5, 格子点的分辨率为 0.1m, 信号的丢弃阈值取 $\rho = 0.5$, 噪声功率取 $\sigma = 0.5$, 两种求解算法均采用 10 核心并行计算, 算法的运行时间及性能见表 1, 可以看出 ADMM 算法的求解速度是显著高于 Mosek 求解器的. 其中重建信号的二次误差指的是利用重建的物体位置还原出的观测信号与原始观测信号之差的二范数, 即 $\|\mathbf{Z} - \mathbf{E}\hat{\alpha}_N\|_2$, 考虑到这里观测信号的总维度是 $N_a \times M = 22016$, 表 1 中两种算法得到的损失已经较小, 可以认为还原效果比较优秀.

表 1: 在图 3 对应的模拟数据集上, 分别在分块压缩感知算法中调用 ADMM 算法以及 Mosek 求解器的运行结果, 均采用 10 核心并行计算.

求解器	cpu 时间 (s)	重建信号的 ℓ_2 误差
Mosek	240.4064	3.3688
ADMM	6.0562	3.6597

2.2.3 超分辨率优化

雷达通常的任务是在广阔空间中探测稀疏分布的目标物, 而目标物具有高度的稀疏性, 这为压缩感知理论在雷达目标定位识别中的应用提供了条件. 相比于传统方法, 压缩感知是一种可以达到更高分辨率的方法, 然而由于压缩感知的分辨率主要由其划分的精细程度而决定, 越精确的划分对应更高的分辨率, 但想要得到更高的分辨率势必会造成运算量极速增加. 例如我们想从 0.1m 的分辨率改进至 0.01m, 则 N 将会增大 100 倍, 也就是说运算量的增加是平方级别的. 如果我们想要达到超分辨率, 必须有足够的运算资源支撑, 并且也难做到快速算法.

此外, 当物体之间的距离小于格子点的划分长度时, 压缩感知没有能力识别多个物体, 这也是压缩感知在“超分辨率”问题上的局限性. 导致该问题的主要原因是由于压缩感知算法只能给出在格点上的信号估计, 而对于不存在格点的空白区域, 压缩感知只能用距离其最近的格点来近似替代它, 且有可能会将真实信号附近的多个格子点均识别为存在物体, 出现形如“云朵”状的信号, 见图 4. 该图展示了离散物体以及重叠物体的压缩感知结果, 显然压缩感知更能够胜任离散型物体的定位与识别的任务.

为解决该问题, 我们考虑对压缩感知还原的信号进行改进, 若出现类似于图 4 的情况, 其中的点状信号表示识别到离散物体, 而云朵状信号说明识别到重叠物体或出现了很大的噪声. 此时由于云朵状信号的存在, 压缩感知重建信号稀疏性没有得到充分体现, 因此我们考虑将压缩感知的解进行进一步稀疏化. 这一步骤有多种实现方式, 根据掌握的信息不同:

1. 物体数 K 已知: 利用 k-means 聚类算法, 其中聚类数目取作物体数 K , 聚类变量为压

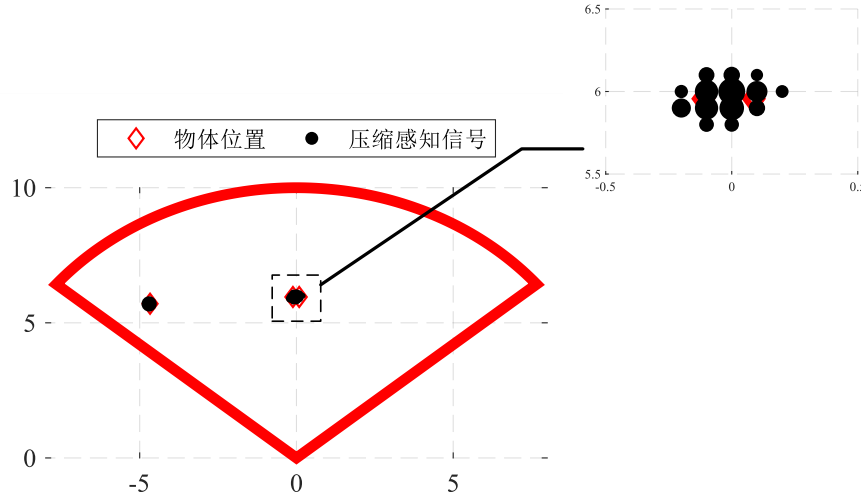


图 4: 压缩感知成功识别离散物体出现的点状信号 (图左侧), 以及在面对近似重叠物体时出现的“云朵状”信号 (图中侧以及小图).

压缩感知重建的物体坐标. 聚类成功后重新对簇中心的坐标进行修改, 修改方式可以为该类中所有信号点坐标的按信号强度加权平均, 或者直接取该类中信号强度最强的点.

2. 物体数 K 未知:

- 利用密度峰值聚类法 [15], 云朵状的信号由信号强度较小的一些格点组成, 可以考虑改进密度峰值点的计算方式, 将点的局部密度修改为局部的信号强度, 同时要求不同簇中心之间的距离应该相对较远, 从而利用密度峰值点的数量可以有效地发现聚类簇的数量.
- 利用社区发现算法 (Community Detection), 社区发现算法类似于聚类算法, 但不需要事先指定聚类数量. 而最先进的社区发现算法考虑重叠社区 [5], 云朵状信号中的信号点可以被认为处于重叠社区中, 因此可以借助重叠社区发现算法进行信号的进一步稀疏化.

本文主要使用了聚类型的算法, 我们在图 4 的算例上分别运行了物体数 $K = 2, 3$ 的 k-means 聚类算法, 结果展示在图 5 中, 其中 $K = 2$ 的 k-means 算法很自然地将云朵状的信号识别为一个物体, 而 $K = 3$ 的 k-means 算法成功将重叠物体识别. 另外, 两个物体的最终观测信号重建误差为 544.9472, 而三个物体的重建误差仅为 0.0010, 二者相差非常大.

为实现超分辨率的目的, 同时考虑到聚类的结果可能不够稳定, 构建连续目标函数进行更新是必须的, 现我们以物体的直角坐标为优化变量, 考虑极小化重构信号误差, 即最小化如下的述目标函数

$$\min_{\{(x_i, y_i)\}_{i=1}^K} \|Z - \mathbf{E}_{\{(x_i, y_i)\}_{i=1}^K} \alpha_K\|_2^2, \quad \text{s.t. } \alpha_K = (\mathbf{E}^\top \mathbf{E})^{-1} \mathbf{E}^\top Z, \quad (8)$$

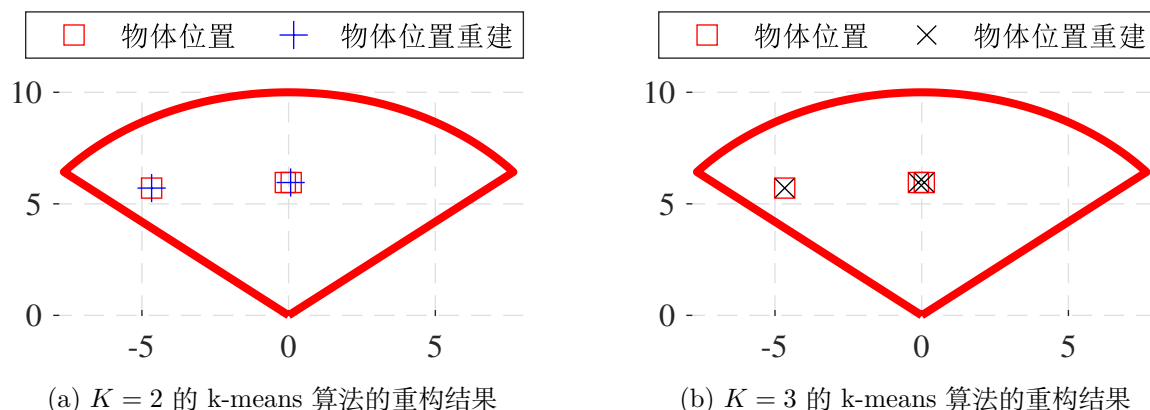


图 5: 在图 4 对应的模拟数据集上, 分别调用 $K = 2, 3$ 的 k-means 算法进行进一步稀疏化.

该极小化问题并不复杂, 其中 K 已被我们确定, 还需要注意的是其中矩阵 \mathbf{E} 依赖于优化变量, 根据式 1 可求得 \mathbf{E} , 而约束条件实际说明的是 α_K 是观测信号 \mathbf{Z} 关于感知矩阵 \mathbf{E} 的最小二乘解, 而 $\mathbf{E}^T \mathbf{E}$ 是 K 阶矩阵, 通常物体数 K 的数值不会太大, 因此大部分场合该矩阵的求逆操作是可接受的. 最关键的一点在于该目标函数实际上考虑的是 \mathbf{Z} 关于感知矩阵 \mathbf{E} 作普通线性回归的残差平方和极小化问题, 调用编程软件自带的求解器也可以很快速地解决. 此外, 在上一步我们聚类后的信号点可以作为该优化问题的初值输入, 使得收敛速度更快. 在推导出目标函数的梯度以及黑塞矩阵以后, 也可以仅用牛顿法进行一次迭代, 以实现更快速的算法.

因此超分辨率压缩感知雷达定位算法的基本框架可以分成三个步骤, 图 6 展示了这三个步骤以及一个算例, 具体来说:

1. **压缩感知重构**: 对研究区域进行格点划分, 利用物体在空间位置上的稀疏性, 通过压缩感知的手段来求解稀疏信号, 其中为了降低算法复杂度我们可以考虑时空分块策略, 求解算法可以采用 ADMM 算法;
2. **聚类或社区发现增大稀疏度**: 考虑到超分辨率场景以及噪声的存在, 压缩感知重建的信号稀疏度可能不够 (但它们的整体分布是稀疏的), 利用密度峰值聚类或重叠的社区发现算法来确定物体 K 的数量以及改进稀疏信号;
3. **连续优化更新**: 为应对超分辨率需求, 在聚类结果的基础上对信号作进一步改进, 以达到超分辨率.

图 6 不仅展示了超分辨率压缩感知雷达定位算法的运行流程, 同时体现了该算法的有效性, 该算例的场景为目标物体数 $K = 10$, 且部分物体较为集中, 也存在处于研究区域边缘的物体, 从结果来看压缩感知重构就已经可以大概分辨出物体的位置, 而后续的聚类以及连续优化均可大幅度增加信号的精确度. 同时我们采用空间分块为 5 块, 时间分块为 10 块

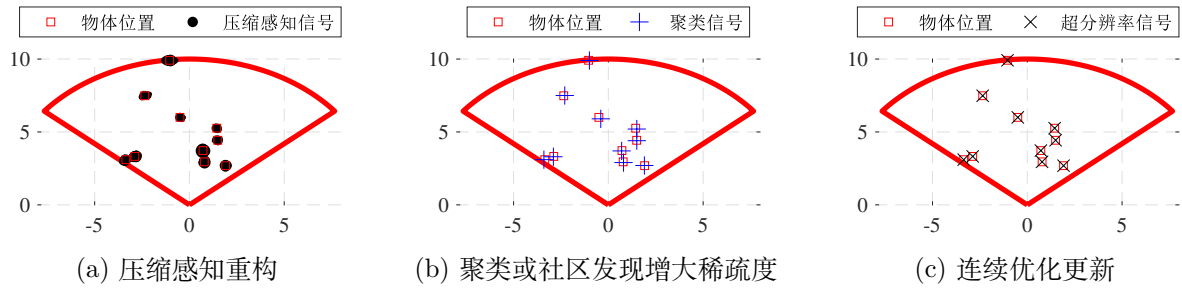


图 6: 超分辨率压缩感知雷达定位算法的基本框架, 以及在物体数 $K = 10$ 的模拟数据上的测试.

的 ADMM 算法, 10 核心并行计算只需花费 8.7691 秒的运算时间, 最终的重建信号误差为 0.0023.

2.3 不同场合的求解结果

本题的第 1、2、4 个子问题均属于静止物体的定位问题, 均可以利用本章提出的算法框架分别对这三个问题进行求解, 需要调整的参数只有噪声的功率 σ . 求解结果发现三个子问题均有二个物体, 且均属于重叠物体, 契合了超分辨率的切意. 图 7 绘制了在极坐标系下这三个问题的求解结果, 表 2 详细描述了求解结果.

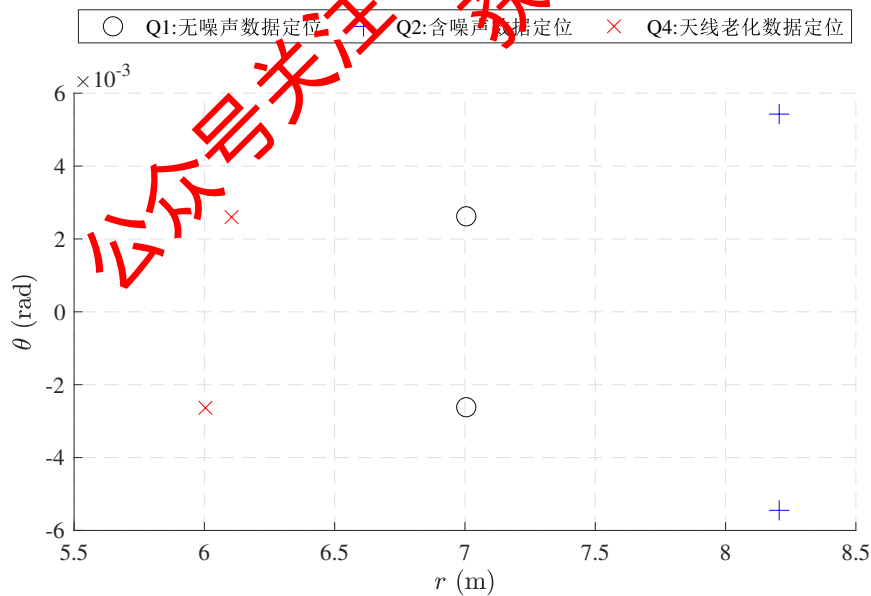


图 7: 第 1、2、4 问的定位结果极坐标图, 其中横坐标为极径 (到原点的距离), 纵坐标为极角 (以 y 轴方向为 0° , 顺时针为负角度, 逆时针为正角度). 黑色 \circ 是 Q1 的无噪声数据定位, 蓝色 $+$ 号是 Q2 的含噪声数据定位, 红色 \times 号是 Q4 的天线老化数据定位.

表 2: 第 1、2、4 问的详细定位结果, 每一问的物体数量皆为 2 个, 给出了物体的直角坐标、物体极坐标、物体的反射系数, 以及利用重建的位置数据生成观测信号与原始观测信号间的 ℓ_2 范数误差.

数据集	物体直角坐标 (x, y)	物体极坐标 (r, θ)	物体反射系数	观测信号的重建误差
1, 无噪声数据	(-0.0183, 7.0048)	(7.0048, -0.0026)	$3.9952 + 3.0058i$	0.0454
	(0.0184, 7.0048)	(7.0048, 0.0026)	$-2.9940 - 4.0046i$	
2, 含噪声数据	(-0.0447, 8.2056)	(8.2057, -0.0054)	$4.3719 + 2.0751i$	209.4977
	(0.0447, 8.2056)	(8.2058, 0.0054)	$-3.8427 - 2.9398i$	
4, 天线老化数据	(0.0159, 6.1042)	(6.1042, 0.0026)	$-2.5414 - 4.2246i$	169.0732
	(-0.0158, 6.0041)	(6.0041, -0.0026)	$3.8456 + 3.0986i$	

公众号关注：建模忠哥
获取更多资源

3 运动物体的超分辨率追踪算法

运动物体的高分辨率追踪是静止物体高分辨率定位的深入问题, 考虑多物体在一帧内的运动, 假设一帧包含 N_f 个 chirp 周期, 每个 chirp 周期均能观测到中频信号组, 第 q 个 chirp 周期观测到的信号向量记作 $\mathbf{Z}_q, q = 0, 1, \dots, N_f - 1$, 它和第 2 章中的定义是类似的. 显然我们需要求解初始状态, 即求解第 0 个 chirp 周期中静止物体的高分辨率定位问题, 这一问题可利用第 2 章的算法框架进行求解, 该过程也可被称为“预热阶段”, 需要耗费一定的时间, 而后续的追踪则必须是高分辨率的在线算法, 否则没有在现实场合运用的可能性.

实际上, 我们需要使用上一 chirp 周期或前几个 chirp 周期已经寻找到的物体位置, 并结合当前采集的信号数据设计快速的物体在线追踪算法. 因此该任务可以归结至预测信息 (前几 chirp 周期的信息) 与观测数据 (当前 chirp 周期采集到的信号) 的同化, 可以考虑结合常用的数据同化方法.

3.1 数据同化方法简介

数据同化 (Data Assimilation) 是指在考虑观测数据和预报误差的同时, 于动态数值模型中集成观测数据, 以优化状态变量的估计. 对于混沌系统, 用优化的状态变量估计值作为初始值进行预报, 可以显著提高预报精度 [13]. 数据同化起源于海洋和数值天气预报, 已被广泛应用于许多科学领域, 如水文、遥感、大气环境和其它地球科学, 也包括自动驾驶等自动化工程.

目前通行的数据同化方法主要有两种: 四维变分 [1] 和集合 Kalman 滤波 [7]. 现设随机向量 $\mathbf{X}_q \in \mathbb{R}^{2K}$ 是 q 时刻或第 q 个周期的物体位置状态变量, 例如当前有 K 个物体需要被追踪, 则

$$\mathbf{X}_q = (x_{1,q}, y_{1,q}, \dots, x_{K,q}, y_{K,q})^\top, \quad (9)$$

其中 $x_{i,q}, y_{i,q}$ 分别表示第 q 个 chirp 周期时第 i 个物体的横、纵坐标. 而 $\mathbf{Z}_q \in \mathbb{C}^L$ 是第 q 个 chirp 周期的一组观测变量, 在当前场景下即为第 q 个 chirp 周期接收到的所有信号, 我们将其拉直成为向量, 因此如果我们使用全部的信号数据, 则有 $L = 2 \times N_a \times M$. 其中 N_a 是天线数量, M 是个 chirp 周期的时刻数, 同时我们将实数信号以及复数信号分开考虑. 此外也可以不使用所有数据, 而对收集到的信号数据进行采样, 只使用部分数据, 此举有助于减少计算量, 当然也有可能造成精度损失.

假设 $\mathbf{X}_q, \mathbf{Z}_q$ 满足下面的预报-观测方程:

$$\begin{aligned} \mathbf{X}_q &= \mathbf{M}_q \mathbf{X}_{q-1} + \mathbf{w}_q, \quad \mathbf{w}_q \sim \mathcal{N}_{2K}(0, \mathbf{Q}_q), \\ \mathbf{Z}_q &= \mathbf{H}_q(\mathbf{X}_q) \cdot \mathbf{X}_q + \mathbf{v}_q, \quad \mathbf{v}_q \sim \mathcal{N}_L(0, \mathbf{R}_q). \end{aligned} \quad (10)$$

其中 $\mathbf{M}_q, \mathbf{H}_q(\mathbf{X}_q)$ 分别为预报算子和观测算子. 在本问题的场景下, 观测算子依赖于状

态变量. $\mathbf{w}_q, \mathbf{v}_q$ 分别为模型误差和观测误差且服从高斯分布. 我们用 $\mathcal{N}_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 表示均值为 $\boldsymbol{\mu}$, 方差为 $\boldsymbol{\Sigma}$ 的 m 维高斯分布.

数据同化的目标是在观测到 \mathbf{Z}_q 的基础上给出 \mathbf{X}_q 的一个优化估计, 记作 \mathbf{X}_q^a , 也称其为分析状态 (analysis state).

Kalman 滤波 [12] 的数学基础是多元正态分布下对条件均值的最小方差无偏估计理论, 其理论假设预报算子和观测算子均是线性的. Kalman 滤波包括两步: 基于模型的预测, 以及结合新的观测之后的更新. 令 $\mathbf{Z}_{1:q-1}$ 为第 q 个周期之前观测的集合, 真实物体状态变量 \mathbf{X}_{q-1} 在历史观测 $\mathbf{Z}_{1:q-1}$ 下的条件分布为 $\mathbf{X}_{q-1} | \mathbf{Z}_{1:q-1} \sim \mathcal{N}_{2K}(\boldsymbol{\mu}_{q-1}^a, \boldsymbol{\Sigma}_{q-1}^a)$, Kalman 滤波首先基于历史观测将 q 时的条件分布预测为 $\mathbf{X}_q | \mathbf{Z}_{1:q-1} \sim \mathcal{N}_{2K}(\boldsymbol{\mu}_q^f, \boldsymbol{\Sigma}_q^f)$, 其中

$$\boldsymbol{\mu}_q^f = \mathbf{M}_q \boldsymbol{\mu}_{q-1}^a, \quad \boldsymbol{\Sigma}_q^f = \mathbf{M}_q \boldsymbol{\Sigma}_{q-1}^a \mathbf{M}_q^\top + \mathbf{Q}_q, \quad (11)$$

这里上标 f 表示预测 (forecast). 第二步则是数据同化的体现, 利用 $(\mathbf{X}_q, \mathbf{Z}_q)$ 的联合正态分布, 将条件分布 $\mathbf{X}_{q-1} | \mathbf{Z}_{1:q-1}$ 更新为 $\mathbf{X}_q | \mathbf{Z}_{1:q} \sim \mathcal{N}_{2K}(\boldsymbol{\mu}_q^a, \boldsymbol{\Sigma}_q^a)$. 其中

$$\boldsymbol{\mu}_q^a = \boldsymbol{\mu}_q^f + \mathbf{K}_q (\mathbf{Z}_q - \mathbf{H}_q \mathbf{X}_q^f), \quad \boldsymbol{\Sigma}_q^a = (\mathbf{I}_q - \mathbf{K}_q \mathbf{H}_q) \boldsymbol{\Sigma}_q^f, \quad \mathbf{K}_q = \boldsymbol{\Sigma}_q^f \mathbf{H}_q^\top (\mathbf{H}_q \boldsymbol{\Sigma}_q^f \mathbf{H}_q^\top + \mathbf{R}_q)^{-1}, \quad (12)$$

这里的 \mathbf{K}_t 被称为 Kalman 增益矩阵.

四维变分 (4D Var) 的分析状态是以下目标函数的最小值点

$$J(\mathbf{X}) = \frac{1}{2} (\mathbf{X}_0 - \mathbf{X}^b)^\top \mathbf{B}^{-1} (\mathbf{X}_0 - \mathbf{X}^b) + \frac{1}{2} \sum_{i=1}^q (\mathbf{H}_i \mathbf{X}_i - \mathbf{Z}_i)^\top \mathbf{R}_i^{-1} (\mathbf{H}_i \mathbf{X}_i - \mathbf{Z}_i) + \frac{1}{2} \sum_{i=1}^q (\mathbf{X}_i - \mathbf{M}_i \mathbf{X}_{i-1})^\top \mathbf{Q}_i^{-1} (\mathbf{X}_i - \mathbf{M}_i \mathbf{X}_{i-1}), \quad (13)$$

其中 $\mathbf{X} = \{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_q\}$, \mathbf{X}^b 是背景状态变量 (通常可取作从上一时刻的预报), \mathbf{B} 是对应的背景误差协方差阵. 目标函数 $J(\mathbf{X})$ 的三项分别反映状态变量与初步估计的误差、与观测变量的误差以及累积误差. 四维变分方法的优势在于可以处理非线性观测算子以及同化预报多时刻的状态变量, 但该方法的缺陷是没有对 \mathbf{B} 和 \mathbf{Q} 这两个方差矩阵的估计方案. 并且 4D Var 需要求解高维非线性优化问题, 计算量较大且可能出现多个局部极小值 [14].

上述两种方法是数据同化领域最常用的方法, 但均未在本问题中起到好的效果, 其中 Kalman 滤波存在的问题是此时观测变量 \mathbf{Z} 的维度 (L) 要远大于状态变量 \mathbf{X} 的维度 ($2K$), 并且观测算子 \mathbf{H} 强非线性, 为超高维向量值函数. 若考虑扩展卡尔曼滤波, 用雅可比矩阵来估计观测算子, 其近似效果将会非常差. 而 4D Var 的主要缺陷在于计算量太大, 并且难以找到背景噪声协方差矩阵 \mathbf{B} 以及累计误差协方差矩阵 \mathbf{Q} 的良好估计, 因此难以运用. 为解决运动物体的高分辨率追踪问题, 我们需要设计更适用于高分辨率目标追踪问题、复杂性更低的数据同化算法.

3.2 超分辨率目标追踪算法

超分辨率目标追踪的本质是快速地重建信号, 而本文第 2 章提供的算法虽然效率优秀, 但仍然不支持在线算法, 其求解一个 chirp 周期内两个物体的重建问题大概需要花费 3 秒, 而一个周期的持续时间是远远小于 3 秒的. 本小节致力于在动态的场合改进前文的算法.

3.2.1 局部压缩感知

第 2 章提到压缩感知算法在超分辨率场合的局限性主要在于其计算复杂度的增长与分辨率的增加是平方级别的关系, 因此无法不加限制地增加分辨率. 但在目标追踪场合, 我们可以同化上一个 chirp 周期的数据, 同时也考虑到相邻周期的物体位置变化不会太剧烈, 利用这一信息可以帮助我们缩小压缩感知的格子点范围, 以便于更快速的找到信号. 具体来说, 假设我们在一帧内的第 q 个周期重建出 K 个物体的坐标为 $\{(x_{i,q}, y_{i,q})\}_{i=1,\dots,K}$. 由于每个物体在相邻两个 chirp 周期上的位置不会相差太大, 因此可以在第 $q+1$ 个周期时缩小压缩感知的搜索范围, 记第 i 个物体的近邻区域为

$$R_{i,q} = [x_{i,q} - \varepsilon_{x,i}, x_{i,q} + \varepsilon_{x,i}] \times [y_{i,q} - \varepsilon_{y,i}, y_{i,q} + \varepsilon_{y,i}], \quad (14)$$

其中 $\varepsilon_{x,i}, \varepsilon_{y,i}$ 由第 i 个物体在横、纵坐标上的运动速度决定. 一般而言, 考虑到两个 chirp 周期的相邻时间极短, 我们通常可以将所有的 $\varepsilon_{x,i}, \varepsilon_{y,i}$ 取作一个较小的常数, 例如 $\varepsilon = 0.5$. 可以认为第 i 个物体在第 $q+1$ 个 chirp 周期时的位置落在 $R_{i,q}$ 内. 因此第 $q+1$ 个 chirp 周期的压缩感知格子点可取作 $\bigcap_{i=1}^K \mathfrak{D}_h(R_{i,q})$, 其中 $\mathfrak{D}_h(R)$ 为将区域 R 按照分辨率 h 进行格子点划分的划分算子.

这一方法可以形象地被称为局部压缩感知, 即利用先验信息来对格子点的范围进行缩减并增大分辨率, 图 8 给出了一个例子, 可以看出相比于原始格子点, 局部压缩感知格子点可以在不提升计算复杂度 (主要依赖于用于求解压缩感知物体的格子点数量) 的前提下, 大幅度提升了分辨率, 并且由于它排除了无效点的干扰, 也更容易快速求解出真实信号.

3.2.2 Dropout

在第 2 章中我们考虑了时空分块的压缩感知算法, 目的是在不丢弃样本信息的同时提升计算的速度, 但实际上相对于稀疏信号的维度 K 来说, 观测信号的维度 $L = N_a \times M$ 显然是冗余的, 是可以适当舍弃的. 另外, 当我们采取了局部压缩感知的策略后, 真实信号的定位变得精确, 愈发不需要过多的冗余数据, 所以可以适当的对原始数据进行删减, 以期望提升算法速度. 在算法 2 的框架下, 借鉴卷积神经网络 (CNN) 中的 Dropout 层, 我们提出分块算法的 Dropout 策略, 在进行第 κ 个分块的 ℓ_1 范数问题求解前, 生成随机数 $\xi_\kappa \in \mathcal{U}(0, 1)$, 则

- 若 $\xi_\kappa < \gamma$, 跳过该块;

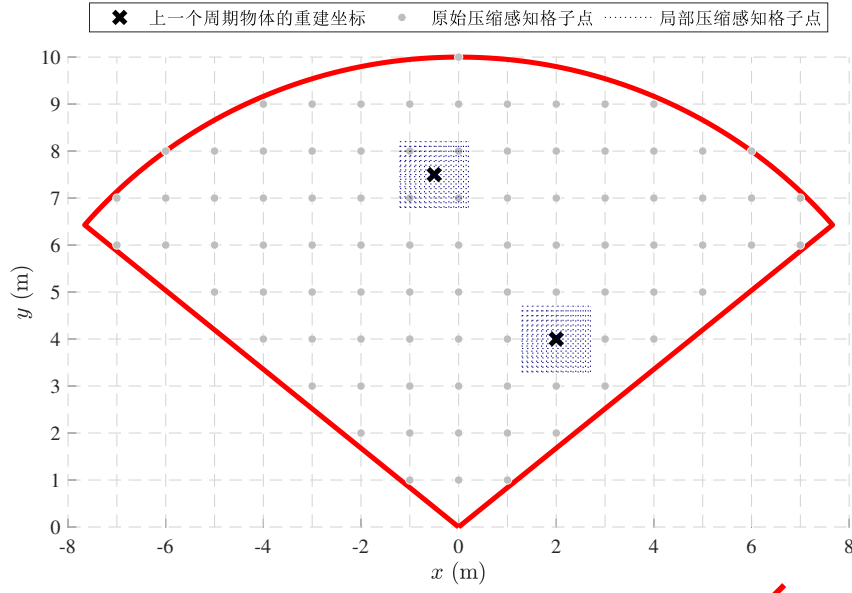


图 8: 局部压缩感知的例子. 其中灰色点代表原始压缩感知格子点, 而黑色 'x' 代表在上一个周期重建的物体坐标 (该例中物体数 $K = 2$), 蓝色 '·' 为局部压缩感知格子点, 它们是对上一个周期重建的两个物体坐标的邻近区域的更细致的划分.

- 若 $\xi_k \geq \gamma$, 运行算法, 并修正该块中求解得到的信号强度: $\alpha_k = \frac{\alpha_k}{1-\gamma}$.

其中 $\gamma \in [0, 1)$ 为 Dropout 参数, 容易证明 Dropout 操作保证了生成信号的无偏性. 算法 3 给出了加入局部压缩感知以及 Dropout 操作后的超分辨率追踪物体算法, 需要指出的是引入局部压缩感知后, ADMM 算法的收敛速度将会更快, 因此我们可以减少每个分块上 ADMM 算法的最大迭代次数.

3.3 任务求解

本小节将算法 3 用在一个模拟数据集上并检验其效果, 然后将该算法用于求解问题所给的一帧数据的.

3.3.1 模拟数据上的结果

我们生成了两个物体在一帧内的非直线运动轨迹 (见图 9a), 并模拟产生一帧的观测信号, 算法 3 的重建结果见图 9b, 结果显示算法的重建效果较好. 该例的运行采用了 20 个时间分块以及 20 个空间分块, Dropout 参数 ρ 取 0.8, ADMM 算法的最大迭代次数取 1, 即仅进行一步迭代. 最终算法在 32 个周期上的平均 ℓ_2 损失是 6.04×10^{-4} , 算法平均运行时间是 0.163 秒. 表 3 汇报了算法的详细运行情况, 包含了每一个周期重建的 ℓ_2 范数损失以及每一个周期上的运行时间以及两个物体的重构位置距其真实位置的欧几里得距离.

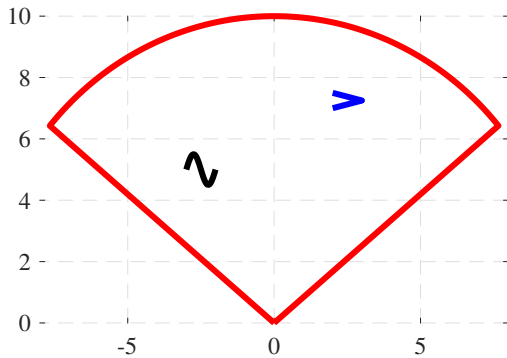
算法 3: 超分辨率在线物体追踪算法

Input: 一帧内 N_f 个 chirp 周期的观测信号 $\mathbf{Z}_0, \mathbf{Z}_1, \dots, \mathbf{Z}_{N_f-1}$, 分块数目 B , 初始每个分块上的感知矩阵 $\mathbf{E}_0^{(1)}, \dots, \mathbf{E}_0^{(B)}$, 参数 μ , ADMM 算法的最大迭代次数 MaxIter , 信号的丢弃阈值 ρ , Dropout 参数 γ , 物体的移动参数 ε .

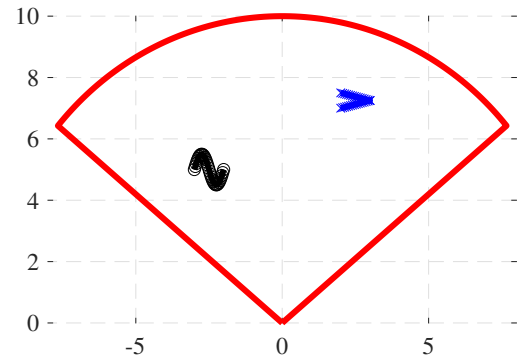
```

1  $q \leftarrow 0$ ;
2 利用初始信号  $\mathbf{Z}_0$ , 构建每个分块上的初始信号向量  $\mathbf{Z}_0^{(1)}, \dots, \mathbf{Z}_0^{(B)}$ ;
3 调用算法 2 求解信号  $\alpha$ , 并利用密度峰值聚类求解  $K$  以及进一步稀疏化  $\alpha$ , 再利用连续优化更新  $\alpha$ ;
4 通过  $\alpha$  的非 0 元位置确定初始物体的坐标  $\{(x_{i,0}, y_{i,0})\}_{i=1, \dots, K}$ ;
5 for  $q = 1, 2, \dots, N_f - 1$  do
6   for  $i = 1, 2, \dots, K$  do
7      $R_{i,q} \leftarrow [x_{i,q} - \varepsilon_{x,i}, x_{i,q} + \varepsilon_{x,i}] \times [y_{i,q} - \varepsilon_{y,i}, y_{i,q} + \varepsilon_{y,i}]$ ;
8     计算压缩感知格子点  $\bigcap_{i=1}^K \mathcal{D}_h(R_{i,q})$ ;
9     构建每个分块上的信号向量  $\mathbf{Z}_q^{(1)}, \dots, \mathbf{Z}_q^{(B)}$ ;
10    利用压缩感知格子点构造分块信号矩阵  $\mathbf{E}_q^{(1)}, \dots, \mathbf{E}_q^{(B)}$ ;
11     $\alpha \leftarrow 0_{L_q}$ , 其中  $L_q$  是当前压缩感知格子点的数量;
12    for  $\kappa = 1, 2, \dots, B$  do
13      if  $\text{rand}(1) \leq \gamma$  then
14        Continue;
15       $\alpha_\kappa \leftarrow \text{ADMM\_CS}(\mathbf{Z}_\kappa, \mathbf{E}_\kappa, \mu, \text{MaxIter})$ ;
16       $\alpha \leftarrow \alpha + \alpha_\kappa / (1 - \gamma)$ ;
17     $\text{maxSignal} \leftarrow \max(\alpha)$ ;
18    for  $l = 1, 2, \dots, L_q$  do
19      if  $|\alpha[l]| \leq \rho \cdot \text{maxSignal}$  then
20         $\alpha[l] \leftarrow 0$ ;
21    利用连续优化更新  $\alpha$ ;
22    通过  $\alpha$  的非 0 元位置确定物体的坐标  $\{(x_{i,q}, y_{i,q})\}_{i=1, \dots, K}$ ;

```



(a) 模拟生成的两个物体一帧内的运动路径



(b) 算法3重建的两个物体一帧内的运动路径

图 9: 利用算法 3 在一帧物体数量 $K = 2$ 的运动物体数据上的求解结果.

表 3: 算法 3 在模拟数据集上的表现, 评价指标有每一个周期上的重建观测信号于真实观测信号间的 ℓ_2 范数误差、每一个周期上的算法求解时间以及两个物体的重构位置距其真实位置的欧几里得距离.

周期	ℓ_2 误差	cpu time	物体 1 差距	物体 2 差距	周期	ℓ_2 误差	cpu time	物体 1 差距	物体 2 差距
0	0.0005	0.0554	9.31E-08	9.78E-09	16	0.0008	0.2074	6.32E-08	7.34E-08
1	0.0007	0.0871	1.88E-07	5.80E-08	17	0.0007	0.0198	1.63E-07	3.73E-08
2	0.0006	0.1563	1.13E-07	7.77E-08	18	0.0008	0.1417	5.26E-08	2.81E-08
3	0.0005	0.2479	1.44E-07	9.30E-08	19	0.0006	0.1471	1.29E-07	2.16E-08
4	0.0006	0.0830	1.17E-07	6.87E-08	20	0.0005	0.2398	1.13E-07	3.21E-08
5	0.0006	0.2624	1.39E-07	7.32E-08	21	0.0006	0.0204	9.60E-08	3.58E-08
6	0.0006	0.2387	9.28E-08	5.25E-09	22	0.0001	0.1284	3.60E-08	5.28E-08
7	0.0007	0.1899	1.35E-07	1.17E-07	23	0.0007	0.1345	1.09E-07	1.31E-07
8	0.0006	0.0804	7.65E-08	1.40E-07	24	0.0006	0.3000	6.23E-08	1.93E-08
9	0.0007	0.2540	1.71E-07	6.57E-08	25	0.0006	0.1331	7.81E-08	5.95E-08
10	0.0007	0.0787	8.26E-08	3.92E-08	26	0.0007	0.0758	2.17E-07	2.82E-08
11	0.0007	0.3613	1.28E-07	8.02E-09	27	0.0007	0.1910	5.42E-08	4.00E-08
12	0.0006	0.0836	8.94E-08	7.49E-08	28	0.0005	0.1295	1.24E-07	5.14E-08
13	0.0006	0.2027	6.38E-08	4.30E-08	29	0.0005	0.1295	1.09E-07	4.72E-08
14	0.0005	0.1955	1.43E-07	1.16E-07	30	0.0005	0.2432	1.35E-07	5.64E-08
15	0.0005	0.2089	2.25E-07	2.17E-08	31	0.0007	0.1901	1.11E-07	2.89E-08

该模拟数据集的算例说明了算法 3 的有效性, 可以以超分辨率成功追踪非线性的多物体运动轨迹, 虽然相较于一个 chirp 周期 (50 毫秒) 而言, 其运行时间仍然有改善空间, 但基本可以对时间分辨率不低于 0.5s 的场合进行在线追踪.

3.3.2 问题数据的结果

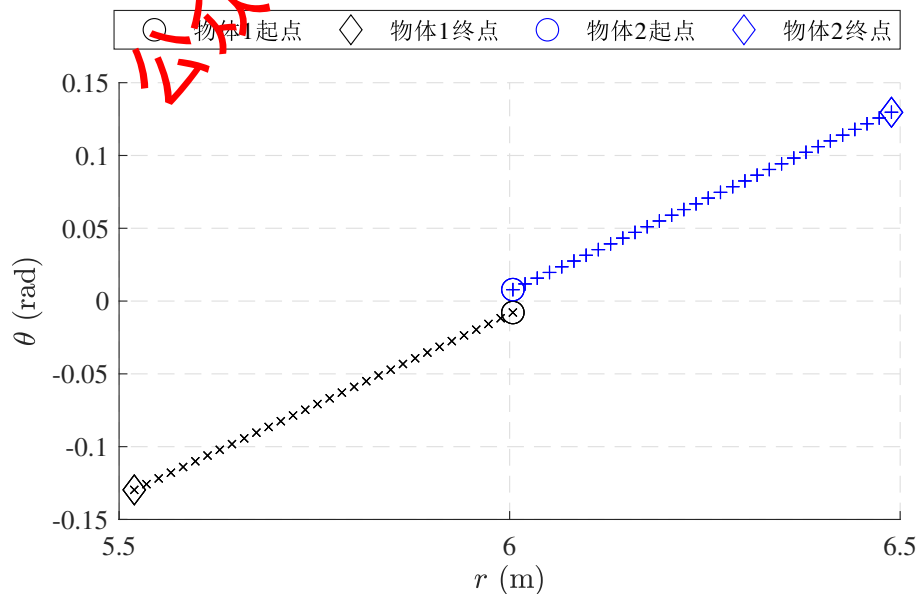


图 10: 第 3 问的物体运动轨迹极坐标图, 其中横坐标为极径 (到原点的距离), 纵坐标为极角 (以 y 轴方向为 0° , 顺时针为负角度, 逆时针为正角度).

使用算法 3 对问题所给的一帧数据进行求解, 解得物体数量 $K = 2$, 图 10 给出了两个物体运动轨迹的极坐标图, 两个物体的初始坐标非常接近, 属于超高分辨率问题的场景. 两个物体的运动轨迹呈现出相反方向的直线趋势. 最终算法在 32 个周期上的平均 ℓ_2 损失是 0.203, 算法平均运行时间是 0.1231 秒. 表 10 汇报了算法的详细运行情况, 包含了每一个周期重建的 ℓ_2 范数损失以及每一个周期上的运行时间.

表 4: 问题 3 的求解结果, 评价指标包含了每一个周期上的重建观测信号于真实观测信号间的 ℓ_2 范数误差以及每一个周期上的算法求解时间.

周期	ℓ_2 误差	cpu time	周期	ℓ_2 误差	cpu time
0	0.1436	0.1133	16	0.2054	0.1254
1	0.1552	0.0621	17	0.2064	0.0507
2	0.2013	0.1065	18	0.2063	0.1792
3	0.2112	0.2972	19	0.2064	0.1192
4	0.2064	0.1357	20	0.2065	0.1219
5	0.2100	0.0235	21	0.2066	0.0214
6	0.2054	0.2510	22	0.2069	0.1089
7	0.2047	0.1201	23	0.2066	0.1808
8	0.2038	0.1635	24	0.2068	0.1412
9	0.2062	0.1131	25	0.2068	0.1422
10	0.2065	0.1367	26	0.2069	0.1126
11	0.2048	0.1412	27	0.2072	0.0625
12	0.2061	0.0570	28	0.2073	0.2261
13	0.2065	0.1152	29	0.2075	0.1066
14	0.2067	0.1038	30	0.2074	0.0457
15	0.2058	0.1072	31	0.2075	0.1489

4 灵敏度分析

为检验算法的鲁棒性, 本章考虑在图 6 对应的模拟数据上增加不同强度的噪声, 并再利用超分辨率雷达定位算法进行求解.

4.1 低噪声

在图 6 对应的物体数 $K = 10$ 的模拟数据上增加信号强度为 1 的噪声, 即在观测信号上施加标准差为 1 的正态分布, 利用超分辨率雷达算法的求解过程以及求解结果由图 11 给出, 可以看出求解结果与图 6 的结果并无太大区别. 因此可以认为该算法在低噪声的场合是有效的. 该场合下的观测信号重建 ℓ_2 范数误差是 149.2470.

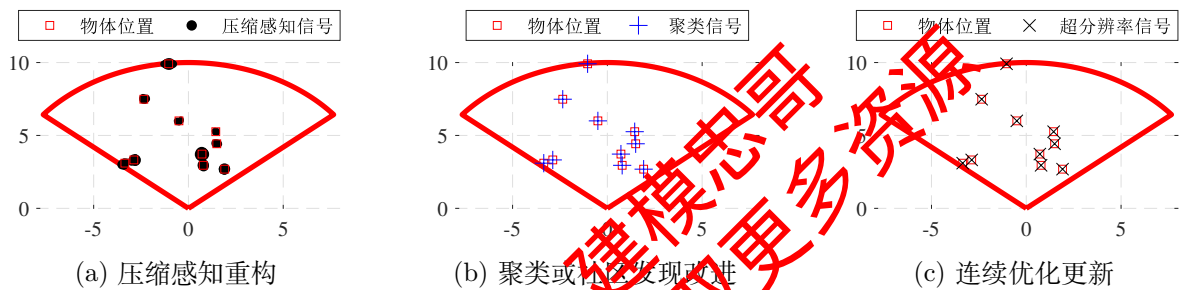


图 11: 超分辨率压缩感知雷达定位算法在物体数 $K = 10$, 噪声强度为 1 的模拟数据上的测试.

4.2 中噪声

在图 6 对应的物体数 $K = 10$ 的模拟数据上增加信号强度为 3 的噪声, 即在观测信号上施加标准差为 3 的正态分布, 利用超分辨率雷达算法的求解过程以及求解结果由图 12 给出, 除了在压缩感知重构结果与图 11 有些许差别外, 聚类以及连续优化更新的结果基本保持一致. 中噪声场合的观测信号重建 ℓ_2 范数误差是 447.7237.

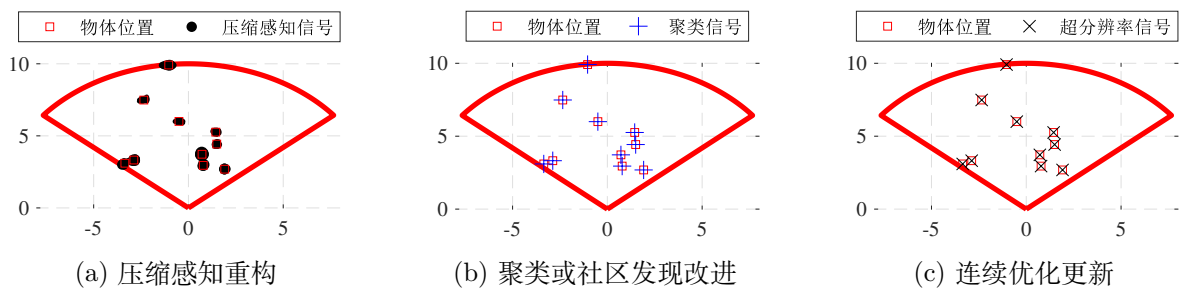


图 12: 超分辨率压缩感知雷达定位算法在物体数 $K = 10$, 噪声强度为 3 的模拟数据上的测试.

4.3 高噪声

在图 6 对应的物体数 $K = 10$ 的模拟数据上增加信号强度为 5 的噪声, 即在观测信号上施加标准差为 5 的正态分布, 注意此时物体的信号强度的模也基本不会超过 5, 所以可以认为该场景的噪声是大于信号强度的, 属于高噪声场合. 利用超分辨率雷达算法的求解过程以及求解结果由图 13 给出, 求解结果与前面两个场景基本一致, 高噪声场合的观测信号重建 ℓ_2 范数误差是 746.1775. 此时需要指出, 在高噪声场合, 使用较小的噪声功率参数 σ 较难得到好的结果, 因此此时我们在算法中增大了噪声功率参数 σ . 此外, 测试中也发现使用较大的噪声功率参数仍然可以很好地求解低噪声与中噪声问题.

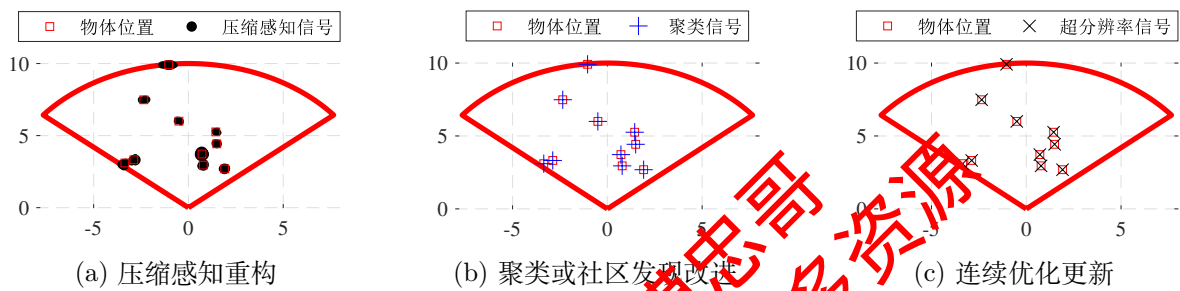


图 13: 超分辨率压缩感知雷达定位算法在物体数 $K = 10$, 噪声强度为 5 的模拟数据上的测试.

本章节的内容从数值上证明了本文算法的有效性与鲁棒性, 对于高、中、低噪声以及无噪声场景下的物体定位问题均可以在不修改太多算法参数的情况下顺利求解.

5 结论与展望

5.1 结论

雷达是国防作战和防御的重要武器, 在军事领域发挥着重要作用, 在民用领域的作用也越来越大. 定位是雷达最基本的功能, 而在人工智能与大数据的时代背景下, 研发超分辨率的雷达定位算法具有重大的意义. 本文基于压缩感知理论, 建立了一套用于雷达定位、追踪的方法理论, 实现了超分辨率定位算法以及低运算量的在线追踪算法.

本文针对连续波雷达 FMCW 的目标定位以及追踪算法进行了研究, 作出的主要贡献包括以下几个方面:

1. 阐述了压缩感知的理论背景以及其在雷达领域中的应用现状及前景, 并指出了其在实现超分辨率方面的局限性.
2. 提出了时空分块的压缩感知算法, 雷达天线的测量信号相对于真实稀疏信号而言实际是冗余数据, 文章中验证了只用部分数据对信号进行重建的可行性, 因此使用时空分块的策略可以大幅度提升效率而不影响求解的效果.
3. 提出了使用聚类或社区发现的方法来对压缩感知的解进一步稀疏化, 这一做法旨在解决物体重叠或噪声太大导致压缩感知的解不够稀疏的问题, 其中改进的密度峰值聚类以及社区发现算法可以不指定物体数量 K .
4. 指出了格点划分对于实现超分辨率尤其是快速算法的不可行性, 从而引出用连续函数优化的方法最后对物体位置进行更新以得到超分辨率的重建结果.
5. 将运动物体的超分辨率追踪算法归结至数据同化问题, 并且指出了目前常用的数据同化方法 (例四维变分, Kalman 滤波等) 在超分辨率追踪问题上难以取得好的效果.
6. 提出了以局部压缩感知为基础的数据同化方法, 将压缩感知的备选格子点的空间范围进行缩减, 并增大压缩感知的分辨率, 这有利于快速、高精度的求解压缩感知问题.
7. 发现了分块压缩感知算法的结构可以引入了深度学习中常用的 Dropout 操作, 此举可以很大程度地降低算法的运行时间, 同时深度学习的经验也告诉我们这有助于减少过拟合.
8. 实现了超分辨率的在线目标追踪算法.

5.2 工作展望

由于雷达系统的复杂性和涉及知识面的广泛性, 本文在许多方面可能存在疏漏并且在很多细节方面未展开叙述, 且压缩感知理论在雷达中的应用还不完全成熟, 仍有很多问题需要被解决. 在此提出本文的不足之处以及未来可以继续研究的方向:

1. 本文用于模拟的雷达回波是在理想环境下产生的, 忽略了外部环境的影响. 因此, 接下来的工作我们将模拟更接近实际环境的雷达回波, 并找到消除干扰信号的有效方法. 或直接通过现场实验采集数据进行算法测试.
2. 超分辨的物体定位算法中的聚类或社区发现这一步骤的稳定性仍然有待更进一步的讨论, 如果未发现正确的物体数 K , 则不可能做到后续的超分辨率重建, 需要改进现有算法或开发更稳定有效的算法来实现该步骤.
3. 将运动物体的超分辨率追踪视作数据同化问题, 但难以将现有的数据同化方法用在该问题上. 后续应该考虑更新的一些数据同化方法.
4. 本文提出的在线算法虽可以达到较高的求解速度, 但想要在一个 chirp 周期 (约为 50 微秒) 内完成求解仍有较大差距.

基于压缩感知理论的雷达定位、追踪算法可以利用物体在空间上的稀疏性, 将定位问题转化为信号重建问题, 这很大程度上简化了雷达信号处理的过程, 只需少量的观测值即可准确估计出目标物体的位置. 这种独特的处理方式势必会在未来的雷达信号处理领域大放异彩. 本文的研究内容很大程度上解决了压缩感知在“超分辨率”以及“低复杂度”这两个方面的局限性, 但想要在真正复杂的场景 (例如自动驾驶、军事领域等) 发挥作用, 仍有一段路要走.

参考文献

- [1] R. N. Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):607–633, 2017. doi: <https://doi.org/10.1002/qj.2982>.
- [2] E.J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. doi: 10.1109/TIT.2005.858979.
- [3] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006. doi: <https://doi.org/10.1002/cpa.20124>.
- [4] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998. doi: 10.1137/S1064827596304010.
- [5] Zhuanlian Ding, Xingyi Zhang, Dengdi Sun, and Bin Luo. Overlapping Community Detection based on Network Decomposition. *Scientific Reports*, 6(1):24115, 2016. ISSN 2045-2322. doi: 10.1038/srep24115.
- [6] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. doi: 10.1109/TIT.2006.871582.
- [7] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994. doi: <https://doi.org/10.1029/94JC00572>.
- [8] Daquan Feng, Chunqi Wang, Chunlong He, Yuan Zhuang, and Xiang-Gen Xia. Kalman-Filter-Based Integration of IMU and UWB for High-Accuracy Indoor Positioning and Navigation. *IEEE Internet of Things Journal*, 7(4):3133–3146, 2020. doi: 10.1109/JIOT.2020.2965115.
- [9] Stefano Fortunati, Raffaele Grasso, Fulvio Gini, and Maria Sabrina Greco. Single snapshot DOA estimation using compressed sensing. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2297–2301, 2014. doi: 10.1109/ICASSP.2014.6854009.
- [10] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Task-Driven Super Resolution: Object Detection in Low-Resolution Images. In Teddy Mantoro, Minhoo Lee, Media Anugerah Ayu, Kok Wai Wong, and Achmad Nizar Hidayanto, editors, *Neural Information Processing*, 2021.

- [11] Matthew A. Herman and Thomas Strohmer. High-Resolution Radar via Compressed Sensing. *IEEE Transactions on Signal Processing*, 57(6):2275–2284, 2009. doi: 10.1109/TSP.2009.2014277.
- [12] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. ISSN 0021-9223. doi: 10.1115/1.3662552.
- [13] Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of Atmospheric Sciences*, 20(2):130–148, March 1963. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.
- [14] Carlos Pires, Robert Vautard, and Olivier Talagrand. On extending the limits of variational assimilation in nonlinear chaotic systems. *Tellus A*, 48(1):96–121, 1996. doi: <https://doi.org/10.1034/j.1600-0870.1996.00006.x>.
- [15] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. doi: 10.1126/science.1242072.
- [16] Fabian Roos, Philipp Hügler, Lizette Lorraine Tovar Tones, Christina Knill, Johannes Schlichenmaier, Claudia Vasanelli, Nils Appel, Jürge Dickmann, and Christian Waldschmidt. Compressed Sensing based Single Snapshot DoA Estimation for Sparse MIMO Radar Arrays. *2019 12th German Microwave Conference (GeMiC)*, pages 75–78, 2019.
- [17] Inam Ullah, Yu Shen, Xin Shao, Christian Esposito, and Chang Choi. A Localization Based on Unscented Kalman Filter and Particle Filter Localization Algorithms. *IEEE Access*, 8:2233–2246, 2020. doi: 10.1109/ACCESS.2019.2961740.
- [18] Peng Xiao, Ping Chen, and Bin Liao. ADMM-based approach for compressive sensing with negative weights. *IET Signal Processing*, 14, 12 2020. doi: 10.1049/iet-spr.2020.0276.
- [19] 李芳华. 基于压缩感知的雷达目标定位算法研究. 2012.
- [20] 杨瑞峰 and 郭晨霞. 基于 UCA 压缩感知的声源定位算法. *电子测量技术*, 44(7):46–49, 2021.

附录

附录 1 RIP 条件

定义 1 (RIP 条件). 如果存在 $\delta_K \in (0, 1)$, 使得

$$(1 - \delta_K) \|\mathbf{x}\|_2^2 \leq \|\Phi \mathbf{x}\|_2^2 \leq (1 + \delta_K) \|\mathbf{x}\|_2^2$$

对所有的 K -稀疏向量 \mathbf{x} 都成立, 则矩阵 Φ 满足 K 阶约束等距条件 (RIP), 满足上式的最小常数 δ_K , 称为约束等距常数。

通常只有在 RIP 条件满足时, ℓ_1 范数问题的解与 ℓ_0 范数问题的解是等价的。

附录 2 复数变量的压缩感知

假设有复数变量压缩感知问题:

$$\hat{\alpha} = \arg \min_{\alpha} \|\alpha\|_1, \quad \text{s.t. } \|\mathbf{Z} - \mathbf{E}\alpha\|_2 \leq \delta,$$

其中 $\alpha \in \mathbb{C}^N$, $\mathbf{Z} \in \mathbb{C}^L$, $\mathbf{E} \in \mathbb{C}^{L \times N}$. 记

$$\tilde{\mathbf{E}} = \begin{pmatrix} \Re\{\mathbf{E}\} & -\Im\{\mathbf{E}\} \\ \Im\{\mathbf{E}\} & \Re\{\mathbf{E}\} \end{pmatrix}, \quad \tilde{\mathbf{Z}} = \begin{pmatrix} \Re\{\mathbf{Z}\} \\ \Im\{\mathbf{Z}\} \end{pmatrix}$$

这里 $\Re\{\cdot\}, \Im\{\cdot\}$ 分别表示取实部与取虚部操作. 而记 $\tilde{\alpha} = \begin{pmatrix} \alpha^r \\ \alpha^i \end{pmatrix}$ 为新的优化变量, 求解实变量的优化问题:

$$\min_{\tilde{\alpha}} \|\tilde{\alpha}\|_1, \quad \text{s.t. } \|\tilde{\mathbf{Z}} - \tilde{\mathbf{E}}\tilde{\alpha}\|_2 \leq \delta.$$

最后令 $\hat{\alpha} = \hat{\alpha}^r + \sqrt{-1}\hat{\alpha}^i$. 需要注意的是该解与原问题的解可能并不是等价的, 但由于我们采用了分块算法并且设置了信号丢弃阈值, 可以认为最终求得的结果是近似等价的。

附录 3 ℓ_1 正则化问题的 ADMM 算法推导

考虑 ℓ_1 正则化问题 (Primal):

$$\min_{\alpha} \frac{1}{2} \|\mathbf{Z} - \mathbf{E}\alpha\|_2^2 + \mu \|\alpha\|_1,$$

它等价于约束优化问题

$$\min_{\alpha} \frac{1}{2} \|\mathbf{r}\|_2^2 + \mu \|\alpha\|_1, \quad \text{s.t. } \mathbf{r} = \mathbf{Z} - \mathbf{E}\alpha.$$

该问题的拉格朗日函数是

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{r}, \boldsymbol{\lambda}) &:= \mu \|\boldsymbol{\alpha}\|_1 + \frac{1}{2} \|\boldsymbol{r}\|_2^2 - \boldsymbol{\lambda}^\top (\boldsymbol{Z} - \boldsymbol{E}\boldsymbol{\alpha}) \\ &= \mu \|\boldsymbol{\alpha}\|_1 - (\boldsymbol{E}^\top \boldsymbol{\lambda})^\top \boldsymbol{\alpha} + \frac{1}{2} \|\boldsymbol{r}\|_2^2 + \boldsymbol{\lambda}^\top \boldsymbol{r} + \boldsymbol{Z}^\top \boldsymbol{\lambda},\end{aligned}$$

因此对偶问题是

$$\max_{\boldsymbol{\lambda}} \boldsymbol{Z}^\top \boldsymbol{\lambda} - \frac{1}{2} \|\boldsymbol{r}\|_2^2, \text{ s.t. } \|\boldsymbol{E}^\top \boldsymbol{\lambda}\|_\infty \leq \mu.$$

而对偶问题又等价于

$$\max_{\boldsymbol{\lambda}} \boldsymbol{Z}^\top \boldsymbol{\lambda} - \frac{1}{2} \|\boldsymbol{r}\|_2^2, \text{ s.t. } \boldsymbol{E}^\top \boldsymbol{\lambda} = \boldsymbol{s}, \|\boldsymbol{s}\|_\infty \leq \mu,$$

构造增广拉格朗日函数

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{s}, \boldsymbol{\alpha}) = -\boldsymbol{Z}^\top \boldsymbol{\lambda} + \frac{1}{2} \|\boldsymbol{r}\|_2^2 + \boldsymbol{\alpha}^\top (\boldsymbol{E}^\top \boldsymbol{\lambda} - \boldsymbol{s}) + \frac{1}{2\mu} \|\boldsymbol{E}^\top \boldsymbol{\lambda} - \boldsymbol{s}\|_2^2.$$

因此有 ADMM 算法的迭代格式如下：

- $\boldsymbol{\lambda}^{(b)} = \arg \min_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{s}^{(b-1)}, \boldsymbol{\alpha}^{(b-1)}) = (\boldsymbol{E}^\top \boldsymbol{E} + \mu \boldsymbol{I})^{-1} (\boldsymbol{E}^\top (\boldsymbol{Z} - \boldsymbol{E}\boldsymbol{\alpha}^{(b-1)}) + \boldsymbol{E}\boldsymbol{s}^{(b-1)}).$
- $\boldsymbol{s}^{(b)} = \arg \min_{\boldsymbol{s}} \mathcal{L}(\boldsymbol{\lambda}^{(b)}, \boldsymbol{s}, \boldsymbol{\alpha}^{(b-1)}), \text{ s.t. } \|\boldsymbol{s}\|_\infty \leq \mu \in \mathcal{P}_{[-\mu, \mu]}(\boldsymbol{E}^\top \boldsymbol{\lambda}^{(b)} + \mu \boldsymbol{\alpha}^{(b-1)}).$
- $\boldsymbol{\alpha}^{(b)} = \boldsymbol{\alpha}^{(b-1)} + \frac{\boldsymbol{E}^\top \boldsymbol{\lambda}^{(b)} - \boldsymbol{s}^{(b)}}{\mu}.$

附录 4 求解 ℓ_2 正则化问题的 ADMM 算法 (Matlab R2020a)

```
1 function xx = ADMM_CS(eps, Hf, mu, M_Iter)
2 [~,N] = size(Hf);
3 D = [real(Hf), -imag(Hf); imag(Hf), real(Hf)];
4 yt = [real(eps); imag(eps)];
5 [n,m] = size(D);
6 invAAmu = inv(D*D' + mu * eye(n));
7 lambda0 = randn([n,1]);
8 s0 = D' * lambda0; s0 = s0 .* (s0 <= mu & s0 >= -mu) + mu * (s0 > mu) - mu * (s0 < -mu);
9 x0 = (1/mu) * (D'*lambda0 - s0);
10 for k = 1:M_Iter
11     lambda0 = invAAmu * (D * s0 - mu * (D * x0 - yt));
12     s0 = D' * lambda0 + mu * x0; s0 = s0 .* (s0 <= mu & s0 >= -mu) + mu * (s0 > mu) - mu * (s0 < -mu);
13     x0 = x0 + (1/mu) * (D'*lambda0 - s0);
14 end
15 xx = x0(1:N) + 1i * x0(N+1:end);
16 end
```