arXiv:2304.03146v1 [cs.DS] 6 Apr 2023

具有一般范数目标的聚类参数化近似方案

Fateme Abbasi* Sandip Banerjee* JarosławByrka*Parinya Chalermsook† AmeetGadekar† Kamyar Khodamoradi‡Dániel Marx§ Roohani Sharma¶ Joachim Spoerhase¶‖

摘要

本文探讨了一种研究已久的算法范式：为 k-聚类问题设计一个 $(1+\epsilon)$-近似算法，其运行时间为 $f(k,\ \epsilon)\text{poly}(n)$（有时称为高效参数化近似方案，简称 EPAS[1]）。此类代表性成果包括高维欧几里得空间中的 k-中心 [Bad˘oiu、Har-Peled、In- dyk；STOC'02]，以及 k-中位数和 k-均值 [Kumar、Sabharwal、Sen；J. ACM 2010]。

我们的主要贡献是一个简洁明了的高效参数化近似方案，它解决了十多个聚类问题（涵盖多个经过充分研究的目标函数以及度量空间），并统一了众所周知的高效多项式时间近似方案。更具体地说，我们的算法在以下设置中给出了高效多项式时间近似方案：

- **聚类目标**：k-均值、k-中心、k-中位数、优先级 k-中心、ℓ-中心点、有序 k-中位数、社会公平 k-中位数（又称鲁棒 k-中位数），或任何其他可表述为最小化点到解的距离的单调（不一定对称！）度量的目标（推广了 Chakrabarty 和 Swamy 在 [STOC'19] 中提出的对称表述）。
- **度量空间**：连续高维欧几里得空间、倍增维数有界的度量、树宽有界的度量以及平面度量。

在我们的研究成果之前，人们只在基础聚类目标（k-均值、k-中位数和 k-中心）中了解高效多项式时间近似方案，并且每种此类算法都是针对特定输入度量和聚类目标而设计的（例如，$\mathbb{R}^d$ 中的 k-均值和 k-中心的高效多项式时间近似方案在概念上截然不同）。相比之下，我们的算法框架以统一的方式适用于广泛的、经过深入研究的目标函数，并且（几乎）完全不依赖于任何特定的度量结构，同时仍能有效利用这些未知结构。特别是，我们的算法不 基于核心集这一（度量和目标函数特定的）技术。

我们的分析关键在于一个新概念，我们称之为有界 ϵ-散射维数——这是度量空间的一种内在复杂度度量，是对标准有界倍增维数 概念的放宽（后者常被用作几何问题具有算法可处理性的依据）。我们的主要技术结果表明，对于任何聚类目标函数，只需满足两个条件，我们的算法就基本上足以在输入度量空间 M 上实现高效参数化近似方案：

(i) 目标函数由单调范数描述；(ii) ϵ-散射维数 M 的上界是 ϵ的某个函数。

*弗罗茨瓦夫大学，波兰（{fateme.abbasi,jby}@cs.uni.wroc.pl，sandip.ndp@gmail.com）
†阿尔托大学，埃斯波，芬兰（{parinya.chalermsook,ameet.gadekar}@aalto.fi
‡不列颠哥伦比亚大学，加拿大（kamyar.khodamoradi@ubc.ca）§CISPA亥姆霍兹信息安全中心，萨尔布吕肯，德国（marx@cispa.de）
¶马克斯·普朗克信息学研究所，萨尔布吕肯，德国（{rsharma,jspoerha}@mpi-inf.mpg.de）
‖谢菲尔德大学，英国（j.spoerhase@sheffield.ac.uk）

目录

1 引言 11.1 范数 k-聚类的高效参数化近似方案 . . . . . 1 1.2 我们的概念与技术贡献.5**2 技术概述 8 3 预备知识 14**4 ϵ**-散射维数 16 5 高效参数化近似方案框架 19**5.1 算法.19 5.2 分析.21 5.2.1 迭代次数上界.22 5.2.2 成功概率上界.23

6 ϵ-散布维数界 26
6.1 有界倍增维数.26 6.2 有界树宽图.27 6.2.1 树宽与蜘蛛图.27 6.2.2 迭代查找蜘蛛图.28 6.2.3 引理 6.5 的证明.28 6.3 通过低树宽嵌入界定 ϵ-散射维数.30 6.4 高维欧几里得空间.31

7 结论与开放问题 **33**

1 引言

在 k-聚类问题中，我们关注将 n 数据点划分为 k 子集，这些子集称为簇，每个簇都由一个中心表示。我们的目标是最小化一个基于数据点与其各自簇中心之间距离的特定目标函数。这是理论和实践中经常出现的最基本优化问题之一，并且受到了包括优化、数据挖掘、机器学习和计算几何在内的多个研究领域的关注。诸如 k-Median、 k- 中心和 k-Means 等基本聚类问题已经被研究了半个多世纪，但从计算的许多角度来看，它们仍然难以解决。

本文考察了针对 k-聚类的一种重要且经典的算法范式，其目标是设计高效参数化近似方案（EPAS）——一种（$1+\epsilon$）近似算法，该算法对于每个 $\epsilon > 0$ 都能在 $h(k,\epsilon)\text{poly}(n)$ 时间内运行。在一般度量空间中，即使对于基本的聚类问题，获得这样的近似方案也是不可能的。因此，过去的研究一直专注于设计适用于结构化 度量空间（例如平面图或欧几里得空间）的算法。在连续高维欧几里得空间中，EPASes 可以说是人们所能期望的“最快”近似方案 [Das08, ACKS15], 因此，过去二十年来，针对聚类问题的 EPASes 研究受到了广泛关注 [HPM04,Mat00, ORSS13, DX20, BJK18, BJKW21, CAPP19].[2]

本文受到以下元问题的启发：

> 对于给定的 k 聚类目标函数和一个（结构化）度量空间，是否存在高效参数化近似方案?

关于这个问题，系统性的理解一直严重不足。尽管早在二十年前，就已证明在连续高维欧几里得空间中， k-中心、 k-中位数和 k-均值等基本聚类问题存在肯定答案 [BHI02, KSS10]（结构化图度量方面的近期结果见 [BBH+20, KLP19, FEKS19, BJKW21, CAPP19]），但对于更复杂的聚类目标函数，我们尚未发现任何此类结果。

本文在全面理解上述元问题方面取得了重大进展。具体而言，我们提出了一种统一的高效参数化近似方案，适用于一大类聚类目标函数（涵盖算法领域曾考虑过的几乎所有基于中心的聚类目标函数，以及一些进一步推广现有问题的新目标函数）和多种度量空间，因此也顺带解决了许多研究充分的独立聚类问题。[3]与现有方法（其中每种算法都针对特定的输入度量和聚类目标函数量身定制）不同，我们的算法框架（几乎）完全不依赖于任何特定的度量结构和目标函数，却能够有效利用这些未知结构。

1.1 高效的参数化范数 k-聚类近似方案

作为（一般） k-聚类问题的输入，我们给定 n 数据点 P、候选中心 F、度量空间 $M=(P\cup F,\delta)$、一个正整数 k 和一个目标函数 $f\colon \mathbb{R}^P \to \mathbb{R}$。当一

[1]简要说明：(i) 高效参数化近似方案与多项式时间近似方案（PTAS）不可比较；(ii) 在高效参数化近似方案这一术语被提出之前，一些研究人员将这类近似方案称为多项式时间近似方案，或简称近似方案（在聚类中，通常假设 k 较小）[KSS10、FMS07], 以及(iii) 高效参数化近似方案和多项式时间近似方案都可由高效多项式时间近似方案（EPTAS）的存在性推出。[2]我们指出，与高效多项式时间近似方案不可比较的多项式时间近似方案，对于连续的 k-均值、 k-中位数和 k-中心并不存在 [CAK19、ACKS15]。[3]聚类问题还有一些变体，会对点如何分配给开放的中-心施加约束（例如容量约束和多样性约束）。我们的目的是处理许多基于中心的聚类目标函数；处理广泛的约束（例如容量约束）超出了本文的范围。

选择了一组 k“开放”中心 $X \subseteq F$，该解会导出一个成本向量 $\boldsymbol{\delta}(P, X) = (\delta(p, X))_{p \in P}$，其中 $\delta(p, X) = \min_{x \in X} \delta(p, x)$表示点 p到 X中最近中心的距离。我们的目标是最小化 $f(\boldsymbol{\delta}(P, X))$。我们将这个问题称为带有成本函数 f的 k-聚类问题。我们可以将函数 f 视为对各点产生的成本进行“聚合”。例如，我们可以通过以下函数来表述基本的 k-聚类目标： $f(\boldsymbol{x}) = \sum_{p \in P} x(p)$（$k$-中位数）、$f(\boldsymbol{x}) = \sum_{p \in P} x(p)^2$（$k$-均值）以及 $f(\boldsymbol{x}) = \max_{p \in P} x(p)$（$k$-中心）。

大多数自然且经过深入研究的聚类目标，都可以用 Chakrabarty 和 Swamy [CS19]提出的范数优化概念（及其推广）来建模。更具体地说，我们关注目标函数 f 是一个范数的情形。范数是一个函数 f: $\mathbb{R}^n \to \mathbb{R}_{\geq 0}, n \in \mathbb{N}$，满足：(i) 对所有 $\boldsymbol{x} \in \mathbb{R}^n$, $f(\boldsymbol{x}) = 0$ 当且仅当 $\boldsymbol{x} = \mathbf{0}$；(ii) $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$: $f(\boldsymbol{x} + \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y})$；以及 (iii) $\forall \boldsymbol{x} \in \mathbb{R}^n, \lambda \in \mathbb{R}$: $f(\lambda \boldsymbol{x}) = |\lambda| f(\boldsymbol{x})$。如果对于任意满足 $\boldsymbol{x} \leq \boldsymbol{y}$的情形，均有 $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$，则称 f 为单调范数。我们所说的范数 k- 聚类，是指目标函数 f: $\mathbb{R}^P \to \mathbb{R}_{\geq 0}$ 为单调范数的 k-聚类问题。尽管 Chakrabarty 和 Swamy [CS19] 还要求 f是对称的[4]，我们的算法框架适用于所有单调范数代价函数。这一类包含以下著名的聚类问题（概述见图2）：

- **从 k均值、 k中心和 k中位数 到 (k、 z)-聚类**：所有基本的聚类问题都可以在 $z \in \{1, 2, \infty\}$ 时由 ℓ_z-范数刻画。事实上，(k、 z)-聚类问题 [HV20、CASS21、CALSS22] （对于常数正整数 z）使用目标函数 $g(\boldsymbol{x}) = \sum_{p \in P} |x(p)|^z$。（该函数本身不是范数，但我们可以改为考虑 ℓ_z-范数 $f(\boldsymbol{x}) = g(\boldsymbol{x})^{1/z}$。）

- **加权 k-中心(或优先级 k-中心)**：加权版的 k-中心[LGW03、BCCN21、Ple87] 推广了 k-中心，使得每个数据点 $p \in P$都关联一个正权重（或优先级） $w(p)$，目标函数是最小化到某个中心的（加权）最大距离。[5] 该问题可以用“加权最大”范数$f(\boldsymbol{x}) = \max_{p \in P} w(p)x(p)$来建模。类似地，也可以定义加权版的 k-中位数和 k-均值（例如参见 [CGK+19]）。我们指出，底层的加权范数并非对称。

- **ℓ-中心**：该问题（有时称为 k-设施 ℓ-中心问题）旨在最小化 ℓ “成本最高”的点（即距离开放中心最远的点）的连接成本之和。该问题将 k-中心问题（$\ell = 1$）和k-中位数问题（$\ell = |P|$） [Tam01]推广为一个统一框架。（有关ℓ-中心问题及下文讨论的更一般的有序 k-中位数问题的更多详情，请参阅书籍 [NP05、LNdG15] 。）该问题可以用 top-ℓ 范数 $f(\boldsymbol{x}) = \sum_{j=1}^{\ell} x^{\downarrow}(j)$ 建模，其中 $\boldsymbol{x}^{\downarrow}$ 表示将向量$\boldsymbol{x}$ 重新排序，使其各项按非递增顺序排列。top-ℓ 范数是对称的。

- **有序 k中位数**：这个问题进一步推广了 ℓ-中心问题，允许对产生最高连接成本的数据点施加灵活的惩罚。更正式地说，目标函数是有序加权范数 $f(\boldsymbol{x}) = \boldsymbol{v}^{\intercal} \boldsymbol{x}^{\downarrow}$，其中 $\boldsymbol{v} \in \mathbb{R}^n_{\geq 0}$ 是一个非递增成本向量，即 $v(1) \geq v(2) \geq \ldots \geq v(n)$。 ℓ-中心问题对应于 $\boldsymbol{v} = (1, \ldots, 1, 0, \ldots 0)$，其中 $\boldsymbol{v}$ 的前 ℓ个条目为 1。几十年来，这个问题已受到一定关注 [BSS18, CS19, BJKW19]。我们指出，此处的 f 是一个单调且对称的范数。

- **具有社会公平性的 k-中位数 (或 鲁棒 k-中位数)**：在具有社会公平性的 k-中位数问题中，除了点集 P，我们还给定了 m 个不同的（不一定互不相交的）子群体，例如

[4]我们称 f 是对称的，如果当 $\boldsymbol{x}'$ 可以通过重新排列 $\boldsymbol{x}$的坐标得到时， $f(\boldsymbol{x}) = f(\boldsymbol{x}')$。[5]为便于表述，我们使用的术语与文献中的术语略有不同。

使得 $P = \bigcup_{i\in[m]} P_i$。我们的目标是找到一组 X 中心，通过最小化所有群体中的最大成本，使各群体承担公平的成本。换言之，

$$\min_{\substack{X\subseteq F\\ |X|=k}} \max_{i\in[m]} \sum_{p\in P_i} \delta(p, X)\,.$$

由于这一聚类变体在至少两个领域中具有不同的应用，近年来得到了广泛研究：(i) 在算法公平性领域 [ABV21、GJ23、MV21、GSV22] ；以及 (ii) 在鲁棒优化的背景下，该问题被称为鲁棒 k-中位数问题，旨在描述这样一类应用：我们无法确定实际可能出现的数据场景（对应于群组 P_i） [AGGN10、BCMN14、BKK+13]。

• **(z, q)-公平聚类：**我们的问题还刻画了一种称为 (z, q)-公平聚类[6] 的聚类问题，该问题由 Chlamtáč 等人提出 [CMV22], ，并推广了社会公平k-中位数问题。

具体而言，可以将社会公平 k-中位数问题的成本函数 f 视为一种"二级"聚合成本：首先，群组 P_i 产生的成本 $\sum_{p\in P_i} \delta(p, X)$ $i \in [m]$ 可以看作加权 ℓ_1-范数 $\boldsymbol{w}_i \boldsymbol{x}$，其中 $\boldsymbol{w}_i = \mathbf{1}_{P_i} \in \{0,1\}^P$ 表示 P_i的特征向量。其次，这些群组成本进一步通过 ℓ_∞ 聚合，即 $f(\boldsymbol{x}) = \max(\boldsymbol{w}_1\boldsymbol{x}, \boldsymbol{w}_2\boldsymbol{x}、\ldots\ldots、\boldsymbol{w}_m\boldsymbol{x})$。

(z, q)-公平聚类允许任意使用 ℓ_z 和 ℓ_q 范数，在两个层级聚合成本。成本函数定义为 $f(\boldsymbol{x}) = g(\boldsymbol{h}(\boldsymbol{x}))$，其中 g 是任意的 ℓ_q-范数函数，而 $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x})\ h_2(\boldsymbol{x})\ldots, h_m(\boldsymbol{x}))$，其中 $h_i(\boldsymbol{x})$ 是加权 ℓ_z-范数，即 $h_i(\boldsymbol{x}) = \left(\sum_{p\in P} w_i(p)x(p)^z\right)^{1/z}$，适用于任意权重向量 $\boldsymbol{w}_i \in \mathbb{R}^P_{\geq 0}$、$i \in [m]$。容易验证，只要 g 和 $\{h_i\}$ 是单调范数，$f(\boldsymbol{x}) = g(\boldsymbol{h}(\boldsymbol{x}))$ 就是单调范数。

• 超越已知问题：我们的（非对称）范数表述使我们能够对更复杂的聚类目标进行建模。这些目标在某些应用场景中可能有用，据我们所知，算法领域尚未研究过。其中一个目标是优先级有序 k-中位数：其代价函数为 $f(\boldsymbol{x}) = \boldsymbol{v}\ \boldsymbol{x}_{\boldsymbol{w}}{}^{\downarrow}$，其中权重向量 $\boldsymbol{v} \in \mathbb{R}^n_{\geq 0}$ 和优先级向量 $\boldsymbol{w} \in \mathbb{R}^P_{\geq 0}$ 作为输入给定，并且$\boldsymbol{x}_{\boldsymbol{w}} = (w(p)x(p))_{p\in P}$。此目标同时推广了优先级 k-中心和有序k-中位数。另一个自然的目标是（多层）级联范数聚类，它推广了 (z, q)-公平聚类，允许多个层级的代价聚合。该问题的代价函数 f 由一个有向无环图（DAG） D 描述，该图有一个汇点和 $|P|$ 个源节点（每个源节点对应 P 中的一个点）。每个非源节点 v 都关联一个范数 ℓ_q，其中某个 q，每条边 (u, v) 的权重为 $w_{u,v}$。给定这样的有向无环图 D，可以按从源节点到汇点的（拓扑）顺序计算 $V(D)$ 中各节点的求值，从而得到 $f(\boldsymbol{x})$ 的值：(i) 源节点 $p \in P$ 处的求值为$\eta(p) = x(p)$；(ii) 对于任何标记有范数 ℓ_q 的非源节点 $v \in V(D)$，我们计算$\eta(v) = \left(\sum_{u\in N_-(v)} w_{u,v}\eta(u)^q\right)^{1/q}$；(iii) $f(\boldsymbol{x})$ 的值即为汇点处的求值。示意图见图 1。当 D 有 3 层且中间层使用相同的范数时，(z, q)-公平聚类就是一个特例。当然，其他基本单调范数（例如 top-ℓ 或有序加权范数）也可以类似地组合成更复杂的范数。

[6]Chlamtáč 等人 [CMV22] 将该问题称为 (p, q)-公平聚类。为与论文其余部分的符号保持一致，我们对名称稍作了调整。

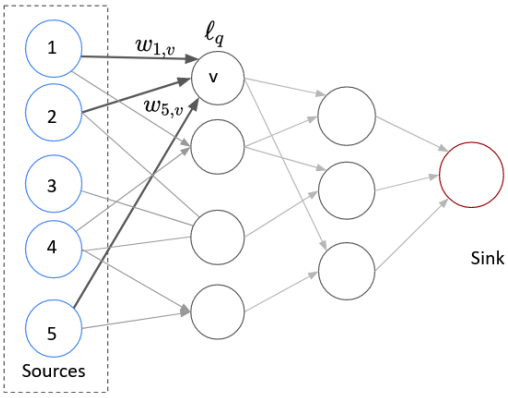

图1：此有向无环图描述了函数 f 的求值。节点 v 标记为 ℓ_q 范数，因此节点 v 处的求值为 $\eta(v) = (w_{1,v}x_1^q + w_{2,v}x_2^q + w_{5,v}x_5^q)^{1/q}$。

我们指出，非对称范数可能会显著增加问题的难度。例如，对于对称范数，存在一个多项式时间的 $\mathcal{O}(1)$-近似算法 [CS19] ；但对于非对称范数，即使是直线度量上的鲁棒 k-中位数这一特殊情形，也很难在 $\Omega(\log n/\log\log n)$ 因子内完成近似 [BCMN14]。

我们的主要结果概括如下定理。

定理1.1。 设 f 是一种可高效计算的单调范数代价函数。那么，对于以下输入度量，代价函数为 f 的 k- 聚类问题存在一种高效参数化近似方案：（*i*）有界倍增维数的度量；（*ii*）任意维数的连续欧几里得空间；（*iii*）有界树宽的度量；以及（*iv*）平面度量。

这里所说的连续欧几里得空间，是指可以选择空间中任意一点作为中心的情形。这与离散欧几里得空间不同，在离散欧几里得空间中，我们将中心限制为从点的特定有限子集中选取。注意，对于固定的 d， $\mathbb{R}^d$ 中的离散欧几里得问题具有有界倍增维数，因此适用于我们的框架。此外，我们的结果未涵盖高维离散欧几里得空间并非其缺陷：在这种情形下， k-中心在近似因子 $\sqrt{3/2} - o(1)$ 以内是 W[1]-困难的（相关证明将另行发表）。

我们的结果特别蕴含以下结论。

推论 1.2。 在所有上述度量空间中，

1. 存在一种针对 n 个点的 有序 k-中位数 问题的 $2^{h(1/\epsilon)\cdot k\cdot polylog(k)}\cdot poly(n)$ 时间高效参数化近似方案。*2.* 存在一种针对包含 n 个点和 m 个群组的 (z,q)-公平聚类 问题的 $2^{h(1/\epsilon)\cdot k\cdot polylog(k)}\cdot poly(n,m)$ 时间高效参数化近似方案。

在我们的研究结果之前，所有这些问题的高效多项式时间近似方案是否存在都是未解决的（ k-均值、 k-中心和 k-中位数除外）。除了这些已知问题之外，我们还针对上文引入并在图2中描绘的新型广义问题获得了高效多项式时间近似方案。相当令人意外的是，与多项式时间近似机制不同，在参数化近似机制下，对称范数聚类问题和非对称范数聚类问题的复杂性“趋于一致”。

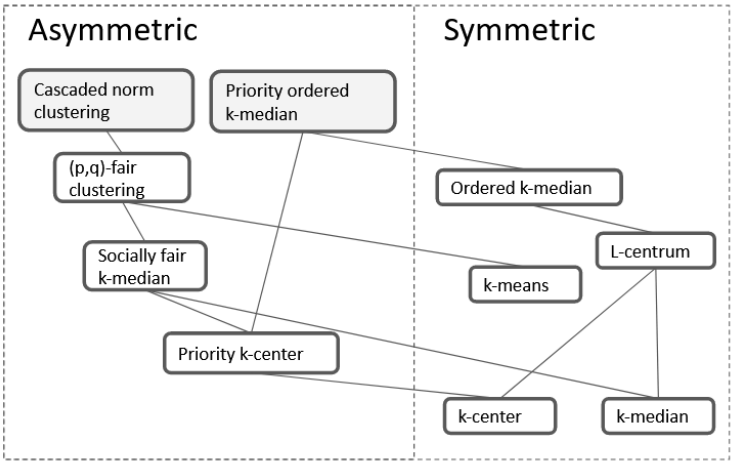

图2：可表述为单调范数最小化的精选聚类目标函数。该线表示推广关系（底部是顶部的特殊情况）。

1.2 我们在概念和技术方面的贡献

我们的主要贡献分为两部分：(i) 度量维数的新概念；以及(ii) 我们的主要技术结果，证明了针对上述所有聚类问题都存在高效多项式时间近似方案。

通过散射维数统一度量空间

我们的核心概念贡献是一种新的有界度量空间维数概念，它放宽了对有界倍增维数这一标准要求，使得定理1.1中提到的度量空间都“存在”于有限维度中。我们首先解释为什么现有的维数概念不适用于这一目的。

度量空间的文献中出现了多种维数概念。算法界最熟悉的或许是倍增维数（又称 Assouad 维数）。粗略地说，度量（M, δ）的倍增维数是 $\mathcal{O}(d)$，当且仅当至多有 $(1/\epsilon)^{\mathcal{O}(d)}$ 个半径为$\epsilon/2$ 的球可以被打包进一个单位球中（这称为 ϵ-打包）。这种性质通常可以在计算上加以利用，从而为许多几何优化问题带来高效算法（运行时间通常以维数为指数）。然而，倍增维数（以及其他流行的维数概念 [Cla06]）由于以下原因并不适合我们：(i) 倍增维数在高维欧几里得空间中可能大到$\Omega(n)$，以及 (ii)它们并不能很好地“利用”结构化图度量，即使星图的维数也是无界的。[7] 总之，任何利用现有维数概念的算法都不太可能得到我们所期望的结果。

我们引入 ϵ-散射维数 的概念。给定度量 $M = (P, F, \delta)$，若序列 $(x_1, p_1) \ldots, (x_\ell, p_\ell) \in F \times P$ 满足：每当 (x, p) 在序列中先于 (x', p') 出现时，$\delta(x, p)$ 和 $\delta(x', p')$ 都大于 $1+\epsilon$ 彼此，而 $\delta(x', p) \leq 1$，则称其为 ϵ-散布。M 的 ϵ-散射维数定义为最长散射的长度减一。

有两种自然的理解。第一种理解是将其视为两名玩家之间的博弈：

[7] 在以 r 为根的 n 节点星图中，单位球 ball(r, 1) 包含整个图。通过选取非根节点，可以得到大小为 $(n-1)$ 的 ϵ-打包。

试图声称自己能用一个单位球覆盖所有点的中心玩家，以及提出反例的点玩家。在第一轮中，中心玩家选择一个中心 $x_1 \in F$，点玩家则通过给出一个点 $p_1 \in P$ 反驳 这一说法；该点与以 x_1 为中心的（闭）单位球的距离至少相差一个因子 $1+\epsilon$，也就是说，$p_1 \notin \text{ball}(x_1,\ 1+\epsilon)$。博弈就这样继续：在第 i 轮中，中心玩家给出 x_i，使得 $\{p_1,\dots,p_{i-1}\} \subseteq \text{ball}(x_i,\ 1)$，而点玩家给出 $p_i \notin \text{ball}(x_i,\ 1+\epsilon)$。两名玩家都希望尽可能延长博弈。$\epsilon$-散射维数就是可能进行的最长博弈的长度。第二种理解是，可以将这样的序列看作一对彼此间距足够大的 ϵ-打包：很容易验证（只需利用三角不等式），$P^* = \{p_1, p_2, \dots, p_{\ell-1}\}$ 和 $F^* = \{x_2, \dots, x_\ell\}$ 分别是以 x_ℓ 和 p_1 为中心的单位（闭）球的 ϵ-打包。这种观点立即表明，在有界倍增度量空间中，ϵ-散射维数有界。

定理 1.3。对于 $\epsilon \in (0,1)$，任何倍增维数为 d 的度量都具有 ϵ-散射维数 $(1/\epsilon)^{\mathcal{O}(d)}$。

我们接下来研究有界树宽图的 ϵ-散射维数。

定理 1.4。对于 $\epsilon \in (0,\ 1)$，树宽为 $\exp\left((1/\epsilon)^{\mathcal{O}(\mathtt{tw})}\right)$ 的图的 ϵ-散射维数为 $\mathtt{tw}$ $s.$

该证明基于一个（精巧的）组合论证：给定图 G、参数 t以及一个长度至少为 t 的双指数级的 ϵ-散布序列，可以构造一个“证书”，证明 G 的树宽大于 t。证明见第 6.2 节。

接下来，我们介绍一种工具，用于对具有有界 ϵ-散射维数的图类进行“引导提升”。这是通过 ϵ-散射维数与低树宽嵌入（一项度量空间嵌入的活跃研究领域）之间的简单联系实现的 [FL22, FEKS19, CAFKL20]。这种联系使我们能够将某个图类中界定 ϵ-散射维数的问题，归约为有界树宽图中的相应问题（从而调用定理 1.4）。

定理 1.5（非正式表述，正式表述见第 6.3 节）。对于任何图类 $\mathcal{G}$，只要它能嵌入到树宽仅取决于 η 的图中，且嵌入的加性失真 η不超过 $\pm\eta\Delta$（其中 Δ 是图的直径），其 ϵ-散射维数就有界。

这种联系 与 [FEKS19],的嵌入结果结合后，推出以下结论。

定理 1.6。对于 $\epsilon \in$（0,1），平面图的 ϵ-散射维数为双指数级（多项式（$1/\epsilon$））。

此外，低树宽嵌入领域的进一步进展，将有望带来更广泛的散射维数有界图类，例如，似乎可以合理地认为，无禁 minor 图能实现这样的嵌入 [FL22]。

遗憾的是，在高维（连续）欧几里得度量中，有界维数这一性质并不成立。[8] 为了处理高维连续欧几里得空间的情形，我们提出了更强版本的 ϵ-散射维数，称为算法 ϵ-散射维数。游戏的设定与之前相同，只是中心玩家会优化 策略，以尽早结束游戏，而点玩家则希望无限期延长游戏。这意味着双方相互对抗。中心策略 是一个函数 σ: $2^P \to F$，它规定了中心玩家根据此前各轮中点玩家选出的点 $p_1, \dots, p_{i-1}$，如何选取中心$x_i = \sigma(\{p_1, \dots, p_{i-1}\})$。$(\sigma,\epsilon)$-散射维数是中心玩家始终采用策略 σ 时的最大回合数，而算法 ϵ-散射维数则是所有策略 σ 中对应的 (σ,ϵ)-散射维数的最小值。需要说明的是，我们实际采用的定义更复杂，因为它考虑了游戏的加权版本。

[8]为说明这一点，考虑序列 $(x_1,p_1)\dots,(x_{d-1},p_{d-1})$，其中，对每个 $i \in [d-1]$，点 $x_i \in \mathbb{R}^d$ 的第 i 个坐标为 $1/\sqrt{2}$，其余坐标均为零。对所有 $i \in [d-1]$定义点 $p_i = -x_i$。不难验证，该序列是一个 $(\sqrt{2}-1)$-散布。这个例子表明，连续欧几里得度量 $\mathbb{R}^d$ 的 ϵ-散射维数至少可以达到 $d-1$（在 ϵ 中无界）。事实上，ϵ-散射维数高达 $(1/\epsilon)^{\Omega(d)}$。

定理 1.7（界定算法散布维数）。连续欧几里得空间（P, F, δ），也就是说，$P \subseteq \mathbb{R}^d$ 有限且 $F = \mathbb{R}^d$，其算法 ϵ-散射维数为 $\mathcal{O}$（$1/\epsilon^4$ 对数 $1/\epsilon$）。

一般范数聚类的高效参数化近似方案：绕过核心集

现在，我们已准备好说明主要技术结果；它将使我们能够针对所有具有有界 ϵ-散射维数的度量获得高效参数化近似方案。

一个一旦存在便可立即推出高效参数化近似方案的通用工具是 ϵ-核心集——它将输入实例（P，F，δ）"压缩"为一个小得多的实例，同时保证任意解的代价在因子（$1 \pm \epsilon$）范围内得到保留。若存在大小仅取决于 ϵ 和 k 的 ϵ-核心集，便可立即推出高效参数化近似方案（但反之不成立）：首先，使用 ϵ-核心集将实例（P，F，δ）压缩为（P'，F'，δ'），其中 $|P'| \leq \gamma(\epsilon, k)$。然后，枚举将 P' 划分为 k 个集合 $P'_1, \ldots, P'_k$ 的所有可能划分（此类划分至多有 $k^{\gamma(\epsilon,k)}$ 种）。对每个集合 $i \in [k]$，计算其最优中心 P'_i。我们选择总代价最低的划分。

遗憾的是，这种通用方法面临着严重的信息论限制：即使对于 k-中心，在高维欧几里得空间中也不存在大小理想的 ϵ-核心集 [FL11]。这些下界意味着，我们无法指望通过核心集这条路径证明我们的（统一）结果：虽然已知对于常数 z [CASS21]的(k，z)-聚类存在核心集——从而能够以统一的方式处理 k-均值和 k-中位数——但不可能将这种方法扩展到 k-中心。对于更复杂的聚类目标，即便在低维情形下，此类高效多项式时间近似方案实际上也尚未出现。例如，Braverman 等人 [BJKW19] 针对有序 k-中位数提出的核心集在 $\mathbb{R}^d$ 中的大小为$\mathcal{O}_{\epsilon,\, d}(k^2 \log^2 n)$，因此即使在低维情形下也无法提供高效参数化近似方案。

Badoiu、Har-Peled 和 Indyk [BHI02] 提出了适用于高维欧几里得空间中 k-中心的高效参数化近似方案（以不同于上述意义的方式绕过核心集）。因此，一个显而易见的开放问题是：能否将他们的技术扩展到其他聚类目标，从而为其提供高效参数化近似方案。遗憾的是，即便对于优先级 k-中心这样简单的目标，这一点也尚未得到证实。事实上，即使是已知的 k-均值 [KSS10]和 k-中心 [BHI02]的高效多项式时间近似方案，在概念上也大不相同；据我们所知，还没有近似方案能以模块化方式处理 k-均值和 k-中心。

我们的主要技术结果由以下定理给出。值得指出的是，我们的技术不依赖任何核心集构造（因此绕过了 k-中心问题的核心集下界）。

定理 1.8。设 $\mathcal{M}$ 是一个度量空间类，且在将距离按正数倍缩放后保持封闭。存在一种随机化算法，对于任意 范数 k-聚类 实例$\mathcal{I} = (M, f, k)$，其度量为 $M = (P, F, \delta) \in \mathcal{M}$，以及任意 $\epsilon \in (0, 1)$，若满足以下两个条件，则该算法能以高概率计算出一个$(1+\epsilon)$-近似解。

(i) 存在一种高效算法，可在 $T(f)$时间内，对任意距离向量 $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ 计算目标函数 $f(\boldsymbol{x})$。

(ii) 存在一个函数 $\lambda\colon \mathbb{R}_+ \to \mathbb{R}_+$，使得对于所有 $\epsilon > 0$，$\mathcal{M}$ 的算法散射维数至多为 $\lambda(\epsilon)$。

该算法的运行时间为 $\exp\left(\widetilde{\mathcal{O}}\left(\binom{k\lambda_{(\epsilon/10}}{\epsilon}\right)\right)\cdot)\underline{\textit{poly}(|M|)} \cdot T(f)$。

请注意，计算 f 的复杂度仅作为运行时间中的线性因子出现。例如，对于社会公平 k-中位数，群组数量 m 只会影响 f 的计算成本，因此运行时间关于 m 是多项式的。

我们的算法简洁、简单，而且完全不依赖目标函数或输入度量的结构。

˜我们的运行时间关于 k 的指数依赖是单指数的（$\exp(\mathcal{O}_\epsilon(k))$）。就 k 而言，我们的运行时间因此与目前已知最快的高维 k-均值 [KSS10] 的高限制性特例的高效参数化近似方案相匹配。此外，通过证明感兴趣的度量空间的 ϵ-散射维数具有更好的界，可以改进指数中对 ϵ 的依赖；˜例如，$\lambda(\epsilon) = \text{poly}(1/\epsilon)$ 意味着高效参数化近似方案的运行时间为 $\exp(\mathcal{O}(k) \cdot \text{poly}(1/\epsilon))$。

2 技术概述

在本节中，我们将非正式地概述论文中出现的技术思想。主要结果将逐步构建：我们认为，理解专门针对加权 k-中心和加权 k-中位数的主要结果本身就很有意义。我们的出发点是Bad˘oiu 等人 [BHI02] 针对无权 k-中心提出的高效参数化近似方案，该方案适用于高维欧几里得空间。我们重新设计并改动了这一算法，以便将其清晰地划分为两个部分：一个简单的分支算法，以及一个关于（算法）ϵ-散射维数这一抽象概念的界。这样一来，我们便在分支算法（它特定于目标函数）与关于 ϵ-散射维数的界（它特定于度量）之间实现了明确的区分。这可以与基于核心集的技术形成对比，后者从本质上同时特定于单一目标函数和单一度量。本文的主要观点是，通过恰当地结合一些额外思想，这一框架可以在目标函数和度量空间两方面得到显著推广。

本节分三步介绍主要的算法思想。

1.无权 k-中心的算法可以以一种并非完全显而易见的方式推广到带权 k-中心。 2. 基于带权 k-中心的算法，我们可以通过预处理和随机选择步骤求解带权 k-中位数。 3.通过考虑由次梯度定义的无穷多个带权 k-中位数实例，可以将带权 k-中位数算法推广到任意单调范数。

虽然前进道路上的一些挑战乍看之下似乎有其他颇具前景的解决思路，但我们想强调的是，要找到能够整合在一起并导出主要结果的一组合适思路，并非易事。特别是对于加权 k-中位数，必须谨慎定义初始上界，从而同时支持高效的随机选择步骤，并推广到任意单调范数。

加权 k-中心具有有界数量的不同权重。 我们的出发点是一种简单的分支算法，其灵感来自Bad˘oiu 等人针对无权k-中心问题提出的高效参数化近似方案。对我们而言，与其进行分支，不如将其表述为一种随机化算法。此外，我们考虑加权 k-中心问题这一更一般的设置：目标函数是寻找一个包含 O 个 k 中心的集合，使 $\max_p w(p)\delta(p, O)$ 最小。首先，在一个简化假设下介绍该算法：w 是定义在点上的权重函数，其值域至多包含 τ 个不同的值。无权问题对应于对每个 $p \in P$ 均有 $w(p) = 1$，因此 $\tau = 1$。假设我们能够（近似地）知道最优值的值将很方便。

我们从任意选取的候选点 k 中开始，为中心设置 $X = \{x_1, \ldots, x_k\}$ k。此外，我们还引入 k 组请求 $Q_1 \ldots Q_k$，其中每个请求的形式为 (p, r)，其中包含一个点

$p \in P$ 和半径 $r > 0$。对于每个 $\kappa \in [k]$，我们施加簇约束，要求对于每个 $(p, r) \in Q_\kappa$，中心 x_κ 到 p 的距离至多为 r。初始时，对于每个 κ，我们设定 $Q_\kappa = \emptyset$，这意味着这些条件显然都能满足。如果 $\max_p w(p)\delta(p, X) > (1+\epsilon)\mathrm{OPT}$，那么我们就可以停止，因为此时我们已经得到一个 $(1+\epsilon)$-近似解。否则，存在一个点 p，使得 $\delta(p, X) \geq (1+\epsilon)\mathrm{OPT}/w(p)$，同时它到假设的最优解 O 中某个中心的距离至多为 $\mathrm{OPT}/w(p)$。因此，算法会均匀随机地选择一个 $\kappa \in [k]$，希望它是最优解 O 中到 p 的距离至多为 $\mathrm{OPT}/w(p)$ 的中心所对应的索引。接着，我们将请求 $(p, \mathrm{OPT}/w(p))$ 加入集合 Q_κ，并选择满足更新后的 Q_κ 中所有请求所定义的簇约束的中心 x_κ。注意，如果每次随机选择都与假设的最优解 O 相容，那么算法总能找到这样的中心，因为 Q_κ 中的请求始终都能由最优解 O 的第 κ 个中心满足。

我们声称，如果该度量的 ϵ-散射维数有界，那么该算法会在有限步数后停止：要么找到一个近似解，要么无法找到满足某个 Q_κ 的簇约束的中心。设 $x_\kappa^{(1)}$、……、$x_\kappa^{(\ell)}$ 是该分支中作为第 κ 个中心的不同候选者。设 $(p_\kappa^{(1)}, r_\kappa^{(1)})$、……、$(p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ 是引入到 Q_κ 的请求：也就是说，对于 $1 \leq j \leq \ell$，所选中心 $x_\kappa^{(j)}$ 与每个 $p_\kappa^{(i)}$ 的距离至多为 $r_\kappa^{(i)}$，但后来发现它与 $p_\kappa^{(j)}$ 的距离至少为 $(1+\epsilon)r_\kappa^{(j)}$。由于输入中至多有 τ 个不同的权重，因此这些请求中至少有 $\ell' = \ell/\tau$ 个具有相同的半径。也就是说，存在一个子序列 $(x^{(s_1)\kappa}, p_\kappa^{(s_1)}, r^{(s_1)\kappa})$、……、$(x^{(s_{\ell'})\kappa}, p_\kappa^{(s_{\ell'})}, r^{(s_{\ell'})\kappa})$，其中对于 $j \in [\ell']$，每个 $r_\kappa^{(s_j)}$ 都是相同的值 $r \geq 0$。这意味着我们有一个子序列 $(\bar{x}_1, \bar{p}_1)$、……、$(\bar{x}_{\ell'}, \bar{p}_{\ell'})$，其具有如下性质：$\delta(\bar{x}_i, \bar{p}_i) > (1+\epsilon)r$，但对于每个 $i < j$，$\delta(\bar{x}_i, \bar{p}_j) \leq r$。将每个距离缩小 r 倍后，这正是一个长度为 ℓ' 的 ϵ-散布。如果我们考虑一个关于缩放封闭的度量类，其中 ϵ-散射维数为 $\lambda(\epsilon)$，那么该序列的长度不可能超过 $\lambda(\epsilon)$，这意味着 $\ell \leq \tau \cdot \lambda(\epsilon)$。因此可以得出，该算法在每个 Q_κ 中至多引入 $\tau \cdot \lambda(\epsilon)$ 个请求，从而该算法执行的迭代次数不会超过 $k \cdot \tau \cdot \lambda(\epsilon)$。

如果算法在每一步都随机选择一个与最优解 O相一致的索引 $\kappa \in [k]$，那么它唯一可能的停止方式就是找到一个近似解。因此，算法成功的概率至少为 $q = k^{-k\cdot\tau\cdot\lambda(\epsilon)}$。通过采用标准方法，将算法重复运行 $\mathcal{O}(1/q)$ 次，可以将成功概率提高到任意接近 1 的常数，使运行时间达到 $k^{k\cdot\tau\cdot\lambda(\epsilon)} \cdot \mathrm{poly}(n)$。

加权 k中心与任意权重。现在我们展示如何扩展该算法，使其能够处理具有任意权重的加权情形。首先注意，如果不同权重的数量 τ 没有上界，那么即使是在非常简单的度量空间中（例如 $\mathbb{R}^1$），我们也无法限制到达给定 Q_κ的请求数量。举例来说，假设到达 Q_κ 的请求为 $(p^{(i)}, (1+2\epsilon)^{-i})$，其中 $i = 1, 2, \ldots\ldots$，且每个 $p^{(i)}$ 都位于原点（或者距离原点非常近）。那么，位于 $(1+2\epsilon)^{1-i}$ 的中心 $x^{(i)}$ 满足前 $i-1$个请求，但对第 i个请求的约束则超出了一个 $(1+\epsilon)$倍。这一序列可以任意长；存在这样的序列，说明如果不同权重的数量没有上界，我们就无法限制到达 Q_κ 的请求数量。尽管如此，我们将证明，只要通过谨慎地播种初始请求来启动算法，就可以限制请求数量。值得一提的是，我们还知道其他一些能够实现这种限制的简单修改，但事实证明，下面描述的技术还可以进一步扩展到加权 k-中值问题和一般范数。

核心思想是用常数因子近似来引导我们的算法。按照 Plesník [Ple87] 的思路，可以得到一个简单的贪心 3-近似算法。我们来考虑一下

对于每个 $p \in P$，所有球ball(p, OPT/$w(p)$)。让我们按照半径的非递减顺序考虑这些球，并标记每个不与此前标记的任何球相交的球；令ball(p_κ, OPT/$w(p_\kappa)$)， $1 \le \kappa \le k'$ 为被标记的球。我们应有 $k' \le k$：否则，我们将拥有超过 k 个两两不相交的球，而每个球都必须包含一个解的中心，这与最优值 OPT 可以用 k 个中心实现的假设相矛盾。对于 $1 \le \kappa \le k'$，令 x_i 为球ball(p_κ, OPT/$w(p_\kappa)$) 中的任意中心，并令 $Q_\kappa = \{(p_\kappa, \text{OPT}/w(p_\kappa)\}$。对于 $k' < \kappa \le k$，我们任意选择 x_i，并令 $Q_\kappa = \emptyset$。让我们注意到，根据对这些 Q_κ 的定义，我们有$\delta(p, X) \le 3\text{OPT}/w(p)$在算法的每次迭代期间都成立。事实上，如果 p 的球被标记，那么 X 总是包含球 ball(p_κ,OPT/$w(p_\kappa)$)中的一个中心；如果 p 的球未被标记，那么它会与一个半径不大于它且包含 X 的中心的已标记球相交。

主要结论是， Q_κ 中两个请求的半径之比可以由 $\mathcal{O}(1/\epsilon)$ 限定。假设 (p, r) 和 (p', r') 是 Q_κ 中的两个请求（按任意顺序引入），且我们有 $r' < \epsilon r/4$。最优解的一个中心同时满足这两个请求，因此我们有 $\delta(p, p') \le r + r'$。如上所示，在算法的每一步中， X 中都存在一个中心，其与 p的距离至多为 $3r'$；令 y 为引入请求 (p, r) 时该步骤中的这样一个中心。于是我们有

$$\delta(p, y) \le \delta(p, p') + \delta(p', y) \le r + r' + 3r' \le (1 + \epsilon)r,$$

这与第一个请求的必要性相矛盾。

我们可以采用标准假设：每个权重都具有形式 $(1+\epsilon)^i$，其中 i为某个整数：将每个权重向下舍入到这种形式下不超过它的最大数值，目标函数最多只会改变 $1+\epsilon$倍。如果每个权重都具有形式 $(1+\epsilon)^i$，那么，上文证明的 $\mathcal{O}(1/\epsilon)$ 界意味着，对于某个固定的 $\kappa \in [k]$，引入 Q_κ 的请求具有 $\mathcal{O}(1/\epsilon \cdot \log 1/\epsilon)$ 个不同的半径。因此，我们可以将请求总数（从而也包括迭代次数）限制为 $\mathcal{O}(\lambda(\epsilon) \cdot k/\epsilon \cdot \log 1/\epsilon)$。由此可得到一个运行时间为 $k^{\mathcal{O}(\lambda(\epsilon)\cdot k/\epsilon \cdot \log 1/\epsilon)} \cdot \text{poly}(n)$、成功概率为常数的随机化算法。

从加权 k-中心到加权 k-中位数。为了实现我们理解一般范数的目标，现在让我们考虑加权 k-中位数问题，其中目标函数是找到一个集合 O 个 k 中心，使 $\sum_p w(p)\delta(p, O)$最小化。我们将尝试通过以下方式解决这一问题：将其解释为定义在加权点集上的加权 k-中心问题，该点集是在算法执行过程中动态发现的。

我们希望将加权 k-中位问题的线性约束 $\sum_p w(p)\delta(p, X) \le$最优值转化为距离约束：某个点 p 到该解的距离应至多为 r。令 X为当前解，并假设 $\sum_p w(p)\delta(p, X) > (1+\epsilon)$最优值。直观上，对于一个最优解 O，$\sum_p w(p)\delta(p, X) > (1+\epsilon)\sum_p w(p)\delta(p, O)$ 意味着相当大比例的点应满足 $\delta(p, X) > (1+\epsilon/3)\delta(p, O)$，也就是说，它们到该解的距离必须改善超过 $1+\epsilon/3$ 倍。更准确地说，简单的平均论证表明，如果以与 $w(p)\delta(p, X)$ 成正比的概率选择一个点 p，那么 p 以概率$\Omega(\epsilon)$ 满足$\delta(p, X) > (1+\epsilon/3)\delta(p, O)$。我们称这样的点 p为 $\epsilon/3$-见证点，用以证明当前解必须得到改进。

假设采样点 p 确实是一个 $\epsilon/3$-见证点，我们便按照带权 k-中心的情形进行处理。我们随机选择一个索引 κ，并将请求 (p, $\delta(p, X)/(1+\epsilon/3)$) 加入 Q_κ，通过要求 x_κ 比当前解中更靠近 p 来更新簇约束。如果存在一个中心满足 Q_κ中的所有请求，我们就更新 x_κ。重复这些步骤，直到得到一个满足 $\sum_p w(p)\delta(p, X) \le (1+\epsilon)$最优值的解 P。

在每一步中，以$\Omega(\epsilon/k)$的概率，该算法会选择一个 $\epsilon/3$-见证点 p 以及一个与某个假设的最优解 O一致的中心 κ。然而，目前尚不清楚如何界定

算法的运行时间。可能出现向 Q_κ 发出的请求的半径越来越小的情况。正如我们在加权 k-中心中所看到的，在这种情形下，即使在 $\mathbb{R}^1$ 我们也无法界定步骤数。对请求中出现的半径序列进行某种控制至关重要。因此，接下来我们将展示如何确保对中心 κ 的请求中的半径保持在有界范围内。

初始上界。对于每个点 p，我们计算其到最优解的距离的一个弱上界 $u(p) \geq \delta(p, O)$。因此，我们不再从任意一组 k 中心开始，而是通过一个近似满足所有这些上界的解来引导算法。我们论证，可以通过一种方式做到这一点，从而确保每个中心 κ 收到的请求中出现的半径都处于一个有界范围内。

如果一个点 p^* 的权重为 $w(p^*)$，那么 $u(p^*) =$ 最优值$/w(p^*)$ 显然是 p^* 到 O的距离的一个上界；否则，我们就有 $\sum_p w(p)\delta(p, O) \geq w(p^*)\delta(p^*, O) >$最优值。对于加权 k-中心问题，这个上界已经足够；但加权 k-中位数问题的性质使我们在许多情况下可以得到强得多的上界。例如，如果有 c 个权重相同的点 w 大致位于同一位置，那么它们到 O的距离都应至多为最优值$/(wc)$。确实，否则这些 c 点对总和的贡献将大于最优值。更一般地，如果存在一个半径 r，使得距离 p 不超过 r 的点的总权重至少为最优值$/r$，那么我们断言，p 到O的距离至多为 $2r$。确实，否则这些点到 O的距离都将大于 r，导致它们的总贡献超过最优值。因此，我们可以定义 $u(p) = 2r$，其中 r 是满足以下性质的最小半径：距离 p 不超过 r 的点的总权重至少为最优值$/r$。注意，$u(p)$ 可以根据各点的权重及其距离矩阵在多项式时间内确定。

与我们的加权 k-中心算法类似，我们首先对由 $p \in P$ 的上界 $u(p)$ 给出的约束进行 3-近似。让我们按照 $u(p)$ 的非递减顺序遍历各点，并贪心地选取球 ball$(p, u(p))$ 的一个极大独立集。我们至多需要找到 k 个这样的球。在每个球中选取一个中心；容易看出，每个点 p 到所选中心的距离至多为 $3u(p)$。如果选取中心 x_κ 作为球 ball$(p, u(p))$ 的中心，那么我们用请求 $(p, u(p))$ 初始化 Q_κ。这保证了在算法的每一步中，每个点 p 到当前解的距离始终至多为 $3u(p)$。

我们使用这个初始解运行 Weighted k-Median 算法。在分析该算法之前，让我们对随机选择进行一项重要的改变。我们已经看到，以 $\Omega(\epsilon/k)$ 的概率，我们会选择一个随机点 p 和 $\kappa \in [k]$，使得对于某个最优解 O，$\delta(p, X) \geq (1 + \epsilon/3)\delta(p, O)$。证明的一个关键主张是，以 $\Omega(\epsilon/k)$ 的概率，同样有 $u(p) \leq 2k\delta(p, X)/\epsilon$（参见引理 5.10）。直观地说，与某个中心 $x_\kappa \in X$ 距离过近的 $\epsilon/3$-见证点的总贡献不可能非常大，因为这样所有这些见证点都会位于一个小球中，这意味着上界 $u(p)$ 应该更小。注意，在证明中，我们正是在此处关键性地利用了 $u(p)$ 的精确定义。有了这一主张，我们可以修改算法，使其以与 $w(p)\delta(p, X)$ 成比例的概率，随机选择一个满足$u(p) \leq 2k\delta(p, X)/\epsilon$的点 p。p 仍然以 $\Omega(\epsilon/k)$ 的概率是一个$\epsilon/3$-见证点。

现在让我们分析该算法，并界定中心 x_κ 被更新的次数。我们希望说明，请求中的半径始终处于有界范围内。假设我们使用请求 (p, r) 和 (p', r')（顺序不限）更新簇 κ，且满足 $r' \ll \epsilon^2 r/k$。如果算法没有失败，那么就存在一个中心 x_κ，同时满足这两个请求。根据三角不等式，这意味着

$\delta(p, p') \leq r + r' < r + \epsilon r/6$。此外，根据我们选择随机点 p' 时施加的约束 $u(p') \leq 2k\delta(p', X)/\epsilon = 2k(1+\epsilon/3)r'/\epsilon$，我们有 $u(p')$ 远小于 $\epsilon r/18$。在算法的每一步中，当前解 X 都以 3-近似满足上界 $u(p')$。因此，在 X 中应当存在一个中心，其位置比 $\epsilon r/6$ 更接近 p'。结合 $\delta(p, p') < r + \epsilon r/6$，可知 X 中始终存在一个中心，其到 p 的距离至多为 $(1+\epsilon/3)r$，这与请求 (p, r) 的必要性相矛盾。

因此，以下两个事实的结合：(1) 上界总是近似得到满足；(2)请求中的半径不会比上界小太多，这意味着请求中的半径始终处于一个有界范围内。接着，我们可以像处理加权 k-中心问题的情况一样进行论证。如果将每个权重都舍入为$(1+\epsilon)$的幂，那么每个簇收到的请求所涉及的不同半径数量是有界的。如果到达了许多请求，那么其中必然存在一个具有相同半径的请求长子序列。这意味着可以利用 ϵ-散射维数的上界来限制该子序列的长度，从而限制所有簇收到的请求总数。

从 加权 k-中位数到使用次梯度的一般范数。 接下来，我们将展示如何通过将其解释为一组需要同时满足的加权 k-中位数实例，来解决任意单调范数下的聚类问题。在算法运行过程中，我们会反复求解这些在动态过程中发现的加权 k-中位数实例。

使用次梯度的概念会很方便。就我们的目的而言，只需讨论单调范数 $f\colon \mathbb{R}^n \to \mathbb{R}$ 下的次梯度。若 $\boldsymbol{g}$ 是 f 在点 $\boldsymbol{x}$处的次梯度，则称 $f(\boldsymbol{x}) = \boldsymbol{g}^\intercal \boldsymbol{x}$ 且对每个 $\boldsymbol{y} \in \mathbb{R}^n$都有 $f(\boldsymbol{y}) \geq \boldsymbol{g}^\intercal \boldsymbol{y}$。众所周知，每个单调范数在每个点 $\boldsymbol{x} \geq 0$处都有一个非负次梯度 $\boldsymbol{g} \geq 0$。判断向量 $\boldsymbol{g}$ 是否为 $\boldsymbol{x}$ 处的次梯度，以及求出 $\boldsymbol{x}$ 处的次梯度，都可以表述为凸优化问题；因此，如果能够高效计算 f [GLS12]，就可以使用椭球法近似求解。

假设当前解为 X，并令 $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ 表示集合 P 中的点到 X的距离向量。假设 X 不是（近似）最优的：$f(\boldsymbol{x}) > (1+\epsilon)\mathrm{OPT}$。我们计算 f在 $\boldsymbol{x}$处的次梯度 $\boldsymbol{g}$；于是有 $\boldsymbol{g}^\intercal \boldsymbol{x} = f(x) > (1+\epsilon)\mathrm{OPT}$，并且对最优解 $\boldsymbol{y}$，有 $\boldsymbol{g}^\intercal \boldsymbol{y} \leq f(\boldsymbol{y}) = \mathrm{OPT}$。也就是说，$\boldsymbol{g}^\intercal \boldsymbol{x} \leq \mathrm{OPT}$ 是最优解满足、而当前解违反的一条线性约束。接着，根据 $\boldsymbol{g}$ 的各坐标定义权重 $w(p)$，便得到一个加权 k-中值问题实例；对于当前解 X，有 $\sum_p w(p)\delta(p, X) > (1+\epsilon)\mathrm{OPT}$。现在可以像处理加权 k-中值问题那样继续：随机选择一个点 p 和簇 κ，向 Q_κ中加入一个新请求，找到一个新中心 x_κ，等等，直到得到一个解 X，满足 $\sum_p w(p)\delta(p, X) \leq (1+\epsilon)\mathrm{OPT}$。如果这个新解 X对于原范数问题仍非最优，也就是说，$f(\boldsymbol{x}) > (1+\epsilon)\mathrm{OPT}$，那么我们可以再次计算次梯度，找出一条被违反的线性约束（可能与上一步相同）。重复这一过程，直到找到一个解，使得 $f(\boldsymbol{x}) \leq (1+\epsilon)\mathrm{OPT}$。

定义上界，并通过一个近似满足上界的解来引导算法启动，这对加权 k-中位数算法的分析至关重要。对于一般范数，只要有了基于被违反的线性约束 $\boldsymbol{g}^\intercal \boldsymbol{x} \leq$最优值的权重 w，我们就能再次定义上界。然而，这些上界对分析并无帮助，因为它们取决于被违反的线性约束，因此会在算法运行过程中发生变化。

直观地，我们可以将约束 $f(\boldsymbol{x}) \leq$ 最优值看作无穷多个加权 k-中位数实例，对应于对于每一个 f 的次梯度 $\boldsymbol{g}$的线性约束 $\boldsymbol{g}^\intercal \boldsymbol{x} \leq$ 最优值。我们希望将 $u(p)$ 定义为可以赋予 p 的最小可能上界

在所有这些无限多个加权 k-中位数实例中。确定这个值似乎是一项艰巨的任务，但实际上答案非常简单。回想一下， $u(p)$ 被定义为使得球(p,r) 包含的总权重至少为 OPT$/r$ 的最小 r 的两倍。因此，要定义上界 $u(p)$，我们需要知道，在所有对应于这些次梯度的无限多个实例中，球(p,r) 内各点的权重最大值是多少。令 $\boldsymbol{b}$ 为球(p,r) 的特征向量（即每个坐标为 1 或 0，取决于某个点是否在球内）。那么问题就是确定所有次梯度 $\boldsymbol{g}$ 中 $\boldsymbol{g}\ \boldsymbol{b}$ 的最大值。容易看出，这个最大值恰好是 $f(\boldsymbol{b})$：如果 $\boldsymbol{g}$ 是 $\boldsymbol{b}$ 处的一个次梯度，那么 $\boldsymbol{g}\ \boldsymbol{b} = f(\boldsymbol{b})$；如果 $\boldsymbol{g}$ 是任意点 $\boldsymbol{y}$ 处的一个次梯度，那么 $\boldsymbol{g}\ \boldsymbol{b} \leq f(\boldsymbol{b})$。由此，我们可以确定任意球的最大权重，并据此定义上界。有了这些定义，加权 k-中位数算法的分析便可推广到一般单调范数。上界的两项主要性质仍然成立：(1) 最优解满足这些上界；(2) 我们可以将对 p 的随机选择限制在这样的点上：这些点到解的距离不比 $u(p)$ 小太多。

在总结中，最终算法由以下步骤组成（见图 3）。首先，我们计算上界 $u(p)$，并以贪心方式找到一个满足这些约束的 3-近似解。然后重复以下步骤，直到得到一个解 X，其距离向量 $\boldsymbol{x}$ 满足$f(\boldsymbol{x}) \leq (1+\epsilon)$OPT。我们计算 f 在 $\boldsymbol{x}$ 处的次梯度 $\boldsymbol{g}$，以获得一个违反的线性约束$\boldsymbol{g}\ \boldsymbol{x} \leq$ OPT。我们随机选择一个点 p （根据上述分布），并要求 p 与该解的距离至多为 $\delta(p,X)/(1+\epsilon/3)$，从而得到一个违反的距离约束。然后，我们随机选择一个簇 $\kappa \in [k]$，并要求该距离约束由中心 x_κ满足。因此，我们将请求 $(p, \delta(p,X)/(1+\epsilon/3)$放入 Q_κ，以便找到一个新的 x_κ，使其满足由 Q_κ中的请求施加的簇约束（如果可能）。我们重复这些步骤，直到得到一个解 X，其距离向量 $\boldsymbol{x}$满足 $f(\boldsymbol{x}) \leq (1+\epsilon)$OPT。我们的分析表明，每一步都以 $\Omega(\epsilon/k)$ 的概率与一个假设的最优解 O 一致。此外，如果 ϵ-散射维数有界，那么算法必定会在有界次数的迭代后找到一个解，或者失败。

（算法） ϵ-散射维数。 在介绍了能够处理任何单调范数目标函数的一般算法之后，我们的第二项主要贡献是界定各种度量类别的 ϵ-散射维数（第6节）。由于篇幅所限，我们不详细介绍这些（主要是组合性的）证明，而只作简要概述。

- **有界倍增维数。** 如引言所述， ϵ-散布中的点集和中心集都构成一个单位球的 ϵ-打包，这意味着任何倍增维数为 d 的度量，其 ϵ-散射维数均为 $(1/\epsilon)^{\mathcal{O}(d)}$。参见定理 1.3。

- **有界树宽图度量。** 对于由有界树宽图的最短路径度量定义的度量，其 ϵ-散射维数界是通过一种精细的组合证明得到的，该证明利用了图的结构和 ϵ-散布的性质。我们得到的界为树宽为 tw 的图上的 $\mathrm{tw}^{1/\epsilon^{\mathcal{O}(\mathrm{tw})}}$，也就是说，对于固定的 ϵ，关于 tw 呈双指数增长。这个界能否改进，仍是一个有趣的开放问题。

- **平面图度量。** 如引言所述，我们可以利用一个已知的度量嵌入结果，将界定平面图的 ϵ-散射维数的问题，化简为界定有界树宽图的 ϵ-散射维数的问题。具体而言，Fox-Epstein、Klein 和 Schild [FEKS19]的结果给出了将平面度量嵌入低树宽度量的（近似）度量嵌入，可用于得到平面图度量的 ϵ-散射维数的 $2^{2^{\mathrm{poly}(1/\epsilon)}}$ 界。

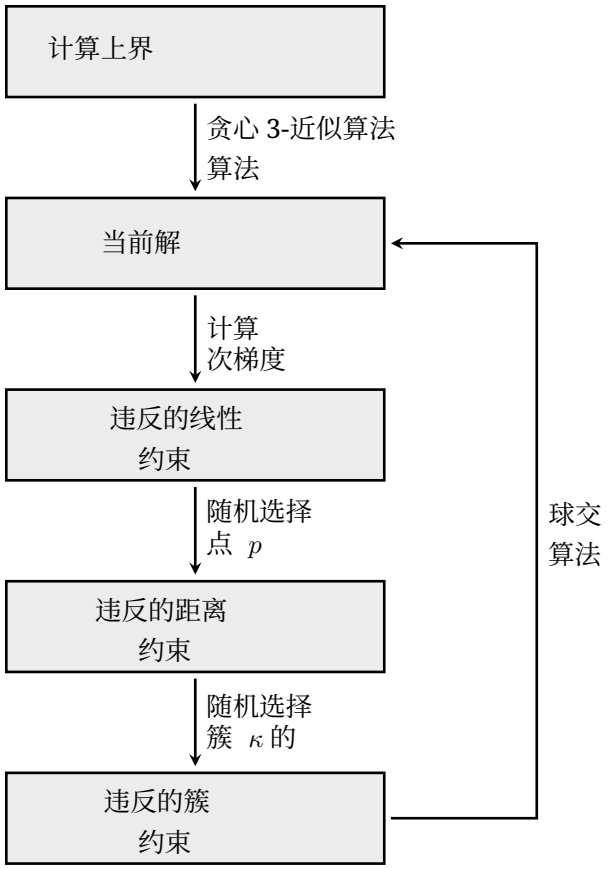

图 3：主算法的整体结构。

• **连续高维欧几里得空间**。 正如引言中所述，高维欧几里得空间不具有有界的 ϵ散射维数。然而，在连续欧几里得空间中，空间中的任意点都可以作为中心，我们可以界定算法 ϵ-散射维数。为此，我们用一个算法“玩家”取代中心玩家，该玩家应用 Kumar 和 Yildirim [KY09] 针对加权 1-中心问题提出的算法。然而，要实现有界的算法 ϵ-散射维数，该算法需要输入请求中的半径具有有界的长宽比。因此，我们证明了一个长宽比条件（即使对于一般度量也成立），它表明只需让算法处理长宽比为 $\mathcal{O}(1/\epsilon)$ 的实例即可。我们将这一结果与 Kumar 和 Yildirim 的算法相结合，证明连续高维欧几里得空间具有有界的算法 ϵ-散射维数，即定理 1.7。

3 预备知识

度量聚类空间的类别。 度量聚类空间（或简写为度量空间）是一个三元组 $M = (P, F, \delta)$，其中 P 是 n 数据点的有限集合， F是簇中心可能位置的（可能是无限的）集合，而 δ 是 $P \cup F$上的度量。集合 P和 F不一定互不相交。（例如，对于聚类问题而言， $P = F$或 $P \subseteq F$是很自然的。）给定度量空间中的任意点 $u \in P \cup F$和半径 $r \in \mathbb{R}_+$，我们将球$_\delta(u, r) = \{ v \in P \cup F \mid \delta(u, v) \leq r \}$ 以下

以 u为中心、半径为 r 的球。如果距离函数从上下文中可以明确，则省略下标 δ 。

我们用 $|M|$ 表示在内存中表示度量空间 M 所需的空间。如果 M 是有限的，那么 $|M|$关于 $|F|$、 $|P|$ 以及分别存储一个点和一个中心所需的空间是多项式的。如果 F 是无限的（例如，在连续欧几里得设置中， $F = \mathbb{R}^d$），那么 $|M|$ 关于 $|P|$ 和存储一个点所需的空间是多项式的。

度量空间的一个类 $\mathcal{M}$ 是度量空间的一个（无限）集合。本文聚焦于在按常数缩放距离后仍保持封闭的度量类。我们考虑以下度量聚类空间类:

- **图度量：** 对于图度量，我们给定一个（带权）图 $G = (V, E)$，以及定义在 V 上的度量 δ_G，其为最短路径度量，即 $\delta_G(u, v)$ 是连接 u 和 v 的路径的最短距离。聚类空间 (P, F, δ_G) 给定为满足 $P, F \subseteq V$。

- **连续欧几里得空间：** 在这种情况下，我们可以从（高维）连续欧几里得空间 $F = \mathbb{R}^d$ 中选取中心。集合 $P \subsetneq \mathbb{R}^d$ 是一个有限点集。

- **倍增度量：** 度量空间 (X, δ) 的倍增维数记作 d，它是满足以下条件的最小 $m > 0$ ：度量空间中的每个半径为 r 的球，都可以被 2^m 个半径为 $\frac{r}{2}$ 的球覆盖。注意， d 维欧几里得度量的倍增维数为 $\mathcal{O}(d)$。

树宽。图 G 的树分解是一个二元组 (T, β)，其中 T 是一棵树， β： $V(T) \to 2^{V(G)}$，$V(T)$ 和 $V(G)$ 分别表示树 T 和 G 的顶点，并满足以下性质。

1. 对于每个 $v \in V(G)$，都存在 $t \in V(T)$，使得 $v \in \beta(t)$，

对于每个 $(u, v) \in E(G)$，存在 $t \in V(T)$ 使得 $u, v \in \beta(t)$，并且

3.对于每个 $v \in V(T)$，由 T 在 $\{t$ 上诱导的子图： $v \in \beta(t)\}$，是连通的。

树分解 (T, β)的宽度 为$\max_{t \in V(T)} |\beta(t)| - 1$。图 G的树宽 是 G的所有树分解中的最小宽度。

范数的次梯度。 我们陈述定义并总结范数次梯度的基本事实，这些内容将在全文中使用。

事实 3.1. 任何范数都是凸函数。

定义 3.2 (次梯度)。 凸函数 f: $\mathbb{R}^n \to \mathbb{R}$ 在任意点 $\boldsymbol{x} \in \mathbb{R}^n$处，是满足以下条件的任意 $\boldsymbol{g} \in \mathbb{R}^n$，对每个 $\boldsymbol{y} \in \mathbb{R}^n$

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\intercal(\boldsymbol{y} - \boldsymbol{x});$$

我们用 $\partial f(\boldsymbol{x})$ 表示 f 在 $\boldsymbol{x}$ 处的次梯度集合。

以下事实总结了专门针对范数函数的次梯度的各种有用性质。由于我们仅将范数目标应用于非负距离向量，因此我们也将范数在 $\mathbb{R}^n_{\geq 0}$ 上的限制称为范数（略微滥用了术语）。

事实 3.3 ([CS19]). 设 $f\colon \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ 是一个范数，且 $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$。如果 $\boldsymbol{g}$ 是 f 在 $\boldsymbol{x}$处的次梯度，那么$f(x) = \boldsymbol{g}\ \boldsymbol{x}$且$^\intercal f(y) \geq \boldsymbol{g}\ \boldsymbol{y}$对所有$\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$均成立。此外，如果 f是单调的，则存在一个次梯度$\boldsymbol{g} \in \partial f(\boldsymbol{x})$ 使得 $\boldsymbol{g} \geq 0$。

以下观察是事实 3.3 的直接推论。

观察 3.4。令 $\partial f = \bigcup_{\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}} \partial f(\boldsymbol{y})$为 f 的所有次梯度构成的集合。那么对于任意$\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$，我们有

$$f(\boldsymbol{x}) = \max_{\boldsymbol{g} \in \partial f} \boldsymbol{g}^\intercal \boldsymbol{x}\,.$$

定义 3.5 (ϵ-近似次梯度)。设 $f\colon \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ 为一个范数，且令 $\epsilon > 0$。我们将集合 $\partial_\epsilon f(\boldsymbol{x})$ 定义为 f 在 $\boldsymbol{x}$ 处的 ϵ-近似次梯度，其中包含所有满足以下两个条件的 $\boldsymbol{g} \in \mathbb{R}^n_{\geq 0}$

(i) $f(\boldsymbol{y}) \geq \boldsymbol{g}\ \boldsymbol{y}$对每个 $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$ 均成立，且

(ii) $f(\boldsymbol{x}) \leq (1+\epsilon)\boldsymbol{g}^\intercal \boldsymbol{x}$.

众所周知，可以通过 Grötschel、Lovasz 和 Schrijver 在其经典著作 [GLS12]中给出的归约，利用函数的（近似）值预言机高效计算凸函数的近似次梯度。尽管 [GLS12] 中的归约似乎至少需要调用预言机 $\Omega(n^{10})$ 次，但在假设凸函数具备额外性质时，存在更快的方法，例如参见 [LV06, LSV18]。具体而言，对于 ℓ_p 范数，已有描述次梯度集合的闭式公式，并且在实践中得到了应用。

一些术语和记号。令 $M = (P, F, \delta)$ 为一个定义在 $n = |P|$ 个数据点上的聚类空间。令 $\boldsymbol{b} \in \mathbb{R}^P_{\geq 0}$ 为一个 n 维向量。我们将 $\boldsymbol{b}$ 解释为给每个点 $p \in P$ 赋予一个非负值，记作 $b(p)$。也就是说，$\boldsymbol{b} = (b(p))_{p \in P}$。例如，给定一个中心子集 $X \subseteq F$，我们将距离向量定义为 $\boldsymbol{\delta}(P, X) = (\delta(p, X))_{p \in P}$。如果 $B \subseteq P$ 是一个点子集，那么 $\mathbf{1}_B \in \{0,1\}^P$ 表示 B 的特征向量，也就是说，它为任意 $b \in B$ 赋值 $\mathbf{1}$，为任意 $p \in P \setminus B$ 赋值 0。如果 $p \in P$ 且 $\alpha \geq 0$，那么我们用 $\mathbf{1}_{p,\alpha}$ 表示二进制向量 $\mathbf{1}_{\mathsf{ball}(p,\alpha) \cap P}$。

4 ϵ-散射维数

本节将正式介绍 ϵ-散射维数的概念，它在我们的算法框架中发挥着核心作用。下面的定义是对引言中所述“中心点博弈”的形式化。

定义 4.1 (ϵ-散射维数)。给定一个有限度量空间类 $\mathcal{M}$ 、其中的一个空间$M = (P, F, \delta)$ 以及某个 $\epsilon \in (0, 1)$中的值 $\mathcal{M}$。一个 ϵ-散射是 M 中的一个序列 $(x_1, p_1), \ldots, (x_\ell, p_\ell)$，由中心—点对 $x_i \in F$, $p_i \in P$, $i \in [\ell]$ 组成，满足

$$\begin{aligned} &\delta(x_i, p_j) \leq 1 && \text{for all } 1 \leq j < i \leq \ell && \text{(covering)} \\ &\delta(x_i, p_i) > 1 + \epsilon && \text{for all } i \in [\ell] && (\epsilon\text{-refutation}) \end{aligned}$$

ϵ-散射维数的 M 是其中一个 ϵ-散布的最大长度。ϵ-散射维数的 $\mathcal{M}$ 是所有 $M \in \mathcal{M}$ 上的 ϵ-散射维数的上确界。

注意，对于任意 ϵ-散射 $(x_1, p_1) \ldots, (x_\ell, p_\ell)$，任意子序列$(x_{i_1}, p_{i_1}) \ldots, (x_{i_{\ell'}}, p_{i_{\ell'}})$，其中 $i_1 < \cdots < i_{\ell'}$ 和 $\ell', i_j \in [\ell], j \in [\ell']$ 也是一个 ϵ-散射。

如定理 1.8 所述，我们证明了有界（算法化） ϵ-散射维数基本上足以在相应的度量空间中为范数 k-聚类提供一种高效参数化近似方案。在第 6.2 节中，我们证明有界树宽图度量、平面图度量以及有界倍增度量具有有界 ϵ-散射维数。由此，我们能够在所有这些度量中获得高效多项式时间近似方案。为了处理高维欧几里得空间，我们采用 ϵ-散射维数的算法化版本。

优化中心策略。回顾引言中的例子，该例表明高维（连续）欧几里得空间 $\mathbb{R}^d$ 的 ϵ-散射维数是无界的。我们构造了一个 ϵ-散布 (x_1, p_1).....，(x_{d-1}, p_{d-1})，其中 x_i 是按$1/\sqrt{2}$缩放的 i 个单位向量,_ 并且对所有 $i \in [d-1]$ 都有 $p_i = -x_i$。请注意，在这个例子中，原点周围的单位球包含序列中的所有点。因此，如果中心玩家改进其策略，上述例子就会失效。这促使我们考虑一种变体：用一个更审慎地计算中心的算法替代中心玩家。此外，采用算法还使我们能够处理无限空间。

球相交问题与算法。为此，我们将中心玩家需要解决的算法问题形式化。我们采用并推广了对中心-点游戏的一种对偶解释：中心玩家试图在点玩家所选各点周围的单位球的交集中找到一个中心。事实上，我们考虑更一般的非均匀球情形，其中散布中的每个点 p 都有其专属半径 r。

设 $\mathcal{M}$ 为度量空间类 (P, F, δ)，其中心集合 F可能是无限集。我们定义如下搜索问题。

球相交
输入： 度量空间 $M = (P, F, \delta) \in \mathcal{M}$，一个由有限个距离约束组成的集合 $Q \subseteq P \times \mathbb{R}_+$。**输出：** 一个点 $x \in F$，使其满足所有距离约束，即对每个 $(p, r) \in Q$，都有 $\delta(x, p) \leq r$；若这样的点不存在，则输出“失败”。

对于有限度量空间，可以通过穷举中心空间 F，高效地求解球交问题。遗憾的是，对于某些无限度量空间，例如高维连续欧几里得空间，我们尚不了解求解球交问题的精确算法。因此，我们采用近似算法。为形式化地定义这一点，若中心 $x \in F$ η-满足 距离约束 $(p, r) \in Q$，误差参数为 $\eta > 0$，则称其满足该约束$\delta(x, p) \leq (1+\eta)r$。设 $\mathcal{C}_\mathcal{M}$ 是一个（确定性）算法，其输入为一个球交问题实例和一个误差参数 $\eta > 0$。如果该算法满足以下条件，则称其为近似球交算法（简称球交算法）。

(i) 算法输出一个对所有距离约束均 η-满足的中心，或者失败。

(ii) 如果存在一个中心能够精确满足所有点的距离约束，那么算法不会失败。

(iii) $\mathcal{C}_\mathcal{M}$ 的运行时间是 poly$(|M|, 1/\eta)$。

我们指出，对于高维欧几里得空间 [KY09], ，存在一个近似球交算法，我们将在第6节.4中使用该算法来证明此度量的有界算法 ϵ-散射维数。

算法的 ϵ-散射维数。 算法的算法的 ϵ-散射维数 的定义基于（$\mathcal{C}_{\mathcal{M}}$，$\epsilon$）-散布的概念，它是 ϵ-散布的一种变体：中心通过（近似）球交算法 $\mathcal{C}_{\mathcal{M}}$ 而不是由对抗性的中心玩家来选择。直观地说，我们维护一个动态的球交实例，并通过逐一添加距离约束（p，r）对其进行扩充。在（$\mathcal{C}_{\mathcal{M}}$，$\epsilon$）-散布的语境中，我们将距离约束（$p$，$r$）称为请求，这些请求由球交算法按顺序满足。

定义 4.2（算法 ϵ-散射维数）。设 $\mathcal{M}$ 是一个度量空间类，具有 球相交算法 $\mathcal{C}_{\mathcal{M}}$；设 $M = (P, F, \delta)$ 是 $\mathcal{M}$ 上的一个度量，并设 $\epsilon \in (0, 1)$。此外，对每个 $i \in [\ell]$，设 $p_i \in P$、$x_i \in F$ 和 $r_i \in \mathbb{R}_+$，其中 ℓ 是一个正整数。序列（x_1，p_1，r_1）...，（x_ℓ，p_ℓ，r_ℓ）称为 （$\mathcal{C}_{\mathcal{M}}$，$\epsilon$）-散射，若满足以下两个条件。

（*i*）对于每个 $2 \le i \le \ell$，都有 $x_i = \mathcal{C}_{\mathcal{M}}$（$M$，{（$p_1$，$r_1$）...，（$p_{i-1}$，$r_{i-1}$）}，$\epsilon/2$）。（序列中的第一个中心 x_1 不受任何要求限制。）

*(ii)*此外，$\delta(x_i, p_i) > (1+\epsilon) r_i$ 对于每个 $i \in [\ell]$。

我们称 $\mathcal{M}$ 具有 算法的 $(\epsilon, \mathcal{C}_{\mathcal{M}})$-散射维数 $\lambda_{\mathcal{M}}(\epsilon)$ 如果任意 $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散布至多包含 $\lambda_{\mathcal{M}}(\epsilon)$ 个具有相同半径值的三元组。 $\mathcal{M}$的算法的 ϵ-散射维数 是在任意 球相交 算法$\mathcal{C}_{\mathcal{M}}$ 上 $\mathcal{M}$的算法的$(\epsilon, \mathcal{C}_{\mathcal{M}})$-散射维数中的最小值。

当上下文中能够明确 $\mathcal{M}$ 这一族时，我们会省略 $\mathcal{M}$ 在 $\lambda_{\mathcal{M}}(\epsilon)$ 和 $\mathcal{C}_{\mathcal{M}}$中的下标。请注意，与 ϵ-散射维数不同，对于算法的 ϵ-散射维数，我们要求每个半径值对应的三元组数量受到限制，而不是序列的总长度。事实上，对于高维欧几里得空间而言，这一更强的要求并不成立，而较弱的（算法的）要求事实证明足以满足我们的结果。另一个值得注意的区别是，$(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散布的子序列不一定本身也是一个$(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散布，因为它可能与算法 $\mathcal{C}_{\mathcal{M}}$ 的行为不一致。

算法与非算法 ϵ-散射维数之间的关系。 下面的引理表明，算法 ϵ-散射维数确实推广了有限度量空间中的 ϵ-散射维数。

引理 4.3。 任何具有 ϵ-散射维数 $\lambda(\epsilon)$ 的有限、显式给定的度量空间类，也具有算法 ϵ-散射维数 $\lambda(\epsilon)$。

证明。 设 $M = (P, F, \delta)$ 是给定类中的一个度量空间，并带有一个由 Q 个距离约束组成的集合。我们的球交算法穷举搜索 F，以找到一个满足所有距离约束恰好的中心 x。如果不存在这样的点，该算法将失败。令 $\mathcal{C}$ 表示该算法。考虑任意一个 $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散布。注意，具有相同半径值的任意三元组子序列都会构成一个 ϵ-散布。因此，对于任意半径值，该序列至多包含 $\lambda(\epsilon)$ 个三元组。

□

算法的长宽比引理 ϵ-散射维数。 </style}下面是有界算法 -散射维数的一个实用推论，我们在证明结果时会用到。它通过限制半径落在长宽比有界的区间内的三元组数量（而不是限制半径值相同的三元组数量），强化了 $(, \epsilon)$-散布的性质。

引理 4.4。 设 $\mathcal{M}$ 是算法 ϵ-散射维数为 $\lambda(\epsilon)$的一类度量空间。那么存在一个 球交 算法 $\mathcal{C}_{\mathcal{M}}$ ，具有以下性质。给定 $\epsilon \in (0,1)$、 $a > 0$ 和 $\tau \geq 2$，任意一个 $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散布都包含 $\mathcal{O}(\lambda(\epsilon/2)(\log \tau)/\epsilon)$ 个半径落在区间 $[a, \tau a]$ 内的三元组。

证明。只需证明一个较弱的结论：区间 $[a,(1+\epsilon/100)a]$中的请求数至多为 $2\lambda(\epsilon/2)$。该结论蕴含此引理，因为区间 $[a, \tau a]$ 可以用$\mathcal{O}((\log \tau)/\epsilon)$个形如 $[(1+\epsilon/100)^j, (1+\epsilon/100)^{j+1}]$, $j \in \mathbb{Z}$的区间覆盖。

设 $\mathcal{A}_{\mathcal{M}}$ 是一个球交算法，且该算法的 $(\epsilon, \mathcal{A}_{\mathcal{M}})$-散射维数为 $\lambda(\epsilon)$。令 $\eta \in (0, 1)$ 为输入误差参数。考虑如下工作的球交算法 $\mathcal{C}_{\mathcal{M}}$ ：对于任意输入请求 (p, r)，我们将 r 舍入为 r'，它是大于 r的最小的 $1+\eta/50$ 次幂。然后，我们使用误差参数 $\eta/2$ ，在舍入后的请求上调用 $\mathcal{A}_{\mathcal{M}}$ ，并输出其返回的中心 $\mathcal{A}_{\mathcal{M}}$。显然，该算法是一个（针对原始请求的）$(1+\eta)$-近似球交算法。

考虑任意算法式 $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散射 $(x_1, p_1, r_1) \ldots, (x_\ell, p_\ell, r_\ell)$。设 r'_i, $i \in [\ell]$ 是由 $\mathcal{C}_{\mathcal{M}}$ 计算得到的取整半径。设 $\epsilon' = \epsilon/2$。设 $1 \leq j < i \leq \ell$。我们有 $\delta(x_i, p_i) > (1+\epsilon)r_i \geq (1+\epsilon)/(1+\epsilon/100)r'_i \geq (1+\epsilon')r'_i$。此外，我们有 $\delta(x_i, p_j) \leq (1+\epsilon'/2)r'_i$。因此，序列 $(x_1, p_1, r'_1) \ldots, (x_\ell, p_\ell, r'_\ell)$ 是一个算法式 $(\mathcal{C}_{\mathcal{M}}, \epsilon')$-散射。请求 (p_i, r_i),$i \in [\ell]$ 中落在区间 $[a,(1+\epsilon/100)a]$ 内的半径，由于 $\mathcal{C}_{\mathcal{M}}$ 是在误差参数为 $\eta = \epsilon/2$ 的条件下调用的，因此经 $\mathcal{C}_{\mathcal{M}}$ 取整后至多只有两个不同的半径值。因此，未取整的序列中，半径位于区间 $[a,(1+\epsilon/100)a]$ 内的三元组数量至多为 $2\lambda(\epsilon') = 2\lambda(\epsilon/2)$。这就完成了该断言的证明，进而完成了引理的证明。 □

5 框架　　高效参数化近似方案的工作

主要结果。 现在我们可以陈述主要结果了。本节余下部分将证明以下定理，该定理在引言中已有陈述。第 5.1 节介绍高效参数化近似方案并提供一些直观说明。第 5.2 节给出完整的技术分析。

定理 1.8。 设 $\mathcal{M}$ 是一类度量空间，在将距离乘以正数后仍然封闭。存在一种随机化算法，对于任意带有度量 $M = (P, F, \delta) \in \mathcal{M}$ 的 范数 k-聚类 实例$\mathcal{I} = (M, f, k)$，以及任意 $\epsilon \in (0、1)$ 中的数，以高概率计算出一个$(1+\epsilon)$-近似解，前提是满足以下两个条件。

(i) 存在一种高效算法，可在 $T(f)$ 时间内，对任意距离向量 $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ 计算目标函数 $f(\boldsymbol{x})$。

(ii) 存在一个函数 λ: $\mathbb{R}_+ \to \mathbb{R}_+$，使得对于所有 $\epsilon > 0$， $\mathcal{M}$ 的算法 ϵ-散射维数至多为 $\lambda(\epsilon)$。

该算法的运行时间为 $\exp\left(\widetilde{\mathcal{O}}\left(\begin{smallmatrix} k\lambda_{(\epsilon/10} \\ \epsilon \end{smallmatrix}\right)\right)$.) $\underline{poly(|M|)} \cdot T(f)$。

5.1 算法

我们的算法在算法 1 中作了正式表述。我们对算法的关键步骤作非正式概述，这些步骤也曾在技术概述中作过部分说明。我们还对分析给出了一些直观解释。

通过标准的枚举技术，我们假设已知最优目标函数值 OPT（或其足够精确的近似值）。我们的目标是满足<style id='1'>凸约束

$f(\boldsymbol{x}) \le (1+\epsilon)\mathrm{OPT}$ 施加在距离向量 $\boldsymbol{x} \in \mathbb{R}^P_{\ge 0}$（其表示可行解 $X \subseteq F$ 所诱导的距离向量$\boldsymbol{\delta}(P,\ X)$）上。根据观察 3.4，该约束等价于（无穷多个）线性约束 $\boldsymbol{w}^\intercal \boldsymbol{x} \le (1+\epsilon)\mathrm{OPT}$，其中 $\boldsymbol{w} \in \partial f$ 是 f 的任一次梯度。

为说明主要思想，我们首先描述一种高度简化但失败的尝试。在while循环的每次迭代中（第8–15 行），我们考虑一个候选解 X。如果 $f(\boldsymbol{x}) \le (1+\epsilon)$ 最优值，那么我们就完成了。否则，我们在第9 行计算 f 在 $\boldsymbol{x}$ 处的一个（$\epsilon/10$-近似）次梯度 $\boldsymbol{w}$。由于$\boldsymbol{w}^\intercal \boldsymbol{x} = f(\boldsymbol{x}) > (1+\epsilon)$最优值，这构成了一个违反的线性约束。考虑以与其对目标函数的贡献 $w(p)\delta(p, X)$ 成比例的概率抽样一个点 $p \in P$，即 $f(\boldsymbol{x}) = \boldsymbol{w}^\intercal \boldsymbol{x}$（第11行）。一个平均论证表明，以$\Omega(\epsilon)$ 的概率，抽样的点 p 满足$\delta(p, X) > (1+\epsilon/3)\delta(p, O)$，其中某个固定的假设的最优解为 O。在此情况下，我们识别出了一个违反的距离约束，并将 p 称为 X的 $\epsilon/3$-见证点。我们将 p 分配给一个均匀随机选取的簇 $\kappa \in [k]$，该簇以概率 $1/k$等于 O 中 p 的正确簇。假设这两个事件都发生，这使我们能够将请求 (p, r)（其半径值为 $r = \delta(p, X)/(1+\epsilon/3)$）添加到施加在索引为 κ的簇上的簇约束 Q_κ 中。（见第 13 行和第14行。）这里，我们将簇 κ 的请求集合 Q_κ 称为 κ的簇约束。

固定簇索引 $\kappa \in [k]$。设 $(p_\kappa^{(1)}, r_\kappa^{(1)}) \ldots, (p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ 是添加到与簇 κ 相关联的簇约束中的请求序列。设 $x_\kappa^{(i)}$，$i \in [\ell]$ 是在将请求 $(p_\kappa^{(i)}, r_\kappa^{(i)})$ 添加到 Q_κ 之前，簇 κ 的中心。关键观察是，三元组序列 $(x_\kappa^{(1)}, p_\kappa^{(1)}, r_\kappa^{(1)}) \ldots, (x_\kappa^{(\ell)}, p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ 构成一种算法式 ϵ-散射。我们希望论证该序列的长度有界，因为算法式 ϵ-散射维数有界。不幸的是，散射维数只能限制每个半径值对应的三元组数量，无法限制序列的总长度。

为了解决这个问题，我们在第 1 行计算任意点 $p \in P$ 的半径初始上界 $u(p)$。在贪心预处理步骤中，我们（近似地）满足所有点的这些初始距离约束（见第 2–7 行）。在主要阶段，我们将这些距离约束作为初始请求加入，从而维持它们（见第 5 行）。上界 $u(p)$ 粗略估计了在任何请求中可能施加于 p 的最小半径 r（$p,\ r$）。我们修改主要阶段的采样过程（见第 11 行），使其仅从一组点中抽样，这些点到 X 的距离并不比其初始上界 $u(p)$ 小很多。通过仔细论证，我们证明所提出的每个请求（$p,\ r$）都以概率 $\Omega(\epsilon/k)$ 与 O 一致。此外，我们还证明，针对特定簇提出的所有请求半径彼此相差不超过因子 $\mathcal{O}(k/\epsilon^2)$。初始上界通过在输入实例中检测“稠密”球（第 1 行）来计算：根据目标范数的某个次梯度，这些球会获得较高权重，因此任何近似最优解都必须在该稠密球附近放置一个中心。

算法 1：范数 k-聚类框架 数据：范数 k-聚类的实例 $\mathcal{I} = ((P, F, \delta)\ k, f\colon \mathbb{R}^P_{>0} \to \mathbb{R}_{\geq 0})$，误差参数$\epsilon \in (0, 1)$，最优值 > 0，依据引理 4.4 的球交算法 $\mathcal{C}$ 结果：若存在代价至多为 OPT 的解，则返回代价至多为 $(1+\epsilon)$OPT 的解 X 1 对每个 $p \in P$，计算 $u(p) = \min\{\alpha > 0 \mid f(\mathbf{1}_{p,\alpha/3}) \geq 3\text{OPT}/\alpha\}$； 2 按 $u(p)$ 非递减顺序对 P 排序； 3 若球 $\text{ball}(p_i, u(p_i))$ 与对每个 $j < i$ 的球 $\text{ball}(p_j, u(p_j))$ 不相交，则标记 $p_i \in P$； 4 令 $p^{(1)}, \ldots, p^{(k')}$ 为这些标记点。；// 引理 5.5 表明 $k' \leq k$5 令 $Q_\kappa = \{(p^{(\kappa)}, u(p^{(\kappa)}))\}$，对所有 $\kappa \in [k']$ 均成立； 6 对所有满足 $k' < \kappa \leq k$ 的 κ，令 $Q_\kappa = \emptyset$； 7 令 $X = (x_1, \ldots, x_k)$ 为任意一组中心，其中 x_κ 满足 Q_κ 中的请求； 8 当 $f(\boldsymbol{\delta}(P, X)) > (1+\epsilon)$OPT 时，执行9 $\boldsymbol{w} \leftarrow \epsilon/10$- f 在 $\boldsymbol{\delta}(P, X)$ 处的次梯度；10 $A \leftarrow \left\{ p \in P \mid \delta(p, X) \geq \frac{\epsilon u(p)}{1000k} \right\}$； 11 抽取一个元素 $p \in A$，其中对于任意 $a \in A$，$\Pr[p = a] = \frac{w(a)\delta(a,X)}{\sum_{b \in A} w(b)\delta(b,X)}$； 12 均匀随机地选取簇 $\kappa \in [k]$，对应于 p；13 $Q_\kappa \leftarrow Q_\kappa \cup \{(p, \delta(p, X)/(1+\epsilon/3))\}$；14 $x_\kappa \leftarrow \mathcal{C}(Q_\kappa, \epsilon/10)$ 若未找到 x_i，则失败； 15 结束 16 返回 X；

5.2 分析

概述。 分析包括确立以下三个事实。第一，如果算法未发生失败便终止，则会计算出一个 $(1+\epsilon)$-近似解。第二，算法会在一定次数的迭代后终止——无论是否发生失败，该次数仅取决于 k 和 ϵ。第三，算法不发生失败的概率也仅取决于 k 和 ϵ。

分析的第一步可直接由 while 循环的停止准则（第 8 行）得出。

观察 5.1 (正确性)。 如果算法成功终止（即未发生失败），则会输出一个$(1+\epsilon)$-近似解。

分析的第二步概括为以下引理，我们将在第 5.2.1 小节中证明该引理。

引理 5.2 (运行时间界)。 算法会在 $\mathcal{O}\left(\frac{k(\log k/\epsilon)}{\epsilon} \lambda\, \epsilon/10\right)$() 次迭代后终止——无论是否发生失败。

有了这两点认识，我们还需要完成第三步，具体如下引理所述，我们将在第 5.2.2 小节中证明该引理。

引理 5.3 （概率界）。 算法以$\exp\left(-\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right)$的概率成功终止（即未发生失败）。

我们将算法重复执行 $\exp\left(\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right)$ 次，因此根据引理 5.3，我们以高概率成功。

本节余下部分将用于证明引理 5.2 和引理 5.3，从而完成主要定理 1.8 的证明。

5.2.1 迭代次数上界

在本小节中，我们证明引理 5.2。证明分为三个步骤。首先，我们论证为每个点 $p \in P$ 计算的初始上界距离 $u(p)$ (i) 与任何最优解一致（引理 5.4），且 (ii) 在算法执行过程中近似得到满足（引理 5.5）。其次，我们证明，针对任意特定簇所提出的请求中的半径彼此相差不超过一个有界因子（长宽比）（引理 5.6）。第三步是证明，对于任意特定簇，请求序列及其对应的中心构成一个长宽比有界的算法式 $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散布。因此，我们可以使用引理 4.4 来界定该序列的长度，从而将迭代次数限制为关于 k 和 ϵ 的函数，由此完成引理 5.2 的证明。

初始上界。我们首先证明，算法中计算的初始上界是保守的，即最优解也满足这些上界。

引理 5.4. 如果 O 是一个最优解，那么 $\delta(p, O) \leq u(p)$ 对于任意 $p \in P$，其中 $u(p)$ 是算法 *1* 第 *1* 行中计算得到的初始上界。

证明。令 $\alpha = u(p)$。为导出矛盾，假设 $\delta(p, O) > \alpha$。根据三角不等式，球$(p, \alpha/3)$中的任何点 $p' \in$ 到 O 的距离至少为 $2\alpha/3$。因此有 $\delta(P, O) \geq (2\alpha/3) \cdot \mathbf{1}_{p,\ \alpha/3}$ 从而 $f(\delta(P, O)) \geq f((2\alpha/3) \cdot \mathbf{1}_{p,\alpha/3}) = (2\alpha/3) f(\mathbf{1}_{p,\alpha/3}) \geq (2\alpha/3) \cdot {}^{3}$最优值$_{\alpha} = 2$最优值，这是一个矛盾。 □

下面的引理表明，在整个算法过程中，我们近似满足所有上界。我们指出，初始化（第 2–7 行）以及分析都是 Plesník 的 Priority k-Center 算法的一种变体；该算法应用于点集 P，其半径为 $u(p)$， $p \in P$。

引理 5.5。 在算法 *1* 的第 *3* 行中标记的点数 k' 至多为 k。此外，在 *while* 循环（第 *8–15* 行）的执行过程中任意时刻，我们都有 $\delta（p, X） \leq 4u（p）$。对于添加到某个簇约束中的任意请求 $（p, r）$，我们有 $r \leq 4u（p）$。

证明。 根据引理 5.4，每个球球$(p^{(\kappa)},\ u(p^{(\kappa)}))$，带有标记 $p^{(\kappa)}, \kappa \in [k']$，至少包含来自某个假设的最优解 O中的一个点。另一方面，根据构造，这些球两两不相交。因此 $k' \leq |O| \leq k$。这也意味着，算法可以在第 7 行初始化$X = (x_1、\ldots\ldots、 x_k)$，其中心满足所有初始簇约束。例如，它可以选取 F 中最接近 $p^{(\kappa)}$的 k' 中心，以及许多其他任意中心。

由于这些初始请求从未被移除，因此每当相应的簇发生变化时，我们都会将它们传递给球交算法（误差参数为 $\epsilon/10$；见第 14 行）。因此，在 while 循环的整个执行过程中，对于任意标记点 p，都有 $\delta(p, X) \leq (1 + \epsilon/10) u(p) \leq 3u(p)/2$。对于任意未标记的点 p'，存在某个标记的 p，使得 ball$(p', u(p'))$ 与 ball$(p, u(p))$ 相交。由于第 3 行按 $u(\cdot)$ 的非递减顺序处理点，因此必有 $u(p) \leq u(p')$。如前所述，ball$(p, 3u(p)/2)$ 在 while 循环期间的任意时刻都保证包含 X 中的一个中心。根据三角不等式，该中心到 p' 的距离至多为 $u(p') + 2 \cdot 3u(p)/2 \leq 4u(p')$。对于第二个结论，请注意，在第 14 行首次处理该请求时，有 $r < \delta(p, X) \leq 4u(p)$。 □

界定请求的长宽比。 以下引理证明，为同一个簇提出的任意两个请求，其半径之比不超过 $\mathcal{O}(k/\epsilon^2)$。直观来说，我们在算法中（见第 10 行）确保只采样半径之比不超过 $\mathcal{O}(k/\epsilon)$ 的点，其中 $u(p)$。假设这两个半径，因此初始界限 $u(p)$、 $u(p')$，

对于同一个簇的两个请求 $(p,\ r)$、$(p',\ r')$，彼此相差很大，比如 $r' \ll r$ 和$u(p') \ll u(p)$。那么这就意味着，在提出请求 $(p,\ r)$ 之前，p 已经（实质上）处于距某个中心 r 的半径范围内，因为根据引理 5.5，距 p' 必有一个中心，其距离在 $4u(p') \ll \epsilon r/3$ 范围内。这与最初提出请求 $(p,\ r)$ 的假设相矛盾。

引理 5.6。 设(p, r) 和(p', r') 是第 *13* 行中按任一顺序加入同一簇约束Q_κ, $\kappa \in [k]$ 的请求。如果 $r' \le \epsilon^2 \cdot r/(10^4 k)$，那么算法在处理这两个请求中的第二个时，会在第 *14* 行失败。

证明。 假设为了导出矛盾，算法没有失败，但找到了中心 x_κ，使得 $\delta(p,\ x_\kappa) \le (1+\epsilon/10)r$ 以及 $\delta(p',\ x_\kappa) \le (1+\epsilon/10)r'$。因此，根据三角不等式，$\delta(p,\ p') \le (1+\epsilon/10)(r+r')$。根据引理 5.5，我们有 $r \le 4u(p)$ 和 $r' \le 4u(p')$。由于我们从第 10 行中定义的集合 A 采样点，因此有 $r \ge \epsilon u(p)/(200k)$ 以及 $r' \ge \epsilon u(p')/(200k)$。

假设 $r' \le \epsilon^2 r/(10^4 k)$。在将 (p, r) 添加到 Q_κ 时，当前候选解 X满足 $\delta(p', X) \le 4u(p') \le 1000kr'/\epsilon$ 根据引理 5.5。因此

$$\begin{aligned}\delta(p, X) &\le \delta(p, p') + \delta(p', X)\\ &\le (1+\epsilon/10)(r+r') + 1000kr'/\epsilon\\ &\le (1+\epsilon/4)r\,.\end{aligned}$$

然而，这是一个矛盾，因为 $\delta(p, X) = (1+\epsilon/3)r$ 在请求 (p, r) 时向 Q_κ，正如第 14 行所示。

□

利用有界算法 ϵ-散射维数。为完成引理 5.2 的证明，我们固定某个簇，并考虑三元组序列 $(x,\ p,\ r)$，其中$(p,\ r)$ 是针对该簇发出的请求，并且 x 是发出请求之前该簇的中心。我们证明该序列构成一个算法$(\mathcal{C}_\mathcal{M},\ \epsilon)$-散布，并使用引理5.6 将该序列中半径的长宽比界定为 $\mathcal{O}(k/\epsilon^2)$。最后通过长宽比引理 4.4 完成证明。

引理 5.2 的证明。固定一个簇索引 $\kappa \in [k]$。设 $(p_\kappa^{(1)}, r_\kappa^{(1)}) \ldots, (p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ 是按添加到 Q_κ （第 13 行）的顺序排列的请求序列。对于任意 $i \in [\ell]$，设 $x_\kappa^{(i)}$ 是在请求 $(p_\kappa^{(i)}, r_\kappa^{(i)})$ 之前，簇 κ 的中心。由于 $r_\kappa^{(i)} = \delta(p_\kappa^{(i)}, X)/(1+\epsilon/3) \le \delta(p_\kappa^{(i)}, x_\kappa^{(i)})/(1+\epsilon/3)$，且 $x_\kappa^{(i)}$ 是通过对 $\{(p_\kappa^{(1)}, r_\kappa^{(1)}) \ldots, (p_\kappa^{(i-1)}, r_\kappa^{(i-1)})\}$ 调用 $\mathcal{C}$ 并使用误差参数 $\epsilon/10$ 计算得到的，因此序列 $(x_\kappa^{(1)}, p_\kappa^{(1)}, r_\kappa^{(1)}) \ldots, (x_\kappa^{(1)}, p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ 是一个算法式 $\epsilon/5$-散布。

根据引理 5.6，$r_\kappa^{(i)} \in R_\kappa = \left[r_{\min}, \frac{10^4 k r_{\min}}{\epsilon^2}\right]$ 对于每个 $i \in [\ell]$，其中 $r_{\min}$ 表示簇 κ的所有请求中的最小半径。将引理 4.4 应用于区间 R_κ，序列的长度为 $\mathcal{O}((\log k/\epsilon)\lambda(\epsilon/10)/\epsilon)$。由于我们的算法在每次迭代中都会向某个簇约束添加一个请求，因此迭代总次数为 $\mathcal{O}(k(\log k/\epsilon)\lambda(\epsilon/10)/\epsilon)$。

□

5.2.2 成功概率的界

引理 5.3 的证明包括两个关键步骤：首先，我们论证，如果算法所作的随机选择与某个假设的最优解“一致”（其更精确的定义将在下文给出），那么算法将成功终止（即不会失败）。其次，我们论证，在每次迭代中，以足够高的概率保持一致性。结合引理 5.2 给出的迭代次数上界，即可完成对主要结果定理 1.8 的证明。

一致性。 非正式地说，我们所说的一致性是指：**(i)** 请求中的点被分配到正确的（最优）簇中；**(ii)** 任何请求（p， r）中的半径 r 都是合理的，也就是说，不小于 p 到其最优中心的距离。

定义 5.7. 考虑一个固定的假设的最优解 $O = (o_1, .., o_k)$。我们称算法 *1* 的当前执行状态（由 $(X, Q_1, ..., Q_k)$ 指定）与一致于 O，若对于任意请求 $(p, r) \in Q_\kappa, \kappa \in [k]$，都有 $\delta(p, o_\kappa) \leq r$。

如果当前状态与最优解 O 一致，那么 O 证明当前施加的簇约束（$Q_1, \ldots, Q_k$）存在解。因此，以下观察是显然的。

观察 5.8. 如果算法的状态在任意一次迭代执行第 *14* 行之前与 O 一致，那么算法在该次迭代中不会失败。

保持一致性的概率。 如果在某次迭代开始时，执行状态与 O 一致，那么在以下两个条件下，它将继续保持一致。首先，本次迭代中采样的点 p 被（随机地）分配到正确的簇。其次， p 到当前候选解的距离足够大于其到 O的距离，从而为第 13 行发出的请求提供充分依据。第二个条件促成了下面的定义。

定义 5.9。 给定一个满足 $f(\delta(P, X)) > (1 + \epsilon)$最优值的候选解 X，若点 $p \in P$满足 $\delta(p, X) > (1 + \epsilon)\delta(p, O)$，则称其为 ϵ-见证点 。

下面的引理表明，在每次迭代中，针对抽样点提出的请求都有 $\Omega(\epsilon)$ 的概率是合理的。这是我们分析的关键部分，因为它将 **(i)** 计算初始上界的具体方式与 **(ii)** 基于这些上界抽取见证点的方式联系起来。正是这种相互作用使我们能够界定针对特定簇的请求中半径的长宽比，从而将每个簇的请求总数限制在 k 和 ϵ 的范围内。

引理 5.10。 考虑算法 *1* 的 *while* 循环中的一次固定迭代，并令 X 为该迭代开始时的候选解。那么，第 *11* 行抽取的点以 $\Omega(\epsilon)$ 的概率是 X 的 $\epsilon/3$-见证点。特别地，第 *10* 行计算的集合 A 非空。

证明。对于任意子集 $S \subseteq P$的点，令 $C_S = \sum_{p \in S} w(p)\delta(p, X)$表示 S对 $w\ \delta(P, X) = C_P$的贡献。 T

令 $W \subseteq P$ 为 X的 $\epsilon/3$-见证点子集。我们断言，贡献 C_W 至少为 $\epsilon C_P/10$。为导出矛盾，假设它们的贡献更少。那么，利用

$0 < \epsilon < 1$,

$$\begin{aligned}
\mathsf{OPT} &\geq \boldsymbol{w}^\intercal \boldsymbol{\delta}(P, O) \\
&\geq \sum_{p \in P \setminus W} w(p)\delta(p, O) \\
&\geq \frac{1}{1 + \epsilon/3} \sum_{p \in P \setminus W} w(p)\delta(p, X) \\
&\geq \frac{1 - \epsilon/10}{1 + \epsilon/3} \sum_{p \in P} w(p)\delta(p, X) \\
&\geq \frac{\boldsymbol{w}^\intercal \boldsymbol{\delta}(P, X)}{1 + \epsilon/2} \\
&\geq \frac{f(\boldsymbol{\delta}(P, X))}{(1 + \epsilon/2)(1 + \epsilon/10)} \\
&\geq \frac{f(\boldsymbol{\delta}(P, X))}{1 + 3\epsilon/4}
\end{aligned}$$

这与 $f(\boldsymbol{\delta}(P, X)) > (1 + \epsilon)$最优值矛盾。

令 W_1、.....、W_k 分别表示在 X 中距离中心 x_1、.....、x_k 最近的见证点子集。

令 $H \subseteq [k]$ 为簇子集 $\kappa \in [k]$，使得 $C_{W_\kappa} \geq \epsilon C_P/(100k)$。任取一个簇 $\kappa \in H$。令 $\{z_1, \ldots, z_\ell\}$ 表示 W_κ 中的见证点，按照它们到最近簇中心 x_κ 的距离 $\delta(z_i, x_\kappa) i \in [\ell]$ 非递减排序。令 $j \in [\ell]$ 为最小指标 j，使得集合 $W_\kappa^- = \{z_1, \ldots, z_j\}$ 的贡献至少为 $C_{W_\kappa}/2$。这也意味着 $C_{W_\kappa^+} \geq C_{W_\kappa}/2$, 其中 $W_\kappa^+ = \{z_j, \ldots, z_\ell\}$。因此， $C_{W_\kappa^-}$ 和 $C_{W_\kappa^+}$ 均至少为 $\epsilon C_P/(200k)$, 因为 $\kappa \in H$。

我们声称 $W_\kappa^+ \subseteq A$，其中 A 的定义与算法 1 第 10 行相同。为此，任取$p \in W_\kappa^+$。我们证明 $u(p) \leq 1000k\delta(p, x_\kappa)/\epsilon$，因此 $p \in A$。注意，球$(p, 2\delta(p, x_\kappa)) \supseteq$球$(x_\kappa, \delta(p, x_\kappa)) \supseteq W_\kappa^-$。另一方面,

$$\frac{\epsilon \mathsf{OPT}}{300k} \leq \frac{\epsilon f(\boldsymbol{\delta}(P, X))}{300k} \leq \frac{\epsilon C_P}{200k} \leq \sum_{q \in W_\kappa^-} w(q)\delta(q, x_\kappa) \leq \delta(p, x_\kappa) \sum_{q \in W_\kappa^-} w(q)\,.$$

设定 $\alpha = 6\delta(p, x_\kappa)$，这意味着

$$f(\mathbf{1}_{p,\alpha/3}) \geq \boldsymbol{w}^\intercal \mathbf{1}_{p,\alpha/3} = \sum_{q \in \mathsf{ball}(p,\alpha/3)} w(q) \geq \sum_{q \in W_\kappa^-} w(q) \geq \frac{\epsilon \mathsf{OPT}}{200k\delta(p, x_\kappa)} = \frac{\epsilon \cdot 3\mathsf{OPT}}{100k\alpha}\,. \tag{1}$$

因此， $u(p) \leq 100k\alpha/\epsilon \leq 1000k\delta(p, x_\kappa)/\epsilon$ 如所声称。至此完成了以下断言的证明：$W_\kappa^+ \subseteq A$ 对于任意 $\kappa \in H$。

如上所示，$\sum_{\kappa \in [k]} C_{W_\kappa} = C_W \geq \epsilon C_P/10$。$H$ 是 $\sum_{\kappa \in [k] \setminus H} C_{W_\kappa} \leq \epsilon C_P/100$ 的。根据定义，我们有 $\sum_{\kappa \in H} C_{W_\kappa} \geq \epsilon C_P/20$。因此。此外，根据上述论证，

$$C_{A \cap W} \geq \sum_{\kappa \in H} C_{W_\kappa^+} \geq \sum_{\kappa \in H} \frac{C_{W_\kappa}}{2} \geq \frac{\epsilon C_P}{40} \geq \frac{\epsilon C_A}{40}\,.$$

由于我们以与其贡献 $C_{\{p\}}$ 成正比的概率从 A 中采样一个点 p，因此在每次迭代中，我们以至少为 $\epsilon/40$ 的概率采样一个见证点。

注意， $C_P \geq f(\boldsymbol{\delta}(P, X))/2 > 0$。因此，公式 1 的左侧必须为正数。这意味着 A 不为空。

□

总体成功概率。现在我们可以证明引理 5.3，从而完成主定理 1.8 的证明。我们将证明，在进入算法 1 的 while 循环之前，执行状态是一致的。结合迭代次数的上界（引理 5.2）和维持一致性的概率下界（引理 5.10），即可完成证明。

引理 5.3 的证明。令 $p^{(1)}, \ldots, p^{(k')}$ 为算法 1 第 3 行标记的点。根据引理 5.4，每个球$(p^{(\kappa)}, u(p^{(\kappa)}))$，$\kappa \in [k']$ 都包含一个来自 O 的点。根据构造，这些球两两不相交。因此，通过重新标记最优中心 $O = (o_1, \ldots, o_k)$，我们可以假设，对于每个标记点 p，都有$\delta(p^{(\kappa)}, o_\kappa) \leq u(p^{(\kappa)})$，其中 $\kappa \in [k']$ 是簇的索引。因此，在 while 循环（第 8–15 行）首次执行之前，算法的执行状态与 O 一致。现在假设在某次 while 循环迭代开始时，状态与 O 一致。根据引理 5.10，在这次迭代中，我们以 $\Omega(\epsilon)$ 的概率采样到一个 $\epsilon/3$-见证点 p。在这种情况下，添加的请求 (p, r) 的半径为 $r = \delta(p, X)/(1 + \epsilon/3) \geq \delta(p, O)$。此外，如果随机选取的簇索引 $\kappa \in [k]$ 与 O 中的索引相同（其发生概率为 $\Omega(1/k)$），那么状态仍与 O 一致。在这种情况下，第 14 行重新计算中心不会失败。根据引理 5.2，算法至多经过 $\mathcal{O}\left(\frac{k(\log k/\epsilon)}{\epsilon} \lambda \epsilon/10\right)$ 次迭代后终止。由于在每次迭代中算法以 $\Omega(\epsilon/k)$ 的概率不失败，因此算法最终成功的概率为 $\exp\left(-\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right)$。

□

6 ϵ-散射维数界

本节旨在为各类度量中的 ϵ-散射维数给出界，并证明引言中的定理 1.3、1.4 和 1.6。

6.1 有界倍增维数

本节将给出任意倍增维数为 d 的度量空间中 ϵ-散射维数的上界，并证明定理 1.3。

散射维数与打包。给定度量 (X, δ)，该度量的一个 ϵ-打包是满足以下条件的点子集 $X' \subseteq X$：对于所有 $i, j \in X'$，均有 $\delta(i, j) \geq \epsilon$。这是度量空间理论中的一个标准概念。我们首先观察到，ϵ-散布与 ϵ-打包之间存在如下联系。

观察 6.1。设 $(x_1, p_1) \ldots, (x_\ell, p_\ell)$ 是度量空间 (P, F, δ)中的一个 ϵ-散布。那么，中心 $X = \{x_2, \ldots, x_\ell\}$ 是在度量 $(P \cup F, \delta)$ 中的一个 ϵ-打包，并且 X 包含在一个单位球内。

证明。根据 ϵ-散射的定义，对于每个 $j \in \{2, \ldots, \ell\}$，都有 $\delta(p_1, x_j) \leq 1$。此外，对于任意 $i, j \in \{1, \ldots, \ell\}$，由于 $\delta(x_j, p_j) > 1 + \epsilon$ 且 $\delta(p_j, x_i) \leq 1$，其中 $j < i$，由三角不等式可得$\delta(x_i, x_j) > \epsilon$。若 $i \neq 1$，则 $\delta(p_{i-1}, x_i)$ 和 $\delta(p_{i-1}, x_j)$ 均 ≤ 1。因此，由三角不等式，$\delta(x_i, x_j) \leq \delta(x_i, p_{i-1}) + \delta(p_{i-1}, x_j) \leq 2$。

□

推论 6.2. 度量 M 中单位球的 ϵ打包大小至多为 ϵ散射维数减一。

众所周知，任意倍增维数为 d 的度量的 ϵ-打包大小至多为 $\mathcal{O}((1/\epsilon)^d)$。将这一点与观察 6.1 结合即可得到定理 1.3。

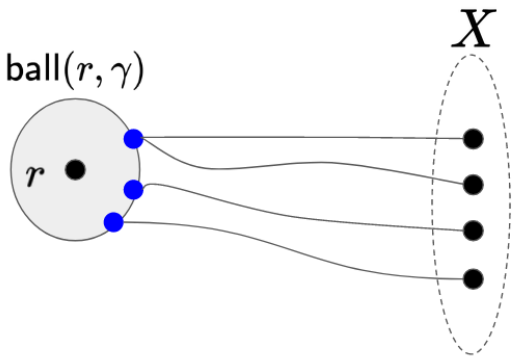

图4： $S =$ 上的一个蜘蛛$_G$球（r， γ）。连接 X 与 r 的路径彼此不相交，但S中的节点除外。

备注： 我们注意到，推论 6.2 的逆命题即使在非常简单的图度量（例如星图）中也不成立。在一个以 r 为根的 n 节点星图中，单位球 球(r,1) 包含整个图。通过选择非根节点，可以得到一个大小为($n-1$)的 ϵ-打包。然而，任何 ϵ-散布的长度至多为 2。

6.2 有界树宽图

本节将证明，任何树宽为 tw 的图都具有 ϵ-散射维数 $\mathrm{tw}^{(1/\epsilon)^{\mathcal{O}(\mathrm{tw})}}$。也就是说，我们针对有界树宽图度量证明定理 1.4。稍后我们将展示，平面图的界可以通过 [FRS19]的一项嵌入结果推导出来。为方便起见，我们用 ball$_G$(r， γ) 表示 ball$_{\delta G}$(r， γ)。

6.2.1 树宽与蜘蛛图

我们的证明依赖于蜘蛛图这一概念。蜘蛛图的存在可以作为“见证点”，表明图 G 的树宽很高。给定一个边权图 G、 $X \subseteq V$（G）和 $\gamma \in$ (0， 1)， γ-*spider on* X 是一个集合 $S =$ 球$_G$(r， γ)，其中某个 $r \in V$(G)满足：存在 $|X|$ 条从 S 到X 的路径，这些路径除 S内的部分外，彼此顶点不相交。若集合 S 是 X 上的一个 γ-蜘蛛图（对于某个 γ），则称其为 X 上的蜘蛛图。图4对此作了示意。

注意，如果 S 是 X 上的 γ-蜘蛛，那么对于任意 $X' \subseteq X$， S 也同样是 X'上的 γ-蜘蛛。下面的引理是我们结果的关键，它大致表明，大量蜘蛛图的存在意味着 G 的树宽很大。

引理 6.3. 设 G 是一个图， k 是一个整数，且 $X \subseteq V(G)$： $|X| > 3k$。若存在一个由 $k+1$ 个两两不相交的 X上蜘蛛组成的族 $\mathcal{S}$，那么 G 的树宽大于 k。

证明。 反设树宽至多为 k。那么，存在一个平衡分隔集$A \subseteq V$(G)，使得 $|A| = k$，以及 $G - A = V_1 \uplus V_2$ 的一个划分，使得 E(V_1， V_2) $= \emptyset$,$|V_1 \cap X|$， $|V_2 \cap X| \geq |X|/3$ [CFK+15] （众所周知）。

我们声称，每个蜘蛛 $S \in \mathcal{S}$必须在分隔集中包含一个顶点，即 $S \cap A \neq \emptyset$。为导出矛盾，设存在 $S \in \mathcal{S}$ 使得 $S \subseteq V_i$ 对于某个 $i \in \{1,2\}$。不失一般性，令 $S \subseteq V_1$。回顾一下， S 是一个以 X为基底的蜘蛛。因此，从 S到 X中互不相同顶点存在 $|X|$ 条内部顶点不相交路径。由于 $|V_2 \cap X| \geq |X|/3 > k$，至少有 $k+1$

从 $S(\subseteq V_1)$ 到 $V_1 \cap X$ 的内部顶点不相交路径。由于 A 是一个 (V_1, V_2)-分隔集，所有这些顶点不相交路径都经过 A。因此， $|A|$ 至少等于这些路径的数量（$k+1$），这就产生了矛盾。由此得出，每个蜘蛛 $S \in \mathcal{S}$ 都与 A 相交。

由于 $\mathcal{S}$ 是一族两两顶点不相交的蜘蛛，因此我们得出 $|A| \geq k+1$，矛盾。

□

6.2.2 迭代寻找蜘蛛图

本节的主要结果由以下定理概括。

定理 6.4. 如果在 G 中存在一个长度至少为$(\mathcal{O}(k/\epsilon))^{(4/\epsilon)^{k+1}}$ 的 ϵ-散布，那么图 G 包含一族两两不相交的蜘蛛，其顶点集大小大于 $3k$。

结合上述结果与引理 6.3，我们可以推出，任意 ϵ-散布的长度至多为 $\mathsf{tw}^{(1/\epsilon)^{\mathcal{O}(\mathsf{tw})}}$，正如所期望的那样。本节余下部分将用于证明该定理。给定 ϵ-散布 σ，我们称由观察 6.1 给出的 ϵ-打包 $X = X(\sigma)$ 为 σ 的规范打包。

引理 6.5. 设 σ 是 $G \subseteq$ 球$_G(r,1)$ 中的一个长度为 ℓ 的 ϵ-散布， $X = X(\sigma)$ 是其典范的 ϵ-打包。那么，存在

- 一只蜘蛛 $S =$ 球$_G(r, \epsilon/3)$ 在 $X' \subseteq X$: $|X'| \geq c_0\epsilon \cdot (\ell/2)^{\epsilon/3}$ 对于某个常数 c_0 和
- 一个图 G'，满足 $S \cap V(G') = \emptyset$，以及一个 ϵ-散射 σ'，它是 σ 的子序列，满足$X(\sigma') = X'$。

在证明这个引理之前，我们说明它如何推出定理6.4。设 $G_0 = G$ 包含一个长度至少为 $\ell_0 = (\frac{k}{c_0\epsilon})^{(4/\epsilon)^{k+1}}$ 且为 $X_0 = X(\sigma_0)$ 的 ϵ-散布 σ_0。该引理允许我们在 X_1 上找到一个大小为 $c_0\epsilon \cdot \ell_0^{\epsilon/3} \geq (\frac{k}{c_0\epsilon})^{(4/\epsilon)^k} = \ell_1$ 的蜘蛛 S_1,一其中 ϵ充分小。此外，我们有图 G_1，它与 S_1 不相交，并且有一个长度为 ℓ_1的 ϵ-散布，它是 σ_1 的子序列。由于(G_1, X_1, σ_1) 满足引理 6.5 的前提条件，我们可以应用该引理得到 (G_2, X_2, σ_2)，依此类推。更正式地，从(G_i, σ_i, X_i)开始，我们应用引理 6.5 得到$(G_{i+1}, \sigma_{i+1}, X_{i+1})$。我们保持以下不变量：序列 σ_i 的长度满足 $\ell_i = |X_i| \geq (\frac{k}{c_0\epsilon})^{(4/\epsilon)^{k+1-i}}$。这使我们能够在 X_{k+1} 上找到互不相交的蜘蛛 $S_1, S_2, \ldots, S_{k+1}$： $|X_{k+1}| > 3k$，正如所需。

6.2.3 引理6.5的证明

令 G 包含于单位球 ball(r,1) 中。证明分为两步。第一步，我们在相对较大的子集 $X'' \subseteq X$ 上找到一个蜘蛛 S。第二步，我们证明，从 G 中移除 S 后得到的图 G' 仍包含一个较大的子序列 σ'，其规范打包是 X'' 的一个子集 X'，且具有所需的基数。

第一步：令 T 为从 r 到 X 的最短路径树（回想一下 $|X| = \ell$ X），因此， 中的顶点出现在这棵树的叶节点处。我们从 T 出发，按如下方式归纳地构造子集 $V \subseteq V$（T）上的“辅助”树 T。令 $B_r =$球$_T$（$r, \epsilon/3$）。从 T 中移除 B_r，得到以 $r_1, \ldots, r_q$ 为根的子树 $T_1, \ldots, T_q$。对于每个 $i \in [q]$，令 $X_i \subseteq X$ 为 T_i 中属于 X 的 r_i 的后代。由于 X 中的顶点位于叶节点处，我们有 $X_i \neq \emptyset$。我们在 上对实例（T_1, X_q）.....（T_q, X_q）归纳地执行此过程，从而得到（T_i, X_i）对应的辅助子树 T_i。现在，通过将 r 连接到 $r_1, \ldots, r_q$（使它们成为 r 的直接子节点），来构造 T。参见图5。

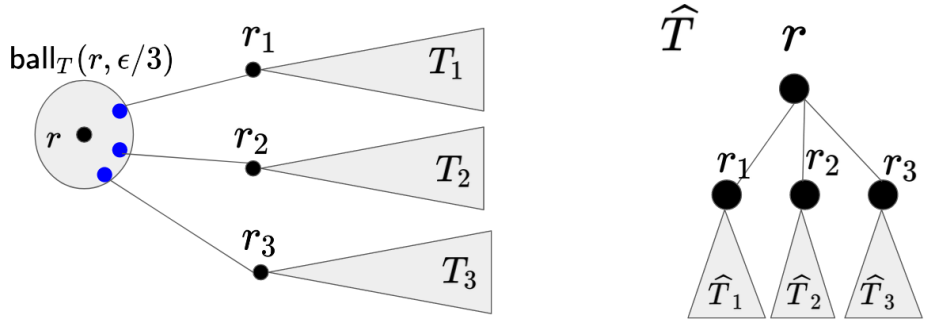

图5：树 T 的递归构造。

对于每个 $v \in V(T)$，用 T_v 表示以 v 为根的 T 的子树，用 B_v 表示通过递归过程构造的球 ball$_{T_v}(v, \epsilon/3)$。注意，$\bigcup_{v \in V(\widehat{T})} B_v \supseteq X$，并且 T 的深度至多为 $(3/\epsilon)$（因为对于所有 $x \in X$，$\delta(r, x) \leq 1$，且每次递归都会使根到叶节点的距离减少 $\epsilon/3$。）

断言6.6。 存在 必须是一个顶点 $r' \in V(\widehat{T})$，使得 r' 至少有 $D = (\ell/2)^{\epsilon/3}$ 个子节点位于 $\widehat{T}$.

证明。假设对于 T中的每个顶点，其子节点数量都小于 D。那么，$\widehat{T}$中的顶点总数小于 $2D^{3/\epsilon}$。对于每个这样的顶点 $v \in V(\widehat{T})$，我们有 $|B_v \cap X| \leq 1$（因为 X ϵ B_v $2\epsilon/3$ $\ell = |X| \leq \bigcup_v |B_v \cap X| \leq$ 是一个 -打包，而的直径至多为）。因此，$D \geq \ell/2^{\epsilon/3}$。这将意味着 $2D^{3/\epsilon}$。 □

令 v 为 V中满足以下条件的节点：在 T 中，它距离根最近，并且至少有 D 个子节点（任意打破平局）。这意味着，在创建 T 的过程中，移除 $B_v =$ 球$_{T_v}(v, \epsilon/3)$ 会得到至少 D 个子树 $T_1, \ldots, T_D$，其中每棵这样的树都包含（任意选择的） $x_i \in X_i$ 作为 T 中的后代。注意，S 是 $X'' = \{x_1, x_2, \ldots, x_D\}$ 上的一只蜘蛛。

第二步： 设 σ'' 为 σ 的子序列，其规范的 ϵ-打包为 X''，即 $X(\sigma'') = X''$。回忆 $|X''| \geq (\ell/2)^{\epsilon/3}$。将蜘蛛 S 记为 $S =$ 球$_G(s, \epsilon/3)$。在这第二步中，我们证明 $G' = G \setminus S$ 仍包含一个较长的 ϵ-散布 σ'，它是 σ'' 的子序列，且使得$X' = X(\sigma') \subseteq X''$ 具有理想的长度。

根据构造，我们有 $X'' \subseteq$ 球$_G(s, 1)$。我们根据到 s 的距离，将 X'' 中的顶点划分为至多 $3/\epsilon$ 个子集，如下所示：对于 $i = 1$、.....、$3/\epsilon$，令 $X''_i = \{x \in X'' : \delta_G(s, x) \in (i\epsilon/3, (i+1)\epsilon/3]\}$。定义 X' 为具有最大基数的集合，因此$|X'| \geq \frac{\epsilon}{3} \cdot (\ell/2)^{\epsilon/3}$。注意，对于所有 $u, u' \in X'$，我们都有 $|\delta(s, u) - \delta(s, u')| < \epsilon/3$。下面的断言说明，$S$ 中的每个点到 X' 中每个点的距离大致相同（参见图 6 进行说明）。

断言 6.7。 设 $v \in S$和 $u, u' \in X'$。则 $|\delta(v, u) - \delta(v, u')| < \epsilon$。

证明。不失一般性，假设 $\delta(v, u) \leq \delta(v, u')$。根据三角不等式，$\delta(v, u') \leq \delta(v, s) + \delta(s, u') \leq \delta(v, s) + \delta(s, u) + \epsilon/3 \leq \delta(s, u) + 2\epsilon/3$。再次应用三角不等式，可得$\delta(s, u) \leq \delta(s, v) + \delta(v, u)$，从而得到所需的界。 □

以下论断将完成证明。

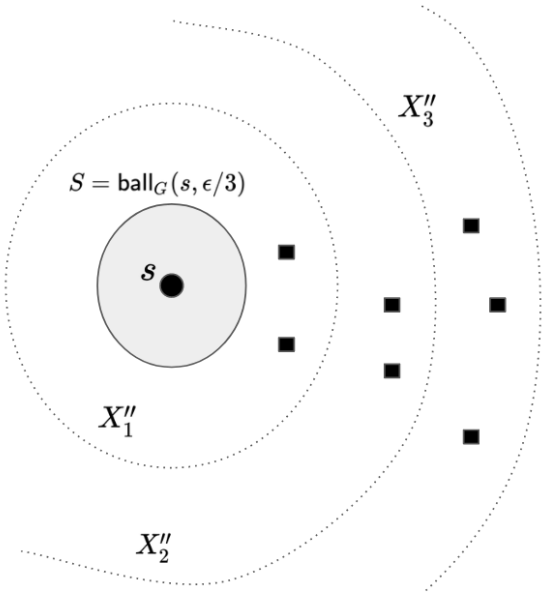

图 6：根据 X'' 中各点到蜘蛛 S 的距离，将其划分为 $\{X_i''\}$ 。矩形点表示 X'' 中的点。

命题 6.8. 令 σ' 为 σ'' 的子序列，其规范打包为 X'。则 σ' 是 $G' = G \setminus S$中的一个有效ϵ-散射。

证明。 我们将 $\delta_{G'}$ 简记为 δ'。设 (x, p) 和 (x', p') 是 σ' 中的两对元素，且 (x, p) 出现在 (x', p') 之前。由于 σ 是散布的，我们有 $\delta(x', p) \leq 1$，而 $\delta(x, p)\ \delta(x', p')\ \delta(x, p') > (1 + \epsilon)$。

注意，移除 S 后，这些点对仍满足反驳性质，即 $\delta'(x, p)\ \delta'(x', p')$,$\delta'(x, p')\ > (1 + \epsilon)$（从图中移除顶点后，距离不会减小）。因此，只需证明 $\delta'(p, x') \leq 1$。为此，我们说明，从 p 到 x'、位于 G 中的任意一条最短路径都不会与球 S 相交。反设存在一条从 p 到 x'、位于 G 中的最短路径 Q，它在某个顶点 $v \in S \cap Q$ 处与 S 相交。注意，$\delta(p, x') = \delta(p, v) + \delta(v, x')$。我们将通过证明 $\delta(p, x) \leq (1 + \epsilon)$ 得出矛盾。由于 $\delta(p, x) \leq \delta(p, v) + \delta(v, x)$，根据命题 6.8，该值至多为 $\delta(p, v) + \delta(v, x') + \epsilon = \delta(p, x') + \epsilon$，这将意味着 $\delta(p, x) \leq (1+\epsilon)$，与反驳性质矛盾。

□

6.3 通过低树宽嵌入界定 ϵ-散射维数

在本节中，我们展示界定 ϵ-散射维数与一个关于加性失真嵌入的活跃研究领域之间的（简单）联系 [FEKS19、FL22、CAFKL20]。这一联系使我们能够给出平面图的 ϵ-散射维数上界。

具体而言，我们称（加权）图类 $\mathcal{G}$ 具有函数 t 的 t-低树宽-直径嵌入：$\mathbb{N} \to \mathbb{N}$ 如果存在一个确定性算法，该算法接收 G 并生成树宽至多为 $t(\eta)$ 的加权图 H 以及一个嵌入 $\phi\colon V(G) \to V(H)$，使得：

$$\delta_G(u, v) \leq \delta_H(\phi(u), \phi(v)) \leq \delta_G(u, v) + \eta D$$

其中 D 是 G的直径。

定理 6.9。 设 $\lambda_{\mathtt{tw}}(\epsilon)$ 表示树宽为 tw 的图的 ϵ-散射维数（由上一节可知，该界至多为关于 tw 的双指数）。如果图类 $\mathcal{G}$ 具有 t-低树宽-直径嵌入，那么 $\mathcal{G}$ 中的每个度量的 ϵ-散射维数至多为 $\lambda_{t(\epsilon/10)}(\epsilon/3)$。

证明。 令 (x_1, p_1)、(x_2, p_2).....(x_ℓ, p_ℓ) 是 G 中的 ϵ-散布。令 $\eta = \epsilon/10$。考虑将 G 嵌入到 H 中的一个嵌入ϕ，使得 H 的树宽至多为 $t(\eta) = t(\epsilon/10)$。注意到

- 对于 $1 \le i < j \le \ell$: $\delta_H(\phi(x_j)\ \phi(p_i)) \le 1 + 2\eta = 1 + \epsilon/5$

  对于 $1 \le i \le \ell$: $\delta_H(\phi(x_i)\ \phi(p_i)) > 1 + \epsilon$

考虑通过将 H 的权重按 $(1+\epsilon/5)$的倍数缩小而得到的（带权）图 H'。我们有 $\delta_{H'}(\phi(x_j)\ \phi(p_i)) \le 1$，而当 $\epsilon > 0$ 足够小时， $\delta_{H'}(\phi(x_i)\ \phi(p_i)) > \frac{1+\epsilon}{1+\epsilon/5} \ge 1 + \epsilon/3$。这意味着嵌入序列在 H中是$(\epsilon/3)$-散布的。因此，根据定理 1.4，长度 ℓ 的上界为 $\lambda_{t(\epsilon/10)}(\epsilon/3)$。

□

现在我们可以使用以下定理。

定理 6.10 （ [FEKS19]的定理 1.3）。 存在一个多项式时间算法，给定一个边带权平面图和一个数 $\eta > 0$，输出该图到一个树宽为 $poly(1/\eta)$ 的平面图的嵌入，且加性误差为 $\eta \cdot D$，其中 D 是输入图的直径。

此实现 ies，换句话说，在我们的术语中，平面图具有低树宽-直径嵌入 g.

花冠 理 6.11. 平面图的 ϵ-散射维数至多为 $\exp(\exp(\mathsf{poly}(1/\epsilon)))$。

6.4 高维欧几里得空间

回顾引言和第4 节可知，高维（连续）欧几里得空间的 ϵ散射维数是无界的。然而，在本节中我们将证明，该度量的算法ϵ散射维数是有界的。

定理1.7 (界定算法散布维数**).** 连续欧几里得空间 (P, F, δ)，即， $P \subsetneq \mathbb{R}^d$ 有限，且 $F = \mathbb{R}^d$，具有算法 ϵ-散射维数 $\mathcal{O}(1/\epsilon^4 \log 1/\epsilon)$。

本节余下部分将用于证明定理 1.7。为了给出连续欧几里得空间中算法散射维数 ϵ 的上界，只需证明存在一个算法 $\mathcal{C}$，使得欧几里得空间中的 $(\mathcal{C}, \epsilon)$-散布维数有界。我们采用 Kumar 和 Yildirim [KY09] 提出的高维欧几里得空间球交算法。他们以加权欧几里得 1-中心问题的术语研究球交问题。他们基于凸优化形式提出了一种球交算法，该算法能够在连续欧几里得空间中，高效地（近似地）求解长宽比有界的权重对应的球交问题。令$\mathcal{C}_{\mathrm{KY}}$ 表示该算法。下面的引理改编自 Kumar 和 Yildirim 的工作，并转换为我们的术语（参见 [KY09]中的引理 4.2）。

引理 6.12. 给定高维欧几里得空间中的一个（P, F, δ）球交实例，为每个 $p \in P$关联半径 $r(p)$，且 $\epsilon \in$（0，1），那么任意（$\mathcal{C}_{\mathrm{KY}}, \epsilon$）-散布的长度至多为 $\mathcal{O}(\tau/\epsilon^2)$，其中 $\tau \ge 1$ 是请求中最大半径与最小半径之比的平方。

注意，对于常数 τ，引理 6.12 即可得出该定理的证明。为完成证明，我们将证明：只要将 ϵ-散布的长度乘以因子 $\mathcal{O}(\log 1/\epsilon)$，就可以假设 τ 是 $\mathcal{O}(1/\epsilon^2)$。

长宽比条件。 以下引理给出了算法的 ϵ-散射维数有界的充分条件，这有助于设计球交算法，以界定算法的 ϵ-散射维数。具体而言，该条件是界定高维连续欧几里得空间的算法 ϵ-散射维数的关键。它可以视为长宽比引理 5.6 的一种强化逆命题。

引理 6.13（长宽比条件）。 设 $\mathcal{M}$ 是一类具有 球交算法 $\mathcal{C}_{\mathcal{M}}$ 的度量空间，且 $\epsilon \in (0, 1)$。若任意 $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-散射 (x_1, p_1, r_1) ..., (x_ℓ, p_ℓ, r_ℓ)，其中$r_i \in [\epsilon/12, 1]$, $i \in [\ell]$，包含相同半径的三元组至多 $\lambda(\epsilon)$ 个，则 $\mathcal{M}$ 的算法散射维数有界于 $\mathcal{O}(\lambda(\epsilon) \log 1/\epsilon)$。

为证明引理6.13，我们假设已获得一个如 $\mathcal{C}_{\mathcal{M}}$所述的球交算法。我们声称，下面这个调用 $\mathcal{C}_{\mathcal{M}}$ 作为子程序的球交算法，根据定义 4.2 的条件，可得到算法性的 ϵ-散射维数 $\mathcal{O}(\lambda(\epsilon)\log 1/\epsilon)$。

算法 2：实现引理 6.13 的球交算法。输入：度量空间 $M = (P, F, \delta) \in \mathcal{M}$，请求 $Q = (p_1, r_1) \ldots (p_\ell, r_\ell)$，其中 $p_i \in P$和$r_i \in \mathbb{R}^+$ 适用于 $i \in [\ell]$；误差参数 $\eta > 0$；如引理 6.13 所示的球交算法 $\mathcal{C}_{\mathcal{M}}$。输出：中心 $x \in F$，使得 $\delta(p_i, x) \le (1+\eta)r_i$ 对所有 $i \in [\ell]$ 成立；否则输出“失败”**1** $\rho \leftarrow \min\{\, 2^{-j} \mid j \in \mathbb{N}_0$ 和 $\min_{i\in[\ell]} r_i \le 2^{-j} \,\}$；**2** $Q' \leftarrow \{(p_i, r_i) \mid i \in [\ell],$ $\eta/3 \cdot r_i \le \rho\}$；**3** $x \leftarrow \mathcal{C}_{\mathcal{M}}(M, Q', \eta)$；4 对每个 $i \in [\ell]$ 执行：5 如果 $\delta(p_i, x) > (1+\eta)r_i$，则输出“失败”；6 结束；7 返回 x；

下面的定义给出了两个请求 $(p,\ r)$，$(p',\ r') \in Q$ 满足何种条件时，只需满足 $(p,\ r)$，便也能满足 $(p',\ r')$。

定义 6.14。 设 $\eta \in (0,\ 1)$ 且 $(p, r),\ (p', r') \in Q$。如果球$(p,(1+\eta)r) \subseteq$ 球$(p',(1+\eta)r')$，则称 p η-蕴含 p'。

引理 6.15。 设$(p, r),\ (p', r') \in Q$ 是两个请求，满足 $r \le \eta/3 \cdot r'$，其中 $\eta \in (0,\ 1)$。如果存在某个中心位于 F 且同时满足这两个请求，那么 p η-蕴含 p。

证明。令 o 为同时满足两个请求的中心，并令 $x \in$ 球$(p,(1+\eta)r(p))$。根据三角不等式

$$\begin{aligned}\delta(p', x) &\le \delta(p', o) + \delta(o, p) + \delta(p, x)\\ &\le r' + r + (1+\eta)r\\ &\le (1+\eta)r'\end{aligned}$$

□

引理 6.16。算法 *2* 是一种球相交算法。

证明。 假设存在某个 $o \in F$，使得 $\delta(p_i, o) \le r_i$ 对所有 $i \in [\ell]$均成立。我们希望证明，对于任意 $i \in [\ell]$，我们的算法不会失败，并输出一个中心 x，使得 $\delta(p_i, x) \le (1+\eta)r_i$。

令 Q'，ρ，x 如算法 2 中所定义。考虑任意一个 $(p_i,\ r_i) \in Q$。我们区分两种情况。首先，假设 $(p_i,\ r_i) \in Q'$。假设 $\mathcal{C}_{\mathcal{M}}$ 是一个正确的球交算法，

我们有 $\delta(p_i, x) \leq (1+\eta)r_i$。另一方面，如果 $(p_i, r_i) \in Q \setminus Q'$，那么存在某个 $(p_j, r_j) \in Q'$，使得 $r_j \leq \rho < \eta/3 \cdot r_i$。因此，根据引理 6.15，$p_j$ η-蕴含 p_i。如上所述，$x \in \text{ball}(p_j, (1+\eta)r_j)$，因此根据定义 6.14，$x \in \text{ball}(p_i, (1+\eta)r_i)$。 □

我们通过证明以下引理来结束引理 6.13 的证明。

引理 6.17。令 $\mathcal{C}$ 表示算法 *2*。任何 $(\mathcal{C}, \epsilon)$-散布随后都有 $\mathcal{O}(\lambda(\epsilon) \log 1/\epsilon)$ 三元组具有相同的半径。

证明。令 Q'、ρ、x 如算法2中所定义。考虑任意一个 $(\mathcal{C}, \epsilon)$-散布 (x_1, p_1, r_1)，....., (x_ℓ, p_ℓ, r_ℓ)，并令 Q 表示其请求序列 (p_1, r_1)....., (p_ℓ, r_ℓ)。注意，假设我们按照长度递增的顺序对 Q 的所有前缀运行算法，则 ρ 的值沿该序列单调递减。我们将该序列划分为多个阶段，每个阶段都是其中值 ρ 不发生变化的极大（连续）子序列。固定一个阶段。注意，在此阶段中，集合 Q' 将按包含关系递增，因为 ρ保持不变。如果 (p_i, r_i) 是在该阶段某个时刻添加到 Q' 中的任意请求，则 $\rho/2 \leq r_i \leq 3\rho/\eta$。将距离按因子 $\eta/(3\rho)$ 重新缩放，并使用 $\eta = \epsilon/2$ 可知，该阶段内所有被请求的半径都落在范围 $[\epsilon/12, 1]$内。因此，根据引理6.13中关于子程序 $\mathcal{C}_\mathcal{M}$ 的假设，在该阶段中，对于每个半径值 r_i，散布至多包含 $\lambda(\epsilon)$ 个三元组。

我们通过指出，对于任何固定的半径值 r_i，至多有 $\log_2(12/\epsilon)$ 个阶段，在此期间，具有该半径的请求会被添加到 Q'，从而完成证明。 □

引理6.12和6.13，连同注意到 $\tau = (12/\epsilon)^2$，给出了定理 1.7 的证明。

7 结论和开放问题

我们提出了一个关于高效参数化近似方案的统一视角，该视角适用于各种聚类目标和度量空间。从复杂性理论的角度来看，这意味着对称和非对称范数聚类问题的可近似性出现了相当令人惊讶的坍缩（在$\mathbf{P}$的情形下，它们的可近似性存在实质性差异，然而在$\mathbf{FPT}$中却会坍缩）。

未来的开放问题可以沿着两个能够独立推进的方向展开。首先，我们能否刻画具有有界散射维数的度量空间类？例如，不含次式的图是否具有有界散射维数？这是一个纯粹的结构性问题，其解决将立即导出一个高效参数化近似方案（通过我们的框架）。由于有界树宽图在我们的方法中发挥着重要作用（通过低树宽嵌入的视角），因此，确定它们的 ϵ-散射维数的精确界将很有意义；一个尤其有趣的具体问题是，该界是否可以降至单指数级。

第二个方向涉及对我们框架的扩展。我们的框架仍未涵盖某些聚类目标。例如，如何处理带离群点的聚类 [KLS18、BOR21、DZ22] （在这种情况下，代价函数 f 反而是一个反范数）？从更概念性的角度来看，我们当前的算法并不关注输入度量的结构，但我们的定理只能讨论是否能够获得高效参数化近似方案。对于这样的框架，是否有可能在所有范围内都给出近似因子（例如，当输入是一个一般的、无结构的度量空间时，能否为 k-中心问题给出$(3 + o(1))$-近似）？“理想结果”或许可以是类似于$(\gamma(\mathcal{M}) + o(1))$ 的 FPT 近似算法，其中 $\gamma(\mathcal{M})$表示度量类 $\mathcal{M}$的 FPT 可近似性。

最后但同样重要的是，本文的范围在于处理多种聚类目标函数和度量空间。许多聚类问题还会对 P 中的点能够如何

被分配给 X中的开放中心，例如容量约束 [ABM+19、DL16、Coh20], 、不同的公平性概念 [BCFN19、CKLV17、CFLM19]以及多样性约束 [LYZ10、TOG21、TGOO22]；在这种情况下，我们的框架不适用。将我们的框架扩展到处理此类约束（或证明在强制施加此类约束时不存在高效多项式时间近似方案）是一个有趣的研究方向。

致谢

Fateme Abbasi、Sandip Banerjee 和 Jarosław Byrka 获得波兰国家科学中心（NCN）项目资助，资助编号为 2020年/39/B/ST6/01641。Parinya Chalermsook 获得欧洲研究委员会（ERC）资助，资助来源为欧盟“地平线 2020”研究与创新计划（资助协议编号 759557）。Dániel Marx 获得欧洲研究委员会（ERC）整合者项目资助，项目编号为 725978 SYSTEMATICGRAPH。

参考文献

[ABM+19] Marek Adamczyk、Jarosław Byrka、Jan Marcinkowski、Syed M Meesum 和 Michał Włodarczyk。带容量限制的 k-中位数问题的常数因子 FPT 近似。载于第 *27* 届欧洲算法研讨会（*ESA'19*），第 1 页。Schloss Dagstuhl-Leibniz 信息学中心有限公司，Dagstuhl 出版社，2019年。[ABV21] Mohsen Abbasi、Aditya Bhaskara 和 Suresh Venkatasubramanian。通过公平的群体代表实现公平聚类。载于*2021*年 *ACM* 公平性、问责制与透明度会议论文集（*FAccT'21*），第 504–514 页，2021年。[ACKS15] Pranjal Awasthi、Moses Charikar、Ravishankar Krishnaswamy 和 Ali Kemal Sinop。欧几里得 k-均值问题的近似难度。载于第 *31* 届国际计算几何研讨会（*SoCG'15*）。Schloss Dagstuhl-Leibniz 信息学中心，2015年。[AGGN10] Barbara Anthony、Vineet Goyal、Anupam Gupta 和 Viswanath Nagarajan。为不确定情况提供的选址指南：极小极大选址问题的近似算法。运筹学数学，35(1)：第 79–101 页，2010年。[BBH+20] Daniel Baker、Vladimir Braverman、Lingxiao Huang、Shaofeng H-C Jiang、Robert Krauthgamer 和 Xuan Wu。树宽有界图上的聚类核心集。载于机器学习国际会议（*ICML'20*），第 569–579 页。PMLR，2020年。[BCCN21] Tanvi Bajpai、Deeparnab Chakrabarty、Chandra Chekuri 和 Maryam Negahbani。重新审视优先级 k-中心问题：公平性与离群点。载于第 *48* 届自动机、语言与编程国际研讨会（*ICALP'21*），第 21 页。Schloss Dagstuhl-Leibniz 信息学中心，2021年。[BCFN19] Suman Kalyan Bera、Deeparnab Chakrabarty、Nicolas Flores 和 Maryam Negahbani。用于聚类的公平算法。载于神经信息处理系统进展 *32*：神经信息处理系统年度会议（*NeurIPS'19*），第 4955–4966 页，2019年。

[BCMN14] Sayan Bhattacharya、Parinya Chalermsook、Kurt Mehlhorn 和 Adrian Neumann。鲁棒 k-中位数问题的新近似性结果。载于《斯堪的纳维亚算法理论研讨会》（SWAT'14），第 50–61 页。Springer，2014 年。[BHI02] Mihai Bad˘oiu、Sariel Har-Peled 和 Piotr Indyk。通过核心集进行近似聚类。载于 John H. Reif 编，《第 34 届 ACM 计算理论研讨会年会论文集》（STOC'02），第 250–257 页。ACM，2002 年。[BJK18] Anup Bhattacharya、Ragesh Jaiswal 和 Amit Kumar。约束 k-均值问题的更快算法。《计算机系统理论》，62(1):93–115，2018 年。[BJKW19] Vladimir Braverman、Shaofeng H-C Jiang、Robert Krauthgamer 和 Xuan Wu。用于有序加权聚类的核心集。载于《国际机器学习会议》（ICML'19），第 744–753 页。PMLR，2019 年。[BJKW21] Vladimir Braverman、Shaofeng H-C Jiang、Robert Krauthgamer 和 Xuan Wu。用于排除小图及更广泛图类中聚类的核心集。载于《2021 ACM-SIAM 离散算法研讨会论文集》（SODA'21），第 2679–2696 页。SIAM，2021 年。[BKK+13] Nikhil Bansal、Rohit Khandekar、Jochen Könemann、Viswanath Nagarajan 和 Britta Peis。关于带度数界的网络设计问题的推广。《数学规划》，141(1):479–506，2013 年。[BOR21] Sandip Banerjee、Rafail Ostrovsky 和 Yuval Rabani。最小和聚类（含离群点）。载于《近似、随机化与组合优化：算法与技术》（APPROX/RANDOM'21），LIPIcs 第 207 卷，第 16:1–16:16 页。施洛斯·达格施图尔—莱布尼茨信息学中心，2021 年。[BSS18] Jarosław Byrka、Krzysztof Sornat 和 Joachim Spoerhase。有序 k 中位数问题的常数因子近似。载于《第 50 届 ACM SIGACT 计算理论研讨会年会论文集》（STOC'18），第 620–631 页，2018 年。[CAFKL20] Vincent Cohen-Addad、Arnold Filtser、Philip N Klein 和 Hung Le。关于轻量化稀疏器、低树宽嵌入以及在排除小图中高效遍历。载于《2020 年 IEEE 第 61 届计算机科学基础年会研讨会》（FOCS'20），第 589–600 页。IEEE，2020 年。[CAK19] Vincent Cohen-Addad 和 CS Karthik。lp 度量中聚类问题的不可近似性。载于《2019 年 IEEE 第 60 届计算机科学基础年会研讨会》（FOCS'19），第 519–539 页。IEEE，2019 年。[CALSS22]Vincent Cohen-Addad、Kasper Green Larsen、David Saulpic 和 Chris Schwiegelshohn。迈向 k-中位数和 k-均值核心集的最优下界。载于《第 54 届 ACM SIGACT 计算理论研讨会年会论文集》（STOC'22），第 1038–1051 页，2022 年。[CAPP19] Vincent Cohen-Addad、Marcin Pilipczuk 和 Michal Pilipczuk。平面图上均匀成本聚类问题的高效近似方案。载于《第 27 届欧洲算法研讨会年会》（ESA'19），第 144 卷，第 33–1 页。施洛斯·达格施图尔—莱布尼茨信息学中心，2019 年。

[CASS21] Vincent Cohen-Addad、David Saulpic 和 Chris Schwiegelshohn。聚类的一种新核心集框架。载于第53届 ACM SIGACT 年度计算理论研讨会（STOC'21）论文集，第169–182页，2021年。[CFK+15] Marek Cygan、Fedor V. Fomin、Lukasz Kowalik、Daniel Lokshtanov、Dániel Marx、Marcin Pilipczuk、Michal Pilipczuk 和 Saket Saurabh。《参数化算法》。Springer，2015年。[CFLM19] Xingyu Chen、Brandon Fain、Liang Lyu 和 Kamesh Munagala。比例公平聚类。载于 Kamalika Chaudhuri 和 Ruslan Salakhutdinov 编，第36届国际机器学习会议（ICML'19）论文集，第97卷，第1032–1041页。PMLR，2019年。[CGK+19] Vincent Cohen-Addad、Anupam Gupta、Amit Kumar、Euiwoong Lee 和 Jason Li。 k-中位数和 k-均值的紧确固定参数可处理近似。载于第46届国际自动机、语言与编程研讨会（ICALP'19）论文集，LIPIcs第132卷，第42:1–42:14页，2019年。[CKLV17] Flavio Chierichetti、Ravi Kumar、Silvio Lattanzi 和 Sergei Vassilvitskii。通过公平子集实现公平聚类。载于《神经信息处理系统进展第30卷：神经信息处理系统年度会议》（NIPS'17），第5029–5037页，2017年。[Cla06] Kenneth L. Clarkson。最近邻搜索与度量空间维数。《用于学习与视觉的最近邻方法：理论与实践》，第15–59页，2006年。[CMV22] Eden Chlamtáč、Yury Makarychev 和 Ali Vakilian。基于级联范数目标的公平聚类近似。载于2022年 ACM-SIAM 离散算法研讨会（SODA'22）论文集，第2664–2683页。SIAM，2022年。[Coh20] Vincent Cohen-Addad。倍增度量下容量约束聚类的近似方案。载于 Shuchi Chawla 编，2020年 ACM-SIAM 离散算法研讨会（SODA'20）论文集，第2241–2259页。SIAM，2020年。[CS19] Deeparnab Chakrabarty 和 Chaitanya Swamy。最小范数与有序优化问题的近似算法。载于第51届 ACM SIGACT 年度计算理论研讨会（STOC'19）论文集，第126–137页，2019年。[Das08] Sanjoy Dasgupta。 k-均值聚类的困难性。技术报告 CS2008-0916，加利福尼亚大学圣迭戈分校，加利福尼亚州圣迭戈，2008年。[DL16] H. Gökalp Demirci 和 Shi Li。容量超限（$1+\epsilon$）时容量约束 k-中位数问题的常数近似。载于第43届国际自动机、语言与编程研讨会（ICALP'16），LIPIcs第55卷，第73:1–73:14页，2016年。[DX20] Hu Ding 和 Jinhui Xu。无局部性约束数据聚类的统一框架。Algorithmica，82(4)：808–852，2020年。[DZ22] Shichuan Deng 和 Qianfan Zhang。带离群点的有序 k-中位数。载于 Amit Chakrabarti 和 Chaitanya Swamy 编，《近似、随机化与组合优化：算法与技术》（APPROX/RANDOM'22），第245卷

LIPIcs，第34:1–34:22页。Schloss Dagstuhl - Leibniz 信息学中心，2022年。[FEKS19] Eli Fox-Epstein、Philip N Klein 和 Aaron Schild。将平面图嵌入低树宽图，并将其应用于度量问题的高效近似方案。载于第三十届 ACM-SIAM 离散算法研讨会（SODA'19）论文集，第1069–1088页。SIAM，2019年。[FL11] Dan Feldman 和 Michael Langberg。数据近似与聚类的统一框架。载于第四十三届 ACM 计算理论研讨会（STOC'11）论文集，第569–578页，2011年。[FL22] Arnold Filtser 和 Hung Le。平面度量和无禁用小子图度量的低树宽嵌入。载于2022 年 IEEE 第 63 届年度计算机科学基础研讨会（FOCS'22），第1081–1092页。IEEE，2022年。[FMS07] Dan Feldman、Morteza Monemizadeh 和 Christian Sohler。基于弱核的 k-均值聚类多项式时间近似方案。载于第二十三届计算几何研讨会（SoCG'07）论文集，第11–18页，2007年。[FRS19] Zachary Friggstad、Mohsen Rezapour 和 Mohammad R Salavatipour。局部搜索为倍增度量下的 k-均值问题提供多项式时间近似方案。SIAM 计算杂志，48(2)：452–480，2019年。[GJ23] Dishant Goyal 和 Ragesh Jaiswal。社会公平聚类的紧确固定参数可处理近似。信息处理快报，第106383页，2023年。[GLS12] Martin Grötschel、László Lovász 和 Alexander Schrijver。《几何算法与组合优化》，第2卷。施普林格科学与商业媒体，2012年。[GSV22] Mehrdad Ghadiri、Mohit Singh 和 Santosh S Vempala。社会公平 k-聚类的常数因子近似算法。arXiv 预印本 arXiv:2206.11210，2022年。[HPM04] Sariel Har-Peled 和 Soham Mazumdar。关于 k-均值和 k-中位数聚类的核心集。载于第三十六届 ACM 计算理论研讨会（STOC'04）论文集，第291–300页，2004年。[HV20] Lingxiao Huang 和 Nisheeth K Vishnoi。欧几里得空间聚类的核心集：重要性抽样近乎最优。载于第52届 ACM SIGACT 年度计算理论研讨会（STOC'20）论文集，第1416–1429页，2020年。[KLP19] Ioannis Katsikarelis、Michael Lampis 和 Vangelis Th Paschos。针对（k, r)-中心的结构参数、紧确界与近似。离散应用数学，264：90–117，2019年。[KLS18] Ravishankar Krishnaswamy、Shi Li 和 Sai Sandeep。通过迭代舍入实现带离群点的 k-中位数和 k-均值问题的常数近似。载于 Ilias Diakonikolas、David Kempe 和 Monika Henzinger 编，第50届 ACM SIGACT 年度计算理论研讨会（STOC'18）论文集，第646–659页。ACM，2018年。[KSS10] Amit Kumar、Yogish Sabharwal 和 Sandeep Sen。适用于任意维度聚类问题的线性时间近似方案。J.ACM，57(2)：1–32，2010年。

[KY09] Piyush Kumar 和 E. Alper Yildirim。加权欧几里得单中心问题的一个算法和核心集结果。《INFORMS Journal on Computing》，21(4):614–629，2009年。[LGW03] Inge Li Gørtz 和 Anthony Wirth。 k-中心变体中的非对称性。载于《近似、随机化与组合优化：算法与技术》（APPROX'03），第59–70页。Springer，2003年。[LNdG15] G. Laporte、S. Nickel 和 F. S. da Gama。《位置科学》。Springer，2015年。[LSV18] Yin Tat Lee、Aaron Sidford 和 Santosh S Vempala。利用成员资格预言机进行高效凸优化。载于《学习理论会议》（COLT'18），第1292–1294页。PMLR，2018年。[LV06] László Lovász 和 Santosh Vempala。对数凹函数的快速算法：采样、舍入、积分与优化。载于《2006年第47届IEEE计算机科学基础年会》（FOCS'06），第57–68页。IEEE，2006年。[LYZ10] Jian Li、Ke Yi 和 Qin Zhang。具有多样性的聚类。载于《自动机、语言与编程：第37届国际学术会议》（ICALP'19），第6198卷，第188–200页。Springer，2010年。[Mat00] Jirı Matoušek。关于近似几何 k-聚类。《离散与计算几何》，24(1):61–84，2000年。[MV21] Yury Makarychev 和 Ali Vakilian。社会公平聚类的近似算法。载于《学习理论会议》（COLT'21），第3246–3264页。PMLR，2021年。[NP05] S. Nickel 和 J. Puerto。《位置理论》。施普林格科学与商业媒体，2005年。[ORSS13] Rafail Ostrovsky、Yuval Rabani、Leonard J Schulman 和 Chaitanya Swamy。Lloyd型方法对于 k-均值问题的有效性。《J.ACM》，59(6):1–22，2013年。[Ple87] Ján Plesník。图中 p-中心问题的一种启发式方法。《离散应用数学》，17(3):263–268，1987年。[Tam01] Arie Tamir。 k-中心多设施选址问题。《离散应用数学》，109(3):293–307，2001年。[TGOO22] Suhas Thejaswi、Ameet Gadekar、Bruno Ordozgoiti 和 Michal Osadnik。具有公平中心表示的聚类：参数化近似算法与启发式方法。载于《第28届ACM SIGKDD知识发现与数据挖掘会议论文集》（KDD '22），第1749–1759页，美国纽约州纽约，2022年。计算机协会。[TOG21] Suhas Thejaswi、Bruno Ordozgoiti 和 Aristides Gionis。多样性感知的 k-中值：具有公平中心表示的聚类。载于《机器学习与数据库中的知识发现：研究专题——欧洲会议》（ECML PKDD'21），西班牙毕尔巴鄂，2021年9月13–17日，论文集，第二部分，第12976卷，第765–780页。Springer，2021年。