计算机科学
COMPUTER SCIENCE

k-均值聚类在高维大数据上的高效算法研究进展

高贵晨, 姜少峰

引用本文

高贵晨, 姜少峰. *k*-均值聚类在高维大数据上的高效算法研究进展[J]. 计算机科学, 2026, 53(4): 24-32.

GAO Guichen, JIANG Shaofeng. Recent Advances in Efficient Algorithms for*k*-Means Clustering on High-dimensional Big Data [J]. Computer Science, 2026, 53(4): 24-32.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.251000037

k-均值聚类在高维大数据上的高效算法研究进展

高贵晨　姜少峰
北京大学计算机学院　北京 100871
(gc.gao@stu.pku.edu.cn)

摘　要　聚类是机器学习中的经典任务,旨在根据相似度度量将数据划分为若干簇。k-均值聚类作为最基本的聚类模型,自提出以来已被深入研究并在众多领域得到广泛应用。聚焦k-均值模型的求解问题,从理论计算机科学的视角出发,介绍k-均值的(接近)线性时间的快速近似算法的研究进展。此外,简要讨论其他相关大数据计算模型中的聚类算法的相关进展,包括动态、数据流与并行计算等计算模型。

关键词:k-均值;欧氏空间;近线性时间算法;亚线性算法
中图分类号　TP18

Recent Advances in Efficient Algorithms for k-Means Clustering on High-dimensional Big Data

GAO Guichen and JIANG Shaofeng
School of Computer Science,Peking University,Beijing 100871,China

Abstract　Clustering is a classic task in machine learning. The goal of clustering is to partition data points into groups,with respect to a similarity measure. As one of the most fundamental models for clustering,k-means has been extensively studied and widely applied. This paper focuses on the computational issue of solving k-means efficiently,and discusses the progress of(near-)linear time approximation algorithms for k-means,from the perspective of theoretical computer science. It also briefly discusses the status of clustering algorithms in various big data computational models,including dynamic,streaming and distributed computing.

Keywords　k-means,Euclidean spaces,Near-linear time algorithms,Sublinear algorithms

1　引言

聚类是无监督学习的核心任务,已被广泛应用于机器学习、数据科学、计算机视觉、自然语言处理等领域。一般的聚类问题旨在将输入数据点按照相似性进行划分,使得类内相似度高、类间相似度低。在各种聚类的具体模型中,k-均值聚类是最基本的聚类模型,自 Lloyd 于 1957 年提出以来便得到了深入研究和广泛应用。k-均值模型的求解也有较为成熟的方法,在包括 scikit-learn 和 Apache Spark MLlib 在内的多种机器学习工具中都有 k-均值求解器,其成为了求解 k-均值相对成熟的手段。

然而,大数据时代对 k-均值的现代应用提出了新的计算挑战。随着表示学习的发展,文本、语音、图像等多模态数据普遍被嵌入到高维向量空间,并在欧氏空间中以距离作为相似度度量。这类高维嵌入向量已成为聚类等数据分析任务的典型输入,其规模往往达到百万量级,维度也超过百维[1-3]。在输入规模日益增长的背景下,相对于输入规模的线性或近线性时间已经成为高效聚类算法的基本要求。

针对 k-均值聚类的计算挑战,本文从理论计算机科学的视角,系统梳理了近线性时间的、具有理论精度保证的 k-均值求解算法。此外,本文还将涉及其他大数据计算模型,包括数据流模型、动态计算模型以及并行计算模型等,并讨论上述计算模型下 k-均值聚类的高效算法。

本文第 2 章介绍欧氏空间 k-均值问题的形式化定义等预备知识;第 3 章综述近线性时间 k-均值近似算法;第 4 章进一步从计算模型的角度出发,讨论动态、数据流、并行计算等大数据计算模型下 k-均值的高效算法;最后总结全文并展望未来研究方向。

2　预备知识

首先,给出欧氏空间中的 k-均值模型的数学优化的表达形式。在该问题中,输入是 n 个 d 维数据点的集合 $P\subset\mathbb{R}^d$ 以及参数 $1\leqslant k\leqslant n$,目标是找到一个大小至多为 k 的中心点集合 $C\subset\mathbb{R}^d$,使得如下目标函数最小化:

$$cost(P,C):=\sum_{p\in P}\min_{c\in C}\|p-c\|_2^2$$

即所有 P 中的数据点到集合 C 中最近点的平方距离之和最

到稿日期:2025-10-13　返修日期:2026-01-20
基金项目:国家自然科学基金(62572006)
This work was supported by the National Natural Science Foundation of China(62572006).
通信作者:姜少峰(shaofeng.jiang@pku.edu.cn)

小。k-均值问题即使在常数维 $d=O(1)$ 的欧氏空间也是 NP-hard 的[4]，因此对于 k-均值问题，多考虑近似算法。即，若对于某个 $\alpha\geqslant 1$ 和（多项式时间）算法 $\mathcal{A}$，$\mathcal{A}$ 对任意输入 P 的输出解 C 都满足：

$$cost(P,C)\leqslant \alpha\cdot OPT$$

则称 $\mathcal{A}$ 为 α-近似，并称最小的 α 为算法的近似比，其中 OPT 为 P 上的最优目标值。若算法 $\mathcal{A}$ 为随机算法，则上述定义改为取数学期望：

$$\mathbb{E}[cost(P,C)]\leqslant \alpha\cdot OPT$$

记 $\widetilde{O}(m):=O(m\,poly\log m)$。

3 欧氏 k-均值的近线性时间近似算法

因为 k-均值的输入是 n 个数据点，所以线性时间自然是针对 n 的，即追求 $\widetilde{O}(n)$ 的运行时间。然而，除 n 之外，还有两个重要参数 k 和 d，这些参数的取值范围会对聚类近似比与运行时间之间的权衡产生显著影响。因此，将按照对 k 和 d 的依赖形式对 k-均值的 $\widetilde{O}(n)$ 时间近似算法分别进行讨论。具体考虑 3 类：适用于固定参数 k 和 d 的近似算法，时间复杂度依赖 $poly(kd)$ 的近似算法，以及适用于一般参数 k 和 d 的常数近似算法。

3.1 固定参数

当聚类数 k 或维度 d 固定时，近线性时间 $\widetilde{O}(n)$ 内可以达到 k-均值的 $(1+\varepsilon)$-近似[5-16]。具体而言：若固定聚类数 k，当前已知的最优结果可在 $O(nd\cdot 2^{(k/\varepsilon)^{O(1)}})$ 的时间复杂度下实现 $(1+\varepsilon)$-近似保证[11]；若固定维度 d，则时间复杂度可达到 $O(n\cdot 2^{(1/\varepsilon)^{O(d^2)}})$ 并保持相同近似比[16]。

尽管上述算法的时间复杂性是近线性 $\widetilde{O}(n)$ 的且达到了 $(1+\varepsilon)$-近似，但其代价却指数依赖于 k 或双指数依赖于 d。事实上，这一指数或双指数依赖是实现 $(1+\varepsilon)$-近似所必须的，因为当参数 k 和 d 是输入的一部分时，已有研究表明，k-均值聚类是 APX-hard 的[17]（大体上，如果一个问题是 APX-hard 的，则在 P≠NP 的假设下，存在一个常数 $c>1$，使得任何多项式时间算法不能比 c-近似更好），即在一般参数 k 和 d 下只有常数近似。

这种对于 d 或者 k 的指数依赖，在很多实际设定中都是不可接受的。因此，接下来讨论更普遍的设定：当 k 和 d 不再是固定参数时，如何实现 $poly(kd)$ 依赖，甚至是更优运行时间的近似算法。

3.2 非固定参数：$poly(kd)$ 依赖

从算法复杂性的角度出发，更合适的设定是允许算法对参数 k 和 d 具有多项式级别的依赖。目前，运行时间为 $\widetilde{O}(n\cdot poly(kd))$ 的欧氏 k-均值近似算法已被大量工作系统研究。具体而言，主流求解器通常基于经典的 Lloyd 迭代算法[18]，配合 k-means++[19] 的初始解选取策略，可在 $O(nkd)$ 时间内完成计算，并在期望意义下达到 $O(\log k)$ 的近似保证。此外，基于局部搜索、随机采样等技术的算法可以实现常数量级的近似保证，同时其时间复杂度仍保持在 $\widetilde{O}(n\cdot poly(kd))$ 量级[20-28]。

更一般地，区别于上述直接构造的算法，本文介绍一个统一的算法框架，可以将任何一个 $poly(n)$ 时间的近似算法在基本保持近似比的情况下，转化成 $\widetilde{O}(n\cdot poly(kd))$ 时间的高效算法[1)]。该框架以黑盒的方式综合利用了已有的数据降维和数据压缩的技术，由两部分构成：1）通过数据降维技术，将维度 d 降低为 $\log(n)$；2）通过核心集构造对输入数据进行摘要，将 n 降低为 $\widetilde{O}(k)$。基于上述框架，任意运行时间为 $poly(n)$ 的 α-近似 k-均值算法，均可被系统性地转换为整体运行时间为 $\widetilde{O}(n\cdot poly(kd))$ 的 $(1+O(\varepsilon))\alpha$-近似算法，其中 $(1+O(\varepsilon))$ 表示框架引入的可控近似损失。

1）数据降维

在欧氏空间中，维度 d 是一个决定算法复杂性的重要参数。对于 k-均值问题，已有一系列工作研究如何在降维后仍近似保持聚类代价[29-32]。当前最优结果表明，将数据随机投影到 $\ell:=O(\log(k/\varepsilon)/\varepsilon^2)$ 维的空间中，即可在高概率下保证 P 的任意划分 $\boldsymbol{C}=(C_1,C_2,\cdots,C_k)$ 的聚类代价在 $(1+\varepsilon)$ 倍以内[32]。具体而言，对于任意点集 $P\subset\mathbb{R}^d$ 和 $\varepsilon\in(0,1)$，Makarychev 等[32] 可以计算一个目标维度是 ℓ 维的随机映射 $f:\mathbb{R}^d\to\mathbb{R}^\ell$，使得对于 P 中任意一个划分 $\boldsymbol{C}=(C_1,C_2,\cdots,C_k)$，都可高概率保证：

$$\begin{aligned}&\sum_{i=1}^{k}\min_{c_i\in\mathbb{R}^d}\sum_{x\in C_i}\|x-c_i\|_2^2\\&\leqslant\sum_{i=1}^{k}\min_{c_i\in\mathbb{R}^\ell}\sum_{x\in f(C_i)}\|x-c_i\|_2^2\\&\leqslant(1+\varepsilon)\cdot\sum_{i=1}^{k}\min_{c_i\in\mathbb{R}^d}\sum_{x\in C_i}\|x-c_i\|_2^2\end{aligned}\tag{1}$$

并且映射 f 对任何点 $x\in\mathbb{R}^d$ 都可在 $O(d\log d)$ 时间内计算。

Makarychev 等[32] 的降维方法在构造形式上与经典的 Johnson-Lindenstrauss（JL）降维[33] 一致，都基于相同的随机线性映射方法，只是目标维度 ℓ 的设定不同。然而，这也带来了降维保证上的不同。具体而言，JL 降维[33] 将点集映射到 $O(\varepsilon^{-2}\log n)$ 维空间，并保证 $\forall\,x,y\in P$，$\|x-y\|_2\leqslant\|f(x)-f(y)\|_2\leqslant(1+\varepsilon)\|x-y\|_2$，即任何两点之间的距离保持在 $(1+\varepsilon)$ 倍以内。而 Makarychev 等[32] 则将数据映射到 $O(\log(k/\varepsilon)/\varepsilon^2)$ 维空间，当 $k\ll n$ 时，该目标维度可以显著低于 JL 的目标维度。然而，Makarychev 等[32] 的降维方法并不保持点对之间的距离，因此无法给出 JL[33] 的点对距离保证。

综上所述，对于欧氏 k-均值，即使 $k=\Theta(n)$，也可以不失一般性地假设 $d=O(\log n)$。由此，近线性时间计算的复杂度瓶颈不再来自维度 d，因为 $poly(d)$ 已退化为 $poly\log(n)$。然而，数据降维技术无法缓解算法对 n 的依赖。因此，在构建高效 k-均值近似算法时，数据降维通常与核心集结合使用，以进一步控制时间复杂度对数据规模的依赖。

2）核心集

在欧氏空间中，聚类算法的主要计算开销来自于对 n 个

1) 没有已知文献对该框架进行过细致讨论，因此此处没有给出标准的文献引用来源。然而，由于该框架较为简洁、黑盒利用了已有结果，目前已经成为了一种标准方法。

d 维数据点的处理。为避免在后续计算中反复访问输入数据点，核心集方法通过构造一个小规模的加权代表性子集，对原始数据进行摘要。对于 k-均值问题，Har-Peled 等[8]首次给出了核心集的定义。具体而言，给定 d 维点集 $P\subset\mathbb{R}^d$ 与精度参数 $\varepsilon\in(0,1)$，若存在加权点集 $S\subset\mathbb{R}^d$ 及权重函数 $w:S\to\mathbb{R}_{\geqslant 0}$，使得对任意 k 个中心构成的集合 $C\subset\mathbb{R}^d$，均满足：

$$\sum_{x\in S}w(x)\cdot\min_{c\in C}\|x-c\|_2^2\in(1\pm\varepsilon)\sum_{p\in P}\min_{c\in C}\|p-c\|_2^2 \quad (2)$$

则称 S 为 P 的一个 ε-核心集。其中，基数 $|S|$ 为核心集的规模。

从核心集的定义可直接推出如下结论：设构造一个 ε-核心集 S 的时间为 T，则对于任意在原始数据集上运行时间为 $poly(n)$、近似比为 α 的 k-均值算法，若将其应用于核心集 S，即可在原始点集上获得一个近似比为 $(1+\varepsilon)\alpha$ 的解，且整体运行时间为 $T+poly(|S|)$。正因如此，为 k-均值构造小规模核心集已成为重要研究方向，且在各类度量空间中得到了广泛探索[8,10,29,34-49]。在欧氏空间中，目前已知可以构造规模为 $\widetilde{O}(k)$ 的 ε-核心集，且其构造时间为 $\widetilde{O}(nd)$[36]。

下面将数据降维与核心集构造作为黑盒模块加以组合，并具体描述一个统一的 k-均值算法框架，如算法 1 所示。算法 1 中，第 1－4 行直接对应了降维和数据摘要的步骤，但仍需要在第 5－6 行中，把将降维和摘要后的数据上的解 C' 转换成原始空间上针对 P 的中心集合 $\hat{C}$。

算法 1 基于数据降维和数据压缩的通用 k-均值算法框架

输入：数据集 $P\subset\mathbb{R}^d$，参数 $1\leqslant k\leqslant n$，$\varepsilon\in(0,1)$，α-近似算法 $\mathcal{A}$

输出：数据集 P 上的一个 $(1+O(\varepsilon))\alpha$-近似 k-均值中心集合 $\hat{C}$

1. 令 $\ell\leftarrow O(\log(k/\varepsilon)/\varepsilon^2)$，构造一个 JL 降维映射 $f:\mathbb{R}^d\to\mathbb{R}^\ell$[32]
2. 令 $P'\leftarrow\{f(x)\mid x\in P\}\subset\mathbb{R}^\ell$ 为降维后的数据集

   //该映射可在 $\widetilde{O}(nd)$ 时间内完成；P 和 P′中的点是一一对应的，并满足式(1)

3. 对 P′构造一个规模为 $\widetilde{O}(k)$ 的 ε-核心集 $S\subset\mathbb{R}^\ell$[36]

   //该核心集可在 $\widetilde{O}(nd)$ 时间内构造，并满足式(2)

4. 在 S 上运行 $\mathcal{A}$，得到解 $C'\subset\mathbb{R}^\ell$
5. 求 P′关于 C′的聚类簇，并令 $\{C_1,\cdots,C_k\}$ 为 P 上对应的 k-划分

   //此处 $\{C_1,\cdots,C_k\}$ 的定义利用了 P 与 P′之间的一一映射

6. 令 $\hat{C}_i\leftarrow\frac{1}{|C_i|}\sum_{x\in C_i}x$ 为簇 C_i $(1\leqslant i\leqslant k)$ 上的聚类中心，返回 $\hat{C}$

算法 1 的总体时间复杂度为 $\widetilde{O}(nd+poly(k)+nk)$，其主要由数据降维、核心集构造、核心集上 k-均值求解及原始空间 P 的 k-均值中心集合计算决定。其中，数据降维和核心集构造的时间开销为 $\widetilde{O}(nd)$，给定算法 $\mathcal{A}$ 的运行时间为 $poly(n)$，则核心集上 k-均值求解的时间开销为 $poly(k)$、计算 P 的 k-均值中心集合的时间开销为 $\widetilde{O}(nk)$，其被计算 P' 到 C' 的最近邻时间开销 $\widetilde{O}(nk)$ 所支配。算法 1 的精度损失主要来自数据降维和数据压缩。在这两个过程中，精度损失均为 $(1+\varepsilon)$。因此，最终得到的中心集合 $\hat{C}$ 是数据集 P 的 $(1+O(\varepsilon))\alpha$-近似解。

综上所述，数据降维与核心集技术提供了一种统一的预处理框架，使得传统在多项式时间内追求更优近似比的 k-均值算法[50-56]（当前最优近似比可达 5.83[56]）都可以在保持近似性能基本不变的前提下，被系统性地转换为近线性时间 $\widetilde{O}(n\cdot poly(kd))$ 算法。

3.3 一般参数设定

当聚类数 k 和维度 d 为自由参数（例如，允许 $k=O(n)$，$d=O(\log n)$）时，前述时间复杂度为 $\widetilde{O}(n\cdot poly(kd))$ 的算法将不再属于近线性时间算法。优化关于参数 k 的依赖成为了显著挑战。一方面，先前介绍的核心集方法不再能用于简化问题，原因在于任何 ε-核心集必须存储 $\Omega(k)$ 个点[45]，当 $k=\Omega(n)$ 时，前述算法的时间复杂度为 $\Omega(n^2)$，所以无法简化问题。除了不能使用核心集外，还有另一项复杂性结果指出：在一般度量空间中，任何具有有限近似比保证的算法都需要 $\Omega(nk)$ 的运行时间[57]，这表明近线性时间的 k-均值近似算法必须要从根本上利用欧氏空间的性质。尽管数据降维已将 $poly(d)$ 优化为 $poly\log(n)$，使得复杂度瓶颈不再来自维度 d，但聚类数 k 对算法复杂度的影响依然存在。

1）高维空间近似算法的基本权衡。另一方面，若对维度 d 不进行任何限制（即需要考虑高维空间），也会导致一些本质的计算困难。具体来说，在高维空间中存在一项普遍的根本障碍：包括聚类问题在内的许多问题很难达到 $\widetilde{O}(nd)$ 运行时间的常数近似，而是只能得到形如 $n^{1+\varepsilon}$ 的接近线性的多项式时间的常数近似。这项根本障碍来源于近似近邻搜索问题，并且具体的权衡来自于局部敏感哈希（Locality Sensitive Hashing，LSH）[58-59]。具体来说，对于最近点对问题，即找到 n 个 d 维数据点最近点对的距离，目前猜想如下的 LSH 权衡基本上是最优的，尤其是不可改进成 $\widetilde{O}(nd)$ 时间的常数近似：对 $t\geqslant 1$，可以在 $O(n^{1+1/t^2})$ 时间做到 $O(t)$ 的近似比[60-62]。最近点对问题足够基本，很多问题都可以规约到该问题，例如最小生成树[59]、设施选址、全局最小权匹配等[63]。k-均值聚类问题也可以直接规约到最近点对问题：令 $k=n-1$，则聚类的最优解直接对应于最近点对的距离。因此，在一般的参数设定下，达到时间-近似比的 LSH 权衡就是目前可以做到的最优的近似算法结果。

2）达到 LSH 权衡的近似算法。近期，Jiang 等[64]针对欧氏 k-均值给出了能达到 LSH 权衡的算法，主要结果如下。本节主要聚焦这项工作，重点介绍该工作的主要算法设计思路。

定理 1[64] 对任何 $t\geqslant 1$，存在针对欧氏 k-均值的 $O(t^2)$-近似的随机算法，运行时间为 $\widetilde{O}(poly(d)n^{1+1/t^2})$。

达到 LSH 权衡的关键是有效利用欧氏空间的性质。这里，首先介绍一种将（高维）欧氏空间转换成稀疏图的一般技术。这项技术广泛适用于各种基于距离的组合优化问题，并且定理 1 就是利用该技术最终达到 LSH 权衡。

定义 1 对于一个欧氏点集 $P\subset\mathbb{R}^d$，定义 P 的欧氏图为欧氏距离导出的完全图，即点集是 P，边集是 $P\times P$，边权是 $w(x,y):=\|x-y\|_2$ 的图。给定欧氏点集 $P\subset\mathbb{R}^d$ 和误差参数 $t\geqslant 1$，t-spanner 是一个欧氏子图 $H(P,E,w)$，使得：

$$\forall x,y\in P,\ \|x-y\|_2\leqslant dist_H(x,y)\leqslant t\|x-y\|_2$$

其中，$dist_H$ 代表图 H 上的最短路距离。

对于确定的 t，希望得到一个尽量稀疏（即边数尽量少）的

spanner H。对于欧氏空间，Har-Peled 等[65]给出了一个基于 LSH 的 t-spanner 构造方法；结合高效的 LSH[62]，该 spanner 边数为 $m=\widetilde{O}(n^{1+1/t^2})$，且可在 $\widetilde{O}(dn^{1+1/t^2})$ 时间内构造，从而达到了 LSH 权衡。因此，在预处理阶段运行这种达到 LSH 权衡的 spanner，就可以将高维欧氏的问题转换成稀疏图上的问题。此时，为了总体上得到 LSH 权衡的近似算法，只需要在（稀疏）图上设计一个 $O(1)$-近似的运行时间为接近边数 m 的算法。

具体到聚类问题，Thorup[66]给出了图上（针对最短路距离的）聚类问题的 $\widetilde{O}(m)$ 时间的 $O(1)$ 近似。因此，Thorup 等[66]考虑的问题自动可以得到欧氏空间上达到 LSH 权衡的近似算法。然而，由于 k-均值目标函数与平方欧氏距离的特殊结构，目前仍不知道 Thorup 的结果是否可以推广到 k-均值。近期，Jiang 等[64]采取与 Thorup 完全不同的技术路线，给出了下列新的图上 k-均值的快速常数近似算法，该结果与 spanner 结合，可以直接推出定理 1。

定理 2[64] 存在一个针对图最短路度量 k-均值的 $O(1)$-近似的随机算法，运行时间为 $m^{1+o(1)}$，其中 m 为边数。

这个算法是经典的 1-交换局部搜索算法的一个高效改进，第一次把局部搜索相关技术在近线性时间进行了实现。在经典的 1-交换局部搜索[67]中，算法从一个初始中心集合 $C\subset\mathbb{R}^d$ 开始迭代。对于当前解 C，穷举数据集中的一个点 x_{in} 以及当前解 C 中的一个点 x_{out}，如果：

$$cost(P,C\backslash\{x_{\text{out}}\}\cup\{x_{\text{in}}\})\leqslant\gamma\cdot cost(P,C)$$

则进行此次交换，即设置 $C:=C\backslash\{x_{\text{out}}\}\cup\{x_{\text{in}}\}$，$\gamma<1$ 是一个阈值。这里 γ 的作用很关键，它直接控制何时进行交换。经典结果[67]证明了，γ 需要取值为一个非常接近 1 的值 $1-1/poly(n)$，从而让局部搜索在 $poly(n)$ 轮迭代后收敛，并且收敛到一个常数近似比。虽然直觉上，γ 取值更小一些，如取成常数，可以得到更少的迭代轮数，但这样很难保证近似比，即很快会收敛到一个近似比很差的局部最优解上。

定理 2 的算法大体上也是 1-交换局部搜索，但关键创新点是采用了可变的阈值 γ，并且这个阈值会根据不同的 x_{in} 和 x_{out} 来设置，即 $\gamma=\gamma(x_{\text{in}},x_{\text{out}})$。由于整个局部搜索过程最耗时的操作就是寻找 x_{in} 和 x_{out}，以及维护交换 x_{in} 和 x_{out} 后的聚类结果，因此 γ 的设置要反映出 $(x_{\text{in}},x_{\text{out}})$ 交换操作所带来的时间上的开销：时间开销越大，则要求 γ 越小，即更显著地降低代价函数 $cost$；反之，则可以使用一个较大的 γ。直觉上，可以认为 $\gamma(x_{\text{in}},x_{\text{out}})$ 衡量/评价了 $(x_{\text{in}},x_{\text{out}})$ 这组交换的质量，这与传统的 $\gamma=1-1/poly(n)$ 具有本质不同。Jiang 等[64]证明了这样的新的局部搜索规则可以让整个搜索过程的总时间开销维持在接近线性的水平，虽然具体到某一步的开销仍可以很大。

下面对定理 1 中的算法进行总结，将前述的 t-spanner 构造与局部搜索过程加以整合，其整体流程如算法 2 所示。算法 2 的总体时间复杂度为 $\widetilde{O}(poly(d)n^{1+1/t^2})$，主要由 t-spanner 构造以及在稀疏图上的聚类求解两部分所决定。具体而言，t-spanner 子图 H 可通过 Har-Peled 等[65]提出的基于 LSH 的方法构造，并结合高效的 LSH 实现[62]，在 $\widetilde{O}(dn^{1+1/t^2})$ 时间内完成，其边数为 $m=\widetilde{O}(n^{1+1/t^2})$。在此基础上，图 H 上的聚类求解可在 $m^{1+o(1)}$ 时间内完成[64]。在近似保证方面，算法 2 的精度损失由 t-spanner 的参数 t 所支配：针对 k-均值的平方距离和目标函数，该算法最终得到 P 的一个 $O(t^2)$-近似中心集合 C。

算法 2 达到 LSH 权衡的 k-均值算法框架

输入：数据集 $P\subset\mathbb{R}^d$，参数 $1\leqslant k\leqslant n$，$t\geqslant 1$

输出：数据集 P 上的一个 $O(t^2)$-近似 k-均值中心集合 C

1. 对 P 构造一个 t-spanner 子图 H，其边数 $m\leftarrow\widetilde{O}(n^{1+1/t^2})$[65]
2. 在 H 上运行下面的局部搜索[64]：选取初始中心集合 $C\subseteq P$，循环检测是否存在 $x_{\text{in}}\in P$ 及 $x_{\text{out}}\in C$ 满足下列条件：
   $cost_H(P,C\backslash\{x_{\text{out}}\}\cup\{x_{\text{in}}\})\leqslant\gamma(x_{\text{in}},x_{\text{out}})\cdot cost_H(P,C)$
   若存在，则令 $C\leftarrow C\backslash\{x_{\text{out}}\}\cup\{x_{\text{in}}\}$，并继续循环；否则结束
   //该过程可在 $m^{1+o(1)}$ 时间内完成；$cost_H$ 表示图上最短路距离的 k-均值代价函数；$\gamma(x_{\text{in}},x_{\text{out}})$ 是一个可变阈值，反映了 $(x_{\text{in}},x_{\text{out}})$ 交换操作所带来的时间上的开销，这与传统的阈值设定具有本质不同
3. 返回 C

La Tour 等[68]针对欧氏 k-均值给出了 $\widetilde{O}(n^{1+1/t^2})$ 时间、$O(t^{12})$ 近似的权衡。虽然该结果并未达到 LSH 权衡，但技术上不依赖高维 spanner、不需要规约到图上进行算法设计，而是直接利用 LSH 来加速某个已有的贪心算法[69]，提供了不同的技术路线。此外，还有一类工作专门研究严格的近线性 $\widetilde{O}(nd)$ 时间的算法，虽然得到的近似比仅为 $poly\log(k)$ 或 $poly(k)$[70-71]，并且无法达到 LSH 权衡，但他们提出的算法框架较为简洁，易于实现，可能会取得较好的实际效果。

4 欧氏 k-均值在其他相关计算模型下的高效算法

除了追求接近线性时间的近似算法，其他大数据计算模型，包括数据流模型、动态计算模型和并行计算模型等，也是理论计算机科学中应对大数据计算挑战的重要研究方向。这些模型因其更新时间、空间远低于输入规模，在理论计算机科学中通常被统称为“亚线性模型”。承接第 3 章中对 k-均值近线性时间近似算法的系统讨论，本章将进一步综述在亚线性计算模型下 k-均值聚类的主要算法结果。

1）动态算法

动态算法是传统近似算法设定的推广。动态算法的输入不再是一个固定的数据集，而是以 $\mathbb{R}^d$ 中数据点的插入/删除来给出，要求在每次插入/删除的更新后，算法都能维护一个 k-均值的近似解 C。此处，“维护”的含义是算法显式维护一个包含 k 个点的集合，每次更新后都需要对应地更新这个集合，以此来保持近似比。注意到动态算法同时也可以作为传统的近似算法来使用：将 n 个数据点依次插入后返回维护的解。所以对于 k-均值，一个理想的目标就是对标近似算法的 LSH 权衡，达到每次均摊更新时间 $\widetilde{O}(n^{1+1/t^2}/n)=\widetilde{O}(n^{1/t^2})$，以及 $O(t^2)$ 的近似比。

近期，Bhattacharya 等[72]针对欧氏空间中的动态 k-均值问题，给出了 $poly(t)$-近似、更新时间为 $\widetilde{O}(k^{1/t^2})$ 的新算法，使得动态 k-均值朝着“LSH 权衡”的目标迈进了一大步。尽管该算法尚未达到真正的 LSH 权衡，但相比一般度量空间中

$\widetilde{O}(n)$的更新时间，其效率提升已是显著突破。达到上述接近LSH权衡的结果需要充分利用欧氏空间的几何结构。相比之下，在一般度量空间中，动态k-均值已得到较为系统的研究[73-78]。其中，Bhattacharya等[78]实现了$\widetilde{O}(n)$更新时间的常数近似算法，并且该结果在已知下界意义下是最优的[57]。

以上结果虽然都适用于一般参数，但无可避免地无法达到$(1+\varepsilon)$-近似。是否能在次线性乃至$poly\log(n)$时间内做到$(1+\varepsilon)$-近似，仍是一个亟需解决的公开问题。

2)数据流算法

数据流设定的输入类似于动态设定，以插入/删除$\mathbb{R}^d$上的更新序列给出[1)]。然而，与动态设定不同的是，这里只需要在数据流结尾时给出答案，但同时额外要求空间复杂度要尽量低，任何时候都必须是$o(n)$的。对于聚类问题，由于输出已经需要$\Omega(k)$的空间，因此不可能寄希望于算法的空间复杂性是$o(k)$的。事实上，这个$\Omega(k)$的空间需求是很本质的：可以证明，即使是在1维，只是分辨数据集上的k-均值的最优值是否为0(而无需输出对应的k点的解)，已经需要$\Omega(k)$的空间。另一方面，目前已有算法可以在$\widetilde{O}(k \cdot poly(d\log n))$的空间内实现$(1+\varepsilon)$-近似[79-80]，然而这个空间复杂性中关于参数$k,d,\log n$的依赖是否还能改进是开放性问题。

不同于动态数据流设定，聚类在仅插入数据流设定下的空间复杂度得到了更细致的研究[8,10,35,37,45,81-86]。这里，在仅插入数据流中，只有对数据点的插入操作而没有删除操作。在这种设定下，可以采用一种被称为merge-and-reduce的通用算法框架[8]，达到与现有核心集大小相近的空间复杂度。设存在一个大小为$S(n,k,d,\varepsilon)$的k-均值的核心集，merge-and-reduce框架将数据分成连续的块后，求得相邻块的核心集后进行合并，之后使用同样的步骤进行迭代。最终，设$\varepsilon':=\Theta(\varepsilon/\log n)$，merge-and-reduce可在$S(n,k,d,\varepsilon')$空间下得到一个$\varepsilon'$-核心集。结合已有的核心集结果，就可以得到运行空间与核心集大小类似的数据流算法。然而，merge-and-reduce框架不可避免地在空间复杂度中引入额外的$poly\log(n)$因子。近期，Cohen-Addad等[86]的工作突破了merge-and-reduce框架的瓶颈，在仅插入数据流模型下实现了$\widetilde{O}(\varepsilon^{-2}kd)\cdot\min(\varepsilon^{-2},k)poly(\log\log n)$的空间上界，其空间复杂度不再依赖$poly\log(n)$因子。

滑动窗口是另一个数据流的常见设定，其目标是在任意时刻维护最近W个数据点的近似解。具体来说，滑动窗口模型的输入仍是一个仅插入数据流，但在任何时候，只有最近的W个数据点保留在数据集中，并且算法需要回答当前W个数据点的聚类近似解。特别地，落在窗口之外的较早到来的数据点会被隐式删除。这种滑动窗口设定是只插入数据流的推广，因为可以将W设置为无穷大，从而使任何元素都不会被删除。然而，滑动窗口与上述的动态数据流模型却是不可比较的，主要原因是动态数据流要求删除操作是由输入显式给出的，而滑动窗口的删除操作是隐式的。近年来，一系列工作给出了滑动窗口模型下的k-均值结果[87-90]。目前最佳的结果实现了空间复杂度为$\varepsilon^{-4}k\cdot poly\log(\varepsilon^{-1}n)$的$(1+\varepsilon)$-近似[90]。值得一提的是，这些$(1+\varepsilon)$-近似的结果均基于核心集技术[89-90]，但核心集的构造方式经过了专门设计，以适应滑动窗口模型的动态特性。

3)并行算法

并行计算有许多模型，这里主要介绍大规模并行计算(MPC)模型[91]——一种脱胎于MapReduce[92]，Hadoop[93]，Spark[94]等，被广泛采用的实用大数据计算软件框架的理论模型。就欧氏空间的设定而言，MPC模型假设每台机器的内存s满足$s\geqslant poly(d)\cdot n^{\sigma}(\sigma\in(0,1))$，机器之间通过若干通信轮来进行计算，并且在每轮中，任意两台机器都可以通信，但要求每台机器的总通信量(即信息总长度)是$O(s)$的。MPC算法设计的目标是得到较好的轮数-近似比权衡，并且通常希望轮数为常数。具体到k-均值，一个关键挑战是k可以很大(如$k=O(n)$)，且机器内存s可以远小于k，这就导致聚类的解C甚至无法存储在一台机器上。

迄今为止，没有任何已知的MPC算法可以在常数轮达到k-均值任何有限近似比，并且已知的结果要么需要$\omega(1)$轮才能达到常数近似[95]，要么只能得到双标准近似[96-97]，即允许使用的聚类中心多于k(目前最好的双标准常数近似，允许使用的聚类中心多于k，但不超过k的常数倍[97])。

此外，若每台机器拥有$\widetilde{O}(n^{\varepsilon}poly(k\varepsilon^{-1}))$的本地内存，则可以采用$n^{\varepsilon}$-分叉的merge-and-reduce框架，并在每一轮使用相应精度的核心集，从而在$O(\varepsilon^{-1})$轮通信内实现k-均值的$(1+\varepsilon)$-近似。然而，由于该方法要求在单轮内合并核心集，其本地空间需求随k和ε^{-1}呈多项式增长，因此在k或精度要求较高的场景下往往难以满足需求，从而限制了其适用范围。

结束语 本文从理论计算机科学视角介绍了k-均值在若干计算设定下的快速算法理论，涵盖了近线性时间近似算法、动态算法、数据流算法与并行计算等。在技术层面，简介了聚类的数据摘要方法核心集、一般的降维Johnson-Lindenstrauss变换、高维欧氏空间上的基本的LSH权衡以及一般的稀疏化方法spanner。这些技术不仅适用于聚类，也可用于其他欧氏空间下的组合优化问题的求解。从研究方向上看，本文所介绍的大研究方向侧重于设计真正高效的近似算法，与传统的主要追求近似比而只要求多项式时间的研究形成鲜明对比。如何将这种研究思路拓展到其他相关机器学习问题上，可以作为理论计算机科学研究的一个未来方向。

本文侧重介绍算法理论结果，但仅有一部分算法的实际效果得到了实际验证。本文以3种参数设定下的近似算法为例进行讨论。1)尽管理论上已存在$(1+\varepsilon)$-近似的k-均值算法，但其时间复杂性对参数k或d的依赖太强，因此主要停留在理论研究层面。2)对于一般维度d，高效k-均值算法可以借助JL变换等降维技术将复杂度控制在$poly(d)$量级。由此可见，当前k-均值求解的效率瓶颈更多取决于参数k。当k较小时，已有大量工作实现了“理论-实践”的双向适配，例如

1) 事实上数据流算法一般需要假定输入是在一个$\{1,\cdots,\Delta\}^d$的大离散格点上，以此来更方便地讨论空间复杂度。本文假定$\Delta=poly(n)$，其中n是数据流中插入数据点的个数，且忽略这个细节并假设数据点来自于$\mathbb{R}^d$。

基于随机采样[19,21,27]、局部搜索[22,24-26,28]、核心集构造[36,38,46,81-82,84]等技术，既能够提供近似保证，也能在真实的大规模数据集上获得较快的运行速度，因此成为当前小 k 场景中的主流方案。3）对于一般的 k 较大的情况，长期以来缺乏高效算法，目前的求解器软件多对较小的 k 有较好的算法，对大 k 的性能较为一般。虽然本文介绍了近期的理论突破，但如何将这个理论上可行的算法进行高效实现并产生实际效果，仍是未来重要的研究方向。

参 考 文 献

[1] JÉGOU H，DOUZE M，SCHMID C. Product Quantization for Nearest Neighbor Search [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2011，33(1)：117-128.

[2] PENNINGTON J，SOCHER R，MANNING C D. Glove：Global Vectors for Word Representation [C]// Proceedings of Conference on Empirical Methods in Natural Language Processing. ACL，2014：1532-1543.

[3] BABENKO A，LEMPITSKY V S. Efficient Indexing of Billion-Scale Datasets of Deep Descriptors [C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas：IEEE Computer Society，2016：2055-2063.

[4] MAHAJAN M，NIMBHORKAR P，VARADARAJAN K R. The Planar k-Means Problem is NP-hard [J]. Theoretical Computer Science，2012，442：13-21.

[5] INABA M，KATOH N，IMAI H. Applications of Weighted Voronoi Diagrams and Randomization to Variance-Based k-Clustering [C]// Proceedings of Annual Symposium on Computational Geometry. New York：ACM，1994：332-339.

[6] MATOUŠEK J. On Approximate Geometric k-Clustering [J]. Discrete & Computational Geometry，2000，24(1)：61-84.

[7] DE LA VEGA W F，KARPINSKI M，KENYON C，et al. Approximation Schemes for Clustering Problems [C]// Proceedings of Annual ACM Symposium on Theory of Computing. New York：ACM，2003：50-58.

[8] HAR-PELED S，MAZUMDAR S. On Coresets for k-Means and k-Median Clustering [C]// Proceedings of Annual ACM Symposium on Theory of Computing. New York：ACM，2004：291-300.

[9] FELDMAN D，MONEMIZADEH M，SOHLER C. A PTAS for k-Means Clustering Based on Weak Coresets [C]// Proceedings of ACM Symposium on Computational Geometry. New York：ACM，2007：11-18.

[10] CHEN K. On Coresets for k-Median and k-Means Clustering in Metric and Euclidean Spaces and Their Applications [J]. SIAM Journal on Computing，2009，39(3)：923-947.

[11] KUMAR A，SABHARWAL Y，SEN S. Linear-Time Approximation Schemes for Clustering Problems in Any Dimensions [J]. Journal of the ACM，2010，57(2)：5：1-5：32.

[12] COHEN-ADDAD V. A Fast Approximation Scheme for Low-Dimensional k-Means [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. New Orleans：SIAM，2018：430-440.

[13] FRIGGSTAD Z，REZAPOUR M，SALAVATIPOUR M R. Local Search Yields a PTAS for k-Means in Doubling Metrics [J]. SIAM Journal on Computing，2019，48(2)：452-480.

[14] COHEN-ADDAD V，KLEIN P N，MATHIEU C. Local Search Yields Approximation Schemes for k-Means and k-Median in Euclidean and Minor-Free Metrics [J]. SIAM Journal on Computing，2019，48(2)：644-667.

[15] ABBASI F，BANERJEE S，BYRKA J，et al. Parameterized Approximation Schemes for Clustering with General Norm Objectives [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE，2023：1377-1399.

[16] COHEN-ADDAD V，FELDMANN A E，SAULPIC D. Near-Linear Time Approximation Schemes for Clustering in Doubling Metrics [J]. Journal of the ACM，2021，68(6)：44：1-44：34.

[17] AWASTHI P，CHARIKAR M，KRISHNASWAMY R，et al. The Hardness of Approximation of Euclidean k-Means [C]// Proceedings of International Symposium on Computational Geometry. Eindhoven：Schloss Dagstuhl-Leibniz-Zentrum für Informatik，2015：754-767.

[18] LLOYD S P. Least Squares Quantization in PCM [J]. IEEE Transactions on Information Theory，1982，28(2)：129-136.

[19] ARTHUR D，VASSILVITSKII S. k-Means++：the Advantages of Careful Seeding [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. New Orleans：SIAM，2007：1027-1035.

[20] METTU R R，PLAXTON C G. Optimal Time Bounds for Approximate Clustering [J]. Machine Learning，2004，56(1/3)：35-60.

[21] BACHEM O，LUCIC M，HASSANI S H，et al. Fast and Provably Good Seedings for k-Means [C]// Proceeding of Annual Conference on Neural Information Processing Systems. Barcelona，2016：55-63.

[22] LATTANZI S，SOHLER C. A Better k-Means++ Algorithm via Local Search [C]// Proceedings of International Conference on Machine Learning. Long Beach：PMLR，2019：3662-3671.

[23] CHOO D，GRUNAU C，PORTMANN J，et al. k-Means++：Few More Steps Yield Constant Approximation [C]// Proceedings of International Conference on Machine Learning. Virtual Event：PMLR，2020：1909-1917.

[24] HUANG J，FENG Q，HUANG Z，et al. FLS：A New Local Search Algorithm for k-Means with Smaller Search Space [C]// Proceedings of International Joint Conference on Artificial Intelligence. Vienna：ijcai. org，2022：3092-3098.

[25] BERETTA L，COHEN-ADDAD V，LATTANZI S，et al. Multi-Swap k-Means++ [C]// Proceeding of Annual Conference on Neural Information Processing Systems. 2023.

[26] HUANG J，FENG Q，HUANG Z，et al. Linear Time Algorithms for k-Means with Multi-Swap Local Search [C]// Proceeding of Annual Conference on Neural Information Processing Systems. 2023.

[27] FAN C，LI P，LI X. LSDS++：Dual Sampling for Accelerated k-Means++ [C]// Proceedings of International Conference on Machine Learning. PMLR，2023：9640-9649.

[28] HUANG J，FENG Q，ZHANG Z，et al. Fast Local Search Algorithms for Clustering with Adaptive Sampling and Bandit Strategies [C]// Proceeding of Annual Conference on Neural Informa-

tion Processing Systems. 2025.

[29] BECCHETTI L, BURY M, COHEN-ADDAD V, et al. Oblivious Dimension Reduction for *k*-Means: Beyond Subspaces and the Johnson-Lindenstrauss Lemma [C]// Proceedings of Annual ACM SIGACT Symposium on Theory of Computing. New York: ACM, 2019: 1039-1050.

[30] BOUTSIDIS C, ZOUZIAS A, DRINEAS P. Random Projections for *k*-Means Clustering [C]// Proceedings of Annual Conference on Neural Information Processing Systems. Curran Associates Inc., 2010: 298-306.

[31] COHEN M B, ELDER S, MUSCO C, et al. Dimensionality Reduction for *k*-Means Clustering and Low Rank Approximation [C]// Proceedings of Annual ACM on Symposium on Theory of Computing. New York: ACM, 2015: 163-172.

[32] MAKARYCHEV K, MAKARYCHEV Y, RAZENSHTEYN I. Performance of Johnson-Lindenstrauss Transform for *k*-Means and *k*-Medians Clustering [C]// Proceedings of Annual ACM SIGACT Symposium on Theory of Computing. New York: ACM, 2019: 1027-1038.

[33] JOHNSON W B, LINDENSTRAUSS J. Extensions of Lipschitz Mappings into Hilbert Space [J]. Contemporary mathematics, 1984, 26(1): 189-206.

[34] LANGBERG M, SCHULMAN L J. Universal Epsilon-Approximators for Integrals [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2010: 598-607.

[35] FELDMAN D, LANGBERG M. A Unified Framework for Approximating and Clustering Data [C]// Proceedings of ACM Symposium on Theory of Computing. New York: ACM, 2011: 569-578.

[36] DRAGANOV A, SAULPIC D, SCHWIEGELSHOHN C. Settling Time vs. Accuracy Tradeoffs for Clustering Big Data [J]. Proceedings of the ACM on Management of Data, 2024, 2(3): 173.

[37] HAR-PELED S, KUSHAL A. Smaller Coresets for *k*-Median and *k*-Means Clustering [J]. Discrete & Computational Geometry, 2007, 37(1): 3-19.

[38] BACHEM O, LUCIC M, KRAUSE A. Scalable *k*-Means Clustering via Lightweight Coresets [C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2018: 1119-1127.

[39] HUANG L, JIANG S H, LI J, et al. Epsilon-Coresets for Clustering(with Outliers) in Doubling Metrics [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE Computer Society, 2018: 814-825.

[40] SOHLER C, WOODRUFF D P. Strong Coresets for *k*-Median and Subspace Approximation: Goodbye Dimension [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE Computer Society, 2018: 802-813.

[41] FELDMAN D, SCHMIDT M, SOHLER C. Turning Big Data into Tiny Data: Constant-Size Coresets for *k*-Means, PCA, and Projective Clustering [J]. SIAM Journal on Computing, 2020, 49(3): 601-657.

[42] HUANG L, VISHNOI N K. Coresets for Clustering in Euclidean Spaces: Importance Sampling is Nearly Optimal [C]// Proceedings of Annual ACM SIGACT Symposium on Theory of Computing. New York: ACM, 2020: 1416-1429.

[43] BRAVERMAN V, JIANG S H, KRAUTHGAMER R, et al. Coresets for Clustering in Excluded-Minor Graphs and Beyond [C]// Proceedings of ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2021: 2679-2696.

[44] COHEN-ADDAD V, SAULPIC D, SCHWIEGELSHOHN C. A New Coreset Framework for Clustering [C]// Proceedings of Annual ACM SIGACT Symposium on Theory of Computing. New York: ACM, 2021: 169-182.

[45] COHEN-ADDAD V, LARSEN K G, SAULPIC D, et al. Towards Optimal Lower Bounds for *k*-Median and *k*-Means Coresets [C]// Proceedings of Annual ACM SIGACT Symposium on Theory of Computing. New York: ACM, 2022: 1038-1051.

[46] COHEN-ADDAD V, LARSEN K G, SAULPIC D, et al. Improved Coresets for Euclidean *k*-Means [C]// Proceedings of Annual Conference on Neural Information Processing Systems. 2022.

[47] HUANG L, LI J, WU X. On Optimal Coreset Construction for Euclidean (k,z)-Clustering [C]// Proceedings of Annual ACM Symposium on Theory of Computing. New York: ACM, 2024: 1594-1604.

[48] BANSAL N, COHEN-ADDAD V, PRABHU M, et al. Sensitivity Sampling for *k*-Means: Worst Case and Stability Optimal Coreset Bounds [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE, 2024: 1707-1723.

[49] COHEN-ADDAD V, DRAGANOV A, RUSSO M, et al. A Tight VC-Dimension Analysis of Clustering Coresets with Applications [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2025: 4783-4808.

[50] JAIN K, VAZIRANI V V. Approximation Algorithms for Metric Facility Location and *k*-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation [J]. Journal of the ACM, 2001, 48(2): 274-296.

[51] KANUNGO T, MOUNT D M, NETANYAHU N S, et al. A Local Search Approximation Algorithm for *k*-Means Clustering [J]. Computational Geometry, 2004, 28(2/3): 89-112.

[52] GUPTA A, TANGWONGSAN K. Simpler Analyses of Local Search Algorithms for Facility Location [J]. arXiv: 0809.2554, 2008.

[53] AHMADIAN S, NOROUZI-FARD A, SVENSSON O, et al. Better Guarantees for *k*-Means and Euclidean *k*-Median by Primal-Dual Algorithms [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE Computer Society, 2017: 61-72.

[54] GRANDONI F, OSTROVSKY R, RABANI Y, et al. A Refined Approximation for Euclidean *k*-Means [J]. Information Processing Letters, 2022, 176: 106251.

[55] COHEN-ADDAD V, ESFANDIARI H, MIRROKNI V S, et al. Improved Approximations for Euclidean *k*-Means and *k*-Median, via Nested Quasi-Independent Sets [C]// Proceedings of Annual ACM SIGACT Symposium on Theory of Computing. New York: ACM, 2022: 1621-1628.

[56] CHARIKAR M, COHEN-ADDAD V, GAO R, et al. An Im-

proved Greedy Approximation for(Metric) k-Means [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE Computer Society, 2025.

[57] BADOIU M, CZUMAJ A, INDYK P, et al. Facility Location in Sublinear Time [C]// Proceedings of Automata, Languages and Programming, International Colloquium. Springer, 2005: 866-877.

[58] INDYK P, MOTWANI R. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality [C]// Proceedings of Annual ACM Symposium on the Theory of Computing. New York: ACM, 1998: 604-613.

[59] HAR-PELED S, INDYK P, MOTWANI R. Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality [J]. Theory of Computing, 2012, 8(1): 321-350.

[60] ANDONI A, INDYK P. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions [C]// Proceedings of Annual IEEE Symposium on Foundations of Computer Science. IEEE Computer Society, 2006: 459-468.

[61] ANDONI A, INDYK P. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions [J]. Communications of the ACM, 2008, 51(1): 117-122.

[62] ANDONI A, RAZENSHTEYN I P. Optimal Data-Dependent Hashing for Approximate Near Neighbors [C]// Proceedings of Annual ACM on Symposium on Theory of Computing. New York: ACM, 2015: 793-801.

[63] GOEL A, INDYK P, VARADARAJAN K R. Reductions among High Dimensional Proximity Problems [C]// Proceedings of Annual Symposium on Discrete Algorithms. Washington: ACM/SIAM, 2001: 769-778.

[64] JIANG S H, JIN Y, LOU J, et al. Local Search for Clustering in Almost-Linear Time [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2026.

[65] HAR-PELED S, INDYK P, SIDIROPOULOS A. Euclidean Spanners in High Dimensions [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2013: 804-809.

[66] THORUP M. Quick k-Median, k-Center, and Facility Location for Sparse Graphs [J]. SIAM Journal on Computing, 2004, 34(2): 405-432.

[67] ARYA V, GARG N, KHANDEKAR R, et al. Local Search Heuristics for k-Median and Facility Location Problems [J]. SIAM Journal on Computing, 2004, 33(3): 544-562.

[68] LA TOUR M D, SAULPIC D. Almost-Linear Time Approximation Algorithm to Euclidean k-Median and k-Means [J]. arXiv: 2407.11217, 2024.

[69] METTU R R, PLAXTON C G. The Online Median Problem [J]. SIAM Journal on Computing, 2003, 32(3): 816-832.

[70] COHEN-ADDAD V, LATTANZI S, NOROUZI-FARD A, et al. Fast and Accurate k-Means++ via Rejection Sampling [C]// Proceedings of Annual Conference on Neural Information Processing Systems. 2020.

[71] CHARIKAR M, HENZINGER M, HU L, et al. Simple, Scalable and Effective Clustering via One-Dimensional Projections [C]// Proceedings of Annual Conference on Neural Information Processing Systems. 2023.

[72] BHATTACHARYA S, COSTA M, FAROKHNEJAD E, et al. Fully Dynamic Euclidean k-Means [J]. arXiv: 2507.11256, 2025.

[73] LATTANZI S, VASSILVITSKII S. Consistent k-Clustering [C]// Proceedings of International Conference on Machine Learning. PMLR, 2017: 1975-1984.

[74] COHEN-ADDAD V, HJULER N, PAROTSIDIS N, et al. Fully Dynamic Consistent Facility Location [C]// Proceedings of Annual Conference on Neural Information Processing Systems. 2019: 3250-3260.

[75] BHATTACHARYA S, COSTA M, LATTANZI S, et al. Fully Dynamic k-Clustering in $\widetilde{O}(k)$ Update Time [C]// Proceedings of Annual Conference on Neural Information Processing Systems. 2023.

[76] LA TOUR M D, HENZINGER M, SAULPIC D. Fully Dynamic k-Means Coreset in Near-Optimal Update Time [C]// Proceedings of Annual European Symposium on Algorithms. London: Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2024: 100: 1-100: 16.

[77] BHATTACHARYA S, COSTA M, GARG N, et al. Fully Dynamic k-Clustering with Fast Update Time and Small Recourse [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE, 2024: 216-227.

[78] BHATTACHARYA S, COSTA M, FAROKHNEJAD E. Fully Dynamic k-Median with Near-Optimal Update Time and Recourse [C]// Proceedings of Annual ACM Symposium on Theory of Computing. New York: ACM, 2025: 1166-1177.

[79] BRAVERMAN V, FRAHLING G, LANG H, et al. Clustering High Dimensional Dynamic Data Streams [C]// Proceedings of International Conference on Machine Learning. PMLR, 2017: 576-585.

[80] SONG Z, YANG L F, ZHONG P. Sensitivity Sampling over Dynamic Geometric Data Streams with Applications to k-Clustering [J]. arXiv: 1802.00459, 2018.

[81] ACKERMANN M R, LAMMERSEN C, MÄRTENS M, et al. Streamkm++: A Clustering Algorithms for Data Streams [C]// Proceedings of the Twelfth Workshop on Algorithm Engineering and Experiments. SIAM, 2010: 173-187.

[82] FICHTENBERGER H, GILLÉ M, SCHMIDT M, et al. BICO: BIRCH Meets Coresets for k-Means Clustering [C]// Proceedings of Annual European Symposium on Algorithms. Springer, 2013: 481-492.

[83] AILON N, JAISWAL R, MONTELEONI C. Streaming k-Means Approximation [C]// Proceedings of Annual Conference on Neural Information Processing Systems. Curran Associates Inc., 2009: 10-18.

[84] ZHANG Y, TANGWONGSAN K, TIRTHAPURA S. Streaming k-Means Clustering with Fast Queries [C]// Proceedings of IEEE International Conference on Data Engineering. IEEE Computer Society, 2017: 449-460.

[85] BRAVERMAN V, FELDMAN D, LANG H, et al. Streaming Coreset Constructions for M-Estimators [C]// Approximation, Randomization, and Combinatorial Optimization. Cambridge:

Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2019: 62: 1-62: 15.

[86] COHEN-ADDAD V, WOODRUFF D P, ZHOU S. Streaming Euclidean k-Median and k-Means with $o(\log n)$ Space [C]// Proceedings of IEEE Annual Symposium on Foundations of Computer Science. IEEE, 2023: 883-908.

[87] BRAVERMAN V, LANG H, LEVIN K D, et al. Clustering Problems on Sliding Windows [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2016: 1374-1390.

[88] EPASTO A, LATTANZI S, VASSILVITSKII S, et al. Submodular Optimization over Sliding Windows [C]// Proceedings of International Conference on World Wide Web. ACM, 2017: 421-430.

[89] EPASTO A, MAHDIAN M, MIRROKNI V S, et al. Improved Sliding Window Algorithms for Clustering and Coverage via Bucketing-Based Sketches [C]// Proceedings of ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2022: 3005-3042.

[90] WOODRUFF D P, ZHONG P, ZHOU S. Near-Optimal k-Clustering in the Sliding Window Model [C]// Proceedings of Annual Conference on Neural Information Processing Systems. 2023.

[91] KARLOFF H J, SURI S, VASSILVITSKII S. A Model of Computation for Mapreduce [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2010: 938-948.

[92] DEAN J, GHEMAWAT S. Mapreduce: Simplified Data Processing on Large Clusters [J]. Communications of the ACM, 2008, 51(1): 107-113.

[93] WHITE T. Hadoop-The Definitive Guide: Storage and Analysis at Internet Scale(4th ed)[M]. O'Reilly, 2015.

[94] ZAHARIA M, CHOWDHURY M, FRANKLIN M J, et al. Spark: Cluster Computing with Working Sets [C]// 2nd USENIX Workshop on Hot Topics in Cloud Computing. Boston: USENIX Association, 2010.

[95] COHEN-ADDAD V, KUHN F, PARSAEIAN Z. An Efficient Massively Parallel Constant-Factor Approximation Algorithm for the k-Means Problem [C]// Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, 2026.

[96] BHASKARA A, WIJEWARDENA M. Distributed Clustering via LSH Based Data Partitioning [C]// Proceedings of International Conference on Machine Learning. PMLR, 2018: 569-578.

[97] CZUMAJ A, GAO G, JIANG S H, et al. Fully-Scalable MPC Algorithms for Clustering in High Dimension [C]// International Colloquium on Automata, Languages, and Programming. Tallinn: Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2024: 50: 1-50: 20.

GAO Guichen, born in 1995, postgraduate, is a member of CCF(No. P3224G). Her main research interests include theoretical computer science and algorithms.

JIANG Shaofeng, born in 1990, Ph. D, assistant professor, Ph. D supervisor, is a member of CCF(No. H3889S). His main research interests include theoretical computer science, algorithms in massive datasets, approximation algorithms and online algorithms.

(责任编辑:李亚辉)