2023 算法设计与分析课程报告

题目：k-中值和 k-均值核集的最优下界

姓名：刘俊杉

学号：2021112078

1.论文介绍

1.1论文背景

给定度量空间中的一组点，(k,z)-聚类问题包括找到一组被认定为中心的 k 个点，从而使每个数据点到其最近中心的距离之和最小化。尽管人们一直在努力了解在各种度量空间中，这两个问题的最佳核心集大小是多少，但在现有技术的上限和下限之间仍然存在显著差距。在本文中，在上界和下界上都取得了进展，获得了几种情况的紧界。

1.2问题定义

k-median 与 k-means 问题两个问题的最佳核心集大小是多少，几种情况下上界和下界的紧界是多少。

1.3算法介绍

【设计思想】

1、在有限 n 点一般度量中，任何核心集都必须由Ω（$k\log n/\varepsilon^2$）个点组成。这改进了Ω（$k\log n/\varepsilon$）的下界并且匹配 Feldman 和 Langberg 提出的 k-中值的上限以及 Cohen Addad、Saulpic 和 Schwiegelshohn 提出的 k-均值高达多对数因子。

2、对于具有加倍常数 D 的加倍度量，任何核心集都必须由Ω（kD/ε^2）个点组成。这与 Cohen Addad、Saulpic 和 Schwiegelshohn 的 k 中值和 k 均值上限相匹配。

3、在 d 维欧几里得空间中，任何核心集都要求有由Ω（k/ε^2）个点。

这改进了 Baker, Braverman, Huang, Jiang, Krauthgamer,和 Wu 提出的下界Ω（k/$\sqrt{\varepsilon}$）。

我们用大小为Ω（k/ε^2）的核心集的构造来补充 d 维欧几里得空间的下界。

【算法执行流程】

首先，我们对近似任何$\|v^s\|_1$的方差进行了定界。假设我们做出简化假设，即所有点都小于 1，并且我们的目标是最多ε·n 的误差。在这种情况下，方差是恒定的，在此基础上应用 Chernoff 界只需要$Var\cdot\varepsilon^{-2}$个样本来近似任何单个$\|v^s\|_1$。

其次，我们必须对所有v^s应用一个并集约束。在欧几里得空间中，一个原始的并界是无用的，因为有无限多的候选解。为了离散 S，先前的工作，无论是隐式的还是显式的，都表明存在一小组向量N^ε，此后称为网络。因此，对于所有$\|v\|_1$，v ∈ N^ε的精确估计足以实现对所有v^s的估计。

我们使用不同尺度的网络，即我们有网络N^1，$N^{\frac{1}{2}}$，$N^{\frac{1}{4}}$ 等等。这些网允许我们将每个v^s写为不同尺度的网络向量的伸缩和。

【例子】可以将这一思想应用于伸缩和的每个连续求和（或者更确切地说，应用于净向量链的每个环节），忽略多对数因子。kmeans 用于数据集内种类属性不明晰，希望能够通过数据挖掘出或自动归类出有相似特点的对象的场景。其商业界的应用场景一般为挖掘出具有相似特点的潜在客户群体以便公司能够重点研究、对症下药。

比如二分 K-Means 聚类算法伪代码

将所有点看成一个簇

当簇数目小于 k 时

对于每一个簇

计算总误差

在给定的簇上面进行 KMeans 聚类（k=2）

计算将该簇一分为二之后的总误差

选择使得误差最小的那个簇进行划分操作

另一种做法是选择 SSE 最大的簇进行划分，直到簇数目达到用户指定的数目位置

1.4理论和实验结论

图表 1 边界和我们的结果之间的比较。用*标记的结果对于 k 中值和 k 均值来说是严格的。

Metric Space	Best upper bound	Best lower bound	Our result
Discrete Metrics	$O(k\varepsilon^{-\max(2,z)}\log n)$ [35]	$\Omega(k\varepsilon^{-1}\log n)$ [6]	$\Omega(k\varepsilon^{-2}\log n)$*
with doubling dimension D	$O(k\varepsilon^{-\max(2,z)}D)$ [35]	-	$\Omega(k\varepsilon^{-2}D)$*
Euclidean k-median	$\widetilde{O}(k\varepsilon^{-4})$ [55]	$\Omega(k\varepsilon^{-1/2})$ [6]	$\widetilde{O}(k\varepsilon^{-3})$ $\Omega(k\varepsilon^{-2})$
Euclidean k-means	$\widetilde{O}(k\varepsilon^{-4})$ [35]	-	$\Omega(k\varepsilon^{-2})$
Euclidean	$\widetilde{O}(k\varepsilon^{-2-\max(2,z)})$ [35] $\widetilde{O}(k^2\varepsilon^{-4})$ [17]	$\Omega(k2^{z/100})$ [55]	$\widetilde{O}_z(k\varepsilon^{-2}\cdot\min(\varepsilon^{-z},k))$ $\Omega(k\varepsilon^{-2})$

图表 2 欧几里得空间中（k,z）-聚类的核集大小的比较。

Reference	Size (Number of Points)
Coreset Bounds in Euclidean Spaces	
Lower Bounds	
Baker, Braverman, Huang, Jiang, Krauthgamer, Wu (ICML'19) [15]	$\Omega(k \cdot \varepsilon^{-1/2})$
Huang, Vishnoi (STOC'20) [55]	$\Omega(k \cdot \min(d, 2^{z/20}))$
This paper	$\Omega(k \cdot \varepsilon^{-2}/z^4)$
Upper Bounds	
Har-Peled, Mazumdar (STOC'04) [50]	$O(k \cdot \varepsilon^{-d} \cdot \log n)$
Har-Peled, Kushal (DCG'07) [49]	$O(k^3 \cdot \varepsilon^{-(d+1)})$
Chen (Sicomp'09) [23]	$O(k^2 \cdot d \cdot \varepsilon^{-2} \cdot \log n)$
Langberg, Schulman (SODA'10) [65]	$O(k^3 \cdot d^2 \cdot \varepsilon^{-2})$
Feldman, Langberg (STOC'11) [41]	$O(k \cdot d \cdot \varepsilon^{-2z})$
Feldman, Schmidt, Sohler (Sicomp'20) [43]	$O(k^3 \cdot \varepsilon^{-4})$
Sohler, Woodruff (FOCS'18) [86]	$O(k^2 \cdot \varepsilon^{-O(z)})$
Becchetti, Bury, Cohen-Addad, Grandoni, Schwiegelshohn (STOC'19) [8]	$O(k \cdot \varepsilon^{-8})$
Huang, Vishnoi (STOC'20) [55]	$O(k \cdot \varepsilon^{-2-2z})$
Bravermann, Jiang, Krautgamer, Wu (SODA'21) [17]	$O(k^2 \cdot \varepsilon^{-4})$
Cohen-Addad, Saulpic, Schwiegelshohn (STOC'21) [35]	$\tilde{O}(k \cdot \varepsilon^{-2-\max(2,z)})$
This paper	$\tilde{O}(k \cdot \varepsilon^{-2} \cdot \min(\varepsilon^{-z}, k))$
General n-point metrics, D denotes the doubling dimension	
Lower Bounds	
Braverman, Jiang, Krauthgamer, Wu (ICML'19) [16]	$\Omega(k \cdot \varepsilon^{-1} \cdot \log n)$
This paper	$\Omega(k \cdot \varepsilon^{-2} \cdot \log n)$
This paper	$\Omega(k \cdot \varepsilon^{-2} \cdot D)$
Upper Bounds	
Chen (Sicomp'09) [23]	$O(k^2 \cdot \varepsilon^{-2} \cdot \log^2 n)$
Feldman, Langberg (STOC'11) [41]	$O(k \cdot \varepsilon^{-2z} \cdot \log n)$
Huang, Jiang, Li, Wu (FOCS'18) [51]	$O(k^3 \cdot \varepsilon^{-2} \cdot D)$
Cohen-Addad, Saulpic, Schwiegelshohn (STOC'21) [35]	$\tilde{O}(k \cdot \varepsilon^{-\max(2,z)} \cdot D)$
Cohen-Addad, Saulpic, Schwiegelshohn (STOC'21) [35]	$\tilde{O}(k \cdot \varepsilon^{-\max(2,z)} \cdot \log n)$

一系列结果[8，10，11，12，25，36，43，44，64，73，86]探讨了使用降维方法进行 k 聚类的可能性，特别关注主成分分析（PCA）和随机投影。降维问题，至少在这些技术方面，目前已经基本解决。

2.领域综述

2.1k-median 与 k-means 问题领域背景

给定度量空间中的一组点，(k,z)-聚类问题包括找到一组被认定为中心的 k 个点，从而使每个数据点到其最近中心的距离之和最小化。特殊情况包括著名的 k-中值问题（z=1）和 k-均值问题（z=2）。k-中值和 k-均值问题是现代数据分析的核心，大量数据应用提出了核心集的概念：输入点集的一个小（加权）子集，将问题的任何解决方案的成本保持在乘法（$1\pm\xi$）之内，从而将问题的输入从大规模减少到小规模。

2.2k-median 与 k-means 问题方法介绍

聚类是数据集 P 的分区，使得同一聚类中的数据点相似，而不同聚类中的点不同。各种聚类问题已经成为组合优化和机器学习问题的重要基石。其中，基于中心的聚类问题可以说是研究和使用最广泛的问题。这里，数据元素位于度量空间中，每个聚类都与一个中心点相关联，数据点的成本是数据点与其分配的聚类之间距离的函数。（k,z）问题通过成本函数实现了这一目标和其他重要目标。

$$\mathrm{cost}(P,\mathcal{S}) := \sum_{p\in P} \min_{s\in\mathcal{S}} d(p,s)^z,$$

其中 z 是正整数，|S| = k，d（,）表示距离函数。对于 z = 1，这是 k-中值问题，对于 z = 2，这是同样需要深入研究的 k-均值问题。

在实践中使用的数据集通常是巨大的，包含数亿个点在空间分布并且随着时间的推移而演变。因此，在这些设置中，经典算法（例如 Lloyd 或 k-means++）失效；数据集的大小禁止对输入数据进行多次传递，而找到输入数据的紧凑表示至关重要。这导致了一种权衡：数据集越小，我们需要的存储就越少，我们在数据集上运行算法的速度就越快，但相反，数据集越少，原始数据的信息就会丢失得越多。

具体来说，给定一个精度参数ξ，k 和 z，一个（ξ，k，z）核心集Ω 是具有权重ω的 P 的子集：Ω→R，它近似于任何候选解决方案 S 的 P 成本，高达（1±ξ）因子，即

$$\forall \mathcal{S}, \quad (1-\varepsilon)\mathrm{cost}(P,\mathcal{S}) \leq \sum_{p\in\Omega} w(p)\mathrm{cost}(p,\mathcal{S}) \leq (1+\varepsilon)\mathrm{cost}(P,\mathcal{S}).$$

因此，一个小的（ξ，k，z）核心集可以很好地压缩初始数据集，因为它保留了任何可能的解决方案的成本。人们可以简单地存储核心集，而不是存储完整的数据集，从而节省内存占用并加快性能。我们注意到，在一些定义中，偏移量Δ被添加到核心集：在这种情况下，解决方案 S 的核心集成本为（1±ξ），成本为（P，S）+Δ。

在输入空间是无限的情况下（例如，欧氏空间），核心集点可以从整个空间中选择，并且不限于是输入的一部分。

尽管许多工作都集中在提高核心集结构的大小上，但我们对核心集下限的理解相对有限，并且在核心集大小上，最佳上限和下限之间存在显著差距。例如，即使对于 Euclidean k-均值，没有什么比下限已知更重要。在这项工

作中，我们试图系统地获得这些问题的下界。

2.3Euclidean Lower Bounds 方法介绍

下界证明由三个单独的步骤组成，它们结合起来证明了点集的任何核心集$P = \{e_1, \cdots e_d\}$在R^d必须有大小Ω。基本方法是证明具有 k 个中心的 P 的任何聚类都有很大的成本，同时，对于任何核心集Ω 使用$o(d)$加权点，可以实现低成本的聚类。将两者结合得出下限。我们分三步进行这个证明。

在第一步中，我们证明了任何使用单位范数中心的 P 聚类的代价至少为 $2d - O(\sqrt{dk})$。

在下一步中，我们展示了对于任何核心集Ω 由 t 点和权重 w 组成：有一种使用单位范数中心的低成本聚类，其成本为 $2d - \Omega\ (\sqrt{\frac{k}{t}} \cdot \sum_{p\in\Omega} \omega(p)\, \|p\|_2)$。

将其与第一步相结合意味着$\sum_{p\in\Omega} \omega(p)\, \|p\|_2 = O(\sqrt{td})$。

在最后一步中，我们展示了任何核心集 C 必须具有$\sum_{p\in\Omega} \omega(p)\, \|p\|_2 = O(d)$，$d = \theta(k \cdot \varepsilon^{-2})$。将此与前面的两个步骤相结合，最终产生$\sqrt{td} = \Omega(d) \rightarrow t = \Omega\ (k \cdot \varepsilon^{-2})$。

3.改进点概述

将分析从加法近似改进为乘法近似导致了几个困难。如果不使用所有点的成本都小于 1 的假设，则方差会增加。事实上，与之前的工作相比，之前的工作使用基于链接的分析来获得单个中心的核心集边界，以及之前的工作使用基于链接启发的方差减少技术，这两种技术都设法获得了恒定的方差，在这种情况下对方差进行边界限制是很大的，需要一些新的想法。我们可以显示的用于估计$\|v^s\|_1$的最低方差仅为 $\min(\varepsilon^{-z}, k)$，导致方差的这个边界是紧的。需要进一步的思考才能达到（推测的）的最优界$\theta(k\cdot\varepsilon^{-2})$。

4.算法与分析

4.1算法设计思想

我们使用以下符号。我们使用$\|P\|_0$来表示 P 中的不同点数。对于解 S，我们将由 S 诱导的$|P|$ 维成本向量v^s 定义为$v_p^S = cost(p, S)$。我们还将使用以下引理来获得 k-均值的三角形不等式的和一般的距离幂。

4.2算法过程描述

【自然语言描述】

我们做出以下三个假设：

假设 1 不同点的数量$\|P\|_0$为 $\mathrm{poly}(k/\varepsilon)$。

假设 2 点的尺寸 d 为$O(\varepsilon^{-2}\log\|P\|)$

假设 3 点集未加权重。

假设这些会大大简化演示。第一个假设可以通过在预处理中计算（潜在加权的）核心集来证明。

第二个假设来自 Narayanan 和 Nelson 关于终端嵌入的结果。这里只需要说，存在一个保留从任意维度到期望目标维度的嵌入的核心集。

最后的假设是按比例缩放权重并将其四舍五入为整数。然后，每个权重都被视为一个点的多重性。请注意，这不会增加不同的点数。

我们现在描述算法。我们首先计算整个实例的一些常数因子近似值 A。设Ci是由 A 生成的第 i 个聚类。Ci的平均损失是$\frac{cost\ (C_i,\mathrm{A})}{|C_i|}$。对于所有的$i,j$，环$R_{i,j}$是点$p \in C_i$的集合。集群的$C_i$由内环$\cup_{j\le z\log(\varepsilon/z)} R_{i,j}$组成。主环$R_M$（Ci）由$Ci$的所有其他点组成。对于每个 j，Rj 被定义为 $\cup_{i=1}^{k} R_{i,j}$。然后我们将输入点集划分为组。

【伪代码叙述】

Algorithm 1 Euclidean Coreset Construction

Compute a $O(2^z)$ approximation $\mathcal{A}$ to P.
Preprocess the instance such that Assumptions 1-3 hold.
Partition the points into groups $\mathcal{G} = \left(\bigcup_j G_{j,\max} \cup \bigcup_b G(j,b) \setminus G_{j,\min}\right) \cup \left(G^O_{\max} \cup \bigcup_b G^O_b \setminus G^O_{\min}\right)$.
for all Groups $G \in \mathcal{G}$ **do**
  Sample $\delta \in k \cdot \log \frac{k}{\varepsilon} \cdot \varepsilon^{-2} \cdot 2^{O(z\log(1+z))} \cdot \log^3 \varepsilon^{-1} \cdot \min(\varepsilon^{-z}, k)$ points Ω_G proportionate to $\frac{\mathrm{cost}(p,\mathcal{A})}{\mathrm{cost}(G,\mathcal{A})}$,
  and weighted by $\frac{\mathrm{cost}(G,\mathcal{A})}{\delta \cdot \mathrm{cost}(p,\mathcal{A})}$.
end for
for all $c_i \in \mathcal{A}$ **do**
  Weigh $c_i \in \mathcal{A}$ by the number of points not in $\mathcal{G} \cap C_i$
end for
Output $\Omega = \mathcal{A} \cup \bigcup_G \Omega_G$.

【伪代码逐行解释】

对于所有的 i,j，点集里包含着环$R_{i,j}$以至于$2^j\Delta C_i \le \mathrm{cost}(p,\mathrm{A}) \le 2^{j+1}\Delta_C$

集群C_i内环 $R_I(C_i) := \cup_{j\le z\log(\varepsilon/z)} R_{i,j}$ 最高成本为$(\varepsilon/z)^z\Delta_C$

对于每一个 j，R_j 被定义为 $\cup^k R_{i,j}$ ，然后，我们将输入点集划分为以下组。

For each j, the rings $R_{i,j}$ are gathered into *groups* G^M:

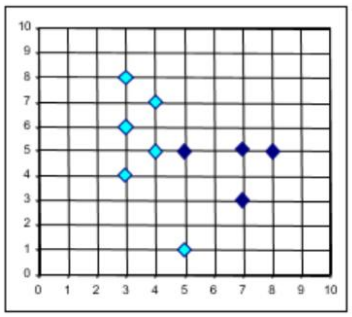
For any j, let $G^M_{j,min} := \cup_{b\le 0} G^M_{j,b}$ be the union of the cheapest groups

外圈中的点也被划分为外部组

【例子说明】

1. 选取我们的初始的中心点的个数
2. 计算剩余的点的距离到初始的中心点的距离
3. 将距离到中心点的距离最短的归为一类
4. 用曼哈顿距离重新计算中心点
5. 重复 3,4 两个步骤，直到中心点不会变化为止

1. (3,8)
2. (3,6)
3. (3,4)
4. (4,5)
5. (4,7)
6. (5,1)
7. (5,5)
8. (7,3)
9. (7,5)
10. (8,5)



图表 3 样例

4.3算法的结果

【需要和第 3 节所述改进点对应，论述为何解决改进点提出的问题】

我们下界背后的一般思想是对随机变量的和使用紧集中界和反集中界

随着网络变得越来越细，差异也越来越小。这种差异直接影响方差的界限，方差从常数降低到大约$2^{2h}\cdot O(1)$。由于有许多差异向量，我们可以通过方差的减少来补偿净大小的增加。将这一思想应用于伸缩和的每个连续求和（或者更确切地说，应用于净向量链的每个环节），会导致$k\cdot\varepsilon^{-2}$的样本总数，忽略多对数因子。将分析从加法近似改进为乘法近似导致了几个困难。如果不使用所有点的成本都小于 1 的假设，则方差会增加。之前的工作使用基于链接的分析来获得单个中心的核心集边界，以及之前的

工作使用基于链接启发的方差减少技术，这两种技术都设法获得了恒定的方差，在这种情况下对方差进行边界限制是很大的，需要一些新的想法。并且需要进一步的想法来达到$\theta(k\cdot\varepsilon^{-2})$的（推测的）最优界