基于随机数三角阵映射的高维大数据二分聚类初始中心高效鲁棒生成算法

李 旻*①② 何婷婷①

①(华中师范大学国家数字化学习工程技术研究中心 武汉 430079)

②(河南大学计算机与信息工程学院 开封 475001)

摘 要：Bisecting K-means算法通过使用一组初始中心对分割簇，得到多个二分聚类结果，然后从中选优以减轻局部最优收敛问题对算法性能的不良影响。然而，现有的随机采样初始中心对生成方法存在效率低、稳定性差、缺失值等不同问题，难以胜任大数据聚类场景。针对这些问题，该文首先创建出了初始中心对组合三角阵和初始中心对编号三角阵，然后通过建立两矩阵中元素及元素位置间的若干映射，从而实现了一种从随机整数集合中生成二分聚类初始中心对的线性复杂度算法。理论分析与实验结果均表明，该方法的时间效率及效率稳定性均明显优于常用的随机采样方法，特别适用于高维大数据聚类场景。

关键词：Bisecting K-means；初始中心生成；三角矩阵映射；随机整数；高维大数据聚类；线性算法

中图分类号：TN911.7; TP391 **文献标识码**：A **文章编号**：1009-5896(2021)04-0948-08

DOI: 10.11999/JEIT200043

An Efficient and Robust Algorithm to Generate Initial Center of Bisecting K-means for High-dimensional Big Data Based on Random Integer Triangular Matrix Mappings

LI Min①② HE Tingting①

①(*National Engineering Research Center for E-Learning* (*Central China Normal University*), *Wuhan* 430079, *China*)

②(*Computer and Information Engineering College*, *Henan University*, *Kaifeng* 475001, *China*)

Abstract: The algorithm of Bisecting K-means obtains multiple clustering results by using a set of initial center pairs to segment a cluster, and then selects the best from them to mitigate the adverse effect of the local optimal convergence on the performance of the algorithm. However, the current methods of random sampling to generate initial center pairs for Bisecting K-means have some problems, such as low efficiency, poor stability, missing values and so on, which are not competent for big data clustering. In order to solve these problems, firstly the lower triangular matrix composed by the pairs of initial centers and the lower triangular matrix composed by serial numbers of the pairs of initial centers are created. Then, by establishing several mappings between the elements and their positions in the two matrices, a linear complexity algorithm is proposed to generate initial center pairs from the set of random integers. Both theoretical analysis and experimental results show that the time efficiency and efficiency stability of this method are significantly better than the current methods of random sampling, so it is particularly suitable for these scenarios of high-dimensional big data clustering.

Key words: Bisecting K-means; Initial center generation; Triangular matrix mapping; Random integer; High-dimensional big data clustering; Linear algorithm

收稿日期：2020-01-13；改回日期：2020-07-28；网络出版：2020-08-21

*通信作者： 李旻 limin_ha139@139.com

基金项目：河南省科技攻关计划(162102210168)

Foundation Item: The Science and Technology Research Plan in Henan Province (162102210168)

1　引言

Bisecting K-means是一种十分经典的聚类算法[1,2]，由于该算法优点众多[3,4]，广泛应用于各种聚类场景[5–7]，且成为各种聚类算法对比的标杆之一[8,9]。然而，该算法也存在局部最优收敛问题，即以不同的两个样本作为初始中心来分割簇，最终得到的“最优聚类模式”可能各不相同[10–12]。为了减轻局部最优收敛的不良影响，最常用的方法是尝试用多个不同的初始中心对来分割簇，以得到多个局部最优聚类模式，然后从中选取质量最好的聚类模式作为最终聚类结果[13–15]。因此，Bisecting K-means算法首要解决的问题便是：如何从待分割簇中，生成一组符合要求的初始中心对。目前，常用的随机采样初始中心生成方法包括两大类：基于样本索引采样的方法、基于样本特征采样的方法。其中，第1类方法的处理过程与样本特征无关，第2类方法的处理过程与样本特征相关，故此基于样本索引采样的方法更适合于处理高维度大数据。在基于样本索引采样的方法中，常用的有以下几种。

(1)随机选取方法：从待分割簇中，随机选取两个不同样本作为初始中心对[14,16]。该方法优点：简单；缺点：多次生成的初始中心对可能有重复。选定初始中心对后，因后续进行的二分聚类操作复杂度较大，特别是对于高维大数据而言。所以应避免重复的初始中心对，以防后续做无谓的大复杂度计算。

(2)随机选取+单点排除方法：选取初始中心对时，把之前曾被选为初始中心的单个样本点排除在备选范围之外。该方法优点：简单、多次生成的初始中心对无重复；缺点：存在缺失值问题(即排除掉从未尝试过的初始中心对)[17]。

(3)随机选取+碰撞检测方法：从待分隔簇中，随机选取2个不同样本，构成待用初始中心对，然后检测其是否已被生成过(即碰撞检测)。若已生成过(即发生碰撞)，则重复生成操作，直至“无碰撞”[1,3]。该方法优点：无重复、无缺失值问题；缺点：碰撞检测效率低、随机碰撞造成时间效率不稳定。

由以上分析可知，现有的随机采样初始中心生成方法存在着各种问题，难以胜任大数据聚类场景。有鉴于此，本文提出了基于随机数三角阵映射的二分聚类初始中心生成方法。

2　随机数三角阵映射方法

2.1　初始中心索引对集合

待分割簇中两个不同样本的索引构成一个样本索引对，所有不重复对构成的集合便是该待分割簇的初始中心索引对集合(the set of Pairs of Indexes of two Initial centers in the Cluster, PIIC)。其对应于待分割簇的初始中心对池，定义如下

$\mathrm{PIIC}=\{\ (i,j)\mid i,j\in Z,\ 1\le i<j\le n,\ n=|C^*|\ \}$;

i,j为待分割簇中的初始中心索引；

C^*为待分割簇。

2.2　初始中心对组合三角阵

若待分割簇有n个样本，可用下三角矩阵将所有可选初始中心对的组合都表示出来，该矩阵称为初始中心对组合三角阵(the lower Triangular Matrix composed by the Pairs of initial centers, TMP)，其具体形式如图1所示。

对于TMP中的任意元素e，其行编号row(e)为第1个初始中心的索引i，其元素大小e为第2个初始中心的索引j，即二元组(row(e), e)与唯一的初始中心对(S_i, S_j)相对应(其中i=row(e), j=e)。由此可知，TMP中的元素与PIIC中的元素可构成一对一映射，该映射定义为

$f:E_{\mathrm{TMP}}\rightarrow\mathrm{PIIC}$;

$E_{\mathrm{TMP}}=\{\ e\mid e\in\mathrm{TMP}\ \}$;

$$(i,j)=f(e)=\begin{cases} i=\mathrm{row}(e)\\ j=e\end{cases}$$

(1) TMP中元素位置到元素大小的映射。

观察图1可知，TMP中元素位置到元素大小可构成一对一映射，该映射定义为

$f:P_{\mathrm{TMP}}\rightarrow\mathrm{E}_{\mathrm{TMP}}$;

$P_{\mathrm{TMP}}=\{\ (x,y)\mid x,y\in Z,\ 1\le x,y\le n-1,\ x+y\le n,\ n=|C^*|\ \}$;

$e=f\ (x,y)=x+y$;

P_{TMP}：TMP中元素位置的集合；

x=row(e)：TMP中元素的行编号；

y=column(e)：TMP中元素的列编号。

(2) TMP中“元素位置”到“初始中心索引对”的映射。

由映射$f:P_{\mathrm{TMP}}\rightarrow E_{\mathrm{TMP}}$和映射$f:E_{\mathrm{TMP}}\rightarrow$PIIC，可得由TMP中元素位置到PIIC中初始中心索引对的映射，该映射定义为

	1	2	3	4	…	n-3	n-2	n-1
n-1	n							
n-2	n-1	n						
n-3	n-2	n-1	n					
n-4	n-3	n-2	n-1	n				
⋮	⋮	⋮	⋮	⋱				
3	4	5	6	…	n			
2	3	4	5	…	n-1	n		
1	2	3	4	…	n-2	n-1	n	

图 1 初始中心对组合三角阵TMP

$f:\mathrm{P_{TMP}}\to\mathrm{PIIC}$;

$$(i,j)=f(x,y)=\begin{cases} i=x \\ j=x+y \end{cases}$$

由此可知：随机生成初始中心对的操作，可转换为从TMP中随机选取元素的操作。然而，直接从TMP中随机选取元素，无法保证每次生成的初始中心对均不重复，为此还需借助于另一个矩阵。

2.3 初始中心对编号三角阵

若待分割簇有n个样本，将所有可选初始中心对从1至$n(n-1)/2$进行编号，然后将这些编号按升序排列成下三角矩阵，便得到初始中心对编号三角阵(the lower Triangular Matrix composed by Serial numbers of the pairs of initial centers, TMS)，其具体形式如图2所示。

TMS中的每个编号都与TMP中相应位置元素所确定的唯一初始中心对相对应。如TMS中第1行第1列的元素1，对应于TMP中第$n-1$行第1列的元素n，故TMS中的“元素1”对应的初始中心对为(S_{n-1}, S_n)。由此可知，TMS中的元素与初始中心对存在映射关系，下面推导该映射。

(1)TMS到TMP的元素位置映射。

观察两矩阵的结构可知，其对应元素的位置满足以下映射关系：

$f:P_{\mathrm{TMS}}\to P_{\mathrm{TMP}}$;

$P_{\mathrm{TMS}}=\{\ (x,\ y)\mid x,\ y\in Z,\ 1\le y\le x\le n-1,\ n=|C^*|\ \}$;

$P_{\mathrm{TMP}}=\{\ (x',\ y')\mid x',\ y'\in Z,\ 1\le x',\ y'\le n-1,\ x'+y'\le n,\ n=|C^*|\ \}$;

$$(x',y')=f(x,y)=\begin{cases} x'=n-x \\ y'=y \end{cases};$$

P_{TMS}：TMS中元素位置的集合。

(2) TMS中元素位置到元素大小的映射。

观察TMS的结构可知，其元素位置到元素大小满足以下映射：

$f:P_{\mathrm{TMS}}\to E_{\mathrm{TMS}}$;

$E_{\mathrm{TMS}}=\{\ e\mid e\in\mathrm{TMS}\ \}$;

$e=f\ (x,\ y)=(x^2-x)/2+y$.

(3) TMS中元素大小到元素位置的映射。

证明：TMS中，任意元素e的行号

$$x=\left\lceil\frac{1+\sqrt{1+8e}}{2}-1\right\rceil \tag{1}$$

不失一般性，证明第$n-1$行中任意元素$e_{(n-1)}$的行号

$$x=\left\lceil\frac{1+\sqrt{1+8e_{(n-1)}}}{2}-1\right\rceil \tag{2}$$

令$f(e)=\dfrac{1+\sqrt{1+8e}}{2}-1$，则$x=\lceil f(e)\rceil$。

$e_{(n-1,n-1)}=c^{n-1}+(n-1)=\dfrac{1}{2}[(n-1)^2+(n-1)]$，则有

$$\begin{aligned} f(e_{(n-1,n-1)}) &= \frac{1+\sqrt{1+4[(n-1)^2+(n-1)]}}{2}-1 \\ &= n-1 \end{aligned} \tag{3}$$

$e_{(n-2,n-2)}=c^{n-2}+(n-2)=\dfrac{1}{2}[(n-2)^2+(n-2)]$

则有

$$\begin{aligned} f(e_{(n-2,n-2)}) &= \frac{1+\sqrt{1+4[(n-2)^2+(n-2)]}}{2}-1 \\ &= n-2 \end{aligned} \tag{4}$$

令$v=1+8e$，则

$$f(v)=\frac{1+\sqrt{v}}{2}-1=\frac{1}{2}v^{\frac{1}{2}}-\frac{1}{2} \tag{5}$$

∵ 函数$f(v)$为单调增加的幂函数，函数$v=1+8e$为单调增加的线性函数，

∴ 函数$f(e)=\dfrac{1+\sqrt{1+8e}}{2}-1$为单调增函数。

∵ $e_{(n-2,\ n-2)}<e_{(n-1)}\le e_{(n-1,\ n-1)}$，

∴ $f(e_{(n-2,\ n-2)})<f(e_{(n-1)})\le f(e_{(n-1,\ n-1)})$，

$n-2<f(e_{(n-1)})\le n-1$，

∴ $\lceil f(e_{(n-1)})\rceil=n-1$。

由以上推导可得：$e_{(n-1)}$的行号满足式(2)。

由映射$f:P_{\mathrm{TMS}}\to E_{\mathrm{TMS}}$及以上推导可知，TMS中元素大小到元素位置满足以下映射：

$f:E_{\mathrm{TMS}}\to P_{\mathrm{TMS}}$;

	1	2	3	4	⋯	$n-3$	$n-2$	$n-1$	
1	1								$c^1=(1^2-1)/2$
2	2	3							$c^2=(2^2-2)/2$
3	4	5	6						$c^3=(3^2-3)/2$
4	7	8	9	10					$c^4=(4^2-4)/2$
⋮	⋮	⋮	⋮		⋱				⋮
$n-3$	$c^{n-3}+1$	$c^{n-3}+2$	$c^{n-3}+3$	⋯	$c^{n-3}+(n-3)$				$c^{n-3}=[(n-3)^2-(n-3)]/2$
$n-2$	$c^{n-2}+1$	$c^{n-2}+2$	$c^{n-2}+3$	⋯	$c^{n-2}+(n-3)$	$c^{n-2}+(n-2)$			$c^{n-2}=[(n-2)^2-(n-2)]/2$
$n-1$	$c^{n-1}+1$	$c^{n-1}+2$	$c^{n-1}+3$	⋯	$c^{n-1}+(n-3)$	$c^{n-1}+(n-2)$	$c^{n-1}+1(n-1)$		$c^{n-1}=[(n-1)^2-(n-1)]/2$

图 2 初始中心对编号三角阵TMS

$$(x,y)=f(e)=\begin{cases}x=\left\lceil\dfrac{1+\sqrt{1+8e}}{2}-1\right\rceil\\ y=e-\dfrac{1}{2}(x^2-x)\end{cases}$$

2.4 初始中心对编号三角阵到初始中心索引对集合的映射

对上述映射进行整理汇总，可用图3表示出从TMS到PIIC的整个映射过程。

由映射f_1至f_4，可得从TMS到PIIC的映射，推导为

设$(i,j)\in$PIIC, $e\in E_{\text{TMS}}$，则有

$$i=x'=n-x=n-\left\lceil\frac{1+\sqrt{1+8e}}{2}-1\right\rceil \tag{6}$$

$$\begin{aligned}j&=x'+y'=(n-x)+y=(n-x)+\left[e-\frac{1}{2}(x^2-x)\right]\\&=n+e-\frac{1}{2}\left(\left\lceil\frac{1+\sqrt{1+8e}}{2}-1\right\rceil^2+\left\lceil\frac{1+\sqrt{1+8e}}{2}-1\right\rceil\right)\end{aligned} \tag{7}$$

由此可得

$f: E_{\text{TMS}}\rightarrow$PIIC;

$$(i,j)=f(e)=\begin{cases}i=n-\left\lceil\dfrac{1+\sqrt{1+8e}}{2}-1\right\rceil\\ j=n+e-\dfrac{1}{2}\left(\left\lceil\dfrac{1+\sqrt{1+8e}}{2}-1\right\rceil^2\right.\\ \qquad\left.+\left\lceil\dfrac{1+\sqrt{1+8e}}{2}-1\right\rceil\right)\end{cases}$$

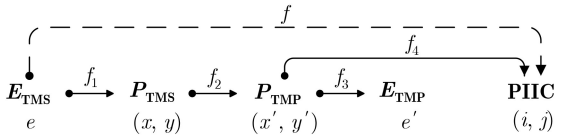

图 3 数据集映射过程汇总

2.5 基于随机数三角阵映射的初始中心生成算法

由映射$f: E_{\text{TMS}}\rightarrow$PIIC可得基于随机数三角阵映射的初始中心生成算法，其步骤如图4所示。

在图4所示的算法步骤中：n为待分割簇所含样本数；m为初始中心对生成数量；randint([1, N], m)表示从[1, N]中“不放回”抽样m个整数；集合PIIC*保存生成好的初始中心索引对。

3 随机数三角阵映射算法的复杂度分析

目前共讨论了4种初始中心随机生成算法，其中随机选取生成的初始中心对可能有重复；随机选取+单点排除会漏掉从未尝试过的初始中心对；随机选取+碰撞检测(简称随机样本碰撞检测)和随机数三角阵映射算法均不存在上述两种缺陷，故只对这两种算法进行复杂度分析。

(1) 时间复杂度：随机数三角阵映射算法的步骤、运算与时耗情况如图4所示。其中，T(·)为运算和操作的计时函数；RS_N为从N个数中不放回抽样操作；MA为内存访问操作。

根据图4所示，统计算法各步骤中所含运算和操作的时间，可知算法所需时耗约为

$$\begin{aligned}\text{T}(A)\approx&m\text{T}(\text{RS}_N)+m[\text{T}(\surd)+4\text{T}(\times)+7\text{T}(+)\\&+\text{T}(\sqcap)+\text{T}(\text{MA})]+2\text{T}(\times)+\text{T}(+)\\\approx&[\text{T}(\text{RS}_N)+\text{T}(\surd)+4\text{T}(\times)\\&+7\text{T}(+)+\text{T}(\sqcap)]m\end{aligned}$$

由此可得算法时间复杂度为$O(\text{T}(A))=O(m)$。

(2) 空间复杂度：算法需维护两个数据结构R和PIIC*: R为随机整数的集合，可包含m个整数，故其所需存储空间为$\text{M}(R)=m\text{M}(\text{Int})$(M(·)为存储空间占用量统计函数；Int为整型数据)；PIIC*

步骤	运算与时耗	
(1) N=n(−1)/2	×,−,÷: 2T(×)+T(+)	
(2) R=randint([1, N], m)	m×RS_N: mT(RS_N)	
(3) For r in R:		
(4) $x=\left\lceil\dfrac{1+\sqrt{1+8r}}{2}-1\right\rceil$	⊓,+, √,+ ,× ,÷ ,−: T(√)+2T(×)+3T(+)+T(⊓)	T(√)+4T(×)+7T(+)+T(⊓)+T(MA)
(5) $y=r-\dfrac{1}{2}(x^2-x)$	−,×: 2T(×)+2T(+)	
(6) $i=n-x$	−: T(+)	
(7) $j=i+y$	+: T(+)	
(8) PIIC*.add((i, j))	MA: T(MA)	

图 4 随机数三角阵映射算法的步骤、运算与时耗分析

为随机初始中心索引对集合，最多包含m个整数二元组，故其所需存储空间为M(PIIC*)=2mM(Int)。

故此，算法所需存储空间约为$\mathrm{M}(A)\approx 3m$M(Int)，空间复杂度为$O(\mathrm{M}(A))=O(m)$。

4 随机样本碰撞检测算法的复杂度分析

(1) 时间复杂度：随机样本碰撞检测算法的步骤、运算与时耗情况如图5所示。

假设在生成m个初始中心对的过程中，共发生了K次碰撞，则算法各层循环次数为

第1层循环总次数：sum(|L1|)=m；

第2层循环总次数：sum(|L2|)=K+m；

第3层平均循环总次数：sum(|L3|)=m(m–1)/2+(m+1)K/3。

据此统计图5中算法各步骤操作所需时间，可得K次碰撞情况下算法的近似时耗为

$$\begin{aligned}\mathrm{T}(A)\approx&\,\mathrm{sum}(|\mathrm{L3}|)\left[\mathrm{T}(\geqslant^{*})+\mathrm{T}(\mathrm{MA})\right]\\&+\mathrm{sum}(|\mathrm{L2}|)\left[2\mathrm{T}(\mathrm{RS}_n)+\mathrm{T}(\mathrm{MA})+\mathrm{T}(\geqslant)\right]\\&+\mathrm{sum}(|\mathrm{L1}|)\,\mathrm{T}(\mathrm{MA})\\\approx&\left(1.125m^2+0.75mK\right)\mathrm{T}(\geqslant)\\&+2(K+m)\mathrm{T}(\mathrm{RS_n})\end{aligned}$$

由此可得算法时间复杂度为$O(A)$=O(T(A))=$O(m^2+mK)$。

(2) 空间复杂度：算法在执行时，需维护一个数据结构PIIC*。故此，算法所需存储空间约为M(A)≈2mM(Int)，空间复杂度为：O(M(A))=O(m)。

总结以上分析结论，可得算法复杂度对比情况如表1所示。

其中，O_{b}, O_{w}, O_{a}分别表示最优、最差、平均复杂度；N=|PIIC|。

5 实验

实验所用评估指标有平均时耗和时耗标准差，即测定算法完成指定初始中心生成任务的运行时间，然后统计多次相同实验所需时耗的平均值及标准差。其中平均时耗用于评估算法的时间效率，时耗标准差用于评估算法的时间效率稳定性。实验用计算机基本配置如下：CPU为Intel core i7 3 GHz；内存为8 GB；操作系统为Windows 7；算法实现语言为Python 2.7。实验分为两部分：第1部分仿真实验，用于验证本文算法时间复杂度理论分析的正确性；第2部分高维数据集实验，用于验证本文算法对高维大数据处理的适用性。

5.1 仿真实验

影响随机数三角阵映射算法性能的因素有两个：数据集所含样本数、初始中心对生成数。故验证该算法的时间复杂度分析结论，只需仿真实验即可。实验举例：当数据集样本数n=4时，因可用初始中心对总数N=6，则依次统计算法生成1～6个初始中心对时的各项评估指标。下文将随机样本碰撞检测算法称为算法1(简写为A1)，将随机数三角阵映射算法称为算法2(简写为A2)。

5.1.1 时间效率实验

(1)实验结果：当3≤n≤10时，对算法1和算法2的平均时耗进行测试，其实验结果如图6和图7所示。

	步骤	运算与时耗
	(1) For i in range(m) :	
	(2) Repeat	
	(3) new_pair = randint([1, n], 2)	RS_n: 2T(RS_n)
	(4) collision=False	MA: T(MA)
L1 L2	(5) For pair in PIIC*:	
L3	(6) If new_pair = = pair :	==: T(⩾*)
	(7) collision=True	MA: T(MA)
	(8) Break	
	(9) Until not collision	≠: T(⩾)
	(10) PIIC*.add(new_pair)	MA: T(MA)

图 5 随机样本碰撞检测算法的步骤、运算与时耗分析

表 1 算法复杂度对比

算法	时间复杂度	空间复杂度
随机数三角阵映射算法	$O(m)$	$O(m)$
随机样本碰撞检测算法	$O(m^2+mK)$ $O_{\mathrm{b}}(m^2)$; $O_{\mathrm{w}}(\infty)$; $O_{\mathrm{a}}(mN\cdot\ln N)$	$O(m)$

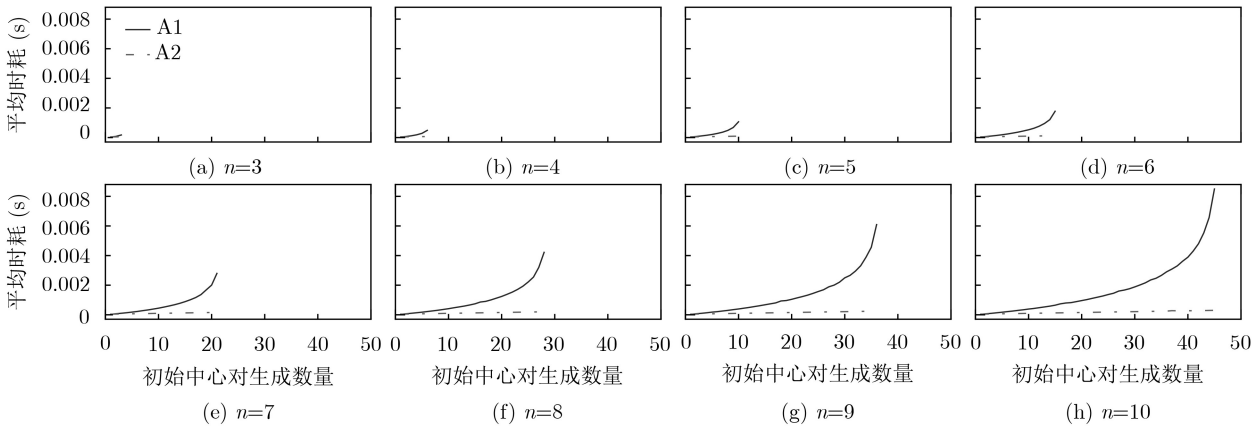

图 6 算法平均时耗分步对比

在图6中，将n取不同值时的平均时耗分开绘制，以便于观察当数据集容量相同而初始中心生成数量不同时的算法时耗对比；图7中，将n取不同值时的平均时耗变化情况绘制在一起，以便于观察当数据集容量不同而初始中心生成数量相同时的算法时耗对比。

(2) 实验结果分析：观察图6和图7可知：随着初始中心对生成数量的增多，算法1的平均时耗加速增长，算法2的平均时耗约呈线性增长；待分割簇的样本数越多，算法1生成相同数量初始中心对的平均时耗越小。以上实验结果与算法时间复杂度分析结论相一致。

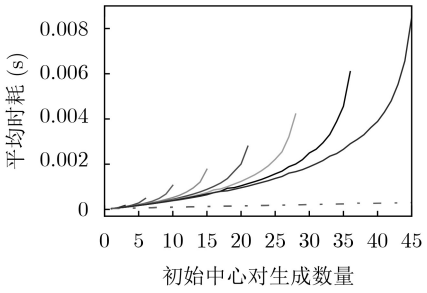

图 7 算法平均时耗叠加对比

5.1.2 时间效率稳定性实验

(1) 实验结果：当3≤n≤10时，对算法1和算法2的时耗标准差进行测试，其实验结果如图8和图9所示。

(2) 实验结果分析：观察图8和图9可知：随着初始中心对生成数量的增多，算法1的时耗标准差随之总体增大，算法2的时耗标准差基本不变；待分割簇的样本数越多，算法1生成相同数量初始中心对的时耗标准差总体越小。以上实验结果与算法时间复杂度分析结论相一致。

5.2 高维数据集实验

参与对比测试的算法有：本文随机数三角阵映射算法(the algorithm based on Random integer Triangular Matrix Mappings, RTMM)、随机样本碰撞检测算法(the algorithm based on Random Sample Collision Detection, RSCD)、特征域均匀采样算法[18](the algorithm based on Feature Range Uniform Sampling, FRUS；当前最流行的基于样本特征采样的算法)。实验引入3个著名高维数据集：20NEWS[19], IMDB[20], MNIST[21]，用于验证本文算法在高维大数据处理领域的优越性。

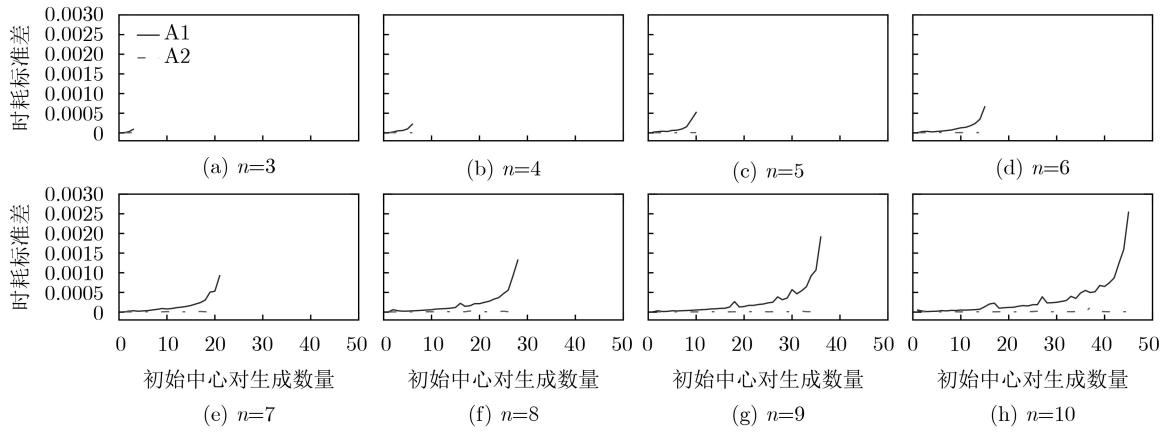

图 8 算法时耗标准差分步对比

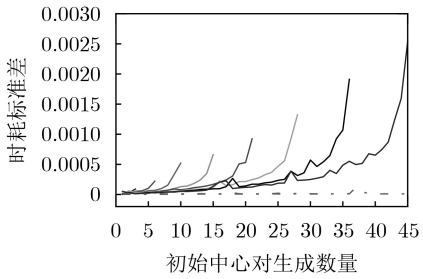

图 9 算法“时耗标准差”叠加对比

其中，20NEWS数据集保存的是网络新闻文本，经数据清洗、特征提取等格式化处理后得到1.8×10^4个样本，每个样本有173451个特征；IMDB数据集保存的是电影评论文本，经处理得到2×10^4个样本、73063个特征；MNIST数据集保存的是手写数字点阵图像，经处理得到1×10^4个样本、784个特征。

(1) 实验结果

测试生成不同数量初始中心对时，算法的运行时耗变化情况，其结果如图10—图12所示。

测试算法在生成大规模初始中心对(1.8×10^5个)时的运行时耗，其结果如表2所示。

(2) 实验结果分析

(a) 数据集维度规模对算法性能的影响：处理20NEWS数据集(约1.7×10^5维)时，RTMM相较于FRUS算法的效率优势最明显；处理IMDB数据集(约7×10^4维)时，效率优势没有在20NEWS数据集上明显；处理MNIST数据集(约7×10^2维)时，两算法的效率基本相当。以上实验结果表明：数据集维度规模对FRUS算法性能的影响显著，而对RTMM和RSCD算法没有影响；数据集维度越高，FRUS算法的效率越低，RTMM算法的效率优势越明显。

(b) 初始中心生成规模对算法性能的影响：随着初始中心生成数量的增加，RSCD算法的运行时耗加速增长，FRUS算法运行时耗约呈线性增长，RTMM算法运行时耗几乎不变。以上实验结果表明：初始中心生成规模对RSCD算法的性能影响最显著，对FRUS算法性能影响次之，对RTMM算法性能影响甚微；初始中心生成数量越多，RSCD和FRUS算法的效率越低，RTMM算法相较于两算法的效率优势越明显。

总结本文实验与分析结果，可得以下结论：FRUS算法更适合于低维数据集上小规模初始中心生成任务；RSCD算法更适合于高维数据集上小规模初始中心生成任务；RTMM算法更适合于高维数据集上大规模初始中心生成任务。

表 2 大规模初始中心生成任务下的算法时耗对比

DataSet	平均时耗(s)		
	RTMM	RSCD	FRUS
20NEWS	15.988381	3304.241926	1398.781894
IMDB	20.109095	3349.822651	567.211075
MNIST	4.805166	3360.242473	6.441524

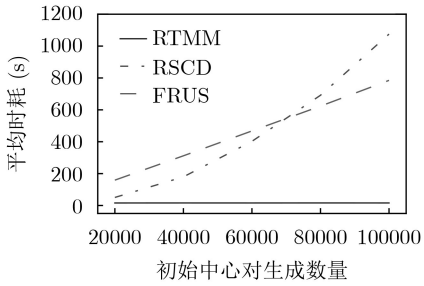

图 10 20NEWS数据集上算法运行时耗

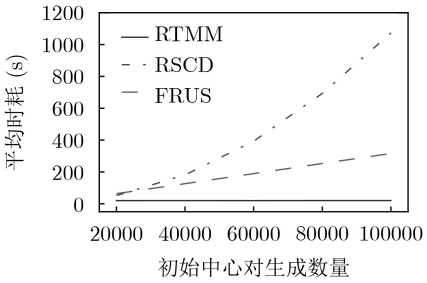

图 11 IMDB数据集上算法运行时耗

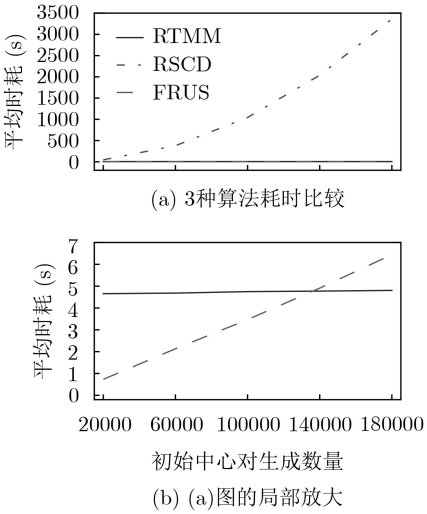

图 12 MNIST数据集上算法运行时耗

6 结束语

本文首先创建出初始中心对组合三角阵和初始中心对编号三角阵，然后通过建立两矩阵中元素及元素位置间的若干映射，从而提出了一种新的二分聚类初始中心生成方法。理论分析与实验结果均表明：随着初始中心对生成数量的增多，新方法的平

均时耗近似于线性增长，且其时耗标准差非常稳定、近似于零。新方法的时间效率及稳定性明显优于常用的随机采样方法，且随着数据集维度规模和初始中心生成规模的增大，其高效性与鲁棒性的优势将更加明显。故此，本文方法特别适用于高维大数据聚类场景。

参 考 文 献

[1] JAIN A K. Data clustering: 50 years beyond K-means[J]. *Pattern Recognition Letters*, 2010, 31(8): 651–666. doi: 10.1016/j.patrec.2009.09.011.

[2] YANG Qiang and WU Xindong. 10 challenging problems in data mining research[J]. *International Journal of Information Technology & Decision Making*, 2006, 5(4): 597–604. doi: 10.1142/s0219622006002258.

[3] ZHAO Wanlei, DENG Chenghao, and NGO C W. K-means: A revisit[J]. *Neurocomputing*, 2018, 291: 195–206. doi: 10.1016/j.neucom.2018.02.072.

[4] KADAM P and MATE G S. Improving efficiency of similarity of document network using bisect K-means[C]. 2017 International Conference on Computing, Communication, Control and Automation, Pune, India, 2017: 1–6. doi: 10.1109/iccubea.2017.8463865.

[5] WEI Zhaolan and XIA Jing. Optimal sensor placement based on bisect k-means clustering algorithm[C]. 2018 3rd International Conference on Materials Science, Machinery and Energy Engineering (MSMEE 2018), Taiyuan, China, 2018: 228–232. doi: 10.23977/msmee.2018.72138.

[6] ABUAIADAH D. Using bisect K-Means clustering technique in the analysis of Arabic documents[J]. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2016, 15(3): 17. doi: 10.1145/2812809.

[7] 王燕, 李晴, 张光普. 长基线/超短基线组合系统抗异常值定位技术研究[J]. 电子与信息学报, 2018, 40(11): 2578–2583. doi: 10.11999/JEIT180056.
WANG Yan, LI Qing, and ZHANG Guangpu. On anti-outlier localization for integrated long baseline/ultra-short baseline systems[J]. *Journal of Electronics & Information Technology*, 2018, 40(11): 2578–2583. doi: 10.11999/JEIT180056.

[8] STEINBACH M, KARYPIS G, and KUMAR V. A comparison of document clustering techniques[C]. KDD Workshop on Text Mining, Boston, USA, 2000: 1–20.

[9] WANG Yong and HODGES J E. A comparison of document clustering algorithms[C]. The 5th International Workshop on Pattern Recognition in Information Systems, Miami, USA, 2005: 186–191. doi: 10.5220/0002557501860191.

[10] BAGIROV A M, UGON J, and WEBB D. Fast modified global *k*-means algorithm for incremental cluster construction[J]. *Pattern Recognition*, 2011, 4: 866–876. doi: 10.1016/j.patcog.2010.10.018.

[11] JAIN A K, MURTY M N, and FLYNN P J. Data clustering: A review[J]. *ACM Computing Surveys*, 1999, 31(3): 264–323. doi: 10.1145/331499.331504.

[12] 赵凤, 孙文静, 刘汉强, 等. 基于近邻搜索花授粉优化的直觉模糊聚类图像分割[J]. 电子与信息学报, 2020, 42(4): 1005–1012. doi: 10.11999/JEIT190428.
ZHAO Feng, SUN Wenjing, LIU Hanqiang, *et al*. Intuitionistic fuzzy clustering image segmentation based on flower pollination optimization with nearest neighbor searching[J]. *Journal of Electronics & Information Technology*, 2020, 42(4): 1005–1012. doi: 10.11999/JEIT190428.

[13] WU Xindong, KUMAR V, QUINLAN J R, *et al*. Top 10 algorithms in data mining[J]. *Knowledge and Information Systems*, 2008, 14(1): 1–37. doi: 10.1007/s10115-007-0114-2.

[14] WITTEN I H, FRANK E, HALL M A, *et al*. Data Mining: Practical Machine Learning Tools and Techniques[M]. 4th ed. Amsterdam: Elsevier, 2017: 97–98.

[15] MARSLAND S. Machine Learning: An Algorithmic Perspective[M]. 2nd ed. Boca Raton: CRC Press, 2015: 197–200.

[16] HAN Jiawei and KAMBER M. Data Mining: Concepts and Techniques[M]. 2nd ed. Amsterdam: Elsevier, 2006: 402–404.

[17] ELKAN C. Clustering with *k*-means: Faster, smarter, cheaper[EB/OL]. http://www.doc88.com/p-347627347988.html, 2004.

[18] KOPEC D. Classic Computer Science Problems in Python[M]. Shelter Island: Manning Publications, 2019: 117–118.

[19] JREN N. The 20 newsgroups data set[EB/OL]. http://qwone.com/~jason/20Newsgroups, 2008.

[20] BO P and LILLIAN L. Movie review data[EB/OL]. http://www.cs.cornell.edu/people/pabo/movie-review-data, 2020.

[21] LECUN Y, CORTES C, and BURGES C J C. The MNIST database of handwritten digits[EB/OL]. http://yann.lecun.com/exdb/mnist, 2020.

李　旻：男，1976年生，副教授，主要研究方向为数据挖掘、自然语言处理、教育信息技术等.

何婷婷：女，1964年生，教授，主要研究方向为网络媒体监测、自然语言处理、教育信息技术等.

责任编辑：马秀强