基于三阶张量的大规模数据谱聚类集成算法

仵匀政[1]，杜韬[1,2]，周劲[1,2]，陈迪[1]，王心耕[1]
1. 济南大学信息科学与工程学院，山东 济南 250024；
2. 山东省网络环境智能计算技术重点实验室，山东 济南 250024

摘要
为了降低大规模数据谱聚类计算负担，进一步提高聚类的准确性和鲁棒性，提出了一种基于三阶张量的大规模数据谱聚类集成算法。首先，提出一种混合代表最近邻近似方法构造数据间的稀疏亲和子矩阵；然后将稀疏亲和子矩阵表示为二部图，通过图分割的方法得到初步聚类结果；最后，提出三阶张量集成方法，将多个聚类结果进行融合，得到最终的聚类结果。在大规模的真实数据集和合成数据集上验证，相较经典的谱聚类算法、聚类集成算法以及近年来对其改进的算法，该算法表现出更优异的性能。

关键词
数据聚类；大规模数据；谱聚类；三阶张量；聚类集成

中图分类号：TP301 文献标志码：A doi: 10.11959/j.issn.2096-0271.2024007

Spectral clustering ensemble algorithm based on three-order tensor for large-scale data

WU Yunzheng[1], DU Tao[1,2], ZHOU Jin[1,2], CHEN Di[1], WANG Xingeng[1]
1. College of Information Science and Engineering, University of Jinan, Jinan 250024, China
2. Shandong Provincial Key Laboratory of Network Based Intelligent Computing, Jinan 250024, China

Abstract
In order to reduce the computational burden of large-scale data spectral clustering and further improve the clustering accuracy and robustness, the spectral clustering ensemble algorithm based on the three-order tensor for large-scale data was proposed. The sparse affinity sub-matrix was first constructed by the mixed representative nearest neighbor approximation method. The sparse affinity sub-matrix was then represented as a bipartite graph. The preliminary clustering results were obtained by Graph Segmentation. Finally, an unified clustering result was obtained by fusing multiple clustering results through the three-order tensor ensemble method. On the real datasets and the synthetic datasets, the proposed algorithm showed a better performance compared to the classical spectral clustering algorithm, the clustering ensemble algorithm, and the improved algorithms in recent years.

Key words
data clustering, large-scale data, spectral clustering, three-order tensor, clustering ensemble

0 引言

聚类作为一种无监督学习方法，只需基于数据本身的特征计算出数据对象间的相似度，即可实现数据的划分[1]。通过聚类处理后，同一簇内的数据应尽可能相似，不同簇间的数据应尽可能相异。目前，聚类算法的研究方向包括基于划分的聚类[2-3]、基于密度的聚类[4-5]、基于网格的聚类[6-7]、基于模型的聚类[8-9]、深度聚类等[10-11]。在大量聚类算法中，谱聚类因其处理非线性可分离数据集的良好能力而受到更多关注[12-15]。然而，传统谱聚类的时间和空间复杂度大，这极大地限制了其在大规模问题中的应用。稀疏亲和子矩阵是现有减少谱聚类方法巨大计算负担的最常用策略。它采用少量数据代表整个数据集，计算负担虽大大降低，但其准确性和鲁棒性普遍不高。为了在降低大规模数据谱聚类计算负担的基础上，进一步提高谱聚类的准确性和鲁棒性，本文提出了一种基于三阶张量的大规模数据谱聚类集成算法（spectral clustering ensemble based on three-order tensor, SCETT）。

首先，针对大规模数据对应的亲和矩阵难以有效分割、难以得到一个效果较好的划分结果的问题，本文提出混合代表最近邻近似方法，首先采用随机和密度峰值聚类算法（density peak clustering algorithm, DPC）[16]相结合的混合代表选择策略来选择p个代表点，然后通过最近邻近似原则选择样本点的K个最近代表，并且将其他代表设置为零以进一步稀疏子矩阵。通过混合代表最近邻近似方法快速有效地构建数据间的稀疏亲和子矩阵，然后将稀疏亲和子矩阵作为交叉亲和矩阵，在数据集和代表集之间构建二部图，通过图分割获得初步聚类结果。最后针对传统聚类集成方法不能有效利用多个基聚类之间的相关性的问题，设计了一种三阶张量集成方法来融合多个基聚类结果，利用张量分解挖掘数据的隐性关系的优势，更全面地挖掘多个基聚类结果之间的互补信息，有效分析基聚类结果之间的关联性，实现数据的准确有效划分。对10个大规模数据集（包括5个人工合成数据集和5个真实数据集）的实验表明，本文提出的SCETT算法在聚类的准确性和鲁棒性方面优于大多数算法。

本文的主要贡献总结如下：

- 提出了混合代表最近邻近似方法，以快速有效地构造数据集和代表集之间的稀疏亲和子矩阵，进而通过图分割方法得出基聚类成员，实现大规模数据的有效划分；
- 将三阶张量的概念引入聚类领域，利用张量分解挖掘数据之间隐性关系的优势，挖掘基聚类之间的互补信息，提高聚类集成的准确性和鲁棒性；
- 提出了基于三阶张量的大数据聚类集成算法，在降低大规模数据谱聚类计算负担的基础上，进一步提高聚类的效果。

1 相关工作

1.1 谱聚类

给定N个对象的数据集，传统的谱聚类[12]首先计算$N\times N$的亲和矩阵。然后，通过计算亲和矩阵的前k个特征值与特征向量，构建特征向量空间。最后，利用k-means或其他经典聚类算法对特征向量空间中的特征向量进行聚类，得到最终的聚类结果。

虽然谱聚类在从复杂数据中发现任意形状的簇方面表现出很好的性能，但其时

间复杂度和空间复杂度极大地限制了其在大规模任务中的应用。为了减少计算量，一些研究人员通过考虑K-最近邻，然后利用一些稀疏特征求解器[12]来求解特征分解问题。然而，这仍然需要计算原始亲和矩阵中的所有元素。

为了避免计算完整的亲和矩阵，采用子矩阵近似代表全亲和矩阵已经成为谱聚类的一个强大而有效的工具。Nystrom算法[13]从数据集中随机选择p个代表，并在N个对象和p个代表之间构建一个$N\times p$的亲和子矩阵。子矩阵构造仅需要$O(Npd)$时间和$O(Np)$内存，远低于全亲和矩阵的构造成本。虽然随机代表选择非常有效，但随机代表的质量通常不稳定。此外，虽然较大的p通常有助于更好地提高聚类稳定性，但子矩阵构造的$O(Np)$内存开销仍然是处理大型数据集的关键瓶颈。为了解决随机代表的潜在不稳定性，Cai等人[14]提出了基于稀疏表示的隐子空间聚类（latent subspace clustering，LSC）算法，该算法首先通过k-means将数据集划分为p个聚类，然后以p个聚类中心为代表。在构造了$N\times p$的亲和子矩阵之后，通过保留每行的k个最近代表并将其他代表置零，进一步使它们变得稀疏。Huang等人[15]扩展了LSC方法，提出了超可伸缩谱聚类算法（ultra-scalable spectral clustering，U-SPEC），该方法结合随机选点和k-means，以p个聚类中心为代表点以提高准确率。

1.2 聚类集成

聚类集成是近年来广泛使用的一种技术。它旨在将多个聚类结果组合成更好、更健壮的一致聚类。现有的聚类集成算法主要包括超图、有限混合模型、证据累积、协方差矩阵和投票。

Fred等人[17]提出了基于证据累积的聚类集成方法（evidence accumulation clustering，EAC），该方法通过计算数据点对被划分到同一个集群中的次数，得到协关联矩阵，然后对协关联矩阵进行层次聚类，得到最终的聚类划分结果。该方法可以识别任意形状和大小的簇，但是它计算所需内存较大，不适合大规模数据集。Hore等人[18]用聚类质心向量来表示聚类成员，降低了算法的时间和空间复杂度，使聚类集成适应数据量大的数据集。Yi等人[19]提出了一种基于矩阵补全技术的聚类集成方法（ensemble clustering by matrix completion，ECMC）。该方法在过滤掉关系不可靠的数据后，对用可靠数据构建的部分观测值的相似矩阵，利用矩阵补全算法获得协方差矩阵，并用谱聚类算法进行聚类，解决了当不确定数据对数量较大时，不可靠数据对中的信息误导聚类算法得出错误的聚类结果的问题。罗晓慧等人[20]提出了一种新的选择性聚类集成方法，利用改进的k-means方法生成锚点，基于锚点的方法生成相似度矩阵，以获得不同维度下的低维流形嵌入；权重衡量标准采用归一化互信息（normalized mutual information，NMI），排除权重较低的划分结果；最后，通过基于权重的选择性投票方法，融合多个不同维度的相似低维流形，得到最终的数据分析结果。

1.3 三阶张量

张量理论是数学的一个分支，在力学中有重要的应用。张量这个术语起源于力学，最初用于表示弹性介质中各点的应力状态。张量理论随后发展为力学和物理学中强有力的数学工具。张量是标量和向量向更高维度的推广。它将一系列具有某些共同特征的数有序地组合起来，表示

一个更一般化的“数”。一阶张量可以定义为向量，二阶张量可以定义为$N\times N$矩阵，三阶张量可以定义为立方体。一个由4个8×2矩阵组成的8×2×4三阶张量样本如**图1**所示。张量分解主要是矩阵分解的高阶推广。后者主要用于隐含关系挖掘、降维处理、缺失数据填充(或稀疏数据填充)，因此张量分解也可以满足这些用途[21]。

2 SCETT算法

SCETT算法依据流程可分为三部分：①提出混合代表最近邻近似(mixed representation nearest neighbor approximation，MRNNA)方法用于快速有效地构造数据集和代表集之间的稀疏亲和子矩阵；②将稀疏亲和子矩阵表示为二部图，通过二部图分割(bipartite graph segmentation，BGS)获得基聚类结果；③提出了一种新的三阶张量集成(three-order tensor ensemble，TTE)方法来融合多个基聚类结果，从而获得更准确的聚类结果。算法的流程如**图2**所示。

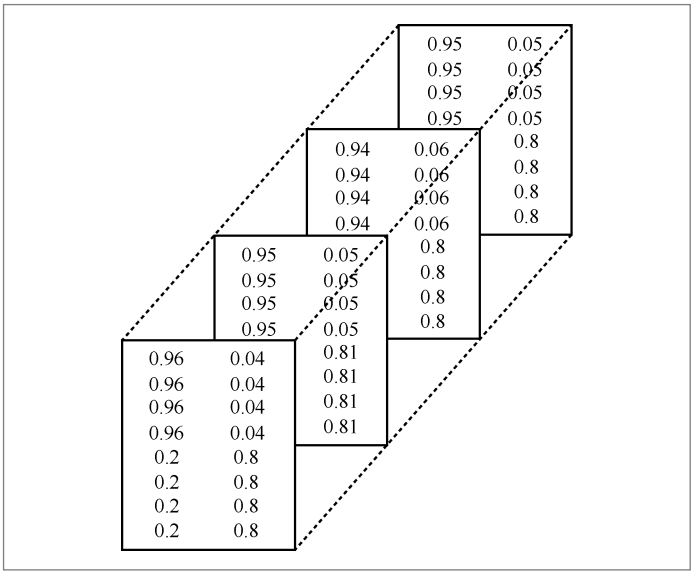

图1 三阶张量样本

2.1 混合代表最近邻近似

对于量大、维度高的数据集，如何快速有效地构造数据之间的亲和矩阵，进而得到多个基聚类结果一直是研究重点。针对这一问题，本文提出了混合代表最近邻近似方法来有效地构造数据集和代表集之间的稀疏亲和子矩阵，然后将稀疏亲和子矩阵作为交叉亲和矩阵，在数据集和代表集之间构建二部图，通过图分割得到初步聚类结果。

混合代表最近邻近似方法首先采用随机和DPC算法相结合的混合代表选择策略从所有数据点中选择出p个代表点，然后通过最近邻近似原则从p个代表点选择出每个数据点的K个最近代表，并且将其他代表设置为零以稀疏亲和子矩阵，可以有效地构造数据集和代表集之间的稀疏亲和子矩阵。

设$X=\{x_1,x_2,\cdots,x_N\}$表示一个包含N个对象的数据集，其中$x_i\in R^d$是第i个对象，d是维数。首先，从数据集中随机选择p'个候选代表，以减少时间成本($p'<<N$)；然后，对于p'个候选点，使用DPC算法获得p个聚类中心点，并使用聚类中心点作为代表点。DPC算法[16]根据ρ_i和δ_i确定聚类中心，ρ_i为数据点的局部密度，δ_i为从数据点到局部密度大于它的最近数据点的距离。

$$\rho_i=\sum_{j\neq i}\chi(d_{ij}-d_c) \tag{1}$$

$$\delta_i=\min_{j:\rho_j>\rho_i}(d_{ij}) \tag{2}$$

$$\gamma_i=\rho_i\cdot\delta_i \tag{3}$$

其中，d_{ij}为数据点i与数据点j之间的距离，d_c为截断距离，χ为逻辑判断函数。得到每个数据点的局部密度ρ和上级点距离δ之后，将二者相乘得到γ值，点的γ值越高，被选取为聚类中心的概率越大。以ρ为横坐标，以δ为纵坐标，构建了Pen-Digits数据集的决策图，如**图3**所示。设定一个参数c，在得到所有数据点的γ值后，选取γ值前c%的点作为聚类中心点，c的取值范围一般为1%~5%。

在形式上，将选定的代表的集合表示为：

$$R=\{r_1,r_2,\cdots,r_p\} \tag{4}$$

在获得p个代表后，下一个目标是对亲和子矩阵进行稀疏化。特别地采用了由粗到细的最近代表快速近似方法。最近代表快速近似的主要思想是，首先在代表点中找到数据点最近的代表（表示为r），然后在r的邻近区域中找到K个最近代表（$K<p$）。

每个样本点x_i的K个最近代表通过以下3个步骤找到：第一步，调整DPC的参数c进一步过滤代表点；第二步，在过滤的代表点内找到x_i最近的代表r；第三步，在r的K'个最近邻中($K'>K$)，找出x_i的K个最近代表。通过调整DPC的参数c进一步过滤代表点，可以使用一次计算来执行两次选择，这降低了算法的计算负担。关于最近代表快速近似方法的更多细节如**图4**所示。

通过得到每个对象的K个最近代表，可以构造$N\times p$的稀疏亲和子矩阵。本文采用高斯核作为相似度核。稀疏亲和子矩阵可以表示为：

$$\boldsymbol{B}=\{b_{i,j}\}_{N\times p} \tag{5}$$

$$b_{ij}=\begin{cases}\exp\left(-\dfrac{\|x_i-r_j\|^2}{2\sigma^2}\right), & r_j\in N_K(x_i)\\ 0, \text{ 其他}\end{cases} \tag{6}$$

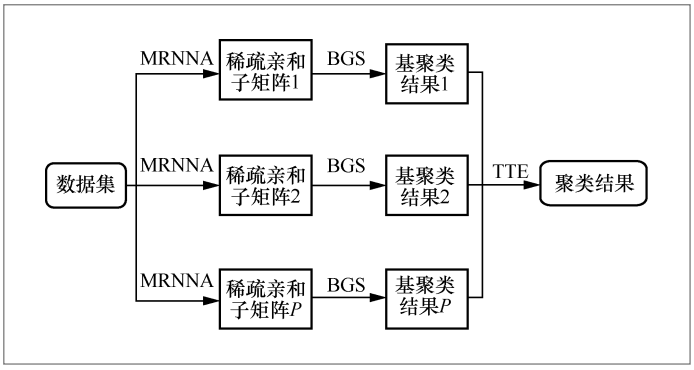

图 2 SCETT 算法的流程

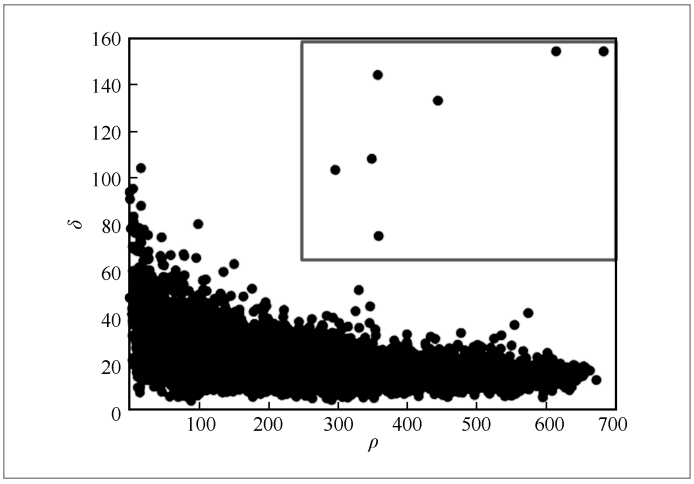

图 3 DPC 选择代表点

其中，$N_K(x_i)$表示x_i的K个最近代表的集合，核参数σ为样本点和它的K个最近代表之间的平均欧氏距离。$\boldsymbol{B}$是一个稀疏矩阵，它只包含$N\times K$个非零项。混合代表最近邻近似方法的具体流程如算法1所示。

算法1 （混合代表最近邻近似）

输入：数据集X，DPC参数c

输出：稀疏亲和子矩阵

Step 1：从数据集中随机选择p'个点；

Step 2：对p'个点进行DPC聚类获得p个代表点；

Step 3：调整DPC参数c，进一步过滤代表点；

Step 4：计算每个样本点与过滤后的

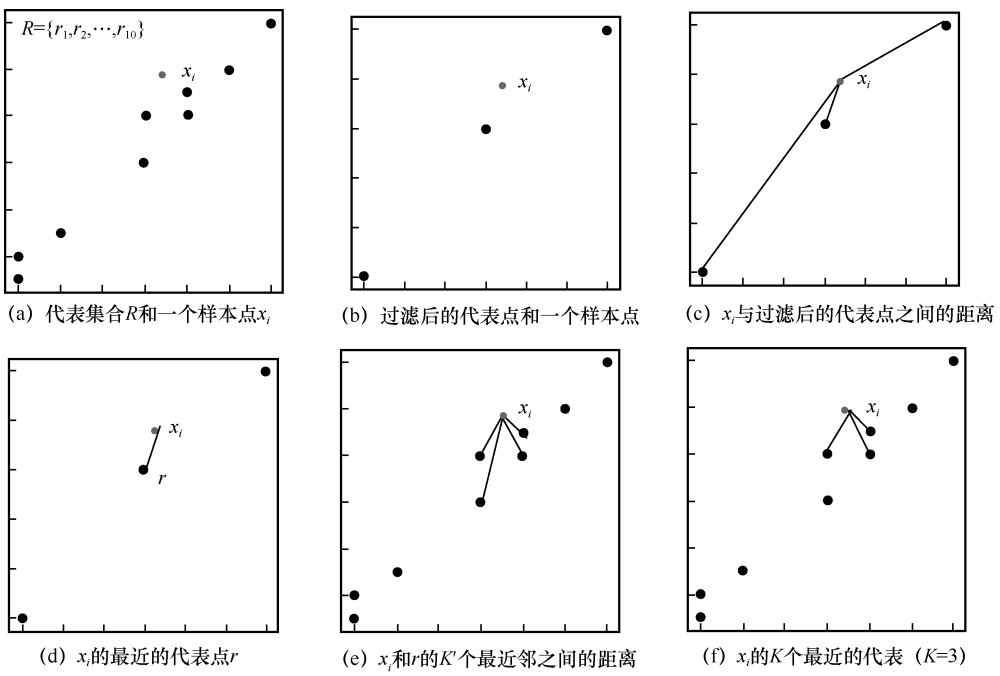

图 4 最近代表快速近似方法

代表点之间的距离；

Step 5：找到每个样本点的最近的代表点；

Step 6：计算p个代表点之间的距离；

Step 7：找到最近的代表点的K'个最近邻；

Step 8：计算样本点与K'个最近邻之间的距离；

Step 9：找出每个样本点的最近的K个代表点；

Step 10：构造稀疏亲和子矩阵。

2.2 二部图分割

稀疏亲和子矩阵$\boldsymbol{B}$反映了X中的数据点与$\boldsymbol{R}$中的代表点之间的关系，$\boldsymbol{B}$是交叉亲和矩阵，可以自然地解释为二部图$G=\{X,R,\boldsymbol{B}\}$，其中$X\cup R$是节点集，$\boldsymbol{B}$是交叉亲和矩阵。利用二部图的结构，可以有效地对图进行划分，得到最终的聚类结果。

首先，如果把图G看作一个具有$N+p$个节点的一般图，那么它的全亲和矩阵可以记为：

$$\boldsymbol{E}=\begin{bmatrix}0 & \boldsymbol{B}^{\mathrm{T}}\\ \boldsymbol{B} & 0\end{bmatrix} \tag{7}$$

聚类试图通过求解以下广义特征问题来对图进行划分：

$$\boldsymbol{L}\boldsymbol{u}=\gamma\boldsymbol{D}\boldsymbol{u} \tag{8}$$

其中，$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{E}$是图$G$的拉普拉斯算子，而$\boldsymbol{D}=R^{(N+p)\times(N+p)}$是度矩阵。将$G$作为一般图，需要$\mathrm{O}((N+p)^3)$时间来解决特征问题（8），它在大规模的数据集上不可行。

因此，采用转移切割[22]来减少特征问题求解时间，求解图G（有$N+p$个节点）的特征问题转移到求解一个更小的图G_R（有p个节点）上的特征问题。具体来

说，图$G_R=\{R, E_R\}$，其中R是代表点集，$\boldsymbol{E}_R=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}_X^{-1}\boldsymbol{B}$是亲和矩阵，$\boldsymbol{D}_X\in R^{N\times N}$是对角矩阵，对角线上的数是$B$对应每行数目的总和。$L_R=D_R-E_R$是拉普拉斯算子，$\boldsymbol{D}_R\in R^{p\times p}$是$\boldsymbol{G}_R$的度矩阵。然后，图$G_R$上的广义特征问题可以表示为：

$$L_R v=\lambda D_R v \tag{9}$$

设特征问题（9）的前k个特征对表示为$\{(\lambda_i,v_i)\}_{i=1}^{k}$ $(0=\lambda_1\leqslant\lambda_2\cdots\leqslant\lambda_k<1)$，特征问题（8）的前$k$个特征对表示为$\{(\gamma_i,u_i)\}_{i=1}^{k}$ $(0=\gamma_1\leqslant\gamma_2\cdots\leqslant\gamma_k<1)$。特征值和特征向量传递计算式为：

$$\gamma_i(2-\gamma_i)=\lambda_i \tag{10}$$

$$u_i=\begin{bmatrix}h_i\\ v_i\end{bmatrix} \tag{11}$$

$$h_i=\frac{1}{1-\gamma_i}\boldsymbol{T}v_i \tag{12}$$

其中，$\boldsymbol{T}=\boldsymbol{D}_x^{-1}\boldsymbol{B}$为转移矩阵。在求解特征问题后，将得到的$k$个特征向量堆叠成一个$(N+p)\times k$矩阵。将矩阵的每一行作为一个新的特征向量，利用对应于N个原始对象的N行，在此基础上进行k-means聚类，得到基聚类结果。

2.3 三阶张量集成

在得到多个基聚类结果的基础上，本文提出了一种基于三阶张量的集成算法，将多个基聚类结果集成到一个统一的集成聚类框架中，以在保持高效率的同时进一步提高聚类的准确性和鲁棒性。三阶张量集成主要利用张量分解的优势来挖掘基聚类结果之间的互补信息，有效分析基聚类结果之间的关联性，从而获得更好、更鲁棒的一致聚类。

每个基聚类都由一定数量的簇组成。将每个基聚类中的簇表示为：

$$C=\{C_1,C_2,\cdots,C_k\} \tag{13}$$

其中，C_i是基聚类中的第i个簇，k是基聚类中的簇数。基聚类的结果可以表示为：

$$W=\{w_{ij}\}_{N\times k} \tag{14}$$

$$w_{ij}=\begin{cases}1, & x_i\in C_j\\ 0, & \text{其他}\end{cases} \tag{15}$$

$\boldsymbol{W}$是一个稀疏矩阵，它只包含N个非零的项。三阶张量集成的具体流程如算法2所示。

算法2 （三阶张量集成）

输入：P个基聚类结果

输出：聚类结果

Step 1：构造三阶张量；

Step 2：对三阶张量进行矩阵展开；

Step 3：对每个展开矩阵进行奇异值分解；

Step 4：构造核心张量；

Step 5：重构张量。

步骤1：张量y的初始构造。可以得到一个初始的三阶张量$y=\phi(y_{ik,1},y_{ik,2},\cdots,y_{ik,P})$，其中$\phi(\cdot)$通过叠加$P$个基聚类结果来构造张量$y\in R^{N\times K\times P}$。

步骤2：张量y的矩阵展开。通过将y对应的单位元素排列为$Y^{(t)}$（t=1、2、3）的列，张量y可以展开为一个矩阵。初始张量y可以折叠到它的3个维度模式：$Y^{(1)}\in R^{N\times KP}$，$Y^{(2)}\in R^{K\times NP}$，$Y^{(3)}\in R^{NK\times P}$。

步骤3：对每个展开矩阵进行奇异值分解（SVD）。3个展开矩阵$Y^{(t)}$的SVD分解：

$$Y^{(t)}=A^{(t)}\sum{}^{(t)}(B^{(t)})^{\mathrm{T}} \tag{16}$$

为了更好地显示不同基聚类结果之间的潜在关联，必须减少左奇异矩阵$\boldsymbol{A}^{(t)}$的

维数。基于$\Sigma^{(t)}$中的奇异值，保持每个对应矩阵$A^{(t)}$中的主导e_t左奇异向量，所得到的矩阵是$A_{e_t}^{(t)}$。参数e_t的值通常是通过$\Sigma^{(t)}$中的知识来选择的。下面通过重构张量来验证选择主导左奇异向量是合理的。

步骤4：核心张量G的构造。3种不同模态之间的相关性可以由核心张量G来主导：

$$G = y \times_1 (A_{e_1}^{(1)})^{\mathrm{T}} \times_2 (A_{e_2}^{(2)})^{\mathrm{T}} \times_3 (A_{e_3}^{(3)})^{\mathrm{T}} \quad (17)$$

其中y是初始张量，$\times_t$表示三阶张量的模乘法，$(A_{e_1}^{(1)})^{\mathrm{T}}$是$A_{e_1}^{(1)}$的转置，$G$是$e_1 \times e_2 \times e_3$的张量。

步骤5：重构张量$y`$：这一步验证了重构的张量与原始张量相似。重构的张量$y`$：

$$y` = G \times_1 A_{e_1}^{(1)} \times_2 A_{e_2}^{(2)} \times_3 A_{e_3}^{(3)} \quad (18)$$

$y`$近似于$y$，这意味着$\|y - y`\|_F^2$很小。此外，不同基聚类结果之间的潜在关联包含在左奇异向量的一个子集中。最后，聚类结果W表示为$A_{e_1}^{(1)}$，即第一模态展开矩阵的单位左奇异向量。三阶张量集成的过程可视化如**图5**所示。

三阶张量集成的结果与模糊聚类的结果相似。矩阵中每一行的值为样本点属于一个簇的概率，一行中最大值对应的聚类为样本点所属的簇。

本文将多个基聚类结果表示为一个三阶张量。设计三阶张量集成方法，核心是利用矩阵分解的优势来挖掘数据的隐性信息，更全面地挖掘基聚类之间的互补信息，融合得到质量更高的聚类划分结果。

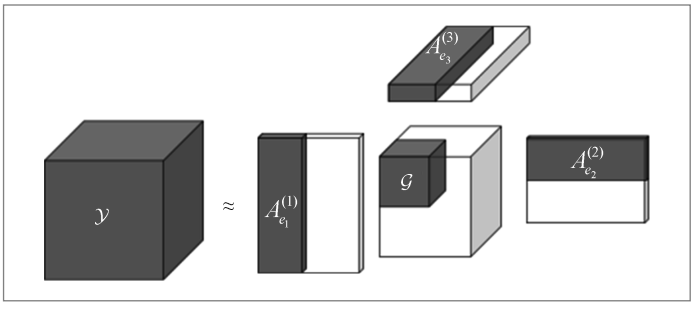

图 5 三阶张量集成过程可视化

3 实验结果与分析

3.1 数据集和评估方法

为了验证算法设计的有效性，本文选择了10个数据分析中常用的数据集来进行实验验证，数据量从1万到2 000万条不等，其中包括5个公开的真实数据集以及5个二维的人工合成数据集。这些数据集具有共同的特点，即存在数据量大、维度高、数据结构不规律、数据密度分布不均匀等问题，这些问题往往会对算法的性能产生较大的影响，因此可以将这些数据集作为算法性能评估的标准。5个真实数据集分别是PenDigits、USPS、Letters、MINIST、Covertype。真实数据集的详细信息见**表1**。

5个人工合成数据集分别是TB-1M（Two Bananas-1M）、SF-2M（Smiling Face-2M）、CC-5M（Concentric Circles-5M）、CG-10M（Circles and Gaussians-10M）、Flower-20M。人工合成数据集的详细信息见**表2**，其可视化表示如**图6**所示。

为了有效评估聚类集成的结果，本文选取了两种聚类评价指标，即准确率（cluster accuracy，CA）和标准化互信息（normalized mutual information，NMI），这两种指标在相关文献中被广泛应用。

CA是聚类分析中常用的评价指标之一，它通过计算被正确聚类的数据点数占总数据点数的百分比来评估聚类的准确性，数值越高，算法的准确性越好。该指标

由TP、FN、FP和TN等参数组成，其中TP表示将正类预测为正类的数据点数，FN表示将正类预测为负类的数据点数，FP表示将负类预测为正类的数据点数，TN表示将负类预测为负类的数据点数。CA计算式如下[23]：

$$CA=\frac{TP+TN}{TP+FN+FP+TN} \quad (19)$$

NMI指标是一种评估聚类算法效果的指标，它可以用来评价聚类算法划分出的类与真实聚类之间的相似度。它将聚类结果与真实簇类归一化后的互信息作为衡量指标，定量地衡量聚类结果的准确性。NMI的值总是介于0到1，值越大表明聚类结果越准确，越接近真实簇类。NMI计算式如下[24]：

$$NMI(X,Y)=\frac{I(X,Y)}{\sqrt{H(X)H(Y)}} \quad (20)$$

其中，X表示实验结果，Y表示真实结果，$I(X,Y)$表示X和Y之间的互信息，$H(X)$和$H(Y)$表示X和Y的熵。

为了排除偶尔侥幸的因素，每种算法进行20次实验，并记录它们的平均NMI、CA和时间成本。NMI和CA的值越大，表示聚类结果越好。对比算法包括经典的谱聚类算法、集成聚类算法以及近年来对其改进的算法。

表1 真实数据集详细信息

数据集	数据量/条	属性数/个	簇数/个
PenDigits	10 992	16	10
USPS	11 000	256	10
Letters	20 000	16	26
MINIST	70 000	784	10
Covertype	581 012	54	7

表2 合成数据集详细信息

数据集	数据量/条	属性数/个	簇数/个
TB-1M	1 000 000	2	2
SF-2M	2 000 000	2	4
CC-5M	5 000 000	2	3
CG-10M	10 000 000	2	11
Flower-20M	20 000 000	2	13

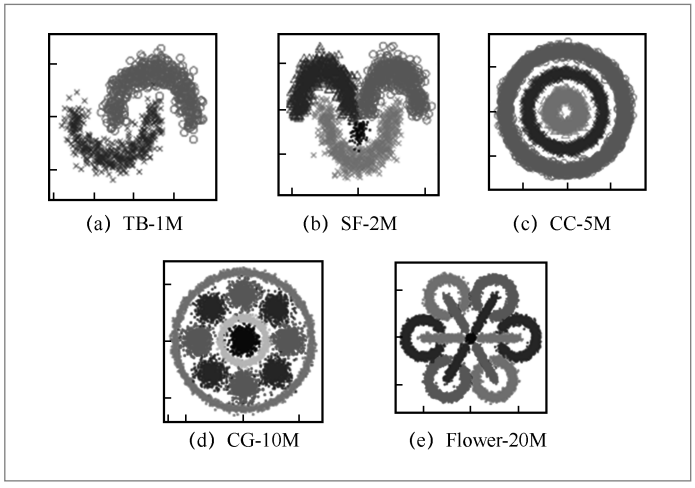

图6 5个合成数据集的可视化（每个数据集只有0.1%的子集被绘制）

3.2 与谱聚类算法的比较

本文选用8种谱聚类算法进行效果对比，包括传统的谱聚类算法和对其改进的算法，选用的算法分别是传统谱聚类(original spectral clustering, SC)[25]、基于图的谱聚类(efficient spectral clustering on graphs, ESCG)[26]、Nystrom谱聚类(Nystrom spectral clustering, Nystrom)[27]、基于k-means地标选择的谱聚类(landmark based spectral clustering using k-means based landmark selection, LSC-K)[28]、基于随机地标选择的谱聚类(landmark based spectral clustering using random landmark selection, LSC-R)[28]、快速直接谱聚类(fast explicit spectral clustering, FastESC)[29]、欧拉谱聚类

（Euler spectral clustering，EulerSC）[30]和U-SPEC[15]。**表3**和**表4**分别展示了本文提出的算法与8种谱聚类算法在实验数据集上的NMI值和CA值。从表中可以看出，大多数谱聚类算法对于量大高维的数据集在计算上是不可行的。由于设备内存无法满足所需容量要求，算法无法顺利运行，最终无法得出正确的结果。具体来说，本文使用N/A来表示内存不足得不出结果。由于构建$N\times N$亲和矩阵的内存消耗，SC和ESCG方法不能处理大于MNIST（由70 000个对象组成）的数据集。Nystrom、LSC-K、LSC-R和FastESC算法可以对包含多达200万个对象的数据集进行聚类，但它们不能处理更大的数据集。在8种谱聚类方法中，只有SCETT、U-SPEC和EulerSC可以处理所有的数据集。与其他算法相比，本文提出的算法通过快速有效地构建数据点间的稀疏亲和子矩阵，其运行所需内存较低，可以对数据进行很好的聚类分析，从而得到正确的数据划分结果。

在**表3**和**表4**中，笔者提供了算法对所有实验数据集的平均分数(Avg-score)，可以综合衡量一个算法对数据集的划分效果。如果一个方法不能处理所有的数据集，它将不会有平均分数。如**表3**和**表4**所

表 3 SCETT 算法与谱聚类算法的 NMI 对比

数据集	SC	ESCG	Nystrom	LSC-K	LSC-R	FastESC	EulerSC	U-SPEC	SCETT
PenDigits	60.54%	77.62%	66.37%	79.65%	78.33%	66.54%	59.72%	80.21%	**84.34%**
USPS	64.35%	49.46%	45.78%	67.96%	59.44%	42.56%	41.51%	64.51%	**75.69%**
Letters	11.52%	36.95%	39.21%	44.35%	41.78%	36.85%	32.85%	42.23%	**46.50%**
MNIST	75.21%	56.97%	46.53%	74.77%	63.36%	44.29%	10.13%	68.02%	**76.24%**
Covertype	N/A	N/A	6.82%	6.65%	6.59%	10.36%	0.03%	7.89%	**12.23%**
TB-1M	N/A	N/A	25.21%	0.12%	0.31%	26.18%	26.94%	95.61%	**96.93%**
SF-2M	N/A	N/A	47.52%	67.75%	60.24%	53.23%	48.08%	75.38%	**78.82%**
CC-10M	N/A	N/A	N/A	N/A	N/A	N/A	0	**99.86%**	99.53%
CG-10M	N/A	N/A	N/A	N/A	N/A	N/A	17.17%	79.61%	**91.53%**
Flower-20M	N/A	N/A	N/A	N/A	N/A	N/A	27.58%	86.78%	**93.51%**
Avg-score	N/A	N/A	N/A	N/A	N/A	N/A	26.40%	70.01%	**75.53%**

表 4 SCETT 算法与谱聚类算法的 CA 对比

数据集	SC	ESCG	Nystrom	LSC-K	LSC-R	FastESC	EulerSC	U-SPEC	SCETT
PenDigits	57.44%	78.21%	72.13%	84.07%	82.79%	70.97%	66.85%	83.21%	**88.21%**
USPS	63.83%	54.32%	52.21%	69.54%	61.61%	49.92%	48.68%	64.64%	**80.06%**
Letters	13.53%	31.45%	33.26%	36.57%	34.97%	30.42%	29.36%	35.81%	**38.62%**
MNIST	75.54%	64.67%	60.83%	80.64%	70.59%	56.61%	25.33%	75.42%	**82.59%**
Covertype	N/A	N/A	49.31%	49.92%	49.51%	49.76%	48.29%	49.65%	**51.73%**
TB-1M	N/A	N/A	80.25%	52.73%	53.38%	78.82%	80.11%	99.46%	**99.65%**
SF-2M	N/A	N/A	70.63%	86.59%	79.42%	75.63%	78.25%	93.47%	**93.71%**
CC-10M	N/A	N/A	N/A	N/A	N/A	N/A	53.85%	**99.99%**	99.97%
CG-10M	N/A	N/A	N/A	N/A	N/A	N/A	33.96%	82.54%	**92.99%**
Flower-20M	N/A	N/A	N/A	N/A	N/A	N/A	34.84%	89.93%	**94.07%**
Avg-score	N/A	N/A	N/A	N/A	N/A	N/A	49.95%	77.41%	**82.16%**

示，本文方法在10个实验数据集上的得分均排在前两名内，这证明了本文算法具有更高的准确性。SCETT算法在CC-10M数据集上虽未表现出最优性能，但是与最优算法的精度差距不大。就本文算法和8种单一聚类算法的平均分数而言，本文提出的算法分别获得了75.53%和82.16%的最佳平均NMI和最佳平均CA分数，而排名第二的U-SPEC方法仅分别实现了70.01%和77.41%的平均NMI和平均CA分数，这证明了本文提出的方法具有更高的鲁棒性。与其他算法相比，SCETT算法通过三阶张量集成融合多个基聚类结果，并利用矩阵分解的优势来挖掘数据的隐藏信息，能更全面地挖掘基聚类之间的互补信息，融合得到质量更高的聚类划分结果，因此聚类结果最优。

3.3 与聚类集成算法的比较

本文选用8种集成聚类算法进行效果对比，选用的算法分别是证据累积聚类（evidence accumulation clustering，EAC）[17]、加权连通三重法（weighted connected triple method，WCT，）[31]、基于k-means的一致性聚类（k-means based consensus clustering，KCC）[32]、基于概率轨迹的图划分（probability trajectory based graph partitioning，PIGP）[33]、基于熵的一致性聚类（entropy based consensus clustering，ECC）[34]、谱聚类集成（spectral ensemble clustering，SEC）[35]、局部加权图划分（locally weighted graph partitioning，LWGP）[36]和超可扩展聚类集成（ultra-scalable ensemble clustering，U-SENC）[15]。**表5**和**表6**分别展示了本文提出的算法与8种集成聚类算法在实验数据集上的NMI值和CA值。从两个表中可看出，EAC和WCT方法不能处理数据量大于MNIST的数据集（包含70 000个对象），运行失败导致无法得到正确结果，这是因为所需内存超过了设备可用容量，具体来说，使用N/A来表示内存不足得不出结果。本文算法和其余的6种集成聚类算法均可处理所有的实验数据集。本文算法的聚类准确度虽然没有每次都最优异，但仍然表现良好。本文提出的算法在5个真实数据集和5个合成数据集上的得分均排在前两名内，这证明了本文算法具有更高的准确性。

同时，**表5**和**表6**还提供了算法对所有实验数据集的平均分数。如果一个方法不

表 5 SCETT 算法与集成聚类算法的 NMI 对比

数据集	EAC	WCT	KCC	PIGP	ECC	SEC	LWGP	U-SENC	SCETT
PenDigits	76.25%	77.48%	58.76%	75.39%	57.92%	47.28%	77.34%	82.26%	**84.34%**
USPS	59.32%	58.68%	49.47%	59.82%	48.96%	39.51%	57.73%	73.65%	**75.69%**
Letters	37.33%	36.72%	33.81%	38.13%	34.42%	31.65%	37.14%	45.81%	**46.50%**
MNIST	66.21%	65.19%	54.47%	60.15%	56.24%	34.07%	65.28%	75.13%	**76.24%**
Covertype	N/A	N/A	5.91%	6.57%	5.86%	5.45%	7.51%	10.63%	**12.23%**
TB-1M	N/A	N/A	23.42%	34.35%	27.06%	10.83%	96.75%	**97.46%**	96.93%
SF-2M	N/A	N/A	42.86%	45.34%	41.51%	27.35%	69.97%	77.49%	**78.82%**
CC-10M	N/A	N/A	33.43%	0.51%	31.75%	17.25%	98.38%	**99.89%**	99.53%
CG-10M	N/A	N/A	64.81%	63.85%	62.93%	49.67%	78.14%	88.35%	**91.53%**
Flower-20M	N/A	N/A	61.08%	67.88%	60.87%	50.62%	78.64%	92.17%	**93.51%**
Avg-score	N/A	N/A	42.80%	45.20%	42.75%	31.36%	66.68%	74.28%	**75.53%**

表 6 SCETT 算法与集成聚类算法的 CA 对比

数据集	EAC	WCT	KCC	PIGP	ECC	SEC	LWGP	U-SENC	SCETT
PenDigits	80.96%	82.85%	63.42%	78.46%	62.21%	51.83%	81.65%	86.64%	**88.21%**
USPS	63.51%	62.83%	53.33%	62.58%	53.49%	45.26%	59.67%	76.82%	**80.06%**
Letters	30.36%	30.05%	26.87%	31.62%	27.49%	26.32%	30.91%	37.25%	**38.62%**
MNIST	73.25%	70.86%	59.78%	65.24%	61.32%	43.25%	71.84%	78.58%	**82.59%**
Covertype	N/A	N/A	49.65%	49.21%	49.73%	49.96%	49.53%	50.53%	**51.73%**
TB-1M	N/A	N/A	70.22%	82.88%	72.65%	60.32%	99.63%	**99.72%**	99.65%
SF-2M	N/A	N/A	67.24%	73.56%	66.82%	55.78%	88.87%	91.69%	**93.71%**
CC-10M	N/A	N/A	66.85%	52.85%	62.71%	61.84%	99.28%	**99.99%**	99.97%
CG-10M	N/A	N/A	66.89%	63.47%	64.83%	58.21%	81.78%	91.85%	**92.99%**
Flower-20M	N/A	N/A	57.63%	63.76%	56.73%	50.68%	81.52%	92.62%	**94.07%**
Avg-score	N/A	N/A	58.19%	62.36%	57.80%	50.34%	74.47%	80.57%	**82.16%**

能处理所有的数据集，它将不会有平均分数。就10个集成聚类算法的平均分数而言，本文提出的算法分别获得了75.53%和82.16%的最佳平均NMI和最佳平均CA分数，而排名第二的集成聚类方法U-SENC仅分别实现了74.58%和81.27%的平均NMI和ACC分数，这证明了本文算法的有效性。

SCETT算法与集成聚类算法的时间成本对比结果见**表7**，从中可看出本文提出的SCETT方法的效率高于大多数集成聚类方法，尤其是在数据量超过百万的大规模数据集上。本文算法通过快速有效地构建数据点间的稀疏亲和子矩阵，运行所需内存较低，可以对数据进行很好的聚类分析，进而通过三阶张量集成融合多个基聚类结果，并利用矩阵分解的优势来挖掘数据的隐藏信息，能更全面地挖掘多个基聚类之间的互补信息，从而融合得到质量更高的聚类划分结果。

4 总结与展望

本文针对大规模谱聚类计算负担大和准确性普遍不高的问题，提出了一种基

表 7 SCETT 算法与集成聚类算法的时间成本对比 /s

数据集	EAC	WCT	KCC	PIGP	ECC	SEC	LWGP	U-SENC	SCETT
PenDigits	7.69	46.01	7.57	9.94	11.76	**3.67**	4.26	15.13	13.56
USPS	10.89	46.45	14.67	57.72	22.43	**8.85**	9.05	27.17	25.07
Letters	23.40	168.11	30.91	124.46	50.04	14.64	**14.08**	21.32	22.79
MNIST	543.71	3384.21	286.57	2 165.18	402.10	247.96	253.46	136.44	**125.60**
Covertype	N/A	N/A	937.75	7 869.71	1 463.43	691.64	672.72	178.48	**163.49**
TB-1M	N/A	N/A	1 296.45	1 264.82	2 098.72	997.30	989.15	321.37	**314.21**
SF-2M	N/A	N/A	2 896.34	2 488.96	4 708.16	2 160.46	2 145.82	689.82	**652.72**
CC-10M	N/A	N/A	6 824.38	5 017.91	11 189.43	5 030.84	5 024.21	1 836.40	**1 785.62**
CG-10M	N/A	N/A	17 296.29	11 485.11	27 482.95	10 927.88	10 865.38	3 796.78	**3 603.26**
Flower-20M	N/A	N/A	33 658.83	20 682.87	53 652.15	21 593.96	21 356.25	7 432.17	**7 265.29**

于三阶张量的大规模谱聚类集成算法。在SCETT算法中，提出了混合代表最近邻近似方法，有效地构造了数据集和代表集之间的稀疏亲和子矩阵。然后，将稀疏亲和子矩阵作为交叉亲和矩阵，在数据集和代表性集合之间构建二部图。利用二部图结构，通过图分割的方法获得初步聚类结果。在此基础上，提出了一种三阶张量集成方法来融合多个聚类结果，挖掘多个聚类结果之间的互补信息，从而得到一个质量更高的聚类结果。最后，在10个大规模数据集上进行实验，通过与谱聚类算法、聚类集成算法以及近年来对其改进的算法进行对比，证明了本文算法在降低大规模数据谱聚类计算负担的基础上，进一步提高了聚类的准确性和鲁棒性。

展望未来，实际应用中的数据量通常非常大，而且可能是动态输入的。为了解决这些问题，可以将数据流聚类和聚类集成相结合进行研究，结合数据动态变化的规律，定量地分析隐藏在数据中的知识演化规律，以实现更有鲁棒性和高效性的数据流划分方法。

参考文献:

[1] 孙林, 秦小营, 徐久成, 等. 基于K近邻和优化分配策略的密度峰值聚类算法[J]. 软件学报, 2022, 33(4): 1390-1411.
SUN L, QIN X Y, XU J C, et al. Density peak clustering algorithm based on K-nearest neighbors and optimized allocation strategy[J]. Journal of Software, 2022, 33(4): 1390-1411.

[2] LU Y, CHEUNG Y M, TANG Y Y. Self-adaptive multiprototype-based competitive learning approach: a k-means-type algorithm for imbalanced data clustering[J]. IEEE Transactions on Cybernetics, 2021, 51(3): 1598-1612.

[3] AHMAD A, KHAN S S. IinitKmix-a novel initial partition generation algorithm for clustering mixed data using k-means-based clustering[J]. Expert Systems with Applications, 2021, 167: 114149.

[4] 胡春安, 王家欣, 毛伊敏. 基于分组和IGSA的并行密度聚类算法[J]. 计算机应用研究, 2021, 38(11): 3293-3299.
HU C A, WANG J X, MAO Y M. Density-based clustering algorithm based on groups and improve gravitational search[J]. Application Research of Computers, 2021, 38(11): 3293-3299.

[5] GUO W J, WANG W H, ZHAO S P, et al. Density peak clustering with connectivity estimation[J]. Knowledge-Based Systems, 2022, 243: 108501.

[6] 江婧婷, 郑朝晖. 面向大规模节点划分的网格密度峰值聚类[J]. 小型微型计算机系统, 2022, 43(3): 498-505.
JIANG J T, ZHENG Z H. Density peak and grid based clustering for large-scale node partition[J]. Journal of Chinese Computer Systems, 2022, 43(3): 498-505.

[7] 徐晓, 丁世飞, 孙统风, 等. 基于网格筛选的大规模密度峰值聚类算法[J]. 计算机研究与发展, 2018, 55(11): 2419-2429.
XU X, DING S F, SUN T F, et al. Large-scale density peaks clustering algorithm based on grid screening[J]. Journal of Computer Research and Development, 2018, 55(11): 2419-2429.

[8] 唐益明, 丰刚永, 任福继, 等. 面向结构复杂数据集的模糊聚类有效性指标[J]. 电子测量与仪器学报, 2018, 32(4): 119-127.
TANG Y M, FENG G Y, REN F J, et al. Fuzzy clustering validity index facing data set with complexity structure[J]. Journal of Electronic Measurement and Instrumentation, 2018, 32(4): 119-127.

[9] 李凯, 张可心. 结构α-熵的加权高斯混合模型的子空间聚类[J]. 电子学报, 2022, 50(3): 718-725.
LI K, ZHANG K X. Structural α-entropy weighting Gaussian mixture model for subspace clustering[J]. Acta Electronica Sinica, 2022, 50(3): 718-725.

[10] 张熠玲, 杨燕, 周威, 等. CMvSC: 知识迁移下的深度一致性多视图谱聚类网络[J]. 软件学报, 2022, 33(4): 1373–1389.
ZHANG Y L, YANG Y, ZHOU W, et al. CMvSC: knowledge transferring based deep consensus network for multi-view spectral clustering[J]. Journal of Software, 2022, 33(4): 1373–1389.

[11] TANG H, ZHU X T, CHEN K, et al. Towards uncovering the intrinsic data structures for unsupervised domain adaptation using structurally regularized deep clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(10): 6517–6533.

[12] VON LUXBURG U. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395–416.

[13] CHEN W Y, SONG Y Q, BAI H J, et al. Parallel spectral clustering in distributed systems[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(3): 568–586.

[14] CAI D, CHEN X L. Large scale spectral clustering via landmark-based sparse representation[J]. IEEE Transactions on Cybernetics, 2015, 45(8): 1669–1680.

[15] HUANG D, WANG C D, WU J S, et al. Ultra-scalable spectral clustering and ensemble clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 32(6): 1212–1226.

[16] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492–1496.

[17] FRED A L N, JAIN A K. Combining multiple clusterings using evidence accumulation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(6): 835–850.

[18] HORE P, HALL L O, GOLDGOF D B. A scalable framework for cluster ensembles[J]. Pattern Recognition, 2009, 42(5): 676–688.

[19] YI J F, YANG T B, JIN R, et al. Robust ensemble clustering by matrix completion[C]//Proceedings of 2012 IEEE 12th International Conference on Data Mining. Piscataway: IEEE Press, 2013: 1176–1181.

[20] 罗晓慧, 李凡长, 张莉, 等. 基于选择聚类集成的相似流形学习算法[J]. 软件学报, 2020, 31(4): 991–1001.
LUO X H, LI F C, ZHANG L, et al. Similar manifold learning based on selective cluster ensemble for image clustering[J]. Journal of Software, 2020, 31(4): 991–1001.

[21] WEI H Q, CHEN L, RUAN K Y, et al. Low-rank tensor regularized fuzzy clustering for multiview data[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(12): 3087–3099.

[22] LI Z G, WU X M, CHANG S F. Segmentation using superpixels: a bipartite graph partitioning approach[C]// Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2012: 789–796.

[23] NGUYEN N, CARUANA R. Consensus clusterings[C]//Proceedings of Seventh IEEE International Conference on Data Mining (ICDM 2007). Piscataway: IEEE Press, 2008: 607–612.

[24] STREHL A, GHOSH J. Cluster ensembles: a knowledge reuse framework for combining multiple partitions[J]. Journal Of Machine Learning Research, 2003, 3: 583–617.

[25] VON LUXBURG U. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395–416.

[26] LIU J L, WANG C, DANILEVSKY M, et al. Large-scale spectral clustering on graphs[C]//Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. New York: ACM, 2013: 1486–1492.

[27] CHEN W Y, SONG Y Q, BAI H J, et al. Parallel spectral clustering in distributed systems[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(3): 568–586.

[28] CAI D, CHEN X L. Large scale spectral clustering via landmark-based sparse

representation[J]. IEEE Transactions on Cybernetics, 2015, 45(8): 1669-1680.
[29] HE L, RAY N, GUAN Y S, et al. Fast large-scale spectral clustering via explicit feature mapping[J]. IEEE Transactions on Cybernetics, 2019, 49(3): 1058-1071.
[30] WU J S, ZHENG W S, LAI J H, et al. Euler clustering on large-scale dataset[J]. IEEE Transactions on Big Data, 2018, 4(4): 502-515.
[31] IAM-ON N, BOONGOEN T, GARRETT S, et al. A link-based approach to the cluster ensemble problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12): 2396-2409.
[32] WU J J, LIU H F, XIONG H, et al. K-means-based consensus clustering: a unified view[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(1): 155-169.
[33] HUANG D, LAI J H, WANG C D. Robust ensemble clustering using probability trajectories[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(5): 1312-1326.
[34] LIU H F, ZHAO R, FANG H S, et al. Entropy-based consensus clustering for patient stratification[J]. Bioinformatics, 2017, 33(17): 2691-2698.
[35] LIU H F, WU J J, LIU T L, et al. Spectral ensemble clustering via weighted K-means: theoretical and practical evidence[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(5): 1129-1143.
[36] HUANG D, WANG C D, LAI J H. Locally weighted ensemble clustering[J]. IEEE Transactions on Cybernetics, 2018, 48(5): 1460-1473.

作者简介

仵匀政(1998-),男,济南大学信息科学与工程学院硕士生,主要研究方向为数据挖掘、数据聚类。

杜韬(1979-),男,博士,济南大学信息科学与工程学院副教授,主要研究方向为数据挖掘、数据聚类。

周劲(1976-),男,博士,济南大学信息科学与工程学院教授,山东省人工智能学会理事,主要研究方向为数据挖掘、数据聚类。

陈迪（1998– ），男，济南大学信息科学与工程学院硕士生，主要研究方向为数据挖掘、数据聚类。

王心耕（1999– ），男，济南大学信息科学与工程学院硕士生，主要研究方向为数据挖掘、数据聚类。

收稿日期： 2023–06–25

基金项目： 国家自然科学基金项目（No.62273164，No.61873324）；山东省自然科学基金项目（No.ZR2019MF040）

Foundation Items: The National Natural Science Foundation of China (No.62273164, No.61873324), The Natural Science Foundation of Shandong Province (No.ZR2019MF040)