

基于标签迭代的聚类集成算法

何玉林^{1,2}, 杨锦², 黄哲学^{1,2}, 尹剑飞²

(1. 人工智能与数字经济广东省实验室(深圳), 广东 深圳 518107;

2. 深圳大学计算机与软件学院, 广东 深圳 518060)

摘要: 现有的“数据相同, 算法不同”式的聚类集成算法训练策略普遍存在处理大规模数据性能受限以及共识函数适应性不强的缺点。为此, 对“数据不同, 算法相同”式的聚类集成算法训练策略进行了研究, 构建了一种基于标签迭代的聚类集成(LICE)算法。首先, 该算法在原始数据集的随机样本划分(RSP)数据块上训练若干基聚类器。接着, 利用最大平均差异准则对聚类簇数相同的基聚类结果进行融合, 并基于标签确定的RSP数据块训练一个启发式分类器。之后, 迭代式地利用启发式分类器对标签不确定的RSP数据块中的样本点进行标签预测, 利用分类标签与聚类标签一致的样本点强化启发式分类器的性能。最后, 通过一系列可信的实验对LICE算法的可行性和有效性进行验证, 结果显示在代表性数据集上, LICE算法对应的标准互信息、调整兰德系数、Fowlkes-Mallows指数以及纯度在第5次迭代时相比于迭代起始分别平均提升了17.23%、16.75%、31.29%和12.37%。与7种经典的聚类集成算法相比, 在选用的数据集上, 这4个指标的值分别平均提升了11.76%、16.50%、9.36%和14.20%。实验证实了LICE算法是一种高效合理的、能够处理大数据聚类问题的聚类集成算法。

关键词: 聚类集成算法; 集成学习; 随机样本划分; 最大平均差异; 标签迭代

中图分类号: TP391.4

文献标志码: A

doi: 10.11959/j.issn.2096-6652.202443

Label iteration-based clustering ensemble algorithm

HE Yulin^{1,2}, YANG Jin², HUANG Zhexue^{1,2}, YIN Jianfei²

1. Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, 518107, China

2. College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China

Abstract: The existing training strategies for clustering ensemble algorithm are generally conducted based on the same data and different base clustering algorithms and commonly have the limitations of low performance for large-scale data and weak adaptability of consensus function. To address these problems, this paper proposed a label iteration-based clustering ensemble (LICE) algorithm which was developed based on the training strategy for clustering ensemble algorithm of different data and same base clustering algorithm. Firstly, multiple base clusterings were trained based on the random sample partition (RSP) data blocks. Secondly, the base clustering results with same cluster numbers were fused with maximum mean discrepancy criterion and then a heuristic classifier was trained based on the RSP data blocks with labels. Thirdly, the sample points without labels were labeled with heuristic classifier which was iteratively enhanced with the labeled sample points having the consistent labeling for clustering and classification. Finally, a series of persuasive experiments were conducted to validate the feasibility and effectiveness of LICE algorithm. The experimental results showed

收稿日期: 2024-08-05; 修回日期: 2024-10-30

通信作者: 何玉林, csylhe@126.com

基金项目: 广东省自然科学基金面上项目 (No.2023A1515011667); 深圳市科技重大专项项目 (No.202302D074); 广东省基础与应用基础研究基金项目 (No.2023B1515120020)

Foundation Items: Natural Science Foundation of Guangdong Province (No.2023A1515011667), Science and Technology Major Project of Shenzhen (No.202302D074), Guangdong Basic and Applied Basic Research Foundation (No.2023B1515120020)

that the normalized mutual information, adjusted Rand index, Fowlkes-Mallows index and purity of LICE algorithm increased by 17.23%, 16.75%, 31.29%, and 12.37% on average at the 5th iteration compared to the initial iteration and these four indexes increased by 11.76%, 16.50%, 9.36%, and 14.20% on average for the representative datasets in comparison with seven state-of-the-art clustering ensemble algorithms and thus demonstrate that LICE algorithm is an efficient and reasonable clustering ensemble algorithm with the potential to handle large-scale data clustering problems.

Key words: clustering ensemble algorithm, ensemble learning, random sample partition, maximum mean discrepancy, label iteration

0 引言

聚类是一种无监督机器学习范式,旨在将相似的样本点分配到同一个数据簇中,将不相似的样本点分配到不同的数据簇中。聚类已成功应用于众多实际领域中,用来探索数据的潜在结构,为后续的数据决策提供支持和参考。聚类集成融合了聚类和集成学习,能将多个聚类结果结合起来,是获得高质量聚类的有效方法之一。已有的实验结果表明,聚类集成能够有效利用基聚类结果之间的差异性,获得比单一聚类更准确、更鲁棒的聚类结果。聚类集成算法的研究是目前机器学习领域的研究热点之一,已在许多领域得到了应用^[1-3]。聚类集成算法主要包括2个部分:基聚类的生成和共识函数的设计。目前,基聚类的生成方法主要可以分为3类:在同一个数据集上训练多个异构的基聚类算法;在同一个数据集上用不同的初始化参数多次训练同构的基聚类算法;在同一个数据集不同的特征空间子集上多次训练同一个聚类算法。共识函数的设计可以分为4类:协关联矩阵;超图分割;投票和重标记;中值划分。基聚类的生成和共识函数的设计可以有很多不同的组合,因此可以产生多种不同类型的聚类集成算法。关于大数据的聚类算法,目前的研究主要有2个方向,分别为大数据串行聚类和大数据并行聚类^[4]。大数据串行聚类可以分为基于样例选择的聚类、基于增量学习的聚类、基于特征子集的聚类和基于特征转换的聚类。而大数据并行聚类可以分为基于 MapReduce 框架的聚类、基于 Spark 框架的聚类、基于 Storm 框架的聚类。

尽管已有的实验结果指出了当前聚类集成算法在特定应用领域的优良性能,然而这些基于“数据相同,算法不同”式的聚类集成算法训练策略普遍存在两点缺陷,即处理大规模数据性能受限与共识函数适应性不强。

第一,现有的聚类集成算法通常是基于距离度

量来判断样本之间的相似性,因此,在基聚类的生成阶段,需要计算所有样本点两两之间的距离,并用一个 $n \times n$ 的矩阵来存储这些距离 (n 为数据集中样本的个数),这一操作的时间复杂度和空间复杂度均为 $O(n^2)$ 。对于聚类集成算法而言,为了生成具有差异性的基聚类成员,就需要对整个数据集进行多次聚类,当数据集规模非常大时,无疑会导致算法训练所需时间复杂度以及空间复杂度的激增。第二,不同的基聚类算法会生成数量不同、形状各异的簇,其中可能会包含与“真实的”聚类结果相差甚远的基聚类,共识函数会将所有的基聚类作为基学习器进行集成,最终的聚类结果势必会受到这些低质基聚类的影响,造成聚类簇数难以确定,聚类融合效果不佳。另外,共识函数本身无法分辨噪声点和非噪声点,当基聚类中存在噪声点时,共识函数通常会将这些噪声点视为某个簇的一部分,使其参与到聚类集成的过程中去,从而导致聚类结果的准确性下降。

为此,本文采用“数据不同,算法相同”式的聚类集成算法训练策略构建了一种基于标签迭代的聚类集成 (label iteration-based clustering ensemble, LICE) 算法。首先,该算法在原始数据集的随机样本划分 (random sample partition, RSP)^[5] 数据块上训练若干基聚类器,其中 RSP 数据块是与原始数据集保持概率分布一致性的数据子集。这样做的优势体现在2个方面:一是相比于传统的整体数据全部使用的训练范式,数据子集的使用缩短了基聚类的生成时间;二是由于 RSP 数据块之间存在的分布一致性,使得基聚类之间更容易保持簇数的相同。接着,设计基于最大平均差异准则 (maximum mean discrepancy criterion, MMD)^[6] 的共识函数对聚类簇数相同的 RSP 基聚类结果进行融合,并基于标签确定的 RSP 数据块训练一个启发式分类器。之后,迭代式地利用启发式分类器对标签不确定的 RSP 数据块中的样本点进行标签预测,采

用分类标签与聚类标签一致的样本点强化启发式分类器的性能,直至所有样本点的标签确定完成。LICE算法只使用簇数相同的基聚类结果来训练初始分类器,可以有效排除与真实聚类结果相差过大的低质基聚类,降低其对聚类集成结果的影响。另外,LICE算法通过基于密度的基聚类算法识别噪声点,使其不参与分类器的训练和迭代,确保了共识函数的性能不会受噪声点的影响。

本文设计了系统的实验对LICE算法的可行性和有效性进行了测试,其中可行性测试验证LICE算法的收敛性和标签迭代的合理性,有效性测试对比与其他7种经典的聚类集成算法的聚类表现。在可行性验证中,随着标签迭代次数的增加,LICE算法对应的4个评价指标,即标准互信息(normalized mutual information, NMI),调整兰德系数(adjusted Rand index, ARI),Fowlkes-Mallows指数(Fowlkes-Mallows index, FMI)以及纯度(Purity)均呈现先上升后收敛的趋势。在每一次的迭代过程中,都会确定若干样本点的标签用作对启发式分类器的训练,且新加入的样本点数量会越来越小,从标签一致性的角度再次证明了LICE算法的收敛。在有效性验证中,本文将LICE算法同7种聚类集成算法进行了对比,分别是局部加权证据累积(locally weighted evidence accumulation, LWEA)算法^[7]、局部加权图分割(locally weighted graph partitioning, LWGP)算法^[7]、证据累积聚类(evidence accumulation clustering, EAC)算法^[8]、超图划分(hybrid bipartite graph formulation, HBGF)算法^[9]、元聚类(meta-clustering, MCL)算法^[10]、基于簇的相似性划分(cluster-based similarity partitioning, CSP)算法^[10]和协关联矩阵的自我提升(ensemble clustering via co-association matrix self-enhancement, EC-CMS)聚类算法^[11]。在8个代表性数据集上,LICE算法均取得了最好的NMI、ARI、FMI以及Purity,进而证明了LICE算法具备高效处理大规模数据聚类问题的能力。

本文的创新点总结如下:采用“数据不同,算法相同”的聚类集成算法训练策略,将原数据集分割为多个RSP数据块,在每个数据块上使用相同的聚类集成算法,而不是像传统的方式那样对整个数据集聚类。这种方式提升了对大规模数据的处理能力,并且使得基聚类之间更容易保持簇数的相同;共识函数的设计没有采用传统的利用所有基聚

类进行集成的方式,而是将基聚类分成了2个部分:簇数一致的和簇数不一致的。选取簇数一致的这部分基聚类来训练启发式分类器,最大限度保证此分类器的预测性能。此后,利用启发式分类器对簇数不一致的RSP数据块的样本点进行标签预测,并利用分类标签与聚类标签一致的样本点再次强化分类器的性能,不断重复“先不一致样本点标注、后分类器训练增强”的过程,直至所有样本点的标签确定完成。

本文结构安排如下:第1节介绍相关工作;第2节介绍与LICE算法相关的预备知识;第3节详细描述了LICE算法的细节;第4节对LICE算法的可行性和有效性进行实验验证;第5节总结全文并给出未来的研究方向。

1 相关工作

1.1 基聚类的生成

聚类集成是一种集成学习方法,它使用常规的聚类算法作为基学习器,因此,基聚类成员的质量至关重要。当基聚类结果“好而不同”时,聚类集成算法往往能够取得较好的结果。关于基聚类生成研究的代表性工作总结如下。

目前,使用不同的聚类算法来生成基聚类成员是一种较为常见的基聚类生成方式。Fred等^[8]使用不同的 k 值运行 k -means^[12]算法来生成基聚类,Strehl等^[10]在同一数据集上采用相同的聚类算法生成了一组基本聚类。Bai等^[13]提出了一种多重 k -means算法,并检测每一个簇标签的可信度。上述研究使用了 k -means算法来生成基聚类,但是 k -means算法的时间复杂度过高,处理大规模数据集的能力有限。为此,Minaei-Bidgoli等^[14-15]使用子采样和bagging生成多个数据子集,然后用 k -means算法对每个数据子集进行聚类,以生成基聚类成员。

对于高维数据集,Topchy等^[16]将其随机投影到一维空间中,然后使用随机超平面将数据空间切割成多个子空间,并使用 k -means在子空间内对数据进行聚类。Fern等^[17]采用随机投影法将高维数据投影到低维空间,为每个投影生成一个聚类,进而得到基聚类成员。

为了充分利用基聚类给出的信息,并降低低质量基聚类所带来的负面影响,基聚类加权是一种当下常见的聚类集成策略。Ilc^[18]提出利用特征选取方法提取出数据集中的冗余特征,根据冗余特征矩

阵计算出权重。Bagherinia等^[19]利用初始的集成结果对基聚类的多样性进行检测，并利用多样性更高的基聚类重新进行集成。

1.2 共识函数的设计

在聚类集成的研究中，选取合适的共识函数来融合基聚类的结果是至关重要的。共识函数的设计大致可以分为四大类，代表性工作分别简述如下。

基于协关联（co-association, CA）矩阵的聚类集成方法首先计算出基聚类中2个样本点出现在同一个簇的频率，将这些频率的值作为CA矩阵的元素。然后，对这个CA矩阵应用传统的聚类算法来得到最终的聚类结果。Fred等^[8]提出了CA矩阵的概念，并以CA矩阵为基础设计了证据累积聚类EAC算法来解决聚类集成问题。Liang等^[20]选取出代表性的样本点来构建出部分CA矩阵，以减少CA矩阵的计算时间。Shi等^[21]结合标签空间和特征空间中的冗余信息，提出了一种针对CA矩阵的优化算法。Huang等^[7]使用集成驱动的聚类指数来对基聚类进行评估和加权，提出了LWEA和LWGP这2个共识函数来提高聚类效果。Bian等^[22]利用原CA矩阵和基聚类不同簇之间的关系，提出了模糊CA矩阵，根据模糊CA矩阵进行聚类集成。Jia等^[11]提出了一种CA矩阵自我优化的模型，利用优化之后的CA矩阵进行集成学习，得到最终的聚类结果。

基于图的聚类集成方法通过构建图来整合基类的信息，然后将集成问题转化为图分割问题。Strehl^[10]提出了3种基于图分割的聚类集成算法：分别为CSP算法、超图分割（hypergraph partitioning, HGP）算法和MCL算法。Fern等^[9]提出了HBGF算法，对相似矩阵进行分解得到最终的聚类集成结果。Zhou等^[23]从多视图特征出发构建了多重图，利用多视图进行谱聚类。

在无监督学习中，聚类标签仅代表分组属性，不能直接在不同的基聚类中进行标签的比较。所以，不同结果下的聚类标签应该被重标记，基于重标记的方法通常和投票策略^[24]相结合。Kuhn等^[25]使用著名的匈牙利算法来解决标签对应问题。Zhou等^[26]将簇之间的样本重叠度作为簇的相似性度量指标，实现了对不同簇的标签对齐。Khedairia等^[27]提出了基于迭代策略的集成算法，根据投票策略将数据点分配到子簇中，而未获得多数票的数据点则在后续的迭代中被重新聚类。

基于中值划分的方法将每个基聚类提供的结果

视为样本点的新特征，聚类集成问题被转化为最优化问题，其目的是找到一个分区，使得该分区与多个基聚类的相似性最大。Cristofor等^[28]使用遗传算法寻找与基聚类结果最相似的分区。Li等^[29]结合了自适应权重和核函数学习，来提高集成的准确性和一致性。Zhang等^[30]通过数据重组的方式来提高基聚类 and 一致性分区的相似性。

2 预备知识

2.1 聚类集成框架

聚类集成问题的形式化定义如下： $X = \{x_1, x_2, \dots, x_n\}$ 是一个由 d 维空间上的 n 个样本点构成的数据集，其中 x_i 表示数据集 X 的第 i 个样本点， n 为数据集中样本点的个数。在基聚类的生成阶段，通过聚类算法生成 M 个基聚类，每个基聚类由多个簇组成。 M 个基聚类可以表示为：

$$\Pi = \{\pi^1, \dots, \pi^M\} \quad (1)$$

其中，

$$\pi^m = \{C_1^m, \dots, C_{n_c}^m\} \quad (2)$$

π^m 表示 Π 中的第 m 个基聚类， C_i^m 表示 π^m 中的第 i 个簇。对于每一个基聚类而言，簇与簇之间不相交，而所有簇的合集即为整个数据集，即 $\forall C_i^m, C_j^m \in \pi^m, i \neq j$, 有 $C_i^m \cap C_j^m = \emptyset$, 且 $\forall \pi^m \in \Pi, \bigcup_{i=1}^{n_c} C_i^m = X$ 。为方便起见， Π 中的所有簇表示为：

$$\mathcal{C} = \{C_1, \dots, C_{n_c}\} \quad (3)$$

其中， C_i 表示第 i 个簇， n_c 表示基聚类中簇的总数量。显然，有 $n_c = n^1 + \dots + n^M$ 成立。

一般而言，聚类集成问题可以表述为：通过给定的基聚类，利用一个共识函数为这些基聚类找到一个一致性分布，即：

$$\Pi^* = \{C_1, C_2, \dots, C_k\} \quad (4)$$

其中， Π^* 为最终的聚类集成结果， C_i 表示 Π^* 中的第 i 个簇。在 Π^* 中，不同簇之间样本点的相似度要尽可能低，而同一个簇内的样本点间相似度要尽可能高。

2.2 最大平均差异准则

对于两个随机分布而言，最大平均差异准则（maximum mean discrepancy criterion, MMD）是衡量这2个分布是否一致的一种有效手段。在聚类

集成的研究中, 通常使用MMD来作为基聚类标签对齐的标准。

$$\text{MMD}(F, p, q) := \sup_{f \in F} (E_p[f(x)] - E_q[f(y)]) \quad (5)$$

其中, $\sup(\cdot)$ 计算内嵌公式的最大值, $E_p(\cdot)$ 表示数学期望, F 为函数域, p 是原域数据 x 的分布, q 是

$$\text{MMD}^2[p, q, H] = \left\| \frac{1}{n^2} \sum_{i=1}^n \sum_{i'=1}^n k(x_i, x_{i'}) - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(x_i, y_j) + \frac{1}{m^2} \sum_{j=1}^m \sum_{j'=1}^m k(y_j, y_{j'}) \right\|_H \quad (6)$$

其中, H 为函数域 F 的再生希尔伯特空间, k 为原始样本空间中的核函数。MMD 的值越小代表 2 个分布越相似, 当 MMD 的值为 0 时, 表示 2 个分布完全一样。

3 LICE 算法

3.1 随机样本划分

随机样本划分 (random sample partition, RSP) 的目的是将大数据集划分为多个更小的数据子集, 使得所有的数据子集和原始数据集的概率分布基本保持一致, 这样不仅使得 LICE 算法处理大规模数据集的能力增强, 而且在基聚类生成阶段, 更容易产生簇数相同的基聚类结果。对原始数据集 D 的一次随机划分约束条件如下: $D_i \cap D_j = \emptyset$, 其中 $i \neq j, i, j \in \{1, 2, \dots, m\}$; $\bigcup_{i=1}^m D_i = D$, 可得 $D = \{D_1, D_2, \dots, D_m\}$ 。普通的随机无放回抽样并不能保证每个数据子集的概率分布与原始数据集保持一致, 因此, Salloum 等^[5]提出了一种随机样本划分的方法, 并通过理论推导验证了对于任意 $i \in \{1, 2, \dots, m\}$, 式 (7) 成立。

$$E[\tilde{F}(D_i)] = F(D) \quad (7)$$

其中, $\tilde{F}(D_i)$ 和 $F(D)$ 分别表示划分之后的数据块 D_i 和原始数据集 D 的概率分布函数。

RSP 的实现过程由 2 个主要部分组成。第一步, 先将原数据集 D 随机分割为一组包含 p 个数据子集的合集, 使得 $D = \{A_1, A_2, \dots, A_p\}$ 。然后, 将 $A_i = \{x_1, x_2, \dots, x_{n_i}\}$ 随机打乱, 再将 A_i 按照打乱后的顺序切分成 m 个数据规模更小的数据块, 得到 $A_i = \{B_1^i, B_2^i, \dots, B_m^i\}$ 。在 2 次切分中, 都要保证切分后的每个数据子集的样本数量尽量相同。第二步, 将切分之后的 A_i 按照顺序依次合并, 得到与 A_i 同等规模的新子集。

$$D_i = \{B_1^i, B_2^i, \dots, B_m^i\}, \text{ 其中 } i \in \{1, 2, \dots, p\} \quad (8)$$

目标域数据 y 的分布, $f(x)$ 和 $f(y)$ 分别是 x 和 y 的映射函数。

假设 x 有 n 个样本, y 有 m 个样本, 因为映射函数 $f(\cdot)$ 直接计算较为困难, 可以将映射函数的内积转换为原始样本空间中的核函数 $k(\cdot, \cdot)$, MMD 的计算可以简化为:

通过以上 2 步得到的 $D = \{D_1, D_2, \dots, D_m\}$ 即为数据集 D 的一个 RSP。

由分布函数的定义可知, 对于 $x \in R$, D 中取值小于等于 x 的样本数为 $n \times F(x)$, 所以 D_i 中取值小于等于 x 的样本数为 $n_i \times F(x)$, 可以得到式 (9)。

$$E[F_i(x)] = \frac{n_i \times F(x)}{n_i} = F(x) \quad (9)$$

其中, $F_i(x)$ 和 $F(x)$ 分别是数据子集 D_i 和原始数据集 D 的概率分布函数, 可知数据子集和原始数据集对应的概率分布满足式 (7), 所以 $D = \{D_1, D_2, \dots, D_m\}$ 是一个合格的 RSP, 即每个数据子集和原始数据集的概率分布一致。

3.2 标签迭代

将数据集切分成 RSP 数据块之后, 可以得到一组概率分布保持一致的数据子集 $D = \{D_1, D_2, \dots, D_m\}$ 。对于每一个 RSP 数据块, 利用基于密度的噪声应用空间聚类 (density-based spatial clustering of applications with noise, DBSCAN) 算法生成基聚类, 以此来获得一组基聚类成员 $\Pi = \{\pi^1, \dots, \pi^M\}$ 。由于 RSP 数据块具有一致的概率分布, 而 DBSCAN 算法又是一种基于密度的聚类算法, 因此在相似的数据分布上, DBSCAN 算法得到的簇数量大致相同。在同一个数据集上多次运行 DBSCAN 算法, 每次运行得到的簇的数量可能会有所区别, 但是只要参数选取合理, 簇的数量都会等于或者接近于某一个值, 而与该值区别过大的簇数明显是不合理的。因此, 在得到基聚类成员之后, 可以将其中簇数相同的基聚类选取出来, 用这些高质量基聚类来训练一个启发式分类器, 之后通过训练好的分类器对剩余的数据点进行标签标定。

为了训练启发式分类器, 利用 2.2 节中介绍的 MMD 来对基聚类进行重标记, 使得在不同的基聚类中, 相似性更高的簇的标签是一致的。对于两个

基聚类

$$\pi^i = \{C_1^i, \dots, C_{n^i}^i\} \quad (10)$$

和

$$\pi^j = \{C_1^j, \dots, C_{n^j}^j\}, \quad (11)$$

需要将其中一个基聚类作为参考来重标记另一个基聚类中簇的标签。对于 π^i 中的任意一个簇 C_m^i 和 π^j 中的任意一个簇 C_n^j , 可以利用式(6)来计算这2个簇之间的MMD值。MMD值越小, 说明两个簇之间的分布越相似, 据此可以将不同基聚类中MMD值更小的簇的标签设定为一致即可。

为了更加清晰地展示上述一致化簇标签的过程, 本文选取Moons数据集作为示例, 计算结果如表1所示。表1展示了Moons数据集通过DBSCAN算法得到的2个基聚类 π^1 和 π^2 不同簇之间的MMD值, 两两簇之间最小的MMD值加粗显示。

表1 Moons数据集2个基聚类不同簇之间的MMD值

簇	C_1^2	C_2^2	C_3^2	C_4^2
C_1^1	0.2610	0.0086	0.3215	0.3823
C_2^1	0.2651	0.3208	0.0097	0.4085
C_3^1	0.0106	0.3951	0.2842	0.3463
C_4^1	0.4152	0.2694	0.4468	0.0124

基于不同基聚类簇之间的MMD值, LICE算法可通过贪心策略获得重标记结果: 首先找到表格中最小的值, 即0.0086, 并将其所在的行和列删除, 将簇 C_2^2 与簇 C_1^1 的标签对齐; 重复执行上述步骤, 直到所有簇的标签均已对齐。

在生成基聚类的过程中, DBSCAN算法可以自动识别噪声点Noise, 从而使得噪声点不会参与标签对齐的过程。在对簇数相等的基聚类进行重标记之后, 就可以得到用来训练分类器的数据子集 D -base, 将簇数不相等的那一部分的数据称为 D -pre, 对于初始数据集 D , 可得 D -base $\cup D$ -pre $\cup$ Noise= D 。确定 D -base之后, LICE算法就可以开始训练启发式分类器。由于K最近邻(K-nearest neighbor, KNN)算法的通用性比较强且易于实现, 因此, 将KNN算法作为启发式分类器。在分类器的训练完成之后, 利用此分类器对 D -pre中的样本点进行标签的标记, 每个样本点会得到一个分类标签, 而在基聚类的生成阶段, D -pre中的样本点会存在一个聚类标签, 利用MMD将此聚类标签与 D -base中的簇对

齐, 若对齐后的标签与聚类标签一致, 则说明聚类结果与分类结果一致, 可以信任此样本点的标记结果并将其加入 D -base中。对 D -pre中的所有样本点完成这样的标签标记之后, 第一轮标签迭代就完成了。继续按照同样的方式, 进行标签迭代, 在每一次迭代中, D -base中的样本点会变多, 而 D -pre中的样本点会相应地变少。为了达到算法执行时间与标签标记效率之间的平衡, 在本文中, 将标签迭代次数的上限设定为5, 5次迭代之后, 利用在 D -pre上训练的分类器对剩余的所有样本点进行标签标记。LICE算法的具体流程如算法1所示。

与目前常见的聚类集成构建策略不同的是, 现有算法大多数采用“数据相同, 算法不同”的模式, 而LICE算法采用的是“数据不同, 算法相同”的模式。数据不同, 即将整个数据集划分为多个概率分布一致的RSP数据块; 算法相同, 即对于每一个RSP数据块, 采用相同的聚类算法来得到基聚类成员。这种训练模式主要有以下2点优势: 其一, 将数据集进行分块能处理更大规模数据集的聚类问题, 且生成基聚类的时间会大大缩短; 其二, 利用相同的基聚类算法, 更容易得到簇数相同的基聚类, 同时降低算法实现的难度。此外, 现有的算法在得到基聚类后, 会将所有的基聚类一起进行集成, 而LICE算法会首先挑选出基聚类簇数相同的RSP数据块, 利用这些标签确定性较强的基聚类训练一个启发式分类器, 再迭代式地利用启发式分类器对标签不确定的RSP数据块中的样本点进行标签预测。

对于不同的数据集, LICE算法初始参数设置不一样, 不同数据集上LICE算法的初始参数如表2所示, 其中, m 表示RSP数据块的个数, ε 表示DBSCAN算法的领域半径, ξ 表示DBSCAN算法的密度阈值。

本文选取Moons数据集对上述的标签迭代过程进行可视化展示, LICE算法运行过程如图1所示。图1展示了迭代开始时和迭代结束后, 数据集每一个簇的变化情况。图1(a)、图1(c)、图1(e)、图1(g)、图1(i)分别表示开始迭代之前数据集的不同簇情况, 图1(b)、图1(d)、图1(f)、图1(h)、图1(j)分别表示经过5次迭代之后数据集的簇情况。相比迭代之前(即第0次迭代), 迭代之后每一个簇确定标签的样本点数量有明显的

算法1: LICE 算法

输入: 数据集 D , 基学习器个数 m , DBSCAN 算法的参数 ε 和 ξ 。

输出: 最终聚类集成结果 $\Pi^* = \{C_1, C_2, \dots, C_k\}$ 。

(1) 将数据集 D 划分为 m 个 RSP 数据块 $\{D_1, D_2, \dots, D_m\}$, 利用 DBSCAN 算法对每个数据块进行聚类。

for $i = 1, 2, \dots, m$ do

$D_i \leftarrow \text{DBSCAN}(\varepsilon, \xi)$;

end for

基聚类结果 $\Pi = \{\pi^1, \dots, \pi^m\}$ 。

(2) 将(1)中识别出的噪声点(聚类标签为-1的点)提取出, 并组成合集 Noise, 选取 $\{\pi^1, \dots, \pi^m\}$ 中簇数相同的基聚类 $\{\pi^1, \dots, \pi^l\}$ ($l \leq m$), 利用 MMD 进行标签对齐得到 D -base。

随机选择 $\Pi^1 = \{\pi^1, \dots, \pi^l\}$ 中的 π^i 作为参考聚类结果;

$\Pi^1 = \Pi^1 - \{\pi^i\}$;

Repeat

随机选择 Π^1 中的 $\pi^j = \{C_1^j, \dots, C_k^j\}$ 以同 π^i 对齐;

for $u = 1, 2, \dots, k$ do

for $v = 1, 2, \dots, k$ do

利用式(6)计算 C_u^i 与 C_v^j 之间的 MMD 值;

end for

end for

得到一个 $k \times k$ 的 MMD 矩阵, 记为 $\Omega = \{(p, q) \mid 1 \leq p, q \leq k\}$;

for $r = 1, 2, \dots, k$ do

$(p, q) = \arg \min_{(u, v) \in \Omega} \text{MMD}[C_u^i, C_v^j, H]$;

将 C_q^j 重标记为 C_p^i ;

$\Omega = \Omega - \{(p, w) \mid (p, w) \in \Omega\} \cup \{(w, q) \mid (w, q) \in \Omega\}$;

end for

$\Pi^1 = \Pi^1 - \{\pi^j\}$;

Until $\Pi^1 = \emptyset$

得到 D -base $= \{C_1, C_2, \dots, C_k\}$, D -pre $= D - D$ -base - Noise, n -pre $= \text{len}(D$ -pre)。

(3) 针对 D -pre, 利用 MMD 与 D -base 的标签对齐得到 Π^2 , 其过程类似于(2)。用 D -base 训练 KNN 分类器 Classifier, 利用此分类器预测 D -pre 中样本点的标签, 若此标签与 Π^2 中标签一致, 则将此样本点移出 D -pre 并加入 D -base。

$\Pi^2 \leftarrow \text{MMD}(D$ -base, D -pre);

得到 $\Pi^2 = \{C_1, C_2, \dots, C_k\}$;

Classifier $\leftarrow$ KNN(D -base); // 基于 D -base 训练启发式分类器 KNN

for $i = 1, 2, \dots, n$ -pre do

$y \leftarrow \text{Classifier}(D$ -pre[i]); // 使用启发式分类器 KNN 对 D -pre 进行标签标记;

找到 D -pre[i] 在 Π^2 中的标签 y' ;

if $y = y'$

D -pre $= D$ -pre - D -pre[i];

D -base $= D$ -base $\cup D$ -pre[i];

end if

end for

(4) 重复步骤(3), 直到迭代次数达到 5 次。

(5) 利用 D -base 训练的启发式分类器预测 D -pre 和 Noise 的标签。

表2 不同数据集上 LICE 算法的初始参数

数据集	m	ε	ξ
SZU	5	0.02	5
SmileFace	5	0.04	5
Moons	5	0.04	16
LS	5	0.42	22
IS	5	0.28	12
Letters	22	0.26	3
ODR	10	1.4	4
SPF	10	1.4	4

次标记出更多的可信样本。

3.3 时间复杂度分析

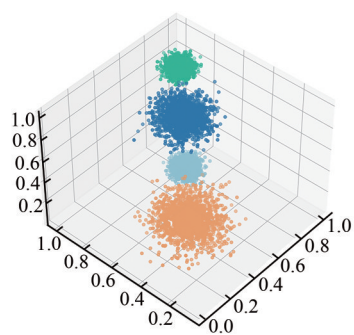
LICE 算法的执行时间由 5 部分组成: 第一, RSP 数据块的生成; 第二, 在每个 RSP 数据块上运行 DBSCAN 算法; 第三, 利用 MMD 进行标签对齐; 第四, 分类器的训练以及样本点标签的多次迭代; 第五, 利用迭代之后的分类器对最后一部分样本点进行打标。接下来将从这 5 个部分对 LICE 算法的时间复杂度进行分析。

RSP 数据块生成的本质是连续 2 次无放回随机抽样, 原数据集的样本点数量为 n , 则每一次的随机抽样时间复杂度为 $O(n)$, 因此, RSP 数据块生成的总体时间复杂度为 $O(n^2)$ 。

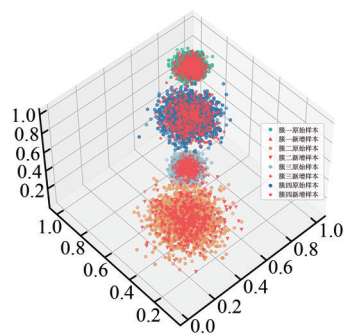
对 m 个 RSP 数据块分别进行 DBSCAN 聚类, 这个过程可以通过多线程并行来实现, 因此, 只需要计算其中一个 RSP 数据块进行聚类的时间复杂度即可。在 RSP 数据划分阶段, 本文将数据集划分成了样本点数量相等的 m 个数据块, 因此, 每个数据块的样本点数量为 n/m 。DBSCAN 算法的时间复杂度由多种因素组成, 首先, 对于每个数据点, 需要找到其 ε -领域内的所有点, 最坏情况下, 领域查询的时间复杂度为 $O(n/m \log_m(n/m))$; 其次, 对于每一个数据点, 需要判断其是否为核心点, 最坏情况下, 这个步骤的时间复杂度为 $O(n/m)$; 最后, 对于每个数据块, 需要确定核心点之间的密度可达关系, 以构建聚类, 由于每个核心点都可能需要与其他所有的核心点进行比较, 这个过程的时间复杂度为 $O((n/m)^2)$ 。综上, 在每个 RSP 数据块上进行 DBSCAN 聚类的时间复杂度为 $O((n/m)^2)$ 。

利用 MMD 进行标签对齐, 时间复杂度主要来源于两个方面, 首先, 对于每一个簇, 需要计算其聚类中心, 样本点总量为 n , m 个数据块中簇的总

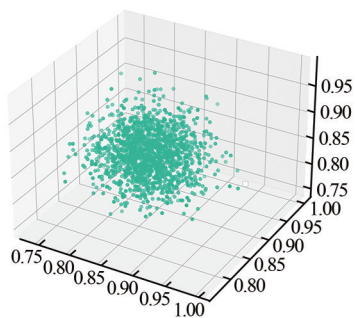
增加, 其中红色的点表示迭代过程中新增的确定标签的样本点。从图 1 中可以发现, 随着标签迭代次数的增加, 可信标签标记的样本点越来越多, 从而使得启发式分类器的性能不断被强化, 进而能够再



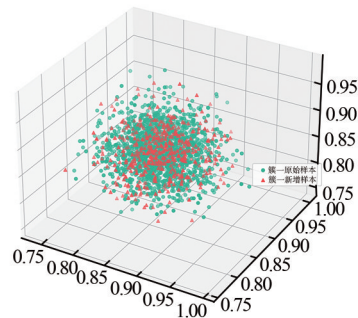
(a) 第0次迭代的聚类情况



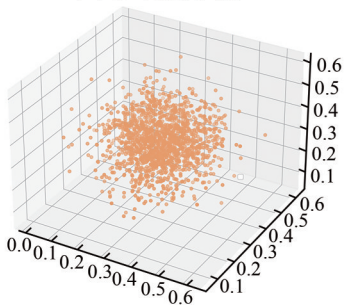
(b) 第5次迭代的聚类情况 (新增样本点3 653个)



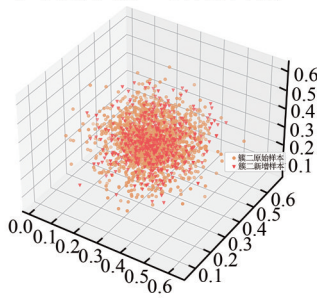
(c) 第0次迭代的簇一



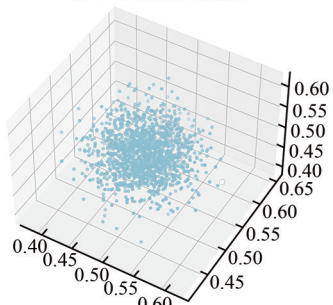
(d) 第5次迭代的簇一 (新增样本点876个)



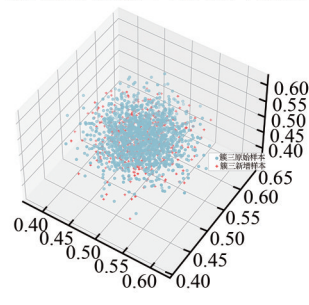
(e) 第0次迭代的簇二



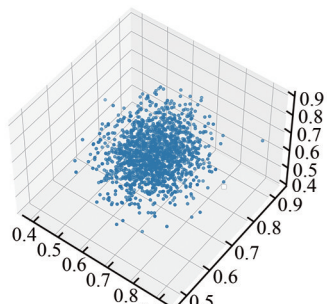
(f) 第5次迭代的簇二 (新增样本点913个)



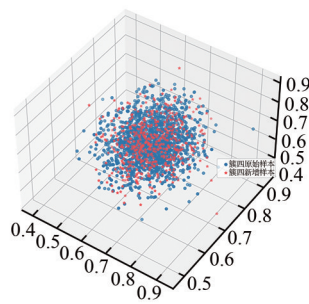
(g) 第0次迭代的簇三



(h) 第5次迭代的簇三 (新增样本点958个)



(i) 第0次迭代的簇四



(j) 第5次迭代的簇四 (新增样本点906个)

图1 Moons数据集聚类的变化

量为 km ，则计算聚类中心的时间复杂度为 $O(kmn)$ ；其次，计算2个分布之间的MMD值的时间复杂度与样本数的平方成正比，即 $O(n^2)$ 。因此，利用MMD进行标签对齐的时间复杂度为 $O(n^2)$ 。

在启发式分类器训练阶段，KNN分类器需要将训练数据存储起来，便于在分类时计算新样本与训练样本之间的距离，并根据距离选择最近的若干个邻居。因此，KNN算法的训练时间可以忽略不计，且这个训练时间不会随着训练数据的增加而增加。在利用分类器生成样本点的标签的过程中，其时间复杂度 $O(dn^2 + dn \log_d(n))$ 取决于样本点的数量和特征维度，其中 d 是特征维度， n 是样本点数量，而迭代次数为LICE算法设定的常数，所以，分类器的训练和标签迭代的过程时间复杂度为 $O(dn^2 + dn \log_d(n))$ 。

利用标签迭代之后的分类器对最后一部分样本点打标，这个过程与上一个过程类似，其时间复杂度也为 $O(dn^2 + dn \log_d(n))$ 。

综上，LICE算法的总时间复杂度为 $O(n^2) + O((n/m)^2) + O(dn^2 + dn \log_d(n)) = O(dn^2)$ 。

4 实验验证与分析

4.1 聚类算法效果评价指标

评价指标用于检测最终聚类集成结果的好坏，是评价聚类是否有效的标准。本文主要采用以下4个指标：NMI、ARI、FMI和Purity。这4个指标的值越大表示聚类效果越接近于真实结果。

NMI值用于测量2个随机分布之间的共享信息，其取值范围是 $[0, 1]$ ，NMI值越大，表示聚类效果越好。其计算式为：

$$NMI(l', l^T) = \frac{\sum_{i=1}^{n'} \sum_{j=1}^{n^T} n_{ij} \log \frac{n_{ij} n'}{n_i n_j^T}}{\sqrt{\sum_{i=1}^{n'} n'_i \log \frac{n'_i}{n} \sum_{j=1}^{n^T} n_j^T \log \frac{n_j^T}{n}}} \quad (12)$$

其中， l' 和 l^T 分别表示聚类的标签和真实的标签， n' 和 n^T 分别表示 l' 和 l^T 中簇的数量， n'_i 是 l' 的第 i 个簇中数据点的个数， n_j^T 是 l^T 的第 j 个簇中数据点的个数， n_{ij} 是 l' 的第 i 个簇和 l^T 的第 j 个簇共享样本点的个数。

ARI值是通过2个簇中一致或不一致的样本点的数量来计算的，它表示的是2个聚类的差异性，

其定义为：

$$ARI(l', l^T) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00} + N_{01})(N_{01} + N_{11}) + (N_{00} + N_{10})(N_{10} + N_{11})} \quad (13)$$

其中， N_{11} 表示 l' 、 l^T 中都出现在同一个簇中样本对的数量， N_{10} 表示在 l' 中出现在同一个簇中，但是在 l^T 出现在不同簇中样本对的数量， N_{01} 表示在 l^T 中出现在同一个簇中，但是在 l' 出现在不同簇中样本对的数量， N_{00} 表示 l' 、 l^T 中都出现在不同簇中样本对的数量。

FMI计算的是两个聚类之间的相似性，它定义为精度与召回率的几何平均值，其定义为：

$$FMI(l', l^T) = \frac{N_{11}}{\sqrt{(N_{11} + N_{01})(N_{11} + N_{10})}} \quad (14)$$

Purity是一种较为简单的评价方法，它直接计算正确聚类的样本点占有所有样本点的比例，其计算式为：

$$Purity(l', l^T) = \frac{1}{N} \sum_k \max_j |w_k \cap c_j| \quad (15)$$

其中， n 表示样本点总数， $l' = \{w_1, w_2, \dots, w_K\}$ 表示聚类得到的簇，而 $l^T = \{c_1, c_2, \dots, c_J\}$ 表示数据集的真实簇。

4.2 实验数据集与实验环境

本次实验选取了8个数据集，其中5个为来自UCI的真实数据集，3个为仿真数据集，数据集信息如表3所示。本实验采取NMI、ARI、FMI和Purity这4个指标作为模型的评判标准，且所有实验结果均是重复10次的独立实验测试的平均值。实验中的算法均使用Python语言实现。所有的实验均在Intel(R) Core(TM) 3.41 GHz i7-6700 CPU以及8 GB内存配置的计算机上进行。

4.3 RSP数据块融合对聚类效果的影响

本节将验证RSP数据块融合对聚类效果的影响。本文将对2种基聚类的处理机制进行对比，一种是对簇数相同的RSP数据块上的基聚类进行标签对齐，另一种是直接对所有RSP数据块上的基聚类结果进行统计，以此来证明选取簇数相同的RSP数据块的基聚类进行融合能够起到改善聚类集成效果的作用。

图2展示了在表3中挑选的6个代表性数据集

表3 数据集信息

数据集	来源	属性个数	样本个数	类标个数
SZU	仿真	2	35 000	3
SmileFace	仿真	2	35 000	5
Moons	仿真	3	35 000	4
LS	UCI	36	6 435	6
IS	UCI	19	2 310	7
Letters	UCI	16	20 000	26
ODR	UCI	64	5 620	10
SPF	UCI	27	1 941	7

上的对比结果, 其中红色柱代表对簇数相同的RSP数据块上的基聚类进行标签对齐之后的聚类结果, 蓝色柱代表直接对所有RSP数据块上的基聚类进行平均之后的聚类结果。通过图2可知, 在所选的6个数据集上, 挑选RSP之后融合基聚类的训练策略对应的NMI值和FMI值均高于直接对所有RSP数据块上的基聚类进行平均的训练策略, 其中, NMI值平均高出11.27%, FMI值平均高出12.36%。这表明, 选择簇数相同的RSP数据块进行融合, 能够有效地排除簇数不同的数据块对标签标注造成的影响, 使得初始分类器能够更加高效地使用不同RSP数据块上的聚类信息, 有效降低样本标注过程中的不确定性, 为样本提供更加确定的标签标注。

4.4 可行性验证

本节实验验证LICE算法的主要部分, 即标签迭代部分的可行性。本节选取了ODR、LS、Letters和IS这4个数据集来验证标签迭代次数对集成效果的影响, 在每一个数据集上进行6组对照实验, 分别将标签迭代的次数设置为0、1、2、3、4、5, 而LICE算法的其他参数则参照表2的设置。在6组实验中, 除了迭代次数有变化之外, 其他参

数均保持一致。实验结果如图3和图4所示, 所有的实验结果均是重复10次独立训练的平均结果。

图3表示在不同的迭代次数下, 4个数据集对应的NMI、ARI、FMI和Purity的变化情况。在ODR和Letters数据集上, 随着迭代次数的增加, 4条折线都呈逐步上升的趋势, 其中, ODR数据集的Purity值在迭代次数为2时就接近收敛, Letters数据集的ARI值在迭代3次时接近收敛, 而这2个数据集的其他指标也都随着迭代次数的增加而稳步提升。对于LS数据集, 除了Purity值在迭代次数为3时发生了微小的扰动以外, 其他的指标也都是逐步上升的趋势, FMI值和ARI值在迭代次数为4时接近收敛。IS数据集的Purity值在迭代的过程中几乎没有变化, 其他的3个指标随着迭代次数的增加而上升。平均而言, 第5次迭代相比第0次迭代, NMI、ARI、FMI和Purity值分别提升了17.23%、16.75%、31.29%和12.37%。

由图4可以看到, 在ODR、LS、Letters和IS这4个数据集上, 在每一次的迭代中, 都会有一部分样本点新加入D-base中, 而随着迭代次数的增加, 每一次新加入D-base的样本点数量会越来越少, 而且变少的速度越来越慢。相应地, D-base总样本总点数量的增加也越来越慢, 当迭代达到5次时, 新加入D-base的样本点数量只有10个以下, 接近于0。

总体而言, 在这4个数据集中, 随着迭代次数的增加, LICE算法在4个评价指标上的表现呈越来越好的趋势。而在迭代次数达到一定的数量时, 部分评价指标的值会接近收敛状态。在每一次的迭代中, 都会有一部分的样本点加入D-base中, 且新加入D-base的样本点都比上一次更少, 说明在

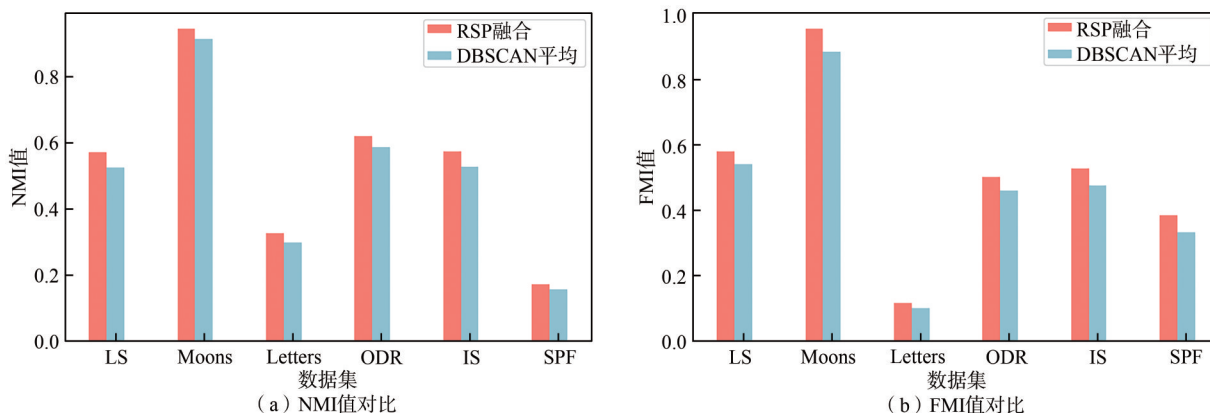


图2 部分RSP融合与全部RSP平均之间的聚类集成效果对比

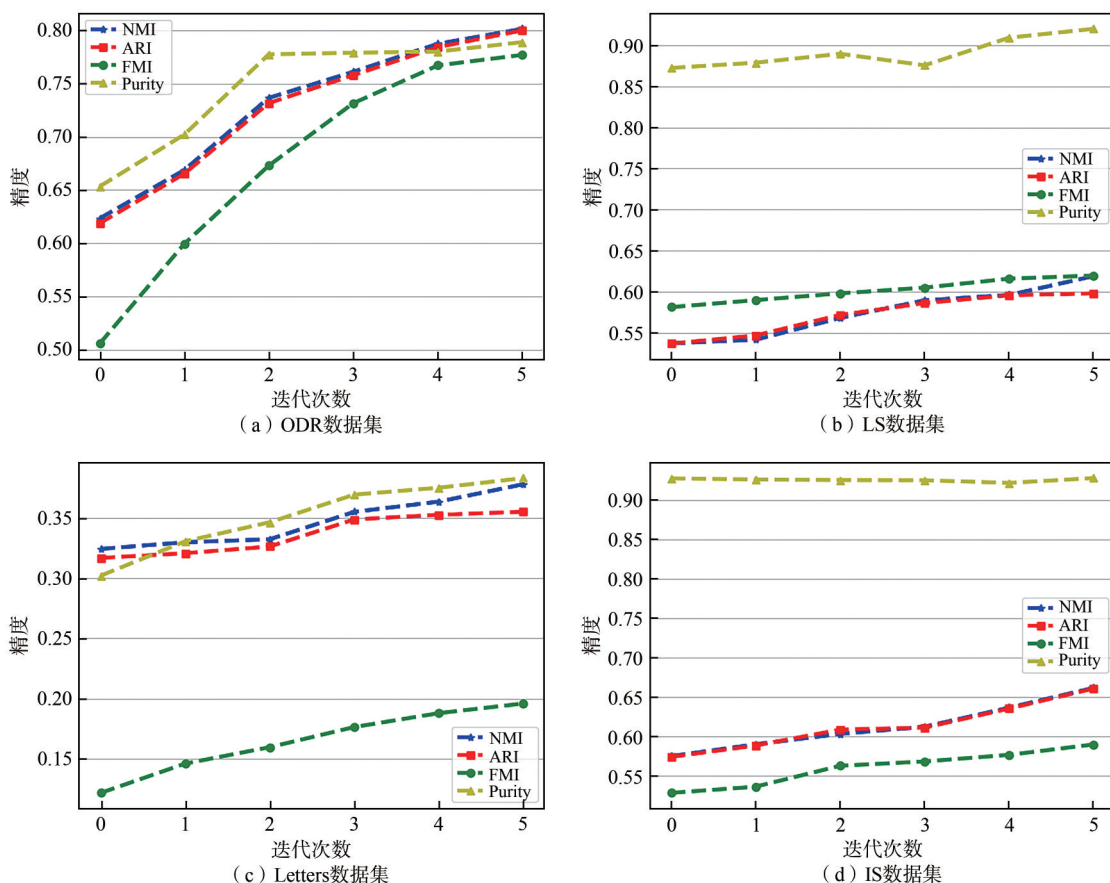


图3 不同迭代次数对应的聚类结果

迭代过程的前期,很大一部分的样本点由于聚类标签和分类标签的一致而被加入 D -base 中,这表明,根据聚类标签与分类标签是否一致来将样本点进行划分是完全可行的。综上,通过测试迭代过程中4个评价指标的变化和样本点的变化证实了LICE算法的可行性。

4.5 有效性验证

为了验证LICE算法的有效性,本文选取了7个经典的聚类集成算法作为对比,分别为LWEA^[7]、LWGP^[7]、EAC^[8]、HBGF^[9]、元聚类算法(meta-clustering algorithm, MCLA)^[10]、基于簇的相似性划分算法(cluster-based similarity partitioning algorithm, CSPA)^[10]和EC-CMS^[11]算法,在8个数据集上进行了对比实验。本部分实验重复10次独立实验并取平均值,采用了NMI、ARI、FMI和Purity这4个指标评估聚类集成算法的性能,实验结果如表4~表7所示。

从表4~表7中可以看到,在所有4个评价指标上,LICE算法在选定的数据集上的表现均优于其他算法。以IS数据集为例,LICE算法的NMI值和

FMI值分别为0.6613和0.5895,相较于CSPA的0.6100和0.5708,分别提升了8.41%和3.28%;对于Letters数据集,LICE算法在Letters数据集上的FMI和Purity值分别为0.1960和0.3828,比LWGP算法分别高出3.43%和15.58%。通过分析评价指标值可以看出,由于图分割算法并不是专门为聚类而设计的^[31],而HBGF、MCLA和CSPA这3个聚类集成算法依赖于图分割算法,聚类质量会受到限制,而LICE算法不存在这样的限制,表明本文提出的LICE算法是一种有效的聚类集成解决方案。表中的N/A表示在本地实验环境下,由于数据量过大而不能得到实验结果,对于SZU、SmileFace和Moons这3个数据集,EC-CMS算法会存在内存溢出的问题,而LICE算法由于对数据集进行了RSP数据块生成,可以有效地处理大规模数据集的聚类问题。

5 结束语

现有的“数据相同,算法不同”式的聚类集成算法训练策略普遍存在处理大规模数据性能受限以

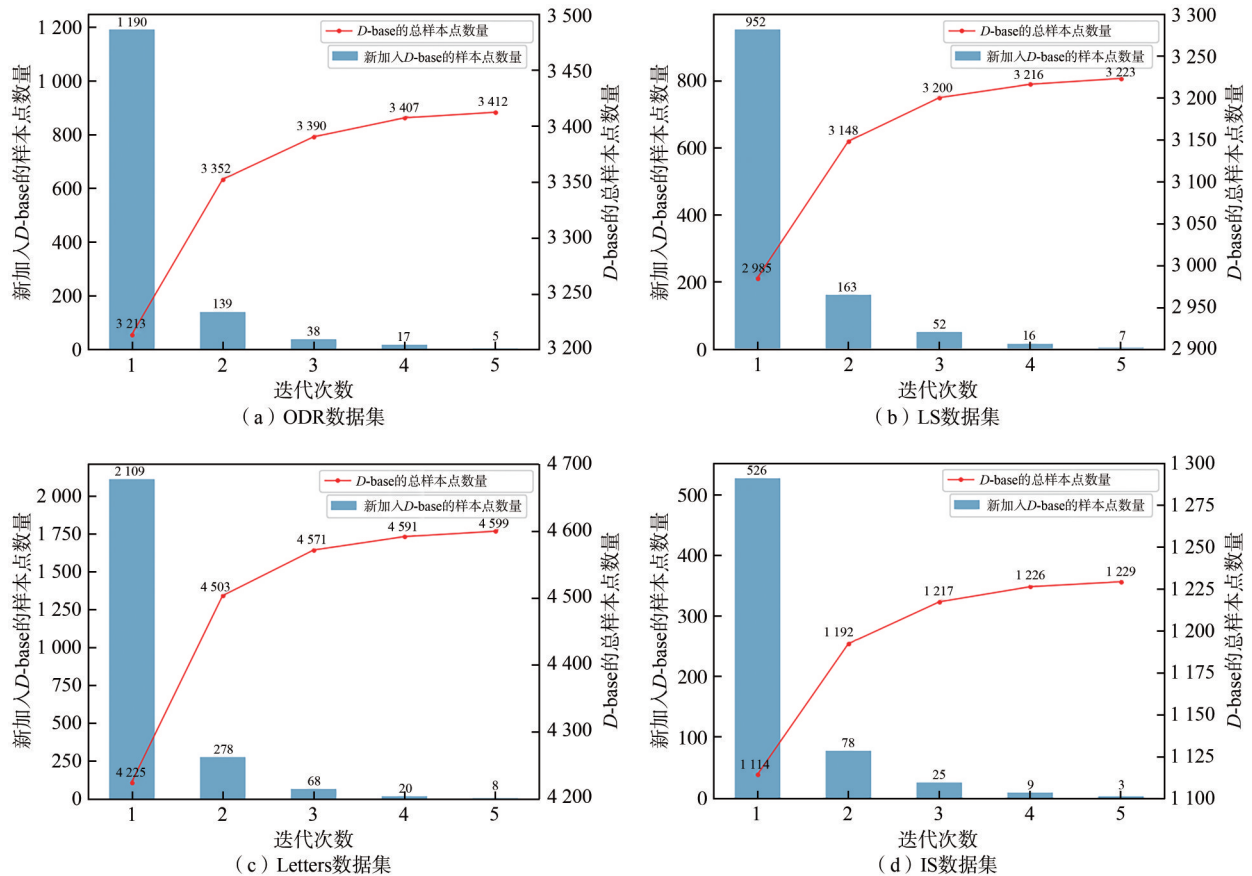


图4 不同迭代次数对应的样本点变化情况

表4 NMI指标下不同聚类算法的结果对比

数据集	LICE	LWEA	LWGP	EAC	HBGF	MCLA	CSPA	EC-CMS
SZU	0.9999	0.9784	0.9858	0.6524	0.6519	0.8175	0.6957	N/A
SmileFace	0.9999	0.9608	0.9688	0.7586	0.9630	0.7454	0.9687	N/A
Moons	0.9718	0.9302	0.9588	0.9403	0.9647	0.9579	0.6957	N/A
LS	0.6191	0.6160	0.6084	0.5810	0.5263	0.5610	0.4764	0.3535
IS	0.6613	0.6210	0.6290	0.5990	0.6398	0.6210	0.6100	0.5976
Letters	0.3777	0.3727	0.3709	0.3719	0.3701	0.3708	0.2800	0.3459
ODR	0.8014	0.7964	0.7868	0.7810	0.6133	0.8011	0.7601	0.7741
SPF	0.1917	0.1670	0.1569	0.1874	0.1572	0.1392	0.1154	0.1869

表5 ARI指标下不同聚类算法的结果对比

数据集	LICE	LWEA	LWGP	EAC	HBGF	MCLA	CSPA	EC-CMS
SZU	0.9999	0.9796	0.9826	0.6654	0.6582	0.8214	0.7053	N/A
SmileFace	0.9999	0.9654	0.9702	0.7652	0.9667	0.7506	0.9775	N/A
Moons	0.9715	0.9276	0.9627	0.9642	0.9659	0.9534	0.7003	N/A
LS	0.5981	0.5680	0.5800	0.5538	0.4424	0.4947	0.3839	0.1756
IS	0.6605	0.5860	0.4730	0.5500	0.1986	0.3615	0.1964	0.3741
Letters	0.3524	0.2000	0.1624	0.1342	0.1378	0.1611	0.0973	0.1654
ODR	0.7999	0.7820	0.7630	0.6981	0.7420	0.7346	0.7016	0.7268
SPF	0.1806	0.0854	0.0857	0.1542	0.1649	0.1728	0.1613	0.0907

表6 FMI指标下不同聚类算法的结果对比

数据集	LICE	LWEA	LWGP	EAC	HBGF	MCLA	CSPA	EC-CMS
SZU	1.0000	1.0000	1.0000	0.7438	0.7411	0.8954	0.7875	N/A
SmileFace	1.0000	0.9709	0.9876	0.7682	0.9771	0.7254	0.9554	N/A
Moons	0.9855	0.9841	0.9813	0.9641	0.9826	0.6701	0.9816	N/A
LS	0.6198	0.5345	0.5455	0.5826	0.5082	0.6002	0.5121	0.4216
IS	0.5895	0.5713	0.5683	0.5672	0.5879	0.5697	0.5708	0.4949
Letters	0.1960	0.1939	0.1895	0.1670	0.1878	0.1864	0.1873	0.1746
ODR	0.7769	0.7420	0.7563	0.6887	0.7654	0.7230	0.6987	0.6851
SPF	0.4155	0.3896	0.3907	0.3368	0.4015	0.3985	0.3941	0.3684

表7 Purity指标下不同聚类算法的结果对比

数据集	LICE	LWEA	LWGP	EAC	HBGF	MCLA	CSPA	EC-CMS
SZU	1.0000	1.0000	1.0000	0.8654	0.8972	0.9358	0.8843	N/A
SmileFace	1.0000	0.9854	0.9935	0.8821	0.9639	0.8726	0.9871	N/A
Moons	0.9927	0.9843	0.9786	0.9653	0.9872	0.8965	0.9687	N/A
LS	0.9198	0.7747	0.7883	0.6612	0.6899	0.7117	0.6338	0.5079
IS	0.9276	0.6842	0.7060	0.7665	0.9236	0.9172	0.9132	0.8874
Letters	0.3828	0.3309	0.3312	0.2963	0.2956	0.3003	0.3048	0.3215
ODR	0.7889	0.7682	0.7625	0.7355	0.7346	0.7348	0.7706	0.7486
SPF	0.7594	0.4678	0.4639	0.5180	0.5080	0.5251	0.5136	0.6795

及共识函数适应性不强的缺点。为此,本文采用“数据不同,算法相同”式的训练策略构建了LICE算法。首先,LICE算法在原始数据集的RSP数据块上训练若干基聚类器,其中RSP数据块是与原始数据集保持概率分布一致性的数据子集。接着,利用最大平均差异准则对聚类簇数相同的基聚类结果进行融合,并基于标签确定的RSP数据块训练一个启发式分类器。之后,迭代式地利用启发式分类器对标签不确定的RSP数据块中的样本点进行标签预测,采用分类标签与聚类标签一致的样本点提升启发式分类器的性能,直至所有样本点的标签确定完成。最后,通过一系列可信的实验对LICE算法的可行性和有效性进行验证,结果显示在代表性数据集上,随着迭代次数的增加,LICE算法对应的NMI、ARI、FMI以及Purity值在第5次迭代时相比于迭代起始分别平均提升了17.23%、16.75%、31.29%和12.37%;与7种经典的聚类集成算法相比,在选用的数据集上,NMI、ARI、FMI和Purity值分别平均提升了11.76%、16.50%、9.36%和14.20%。实验证实了LICE算法是一种高效合理的、能够处理大数据聚类问题的聚类集成算法。

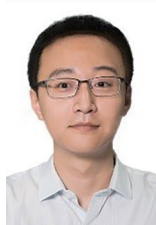
未来,将从以下3个方面进行更深入的研究,以进一步改进LICE算法的性能。第一,探索更加高效的数据分布一致性度量方法,用于进行标签对齐;第二,引入概率模型构建启发式分类器,以增强标签标定的可信性;第三,实现Spark版本的LICE算法,使其具备处理大规模数据聚类问题的能力。

参考文献:

- [1] LIU H Q, ZHANG Q, ZHAO F. Interval fuzzy spectral clustering ensemble algorithm for color image segmentation[J]. *Journal of Intelligent and Fuzzy Systems*, 2018, 35(5): 5467-5476.
- [2] 柯善军, 聂成洋, 王钰苗, 等. 基于集成学习与聚类联合标注的多模态个体情绪识别[J]. *智能科学与技术学报*, 2024, 6(1): 76-87.
KE S J, NIE C Y, WANG Y M, et al. Multimodal individual emotion recognition with joint labeling based on integrated learning and clustering[J]. *Chinese Journal of Intelligent Science and Technology*, 2024, 6(1): 76-87.
- [3] STOLZ T, HUERTAS M E, MENDOZA A. Assessment of air quality monitoring networks using an ensemble clustering method in the three major metropolitan areas of Mexico[J]. *Atmospheric Pollution Research*, 2020, 11(8): 1271-1280.
- [4] 何玉林, 黄哲学. 大规模数据聚类算法的研究进展[J]. *深圳大学学报(理工版)*, 2019, 36(1): 4-17.
HE Y L, HUANG Z X. A review on clustering algorithms for large-scale data sets[J]. *Journal of Shenzhen University (Science and Engineering)*, 2019, 36(1): 4-17.
- [5] SALLOUM S, HUANG J Z, HE Y L. Random sample partition: a distributed data model for big data analysis[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(11): 5846-5854.
- [6] GRETTON A, BORGWARDT K M, RASCH M J, et al. A kernel two-sample test[J]. *The Journal of Machine Learning Research*, 2012, 13(1): 723-773.
- [7] HUANG D, WANG C D, LAI J H. Locally weighted ensemble clustering[J]. *IEEE Transactions on Cybernetics*, 2018, 48(5): 1460-1473.
- [8] 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.
- [9] FERN X Z, BRODLEY C E. Solving cluster ensemble problems by bipartite graph partitioning[C]//*Proceedings of the Twenty-First International Conference on Machine Learning*. New York: ACM Press, 2004: 36.
- [10] STREHL A, GHOSH J. Cluster ensembles - a knowledge reuse framework for combining partitionings[J]. *Journal of Machine Learning Research*, 2002, 3: 583-617.
- [11] JIA Y, TAO S, WANG R, et al. Ensemble clustering via co-association matrix self-enhancement[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, 35(8): 11168-11179.
- [12] JAIN A K. Data clustering: 50 years beyond K-means[J]. *Pattern Recognition Letters*, 2010, 31(8): 651-666.
- [13] BAI L, LIANG J Y, CAO F Y. A multiple k-means clustering ensemble algorithm to find nonlinearly separable clusters[J]. *Information Fusion*, 2020, 61: 36-47.
- [14] MINAEI-BIDGOLI B, TOPCHY A, PUNCH W F. Ensembles of partitions via data resampling[C]//*Proceedings of the International Conference on Information Technology: Coding and Computing*. Piscataway: IEEE Press, 2004: 188-192.
- [15] FISCHER B, BUHMANN J M. Path-based clustering for grouping of smooth curves and texture segmentation[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 25(4): 513-518.
- [16] TOPCHY A, JAIN A K, PUNCH W. Combining multiple weak clusterings[C]//*Proceedings of the Third IEEE International Conference on Data Mining*. Piscataway: IEEE Press, 2003: 331-338.
- [17] FERN X Z, BRODLEY C E. Random projection for high dimensional data clustering: a cluster ensemble approach[C]//*Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. New York: ACM Press, 2003: 186-193.
- [18] ILC N. Weighted cluster ensemble based on partition relevance analysis with reduction step[J]. *IEEE Access*, 2020, 8: 113720-113736.
- [19] BAGHERINIA A, MINAEI-BIDGOLI B, HOSSEINZADEH M, et al. Reliability-based fuzzy clustering ensemble[J]. *Fuzzy Sets and Systems*, 2021, 413: 1-28.
- [20] LIANG Y N, REN Z G, WU Z Z, et al. Scalable spectral ensemble clustering via building representative co-association matrix[J]. *Neurocomputing*, 2020, 390: 158-167.
- [21] SHI Y F, YU Z W, CHEN C L P, et al. Consensus clustering with co-association matrix optimization[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, 35(3): 4192-4205.
- [22] BIAN Z K, QU J, ZHOU J, et al. Weighted adaptively ensemble clustering method based on fuzzy co-association matrix[J]. *Information Fusion*, 2024, 103: 102099.
- [23] ZHOU P, WANG X, DU L, et al. Clustering ensemble via structured hypergraph learning[J]. *Information Fusion*, 2022, 78: 171-179.
- [24] DUDOIT S, FRIDLAND J. Bagging to improve the accuracy of a

- clustering procedure[J]. Bioinformatics, 2003, 19(9): 1090-1099.
- [25] KUHN H W. The Hungarian method for the assignment problem[J]. Naval Research Logistics Quarterly, 1955, 2(1/2): 83-97.
- [26] ZHOU Z H, TANG W. Clusterer ensemble[J]. Knowledge-Based Systems, 2006, 19(1): 77-83.
- [27] KHEDAIRIA S, KHADIR M T. A multiple clustering combination approach based on iterative voting process[J]. Journal of King Saud University - Computer and Information Sciences, 2022, 34(1): 1370-1380.
- [28] CRISTOFOR D, SIMOVICI D. Finding median partitions using information-theoretical-based genetic algorithms[J]. Journal of Universal Computer Science, 2002, 8(2): 153-172.
- [29] LI T Y, SHU X Y, WU J, et al. Adaptive weighted ensemble clustering via kernel learning and local information preservation[J]. Knowledge-Based Systems, 2024, 294: 111793.
- [30] ZHANG H S, WANG Y, CHEN Y P, et al. Consistency-oriented clustering ensemble via data reconstruction[J]. Applied Intelligence, 2024, 54(20): 9641-9654.
- [31] 周志华. 集成学习：基础与算法[M]. 北京：电子工业出版社, 2020: 14-16.
- ZHOU Z H. Ensemble learning[M]. BeiJing: Publishing House of Electronics Industry, 2020: 14-16.

[作者简介]



何玉林（1982-），男，博士，人工智能与数字经济广东省实验室（深圳）研究员、高级工程师。主要研究方向为数据挖掘与机器学习算法、大数据处理与分析方法、大数据近似计算技术、多样本统计分析理论与方法。



杨锦（1999-），男，深圳大学计算机与软件学院硕士研究生。主要研究方向为数据挖掘与机器学习算法及其应用。



黄哲学（1959-），男，博士，人工智能与数字经济广东省实验室（深圳）教授、博士生导师。主要研究方向为数据挖掘、机器学习、大数据处理与分析、大数据系统计算技术。



尹剑飞（1974-），男，博士，深圳大学计算机与软件学院副教授。主要研究方向为数据挖掘、机器学习、多目标优化、人工智能。