

## 近似质心在区间二型模糊聚类中的应用

孙鑫, 郑婷婷, 李清, 王志强  
安徽大学 数学科学学院, 合肥 230601

**摘要:**针对现有质心求解算法仍具有较高计算复杂度, 导致区间二型模糊 C 均值聚类算法(Interval Type-2 Fuzzy C-Means, IT2FCM)运行速度不理想问题, 提出了半数迭代法和一次迭代法两种近似质心求解算法。首先, 在直接求解转换点问题质心求解算法(A Direct Approach for Determining the Switch Points in the Karnik - Mendel Algorithm, DA)的基础上, 借助二分查找思想, 构造出基于二分查找的质心求解算法; 接着, 以该算法为基础, 通过限制查找范围, 考虑两个转换点之间关系的性质和计算差值的技巧得到半数迭代法; 最后, 考虑只进行一次查找得到一次迭代法。在 UCI 上的 5 个数据集上(IRIS、SEEDS、WINE、WIFI\_LOCALIZATION 和 HTRU2)验证了两种算法的聚类性能并没有因为求解的是近似质心而降低; 进一步在 ANURAN CALLS 数据集上构造了 8 组数据量递增数据用于验证基于不同质心求解算法的 IT2FCM 和基于提出的近似质心求解算法的 IT2FCM 运行速度, 实验结果表明: 基于近似质心求解算法的 IT2FCM 运行速度较快, 所以提出的近似质心求解算法能够在一定程度上缓解 IT2FCM 复杂度过高的问题。

**关键词:**区间二型模糊 C 均值聚类算法; 二分查找; 近似质心

**中图分类号:**TP391 **文献标识码:**A **doi:**10.16055/j.issn.1672-058X.2023.0002.013

### Application of Approximate Centroid in Interval Type-2 Fuzzy Clustering

SUN Xin, ZHENG Tingting, LI Qing, WANG Zhiqiang

School of Mathematical Sciences, Anhui University, Hefei 230601, China

**Abstract:** In view of the high computational complexity of the existing centroid solving algorithms leading to the unsatisfactory running speed of interval type-2 fuzzy C-means (IT2FCM) clustering algorithm, two approximate centroid solving algorithms, including half iterative method and one-time iterative method, were proposed. Firstly, based on a direct approach for determining the switch points in the Karnik-Mendel algorithm, with the help of the idea of binary search, a centroid solving algorithm based on binary search was constructed. Based on this algorithm, by limiting the search range, and considering the nature of the relationship between the two conversion points and the skill of calculating the difference, the half iterative method was obtained. Finally, the one-time iterative method was obtained by considering only one search. Five data sets (IRIS, SEEDS, WINE, WIFI\_LOCALIZATION and HTRU2) on UCI verified that the clustering performances of the two algorithms were not reduced by solving approximate centroid. Further, eight groups of incremental data were constructed on ANURAN CALLS data set to compare the running speed of IT2FCM based on different centroid solving algorithms and IT2FCM based on the proposed approximate centroid solving algorithm. The experimental results showed that IT2FCM based on approximate centroid solving algorithm ran faster. Therefore, the proposed approximate centroid algorithm can alleviate the high complexity of IT2FCM to a certain extent.

**Keywords:** interval type-2 fuzzy C-means clustering algorithm; binary search; approximate centroid

**收稿日期:**2022-02-22 **修回日期:**2022-03-25 **文章编号:**1672-058X(2023)02-0085-09

**基金项目:**国家自然科学基金项目资助(61806001).

**作者简介:**孙鑫(1996—),男,安徽广德人,硕士研究生,从事模糊集与粒计算研究.

**通讯作者:**郑婷婷(1978—),女,博士,教授,硕士生导师,从事粒计算和知识发现研究. Email: tt-zheng@163.com.

**引用格式:**孙鑫,郑婷婷,李清,等. 近似质心在区间二型模糊聚类中的应用[J]. 重庆工商大学学报(自然科学版),2023,40(2): 85—93.

SUN Xin, ZHENG Tingting, LI Qing, et al. Application of approximate centroid in interval type-2 fuzzy clustering[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2023, 40(2): 85—93.

## 1 引言

聚类是将一组样本数据划分为不同簇的过程,目的是使簇类数据尽可能相似,簇间数据尽可能相异。由于没有外部标签,聚类算法是一类无监督学习算法。经典的聚类算法都是“硬划分”算法,即每个样本只能属于某一个簇,具有非此即彼性,所以无法处理现实中大量存在的模糊问题。随着 Zadeh<sup>[1]</sup> 模糊集理论的提出,有了处理模糊问题的工具,开始出现模糊聚类,其中最经典的是基于求解带约束非线性规划得出的 FCM 聚类算法<sup>[2]</sup>。FCM 由于能够处理样本数据亦此亦彼的特性,在图像分割、模式识别、信道均衡等领域均有着重要运用。FCM 中引入模糊系数  $m$ ,而  $m$  的选取对聚类性能的影响很大。为了管理和控制  $m$  的不确定性, Huang 等<sup>[3]</sup> 将 FCM 扩展到 IT2FCM;袁飞等<sup>[4]</sup> 扩展了基于点数据集的 IT2FCM,将其扩展到区间不确定数据的聚类中,提出了区间数的区间二型模糊 C 均值聚类算法。随着  $\alpha$ -plane 理论<sup>[5]</sup> 的出现,一般二型模糊集可以被  $\alpha$ -plane 截成许多特殊的区间二型模糊集,使得一般二型模糊集计算难度降低。Linda 等<sup>[6]</sup> 进一步将其扩展到一般二型模糊集上的模糊 C 均值聚类算法 (General Type-2 Fuzzy C-Means, GT2FCM)。

经典的 IT2FCM 选用的是 Karnik-Mendel Algorithm (KM)算法<sup>[7]</sup> 求解质心,由于用 KM 算法求解质心有很多重复计算,导致 IT2FCM 运行速度不理想。而质心是区间二型模糊集中重要的概念,之后出现了很多改进的算法:增强的 KM 算法<sup>[8]</sup> (Enhanced Karnik-Mendel Algorithms, EKM),相较于 KM 算法,它能够减少迭代次数,从而减少计算量达到节约计算成本的目的;DA 算法<sup>[9]</sup>,其最大特点是没有迭代过程,使得它在一些情况下优于之前的迭代算法;DAND 算法<sup>[10]</sup>,一种新的 DA 算法,与 DA 算法一样没有迭代过程,但是与 DA 算法最大的不同在于不再需要计算偏导数等。

在以往的研究中,研究者专注于对模糊系数  $m$  的建模与管理,从而使得模糊聚类性能得到提高,聚类效果更稳定,但也相应提高了算法的复杂度,使得相关模糊聚类算法只能处理小的数据集,无法应用于现实中大量数据的情况。影响算法复杂度的关键在于对质心的求解,降低质心求解的复杂度就能提高聚类算法的运算效率。

基于上述启发,邱存勇等<sup>[11]</sup> 提出一种基于加权平均初始化同时参考 EKM 算法的聚类算法。其主要创新之处在于:参考一些新改进的质心求解算法,最终选择从 DA 算法出发,改造质心求解算法;实验验证了

IT2FCM 对于近似质心不敏感,计算近似质心效率显然高于计算精确质心,受此启发提出两种近似质心求解算法。

本文其余部分组织如下:第一部分回顾区间二型模糊集及其质心、KM 算法、DA 算法和 IT2FCM 算法相关概念与内容;第二部分介绍两种通过 DA 算法改进的近似质心求解算法;第三部分通过几个经典的数据集来说明所提出算法的有效性;最后在第四部分中给出结论。

## 2 知识回顾

### 2.1 区间二型模糊集及其质心

一般二型模糊集的概念最初由 Zadeh<sup>[1]</sup> 提出,目的是为模糊逻辑系统提供额外的自由度。一般二型模糊集和一型模糊集差别在于二型模糊集的隶属度是用一个一型模糊集描述。一般二型模糊集的计算复杂度过高,Mendel 等<sup>[12]</sup> 由此提出了区间二型模糊集。

**定义 1<sup>[12]</sup> (区间二型模糊集)** 区间二型模糊集  $\tilde{A}$  是一个次隶属度为 1 的二型模糊集,其定义为  $\tilde{A} = \int_{x \in X} \int_{\mu \in J_x} 1/(x, \mu) = \int_{x \in X} [\int_{\mu \in J_x} 1/\mu]/x$  或  $\tilde{A} = \sum_{x \in X} [\sum_{\mu \in J_x} 1/\mu]/x$ ,这里  $J_x \subseteq [0, 1]$  是元素  $x$  关于  $\tilde{A}$  的主隶属度,  $\mu$  是  $J_x$  中某个主隶属度值。

显然,一个区间二型模糊集可视为所有嵌入式一型模糊集  $\sum_{x \in X} \mu/x$  组成的集合,这里  $\mu \in J_x$ 。

**定义 2<sup>[12-13]</sup> (不确定性足迹)** 区间二型模糊集主隶属函数的不确定性构成的有界区域叫作不确定性足迹(图 1),记作  $FOU$ ,即  $FOU(\tilde{A}) = \bigcup_{x \in X} x \times J_x$ 。

**定义 3<sup>[12]</sup> (上下隶属度)** 区间二型模糊集  $\tilde{A}$  的下隶属函数 ( $LMF(\tilde{A})$ ) 和上隶属函数 ( $UMF(\tilde{A})$ ) 是两个一型隶属函数,它们分别是  $FOU(\tilde{A})$  的下、上边界(图 1),下隶属度是  $FOU(\tilde{A})$  中所有最小隶属度  $\mu_{-A}(x)$  的并集;上隶属度函数是  $FOU(\tilde{A})$  中所有最大隶属度  $\mu_{+A}(x)$  的并集,即

$$LMF(\tilde{A}) = \{\mu_{-A}(x) | x \in X\}$$

$$UMF(\tilde{A}) = \{\mu_{+A}(x) | x \in X\}$$

对于离散型论域,不妨简记  $x_i$  关于区间二型模糊集  $\tilde{A}$  的最小隶属度  $\mu_{-A}(x_i)$  为  $\mu_{-i}$ ,最大隶属度  $\mu_{+A}(x_i)$  为  $\mu_{+i}$ ,  $\forall x_i \in X$ 。

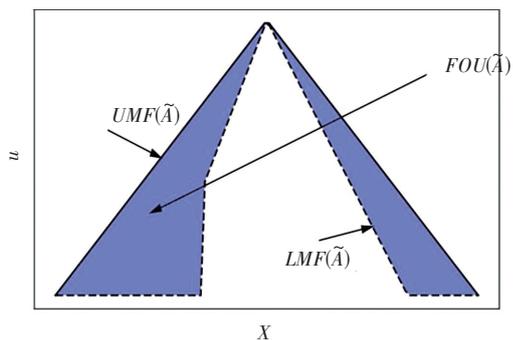


图 1 上下隶属函数和 FOU

Fig. 1 Upper and lower membership functions and FOU

质心是模糊聚类算法中的一项重要参数<sup>[14]</sup>。区间二型模糊集的质心由所有嵌入式一型模糊集的质心组合而成。

**定义 4<sup>[9]</sup>(一型模糊集质心)** 设论域  $X = \{x_1, x_2, \dots, x_N\}$ ,  $\tilde{A}$  是  $X$  上的一个区间二型模糊集, 其中的一个嵌入式一型模糊集记为  $\sum_{i=1}^N \mu_i/x_i$ , 这里  $\mu_i \in J_{x_i}$ , 则这个一型模糊集的质心  $v$  定义如下:

$$v = \frac{\sum_{i=1}^N x_i \mu_i}{\sum_{i=1}^N \mu_i} \quad (1)$$

如果每个嵌入式一型模糊集都如式(1)计算质心, 则整个区间二型模糊集的质心可用区间表示, 记为  $\tilde{v} = [v_l, v_r]$ 。其中,

$$v_l = \inf_{\mu_i \in J_{x_i}} \frac{\sum_{i=1}^N x_i \mu_i}{\sum_{i=1}^N \mu_i}, v_r = \sup_{\mu_i \in J_{x_i}} \frac{\sum_{i=1}^N x_i \mu_i}{\sum_{i=1}^N \mu_i}$$

取这个区间的中点即得到整个质心的中心  $v = \frac{v_l + v_r}{2}$ 。

### 2.2 KM 算法

KM 算法<sup>[7]</sup>是一种求解区间二型模糊集质心的算法。它指出在计算质心区间时, 不需要遍历计算每个嵌入式一型模糊集的质心, 而只需要找到数据集中的转换点  $L$  和  $R$ , 就有

$$v_l = \frac{\sum_{i=1}^L x_i \bar{\mu}_i + \sum_{i=L+1}^N x_i \underline{\mu}_i}{\sum_{i=1}^L \bar{\mu}_i + \sum_{i=L+1}^N \underline{\mu}_i} \quad (2)$$

$$v_r = \frac{\sum_{i=1}^R x_i \underline{\mu}_i + \sum_{i=R+1}^N x_i \bar{\mu}_i}{\sum_{i=1}^R \underline{\mu}_i + \sum_{i=R+1}^N \bar{\mu}_i} \quad (3)$$

可以发现, 它们满足

$$x_L \leq v_l \leq x_{L+1}, x_R \leq v_r \leq x_{R+1}$$

KM 算法能通过迭代的方式计算出  $v_l$  和  $v_r$ 。

### 2.3 DA 算法

DA 算法<sup>[9]</sup>是一种改进的求解质心的算法。通过式(1)求出所有偏导数  $\frac{\partial v}{\partial \mu_j}$ , 在偏导数为零的点的附近即是可能的转换点, 进而计算出质心, 实验仿真结果表明该方法优于其他迭代算法, 比如 KM 算法以及它的改进算法 EKM 等。

DA 算法核心在于如何表示对式(1)的偏导数  $\frac{\partial v}{\partial \mu_j}$ ,

该算法将偏导数表示成正负两部分之和:  $\frac{\partial v}{\partial \mu_j} = (pos_j +$

$neg_j) (\sum_{i=1}^N \mu_i)^2$ 。  $pos_j$  和  $neg_j$  分别表示如下:

$$pos_j = \begin{cases} 0, & j = 1 \\ (\sum_{i=1}^j \mu_i) \delta x_j + [(\sum_{i=1}^2 \mu_i) \delta x_3 + \dots + (\sum_{i=1}^{j-1} \mu_i) \delta x_j] \\ j \in [2, N] \end{cases}$$

$$neg_j = \begin{cases} - [(\sum_{i=j+1}^N \mu_i) \delta x_{j+1} + \dots + (\sum_{i=N-1}^N \mu_i) \delta x_{N-1}] - \\ (\sum_{i=N}^N \mu_i) \delta x_N, & j \in [1, N-1] \\ 0, & j = N \end{cases}$$

其中,  $\delta x_j = x_j - x_{j-1}$ , 这里  $\{x_j\}_{j=1}^N$  按升序排列。由于  $(\sum_{i=1}^N \mu_i)^2 \geq 0$ , 所以  $\frac{\partial v}{\partial \mu_j}$  的符号主要取决于  $pos_j + neg_j$ 。

### 2.4 区间二型模糊集上的模糊 C 均值聚类

IT2FCM<sup>[3]</sup>是基于 FCM 的扩展, 它采用两个模糊系数  $m_1, m_2$ , 从而获得随模糊系数变化的不确定域, 从而提高了模糊聚类算法处理不确定性的能力。结果也证明 IT2FCM 算法能够获得更好的聚类结果。

#### 算法 1 (IT2FCM)

输入: 数据集  $X = \{x_1, x_2, \dots, x_N\}$ , 其中  $x_i = (x_{i1}, x_{i2}, \dots, x_{iM})$  是一个  $M$  维向量 ( $i = 1, 2, \dots, N$ )。

输出:  $v$  和  $label$ , 分别表示最终聚类中心和每个数据样本的聚类类别。

**Step1** 给定初始聚类数  $c, 2 \leq c \leq N$ ; 设置模糊系数  $m_1$  和  $m_2, m_1$  和  $m_2$  选择没有理论上的参考值, 在实际聚类问题中取值范围一般为  $[1.5, 2.5]$ <sup>[15]</sup>; 设置迭代阈值  $\varepsilon$ , 随机初始化聚类中心  $v_j = (v_{j1}, v_{j2}, \dots, v_{jM}), j = 1, 2, \dots, c$ 。

**Step2** 遍历  $j$  从 1 到  $c$ , 使用式(4)、式(5)计算隶属度:

$$\mu_j(\mathbf{x}_i) = \min \left( \frac{1}{\sum_{k=1}^c \left( \frac{d_{ji}}{d_{ki}} \right)^{\frac{2}{m_1-1}}}, \frac{1}{\sum_{k=1}^c \left( \frac{d_{ji}}{d_{ki}} \right)^{\frac{2}{m_2-1}}} \right) \quad (4)$$

$$\bar{\mu}_j(\mathbf{x}_i) = \max \left( \frac{1}{\sum_{k=1}^c \left( \frac{d_{ji}}{d_{ki}} \right)^{\frac{2}{m_1-1}}}, \frac{1}{\sum_{k=1}^c \left( \frac{d_{ji}}{d_{ki}} \right)^{\frac{2}{m_2-1}}} \right) \quad (5)$$

这里把每个聚类中心  $\mathbf{v}_j$  对应一个区间二型模糊集,  $\mu_j(\mathbf{x}_i)$  和  $\bar{\mu}_j(\mathbf{x}_i)$  分别表示  $\mathbf{x}_i$  属于第  $j$  个聚类中心的最小、最大隶属度值。其中  $d_{ji}$  表示第  $i$  个数据样本到第  $j$  个聚类中心的距离, 一般选择欧氏距离。

**Step3** 遍历  $s$  从 1 到  $M$ , 将  $\{x_{is}\}_{i=1}^N$  中元素重新按升序排列, 记为  $\{x'_{is}\}_{i=1}^N$ , 并用质心求解算法得  $\{x'_{is}\}_{i=1}^N$  的质心, 用区间  $[v_{jsl}, v_{jsr}]$  表示, 其对应的转换点分别记为  $L_{js}$  和  $R_{js}$ 。

**Step4** 通过式  $v'_{js} = (v_{jsl} + v_{jsr})/2, s = 1, 2, \dots, M$  降型得到  $\mathbf{v}'_j = (v'_{j1}, v'_{j2}, \dots, v'_{jM})$ , 若  $\|\mathbf{v}'_j - \mathbf{v}_j\|_2 < \varepsilon$ , 转到 Step5; 否则以  $\mathbf{v}'_j$  代替  $\mathbf{v}_j$ , 转到 Step2 重新计算  $\mu_j(\mathbf{x}_i)$  和  $\bar{\mu}_j(\mathbf{x}_i)$ 。

**Step5** 隶属度降型。计算公式如下:

$$\mu_j(\mathbf{x}_i) = \frac{1}{2M} \sum_{s=1}^M [\mu_{js}^R(\mathbf{x}_i) + \mu_{js}^L(\mathbf{x}_i)]$$

$$\mu_{js}^R(\mathbf{x}_i) = \begin{cases} \mu_j(\mathbf{x}_i), & x_{is} \leq x'_{R_{js}} \\ \bar{\mu}_j(\mathbf{x}_i), & x_{is} > x'_{R_{js}} \end{cases}$$

$$\mu_{js}^L(\mathbf{x}_i) = \begin{cases} \bar{\mu}_j(\mathbf{x}_i), & x_{is} \leq x'_{L_{js}} \\ \mu_j(\mathbf{x}_i), & x_{is} > x'_{L_{js}} \end{cases}$$

为了根据隶属度划分数据样本类别, 这里将聚类中心对应的区间二型模糊集降型为一型模糊集, 这里  $\mu_j(\mathbf{x}_i)$  表示  $\mathbf{x}_i$  属于第  $j$  个聚类中心的隶属度, 是一型模糊集的隶属度。

**Step6** 硬划分: 若  $\mu_j(\mathbf{x}_i) = \max_{k=1, \dots, c} \{\mu_k(\mathbf{x}_i)\}$ , 则把数据样本  $\mathbf{x}_i$  划分到第  $j$  个类别中。

### 3 近似质心求解算法

#### 3.1 基于二分查找的质心求解算法

在 DA 算法中, 首先,  $pos_j$  和  $neg_j$  的表示法相对复杂; 其次, DA 算法是一种蛮力的方法, 它需要计算出所有偏导数。下面介绍一种简化  $pos_j$  和  $neg_j$  的表示并且无需遍历所有  $N$  个  $\frac{\partial v}{\partial \mu_j}$  即可求解质心的算法。由于

$$\frac{\partial v}{\partial \mu_k} = \frac{\mathbf{x}_k \sum_{i=1}^N \mu_i - \sum_{i=1}^N \mathbf{x}_i \mu_i}{\left( \sum_{i=1}^N \mu_i \right)^2}$$

对于上述偏导数, 由于只关心它的正负, 而分母是恒大于 0 的, 所以只需关注分子, 记为  $F(k)$ 。即

$$F(k) \triangleq \mathbf{x}_k \sum_{i=1}^N \mu_i - \sum_{i=1}^N \mathbf{x}_i \mu_i = \sum_{i=1}^N (\mathbf{x}_k - \mathbf{x}_i) \mu_i = \sum_{i=1}^{k-1} (\mathbf{x}_k - \mathbf{x}_i) \mu_i + \sum_{i=k+1}^N (\mathbf{x}_k - \mathbf{x}_i) \mu_i \quad (6)$$

由于  $\{x_i\}_{i=1}^N$  按升序排列, 所以  $\sum_{i=1}^{k-1} (\mathbf{x}_k - \mathbf{x}_i) \mu_i \geq 0$ ;

$\sum_{i=k+1}^N (\mathbf{x}_k - \mathbf{x}_i) \mu_i \leq 0$ ,  $F(k)$  显然是单调增的。

DA 算法关键点在于找到偏导数的转换点  $L$  和  $R$ 。找到  $L$  点, 则可找到区间二型模糊集质心的最小值  $v_l$ 。

式(6)中: 当  $k \leq L$  时,  $F(k) \leq 0$ , 此时  $\frac{\partial v}{\partial \mu_k} \leq 0$ ,  $v$  可看作关于  $\mu_k$  的减函数, 故要想取得  $v$  的最小值  $v_l$ , 当  $i \leq k$  时, 令  $\mu_i$  取  $\bar{\mu}_i$ ; 当  $k > L$  时,  $F(k) > 0$ , 此时  $\frac{\partial v}{\partial \mu_k} > 0$ ,  $v$  可看作关于  $\mu_k$  的增函数, 故要想  $v$  取得最小值  $v_l$ , 当  $i > k$  时, 令  $\mu_i$  取  $\mu_i$ 。故可将式(6)转化成

$$F_1(k) \triangleq \sum_{i=1}^{k-1} (\mathbf{x}_k - \mathbf{x}_i) \bar{\mu}_i + \sum_{i=k+1}^N (\mathbf{x}_k - \mathbf{x}_i) \mu_i$$

于是  $F_1(L) \leq 0, F_1(L+1) > 0$ , 从而可得  $v_l$ 。同理, 要想找到转换点  $R$ , 可将式(6)转化成

$$F_2(k) \triangleq \sum_{i=1}^{k-1} (\mathbf{x}_k - \mathbf{x}_i) \mu_i + \sum_{i=k+1}^N (\mathbf{x}_k - \mathbf{x}_i) \bar{\mu}_i$$

于是  $F_2(R) \leq 0, F_2(R+1) > 0$ , 从而求出  $v_r$ 。

显然  $F_1(k) \geq F_2(k)$ , 对任意  $k = 1, 2, \dots, N$ 。并且有

**性质 1**  $F_1(k)$  和  $F_2(k)$  均是  $k$  的单增函数。

**证明** 以  $F_1(k)$  为例, 设  $k_1 > k_2$ , 有

$$F_1(k_1) - F_1(k_2) =$$

$$\begin{aligned} & \sum_{i=1}^{k_1-1} (\mathbf{x}_{k_1} - \mathbf{x}_i) \bar{\mu}_i + \sum_{i=k_1+1}^N (\mathbf{x}_{k_1} - \mathbf{x}_i) \mu_i - \\ & \sum_{i=1}^{k_2-1} (\mathbf{x}_{k_2} - \mathbf{x}_i) \bar{\mu}_i - \sum_{i=k_2+1}^N (\mathbf{x}_{k_2} - \mathbf{x}_i) \mu_i = \\ & \sum_{i=1}^{k_2-1} (\mathbf{x}_{k_1} - \mathbf{x}_{k_2}) \bar{\mu}_i + \sum_{i=k_1+1}^N (\mathbf{x}_{k_1} - \mathbf{x}_{k_2}) \mu_i + \\ & \sum_{i=k_2}^{k_1-1} (\mathbf{x}_{k_1} - \mathbf{x}_i) \bar{\mu}_i - \sum_{i=k_2+1}^{k_1} (\mathbf{x}_{k_2} - \mathbf{x}_i) \mu_i = \end{aligned}$$

$$(x_{k_1} - x_{k_2}) \left( \sum_{i=1}^{k_2-1} \bar{\mu}_i + \sum_{i=k_1+1}^N \mu_i \right) + \sum_{i=k_2}^{k_1-1} (x_{k_1} - x_i) \bar{\mu}_i + \sum_{i=k_2+1}^{k_1} (x_i - x_{k_2}) \mu_i \quad (7)$$

对于式(7),三项中每一项都大于 0,所以最后原式大于 0,即  $F_1(k)$  单调增。 $F_2(k)$  单调增类似可证,证毕。

基于二分查找的质心求解算法原理:由性质 1 知  $F_1(k)$  和  $F_2(k)$  均是  $k$  的单增函数,  $k=1,2,\dots,N$ 。以求解  $v_l$  为例,要想找到  $v_l$  的转换点  $L$ ,首先考虑中间位置的数值(以  $mid$  表示中间位置索引下标),即  $F_1(mid)$ ,若  $F_1(mid) \leq 0$ ,则下次查找范围在  $[mid, N]$ ,继续计算这部分范围中间位置的数值;若  $F_1(mid) > 0$ ,则下次查找范围在  $[1, mid]$ ,继续计算这部分范围中间位置的数值。

**算法 2(基于二分查找的质心求解算法)**

输入:数据集  $X = \{x_1, x_2, \dots, x_N\}$ , 这里  $\{x_i\}_{i=1}^N$  按升序排列,且每个  $x_i$  对应的区间二型模糊集最小、最大隶属度分别记为  $\mu_i, \bar{\mu}_i (i=1,2,\dots,N)$ 。

输出: $L$  和  $v_l$  与  $R$  和  $v_r$ 。

**Step1** 令  $l=1, r=N, l$  和  $r$  为下标索引。

**Step2** 找到最大的  $l \in [1, N-1]$ , 满足  $F_1(l) \leq 0$  (或  $F_2(l) \leq 0$ )。

以查找  $L$  为例,规则如下:当  $l < r$  时,若  $F_1(\lfloor (l+r)/2 \rfloor) \leq 0$ ,则令  $l = \lfloor (l+r)/2 \rfloor$ ;若  $F_1(\lfloor (l+r)/2 \rfloor) > 0$ ,则令  $r = \lfloor (l+r)/2 \rfloor$ 。继续计算  $F_1(\lfloor (l+r)/2 \rfloor)$ ,返回循环;当  $l+1=r$  时,查找结束,令  $L=l$ 。

若查找  $R$ ,则将上述  $F_1(\lfloor (l+r)/2 \rfloor)$  改成  $F_2(\lfloor (l+r)/2 \rfloor)$ ,其余规则不变。查找结束后,令  $R=l$ 。

**Step3** 将  $L$  代入式(2)计算  $v_l$ (将  $R$  代入公式(3)计算  $v_r$ )。

二分查找的时间复杂度是  $O(\log_2 N)$ ,每次循环的时间复杂度是  $O(N)$ ,所以最后该算法的时间复杂度为  $O(N \log_2 N)$ 。由于算法复杂度仍然不够理想,同时计算  $F_1(k)$  或  $F_2(k)$  是个相对耗时的过程,所以有必要减少计算量,降低算法的时间复杂度。

**3.2 半数迭代法**

**3.2.1 转换点  $L$  和  $R$  的关系**

**性质 2**  $L \leq R$ 。

**证明** 因为对任意  $k=1,2,\dots,N; F_1(k) \geq F_2(k)$ 。 $L$  是满足  $F_1(k) \leq 0$  中最大的点,所以  $F_2(L) \leq F_1(L) \leq 0$ 。 $R$  是满足  $F_2(k) \leq 0$  中最大的点,所以  $0 < F_2(R+1) \leq$

$F_1(R+1)$ 。因此  $F_1(L) < F_1(R+1), F_2(L) < F_2(R+1)$ 。由性质 1,  $F_1(k)$  和  $F_2(k)$  均是  $k$  的单增函数,所以有  $F_1(L) \leq F_1(R), F_2(L) \leq F_2(R)$ ,从而  $L \leq R$ ,证毕。

**3.2.2 计算差值**

在算法 2 中,计算  $F_1(k)$  (或  $F_2(k)$ ) 是相对比较耗时的,所以可以考虑只计算初始的  $F_1(\lfloor (l+r)/2 \rfloor)$  (或  $F_2(\lfloor (l+r)/2 \rfloor)$ ),之后  $F_1(k)$  (或  $F_2(k)$ ) 函数值的计算用初始值加上差值即可,差值见式(7),计算差值比计算出完整的  $F_1(k)$  (或  $F_2(k)$ ) 耗时短。以性质 1 的证明为例:假设已计算出  $F_1(k_2)$ ,需要再计算  $F_1(k_1)$ ,直接计算  $F_1(k_1)$  需运行加法  $N$  次,减法  $N-1$  次,乘法  $N-1$  次;而计算差值时,运算加法  $N+k_1-k_2+1$  次,减法  $2 \times (k_1 - k_2 + 1)$  次,乘法  $2 \times (k_1 - k_2) + 1$  次。若  $F_1(\lfloor (1+N)/2 \rfloor) \leq 0$ ,则当  $N$  足够大时,第一次计算的差值运算的乘法约为  $(N-1)/2+1$  次,要小于完整计算  $F_1(k)$  时乘法的运算量  $N-1$  次,之后若循环还没结束,则差值乘法的运算次数就更少了,而计算  $F_1(k)$  乘法运算次数不变。在计算机中,乘法的运算时间要慢于加减法,所以运算更少的乘法保证了计算差值比计算完整的  $F_1(k)$  (或  $F_2(k)$ ) 耗时要短。

**半数迭代法原理:**在基于二分查找的质心求解算法基础上,考虑到计算差值和转换点  $L$  和  $R$  的关系,只考虑  $F_1(mid) \leq 0$  (或  $F_2(mid) \leq 0$ ) 的情形,因此求出的质心不再是精确的质心,而是近似质心,并且显然近似质心区间包含在精确质心区间内。随着 IT2FCM 算法的迭代,去模糊化的近似质心能够慢慢收敛。

值得注意的是,近似转换点  $\tilde{L}$  和  $\tilde{R}$  不再有性质 2 的结论,但仍在限制  $\tilde{L} \leq \tilde{R}$  的基础上求解近似质心。

**算法 3(半数迭代法)**

输入:数据集  $X = \{x_1, x_2, \dots, x_N\}$ , 这里  $\{x_i\}_{i=1}^N$  按升序排列。且每个  $x_i$  对应的区间二型模糊集最小、最大隶属度分别记为  $\mu_i, \bar{\mu}_i (i=1,2,\dots,N)$ 。

输出: $\tilde{R}$  和  $v_r$  与  $\tilde{L}$  和  $v_l$ 。

(1) 求  $\tilde{R}$  和  $v_r$

**Step1** 令  $l=1, r=N, l$  和  $r$  为下标索引。

**Step2** 如果  $F_2(N) \leq 0$ ,则令  $\tilde{R}=N$ ,转到 Step4;否则,转到 Step3。

**Step3** 查找到近似转换点  $\tilde{R}$ ,规则如下:当  $l < r$  时,若  $F_2(\lfloor (l+r)/2 \rfloor) \leq 0$ ,则令  $l = \lfloor (l+r)/2 \rfloor$ ;若  $F_2(\lfloor (l+r)/2 \rfloor) > 0$ ,则提前跳出循环。若未提前跳出循环,继续计算  $F_2(\lfloor (l+r)/2 \rfloor)$  (通过差值计算),返回循

环;当  $l+1=r$  时查找结束,令  $\tilde{R}=\lfloor (l+r)/2 \rfloor$ 。

**Step4** 将  $\tilde{R}$  代入式(3)计算  $v_r$  同时保存  $\tilde{R}$ 。

(2) 求  $\tilde{L}$  和  $v_l$

**Step1** 令  $l=1, r=\tilde{R}, l$  和  $r$  为下标索引,  $r$  初始化指向  $v_r$  的近似转换点  $\tilde{R}$ 。

**Step2** 如果  $F_1(N) \leq 0$ , 则令  $L=N$ , 转到 Step4; 否则, 转到 Step3。

**Step3** 查找到近似转换点  $\tilde{L}$ , 规则如下: 当  $l < r$  时, 若  $F_1(\lfloor (l+r)/2 \rfloor) \leq 0$ , 则令  $l = \lfloor (l+r)/2 \rfloor$ ; 若  $F_1(\lfloor (l+r)/2 \rfloor) > 0$ , 则提前跳出循环。若未提前跳出循环, 继续计算  $F_1(\lfloor (l+r)/2 \rfloor)$  (通过差值计算), 返回循环; 当  $l+1=r$  时查找结束, 令  $\tilde{L}=\lfloor (l+r)/2 \rfloor$ 。

**Step4** 将  $\tilde{L}$  代入式(2)计算  $v_l$ 。

### 3.3 一次迭代法

一次迭代法原理: 在基于二分查找的质心求解算法基础上, 只计算第一次  $F_1(\text{mid})$  (或  $F_2(\text{mid})$ ), 然后将根据  $F_1(\text{mid})$  (或  $F_2(\text{mid})$ ) 的正负选择的第二次中间位置作为转换点的近似, 此时不必再计算  $F_1(\text{mid})$  (或  $F_2(\text{mid})$ )。

一次迭代法的速度是最快的, 求出的近似质心与精确质心差距相较于半数迭代法求出的近似质心与精确质心差距更大, 但是没能求出精确质心的上下界导致一次迭代法求出的近似质心区间同样包含在精确质心区间中, 随着 IT2FCM 算法的迭代, 近似质心去模糊化后同样会慢慢收敛。

**算法 4** (一次迭代法)

输入: 数据集  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , 这里  $\{\mathbf{x}_i\}_{i=1}^N$  按升序排列, 且每个  $\mathbf{x}_i$  对应的区间二型模糊集最小、最大主隶属度分别记为  $\underline{\mu}_i, \bar{\mu}_i (i=1, 2, \dots, N)$ 。

输出:  $\tilde{L}$  和  $v_l$  与  $\tilde{R}$  和  $v_r$ 。

**Step1** 令  $l=1, r=N$ 。

**Step2** 如果  $F_1(N) \leq 0$ , 则令  $\tilde{L}=N$ , 转到 Step4; 否则, 转到 Step3。

**Step3** 以查找近似转换点  $\tilde{L}$  为例, 规则如下: 若  $F_1(\lfloor (l+r)/2 \rfloor) \leq 0$ , 则令  $l = \lfloor (l+r)/2 \rfloor$ ; 若  $F_1(\lfloor (l+r)/2 \rfloor) > 0$ , 则令  $r = \lfloor (l+r)/2 \rfloor$ ; 最后令  $\tilde{L}=\lfloor (l+r)/2 \rfloor$  为近似转换点。

若查找近似转换点  $\tilde{R}$ , 则将上述  $F_1(\lfloor (l+r)/2 \rfloor)$  改成

$F_2(\lfloor (l+r)/2 \rfloor)$ , 其余规则不变, 最后令  $\tilde{R}=\lfloor (l+r)/2 \rfloor$ 。

**Step4** 将  $\tilde{L}$  代入式(2)计算  $v_l$  (将  $\tilde{R}$  代入公式(3)计算  $v_r$ )。

### 3.4 两种近似算法复杂度分析

对于半数迭代法, 由于它基于二分查找, 所以它的迭代上限是  $\log_2 N$  次, 每次迭代主要是计算  $F_1(k)$  (或  $F_2(k)$ ), 计算它们的时间复杂度是  $O(N)$ , 所以半数迭代法最终的时间复杂度是  $O(N \log_2 N)$ ; 一次迭代法的迭代次数为 1 次, 每次迭代的时间复杂度同样是  $O(N)$ , 所以一次迭代法最终的时间复杂度为  $O(N)$ 。

## 4 实验设计与验证

本节中, 为了比较基于近似质心求解算法和基于精确质心求解算法的 IT2FCM 的运行速度, 实验分为两个部分。第一部分主要比较基于精确质心求解算法 (以 KM 算法为例) 的 IT2FCM 和基于近似质心的 IT2FCM 的聚类性能, 同时记录在不同数据集中, 基于半数迭代法的 IT2FCM 计算一次近似质心迭代的次数范围 (对数据每一个分量都做一次半数迭代法, 找到其中最少迭代次数和最大迭代次数), 用来估计半数迭代法的迭代次数; 第二部分比较这两类算法的速度。模糊系数  $m_1$  和  $m_2$  分别固定为 1.5 和 2.5, 实验数据都来自 UCI。实验平台为 Intel I7 处理器, 系统为 Win10 家庭版, 编程语言及版本为 python3.8, 时间成本由 python 的 time 库中的 perf\_counter() 方法计算得出。

### 4.1 聚类性能

由于下面实验的数据集都是带有标签的, 所以可以选择聚类性能度量中的“外部指标”类, 常用的外部指标有 Jaccard 系数、FM 指数和 Rand 指数<sup>[16]</sup>等。这些性能度量的结果值都在  $[0, 1]$  区间, 值越大越好。为了计算简单, 这里选择的外部指标为 Rand 指数 (简记 RI)。

对数据集  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N\}$ , 其中  $\mathbf{x}_i$  是向量, 表示一个数据样本, 若标签给出的簇划分  $C = \{C_1, C_2, \dots, C_k\}$ , 聚类得出的簇划分为  $C^* = \{C_1^*, C_2^*, \dots, C_s^*\}$ , 相应地, 令  $\lambda$  和  $\lambda^*$  分别表示与  $C$  和  $C^*$  对应的簇标记向量。记

$$SS = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}$$

$$DD = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}$$

其中  $SS$  表示在  $C$  中属于相同簇且在  $C^*$  中也属于相同簇的样本对的集合;  $DD$  表示在  $C$  中属于不同簇且在  $C^*$  中也属于不同簇的样本对的集合。  $a = |SS|, d =$

$|DD|$  分别表示两个集合的基数。

定义 5<sup>[16]</sup> (Rand 指数) 记为

$$RI = \frac{2(a+d)}{N(N-1)}$$

这部分实验选用 UCI 上 5 个数据集,其中有 3 个小数据集:IRIS、SEEDS 和 WINE,2 个较大数据集:WIFI\_LOCALIZATION 和 HTRU2。数据描述见表 1,实验结果见表 2。

表 1 数据描述

Table 1 data description

数据集	IRIS	SEEDS	WINE	WIFI_LOCALIZATION	HTRU2
类别	3	3	3	4	2
每类包含数据量	50/50/50	70/70/70	59/71/48	500/500/500/500	1 639/16 259

表 2 Rand 指数

Table 2 Rand index

算 法	数据集				
	IRIS	SEEDS	WINE	WIFI_LOCALIZATION	HTRU2
KM 算法	0.862 3	0.878 9	0.933 1	0.940 7	0.847 2
半数迭代法	0.873 7	0.873 0	0.939 8	0.920 8	0.846 1
一次迭代法	0.867 9	0.889 5	0.927 4	0.940 2	0.843 6

从表 2 可以看出:无论是小的数据集还是较大的数据集,本文提出的两种基于近似质心求解的 IT2FCM 算法的聚类性能与传统的基于 KM 算法求解精确质心的 IT2FCM 算法相差无几,由此说明这两种基于近似质心求解算法的 IT2FCM 算法的有效性。于是在保证聚类性能的基础上,就可以进一步比较算法运行的速度。

### 4.2 半数迭代法迭代次数估计

在 3.2 节得出的半数迭代法理论上的时间复杂度是  $O(N \log_2 N)$ ,实际上它的迭代次数被高估了。在 KM 算法中,它有一个被证明的迭代上限<sup>[17]</sup>: $(N-1)/2$ ,但在实验仿真中,迭代次数位于 2—6 之间,与  $N$  大小无关。借助同样的想法,这部分实验测试了在不同数据集下的迭代次数,结果(表 3)表明它的迭代次数位于 1 到 5 之间,但基本 2 次迭代就跳出循环。所以最后,无论是 KM 算法还是本文提出的两种近似质心求解算法时间复杂度都是  $O(N)$ 。

表 3 迭代次数

Table 3 Number of iterations

数据集	IRIS	SEEDS	WINE	WIFI_LOCALIZATION	HTRU2
迭代次数范围	1—3	1—3	1—3	1—5	1—5
众数	2	2	2	2	2

### 4.3 算法速度比较

这部分实验数据截取自 ANURAN CALLS,原本的数据集包含 3 列类别标签。这里只考虑 Genus 标签,同

时只考虑其中 Adenomera 和 Hypsiboas 这两类数据。截取数据量为 200 的数据集作为第一组数据,两类各占一半,通过倍增方式(将该小数据集复制成需要的倍数再合并成一组数据集)构造出具有 400、600、800、1 000、1 200、1 400 和 1 600 数据量的其他 7 组数据,共 8 组数据。这样能够保证聚类性能不变的同时排除非数据增长因素对聚类时间造成的影响。每种算法在每组数据下运行 12 次,部分结果见表 4—表 9。

表 4 KM 算法运行时间

Table 4 Running time of KM algorithm

第 $i$ 次实验	组 别							
	1	2	3	4	5	6	7	8
1	1.92	4.38	7.87	12.34	18.09	27.54	27.60	44.76
2	1.89	3.71	8.01	14.01	17.80	24.44	31.41	39.99
3	1.89	5.66	10.15	12.40	20.16	24.22	39.43	40.27
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	2.10	4.37	8.80	11.00	22.64	27.50	31.32	39.61
12	1.96	4.93	8.84	13.68	18.05	27.51	35.70	39.31

表 5 EKM 算法运行时间

Table 5 Running time of EKM algorithm

第 $i$ 次实验	组 别							
	1	2	3	4	5	6	7	8
1	0.81	1.87	2.70	3.65	4.25	5.80	7.70	9.57
2	0.86	1.97	2.91	5.62	7.15	5.71	5.95	8.26
3	0.86	1.79	2.34	4.30	5.60	5.06	6.71	8.34
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	1.00	2.47	3.31	4.85	5.10	8.23	7.32	7.87
12	0.85	1.51	2.31	5.15	5.14	5.03	6.66	8.12

表 6 DA 算法运行时间

Table 6 Running time of DA algorithm s

第 $i$ 次实验	组 别							
	1	2	3	4	5	6	7	8
1	1.31	2.29	3.46	5.39	6.79	7.38	8.83	10.17
2	1.35	2.01	3.41	3.92	6.88	7.48	8.58	11.27
3	1.24	1.97	4.28	4.62	6.03	8.23	7.71	10.22
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	1.08	2.65	3.41	3.50	6.76	7.47	8.63	12.72
12	1.22	2.30	3.91	4.56	6.06	8.15	9.57	9.98

表 7 DAND 算法运行时间

Table 7 Running time of DAND algorithm s

第 $i$ 次实验	组 别							
	1	2	3	4	5	6	7	8
1	1.32	2.30	2.89	4.76	6.28	8.40	7.83	10.42
2	0.86	2.43	3.63	6.51	6.32	7.38	9.16	9.16
3	1.12	2.61	3.46	5.17	6.94	6.66	8.02	11.50
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	1.17	2.31	3.11	4.88	6.17	6.72	8.01	10.49
12	0.89	2.31	3.13	4.27	6.11	6.46	9.08	11.84

表 8 半数迭代法运行时间

Table 8 Running time of half iteration algorithm s

第 $i$ 次实验	组 别							
	1	2	3	4	5	6	7	8
1	1.43	2.04	3.03	4.15	6.98	8.41	8.78	12.69
2	1.50	2.30	3.47	4.69	6.75	9.31	8.52	8.97
3	1.29	2.12	3.87	4.75	6.02	7.68	10.63	10.11
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	2.13	1.97	3.07	5.43	7.61	7.30	8.64	10.26
12	1.23	2.97	3.93	4.90	5.98	7.26	9.02	11.60

表 9 一次迭代法运行时间

Table 9 Running time of one iteration algorithm s

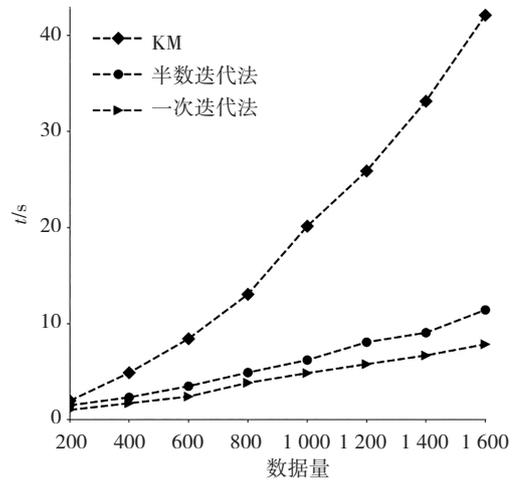
第 $i$ 次实验	组 别							
	1	2	3	4	5	6	7	8
1	1.36	1.64	2.73	3.22	4.21	5.90	6.26	7.60
2	0.93	1.94	2.44	4.92	4.32	5.21	6.52	7.38
3	0.93	1.56	2.46	4.77	5.31	5.82	8.75	7.39
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	1.06	1.36	2.13	3.79	5.39	5.76	7.59	7.17
12	0.87	1.98	2.78	3.80	4.71	7.05	5.46	8.54

然后对每种算法在每组数据下的 12 个值,去掉最大值和最小值再取平均得到表 10,最后画出图 2。

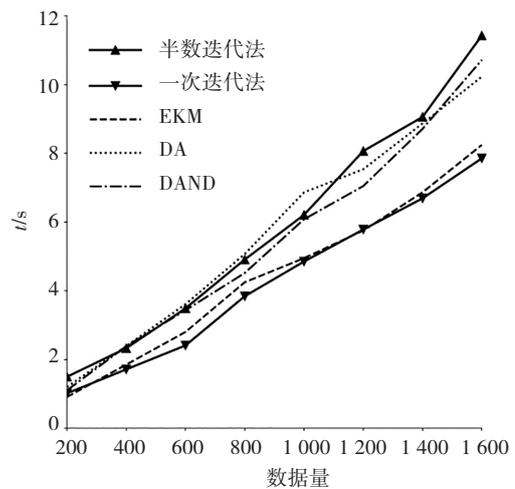
表 10 时间比较

Table 10 Time comparison s

第 $i$ 次实验	组 别							
	1	2	3	4	5	6	7	8
KM 算法	1.97	4.88	8.43	13.05	20.14	25.89	33.15	42.11
EKM 算法	0.91	1.85	2.80	4.25	4.95	5.75	6.87	8.25
DA 算法	1.20	2.39	3.61	5.07	6.86	7.54	8.87	10.23
DAND 算法	1.11	2.39	3.44	4.52	6.08	7.05	8.71	10.72
半数迭代法	1.50	2.33	3.49	4.91	6.21	8.07	9.06	11.43
一次迭代法	1.02	1.71	2.41	3.84	4.85	5.78	6.69	7.85



(a) 时间比较 1



(b) 时间比较 2

图 2 时间比较

Fig. 2 Time comparison

从图 2(a)中可以看出:对于相同的数量,基于 KM 算法的 IT2FCM 运行时间最长,基于半数迭代法的

IT2FCM 运行时间其次,基于一次迭代法的 IT2FCM 运行时间最短。随着数据量的增加,基于近似质心求解的两类 IT2FCM 与基于 KM 算法的 IT2FCM 运行时间差距越来越大。从图 2(b)中可以看出:基于一些改进的质心求解算法的 IT2FCM 的运行时间基本位于基于本文提出的两种近似算法的 IT2FCM 运行时间之间。

本文在 DA 算法的基础上设计了两种近似质心求解算法,最后通过实验比较了基于近似质心求解算法的 IT2FCM 和基于 KM 算法的 IT2FCM 以及一些基于改进质心求解算法的 IT2FCM。实验结果表明:相较于基于现有的一些精确质心求解算法的 IT2FCM,基于近似质心求解算法的 IT2FCM 能够在保证聚类性能的同时提高运行速度。

## 5 结束语

IT2FCM 是一种经典的模糊聚类算法,它的聚类性能高,结果稳定,但由于其计算复杂度较高,需要降低算法复杂度才能更好运用。本文的实验结果发现 IT2FCM 对质心不太敏感,近似质心并不影响聚类性能,所以考虑从计算近似质心角度出发降低算法复杂度,但本文没有考虑选用近似质心对 IT2FCM 外围质心迭代次数的影响,所以如何权衡二者,进而进一步提高算法速度是个值得考虑的方向。

## 参考文献(References):

- [1] ZADEH L A. Fuzzy sets[J]. Information and Control, 1965, 8(3): 338—353.
- [2] BEZDEK J C, EHRLICH R, FULL W. FCM: the fuzzy c-means clustering algorithm [J]. Computer & Geosciences, 1984, 10(2): 191—203.
- [3] HWANG C, RHEE F C H. Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means [J]. IEEE Transactions on Fuzzy Systems, 2007, 15(1): 107—120.
- [4] 袁飞,詹宜巨,王永华. 区间数的区间 II 型模糊 c 均值聚类算法[J]. 自动化与信息工程, 2013, 34(4): 1—5.  
YUAN Fei, ZHAN Yi-ju, WANG Yong-hua. Interval type-2 fuzzy c-means clustering method for interval data [J]. Automation & Information Engineering, 2013, 34(4): 1—5.
- [5] MENDEL J M, LIU F, ZHAI D.  $\alpha$ -plane representation for type-2 fuzzy sets: theory and applications [J]. IEEE Transactions on Fuzzy Systems, 2009, 17(5): 1189—1207.
- [6] LINDA O, MANIC M. General type-2 fuzzy c-means algorithm for uncertain fuzzy clustering[J]. IEEE Transactions on Fuzzy Systems, 2012, 20(5): 883—897.
- [7] KARNIK N N, MENDEL J M. Centroid of a type-2 fuzzy set[J]. Information Sciences, 2001, 132(1): 195—220.
- [8] WU D, MENDEL J M. Enhanced Karnik-Mendel algorithms[J]. IEEE Transactions on Fuzzy Systems, 2009, 17(4): 923—934.
- [9] CHEN C, JOHN R I, TWYXCROSS J, et al. A direct approach for determining the switch points in the Karnik-Mendel algorithm [J]. IEEE Transactions on Fuzzy Systems, 2018, 26(2): 1079—1085.
- [10] CHEN C, WU D, GARIBALDI J M, et al. A comprehensive study of the efficiency of type-reduction algorithms [J]. IEEE Transactions on Fuzzy Systems, 2021, 29(6): 1556—1566.
- [11] 邱存勇,肖建,韩璐. 增强型区间二型 FCM 算法[J]. 控制与决策, 2014, 29(3): 465—469.  
QIU Cun-yong, XIAO Jian, HAN Lu. Enhanced interval type-2 fuzzy c-means algorithm [J]. Control and Decision, 2014, 29(3): 465—469.
- [12] MENDEL J M, JOHN R I, LIU F. Interval type-2 fuzzy logic systems made simple [J]. IEEE Transactions on Fuzzy Systems, 2006, 14(6): 808—821.
- [13] 莫红,王飞跃. 区间二型模糊集合的表述与应用[J]. 模糊系统与数学, 2014, 28(2): 167—174.  
MO Hong, WANG Fei-yao. Representation of interval type-2 fuzzy sets [J]. Fuzzy Systems and Mathematics, 2014, 28(2): 167—174.
- [14] WU D, MENDEL J M. Uncertainty measures for interval type-2 fuzzy sets [J]. Information Sciences, 2007, 177: 5378—5393.
- [15] PAL N R, BEZDEK J C. On cluster validity for the fuzzy c-means model [J]. IEEE Transactions on Fuzzy Systems, 1995, 3(3): 370—379.
- [16] 周志华. 机器学习 [M]. 第一版. 北京: 清华大学出版社, 2016: 198—199.  
ZHOU Zhi-hua. Machine learning [M]. 1st edn. Beijing: Tsinghua University Press, 2016: 198—199.
- [17] LIU F, MENDEL J M. Aggregation using the fuzzy weighted average as computed by the Karnik-Mendel algorithms [J]. IEEE Transactions on Fuzzy Systems, 2008, 16(1): 1—12.