DOI:10.11992/tis.201507013
网络出版地址:http://www.cnki.net/kcms/detail/23.1538.TP.20160315.1239.014.html

适合大规模数据集的增量式模糊聚类算法

李滔,王士同
(江南大学 数字媒体学院,江苏 无锡 214122)

摘 要:FCPM 算法已被成功地应用到模糊系统建模上,但其在某一类的聚类中心已知的大规模数据上的聚类性能较差。为了避免这个缺点,参照单程模糊 c 均值(SPFCM)聚类算法、在线模糊 c 均值(OFCM)聚类算法,提出了适合大规模数据集的增量式模糊聚类算法(Incremental fuzzy (c+p)-means clustering ,IFCM(c+p))。通过在每个数据块中使用 FCPM 算法进行聚类,把每个数据块的聚类中心及其附近的一些样本点加入到下一个数据块参与聚类,同时添加平衡因子以提高算法聚类性能。同 SPFCM、OFCM 以及 rseFCM 算法相比,IFCM(c+p)对初始聚类中心不敏感。实验表明在没有花费很多运行时间的情况下,IFCM(c+p)算法的聚类性能比 SPFCM 算法和 rseFCM 算法更具优势,因此该算法更适合处理某一类聚类中心已知的大规模数据集。

关键词:增量式模糊聚类;FCPM;IFCM(c+p);平衡因子;大规模数据集
中图分类号: TP391.4 **文献标志码**:A **文章编号**:1673-4785(2016)02-0188-12

中文引用格式:李滔,王士同. 适合大规模数据集的增量式模糊聚类算法[J]. 智能系统学报, 2016, 11(2): 188-199.
英文引用格式:LI Tao, WANG Shitong. Incremental fuzzy (c+p)-means clustering for large data[J]. CAAI transactions on intelligent systems, 2016, 11(2): 188-199.

Incremental fuzzy (c+p)-means clustering for large data

LI Tao, WANG Shitong
(School of Digital Media, Jiangnan University, Wuxi 214122, China)

Abstract: FCPM has been demonstrated to be successful in fuzzy system modeling, however, it will be ineffective for large data clustering tasks where the cluster centers of one class are known. In order to circumvent this drawback, referring to single-pass fuzzy c-means (SPFCM) clustering algorithm and online fuzzy c-means (OFCM) clustering algorithm, the incremental fuzzy clustering algorithm for large data called IFCM(c+p) is proposed in this paper. FCPM algorithm is used to cluster for each data block at first, and then the clustering centers of data block and some of the sample points being near them are joined into the next block to be clustered, meanwhile the balance factor is given to enhance the clustering performance. In contrast to SPFCM, OFCM and rseFCM, IFCM(c+p) is not sensitive to the initial cluster centers. The experiments indicate the proposed clustering algorithm IFCM(c+p) is competitive to the clustering algorithms SPFCM and rseFCM in the clustering performance without the loss of running time a lot, hence it is especially suitable for large data clustering tasks where the cluster centers of one class are known.

Keywords: incremental fuzzy clustering; FCPM; IFCM(c+p); balance factor; large data

聚类就是将物理或抽象的对象按照自己的某些属性聚集成类的过程,并尽可能使得类(或者簇)之间对象的差异程度最大,而类内(或者簇内)的相似程度达到最大。聚类过程没有先验知识指导,仅凭对象间的相似程度作为类属划分的准则,是无监督

收稿日期:2015-07-06. **网络出版日期**:2016-03-15.
基金项目:国家自然科学基金项目(61272210).
通信作者:李滔. E-mail:chasingdream119@163.com.

分类学习的一部分。最为经典的模糊聚类算法之一就是 J.C.Bezdek 教授在 20 世纪 80 年代提出的模糊 c 均值聚类算法[1],该算法被成功地应用到了在诸多问题的解决上。

随着科学技术的发展,数据库中的数据更新速度日益加快、数据容量不断增大,若仍然采用原来的聚类算法对这样的大规模数据进行聚类将产生以下几个问题:1)数据更新前得到的聚类结果可能与数据更新后的聚类结果不匹配;2)对更新后的数据进行重新聚类会导致较高的时间复杂度和计算资源的浪费;3)还可能由于系统内存不足的原因而导致该算法失效。鉴于这些问题,Fazli Can 教授在 1990 年提出的增量式聚类算法[2]使得这些问题得以解决。所谓增量式聚类是指利用前期数据已取得的聚类结果,对新增数据进行分批或者逐批次地进行聚类的过程。研究增量式模糊聚类算法对于避免重复聚类造成的计算资源浪费,提高聚类性能等都具有十分重要的意义。

近几年,研究者们提出了很多关于增量式聚类的算法。这些算法大致可以被分为 3 类:1)对大数据进行随机抽样获取小样本进行计算,例如,L. Kaufman 等提出的 CLARA[3],S. Guha 等[4]提出的 CURE;2)按序将小样本加载进内存的单程算法(single-pass),具有代表性的有 F. Can 在文献[5]和[6]中提出的增量式算法;3)采取类图表结构的数据转换算法,如 T. Zhang 等提出的 BIRCH[7]和 R. Ng 等[8]提出的 CLARANS,对于增量式模糊聚类算法;B. U. Shankar 等[9]提出了快速模糊 c 均值算法 FFCM,T. Cheng[10]提出了多阶段的随机模糊 c 均值算法 MRFCM,J. F. Kolen 等[11]提出了随机抽样模糊 c 均值算法 rsFCM,Dhanesh Kothari 等[12]提出了将随机抽样的结果扩展到整个数据集上的扩展随机抽样模糊 c 均值算法 rseFCM。除此之外,还有基于 FCM 的单程模糊 c 均值算法 SPFCM[13]、在线模糊 c 均值算法 OFCM[14],以及在这基础上发展的基于核的模糊 c 均值算法 spkFCM 和 okFCM[15],Yangtao Wang 等[16]提出的基于多重中心的增量式模糊聚类算法在相关性大数据上的应用。最近 Böhm 等[17]受到动力学中同步现象的启发提出了一种新颖的同步聚类算法 Sync,但是这种算法在大规模数据集上的聚类受到了相当大的限制,基于此应文豪等[18]在此基础上提出了快速自适应同步聚类算法 FAKCS。

传统的 FCM 算法对初始聚类中心敏感且容易陷入局部最优,同时也忽略了类间的相互影响。Jacek M. Leski 对 FCM 算法进行了改进,提出了模糊 c+p 均值聚类算法 FCPM,并采用了新的方法初始化聚类中心[19]。对于某一类的聚类中心,它能吸引属于该类的样本并排斥属于其他类的样本,这样更清楚地确定了样本的“归属”问题。对于小样本数据,FCPM 算法可以保持不错的聚类性能,但其在大规模数据集上的聚类性能明显降低而且有较大的时间花费,甚至可能由于无法加载进内存而导致算法失效。对于以往的增量式模糊聚类算法,比如 SPFCM 算法和 OFCM 算法都是通过对样本加权以影响每个数据块产生的聚类中心,但数据块间聚类中心的相互影响程度不明显甚至可能会由于上一次聚类结果的加入而干扰新的数据进行聚类。为了解决以上问题,通过 FCPM 算法计算每个数据块的聚类中心,把离聚类中心最近的一些样本点连同聚类中心一起加入到下一个数据块中参与聚类,同时添加平衡项以提高聚类性能,文中提出了适合大规模数据集的增量式聚类算法 IFCM(c+p)。

1　相关算法

设 N 元样本集合 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$,$\boldsymbol{x}_k(k=1,2,\cdots,N)$ 表示其中的某一个样本,其中每一个样本都有 $\boldsymbol{D}=\{\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n\}\subset\boldsymbol{R}^n$ 一共 n 个特征,$\boldsymbol{d}_j(j=1,2,\cdots,n)$ 表示其中的某一个特征。FCM 算法将 N 个样本按照它所固有的特征划分成 c 簇,用 μ_{ik} 表示第 k 个样本隶属于第 i 簇的程度,那么划分成 c 簇后得到的隶属度矩阵是 $\boldsymbol{U}=\{\mu_{ik}\}\subset\boldsymbol{R}^{c\times N}$,$i\in[1,c]$,$k\in[1,N]$。对于模糊划分而言,所有的样本都需要满足下面的条件:

$$\begin{cases} M_{fcN}=\{\boldsymbol{U}\in\boldsymbol{R}^{c\times N}\mid\mu_{ik}\in[0,1],\\ \qquad\forall i\in[1,c],k\in[1,N]\};\\ \sum_{i=1}^{c}\mu_{ik}=1,\\ \qquad\forall k\in[1,N];\sum_{k=1}^{N}\mu_{ik}\in(0,N),\forall i\in[1,c] \end{cases}$$

由此可见,模糊划分矩阵 $\boldsymbol{U}$ 的每一列的和都必须等于 1,这样才能确保每一个样本都能够被完整地划分到它所属的簇中。

通过使用欧式距离寻求最小均方误差,可以得到 FCM 模型的目标函数(其中 m 为模糊指数):

$$J(\boldsymbol{U},\boldsymbol{V})=\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}\parallel\boldsymbol{x}_k-\boldsymbol{v}_i\parallel^2 \qquad (1)$$

在式(1)的条件下通过拉格朗日乘子法可以得出隶属度矩阵 $\boldsymbol{U}$ 和聚类中心 $\boldsymbol{V}$ 的更新公式。由于

篇幅有限,FCM 算法的具体更新公式以及计算步骤在此不做赘述。

传统的 FCM 算法让聚类中心尽可能地靠近样本点,概率约束也只考虑了聚类中心之间的排斥力,所有的样本重要性相同,同时对初始聚类中心敏感、容易陷入局部最优,得到的聚类结果往往不理想。Jacek M. Leski 考虑了类别间的相互影响,利用了新的方法初始化聚类中心,采用固定一类求其他类的方法,在 FCM 算法的基础上提出了模糊 c+p 均值聚类算法 FCPM。

FCPM 算法中来自其他类的样本对本类的聚类会产生影响,在某一类中,聚类中心应该吸引属于该类的样本,而排斥其他类的样本。设有 c 个聚类中心来自一类,而 p 个聚类中心来自另一类,该算法把 N 个样本划分成为 c 簇,可得目标函数为

$$J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V}) = \sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m} \parallel \boldsymbol{x}_k - \boldsymbol{v}_i \parallel^2 + \sum_{j=1}^{p}\sum_{k=1}^{N}\zeta_{jk}^{m} \parallel \boldsymbol{x}_k - z_j \parallel^2 \tag{2}$$

式中:$\boldsymbol{V}_i$ 表示第 i 簇的聚类中心,z_j 表示已知的聚类中心。对所有的样本而言,都应该满足如下关系:

$$\sum_{i=1}^{c}\mu_{ik} + \sum_{j=1}^{p}\zeta_{jk} = 1, \mu_{ik} \in [0,1], \zeta_{jk} \in [0,1], \forall k \in [1,N] \tag{3}$$

式中:μ_{ik} 表示第 k 个样本属于第 i 簇的程度,ζ_{jk} 表示第 k 个样本属于第 j 簇的程度, 利用拉格朗日乘子法,可以得到划分矩阵 $\boldsymbol{U}$、$\boldsymbol{T}$ 以及聚类中心 $\boldsymbol{V}$ 的更新公式:

$$\mu_{ik} = \frac{\parallel \boldsymbol{x}_k - \boldsymbol{v}_i \parallel^{\frac{2}{1-m}}}{\sum_{l=1}^{c} \parallel \boldsymbol{x}_k - \boldsymbol{v}_l \parallel^{\frac{2}{1-m}} + \sum_{r=1}^{p} \parallel \boldsymbol{x}_k - \boldsymbol{z}_r \parallel^{\frac{2}{1-m}}} \quad \forall k \in [1,N], i \in [1,c] \tag{4}$$

$$\zeta_{jk} = \frac{\parallel \boldsymbol{x}_k - \boldsymbol{z}_j \parallel^{\frac{2}{1-m}}}{\sum_{l=1}^{c} \parallel \boldsymbol{x}_k - \boldsymbol{v}_l \parallel^{\frac{2}{1-m}} + \sum_{r=1}^{p} \parallel \boldsymbol{x}_k - \boldsymbol{z}_r \parallel^{\frac{2}{1-m}}} \quad \forall k \in [1,N], j \in [1,p] \tag{5}$$

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N}\mu_{ik}^{m}\boldsymbol{x}_k}{\sum_{k=1}^{N}\mu_{ik}^{m}}, \forall i \in [1,c] \tag{6}$$

针对 FCM 算法对初始聚类中心敏感的问题,FCPM 算法采用了新的方法初始化聚类中心。通过该方法初始化未知类的聚类中心 $\boldsymbol{V}$,使用 FCM 算法初始化已知类的聚类中心 $\boldsymbol{Z}$,再依次通过式(4)、(5)和(6)获取模糊划分矩阵 $\boldsymbol{U}$ 和聚类中心 $\boldsymbol{V}$。文献[19]详细介绍了新的聚类中心初始化方法及 FCPM 算法,此处不再赘述。

如文献[19]所示,FCPM 算法在模糊系统建模上得到了很好的应用。该算法采用新的初始化聚类中心的方法有效地避免了 FCM 算法对初始聚类中心敏感的问题,通过先确定已知类聚类中心来求未知类聚类中心的方法以提高算法的聚类性能。通过实验可以发现,FCPM 算法对一类已知的小样本数据集有着不错的聚类性能,但对现实中的大规模数据集而言,该算法的聚类性能会下降、算法效率会大大降低甚至会由于样本过大而导致算法失效。基于这些问题,本文提出了适合大规模数据集的增量式模糊聚类算法 IFCM(c+p)。

2 适合大规模数据集的增量式模糊聚类算法 IFCM(c+p)

2.1 IFCM(c+p)算法

在增量式模糊聚类算法中,对每一个数据块进行聚类的算法起着举足轻重的作用。针对以往基于 FCM 的增量式模糊聚类算法对初始聚类中心敏感的问题,文中采用了 FCPM 算法中提到的特别的方法初始化聚类中心。另外在传统的增量式模糊聚类算法中,不管是静态的还是动态的、单程的还是在线的、一个中心或者是多个中心(多个中心形成了一个约束对)等等的方法,都没有考虑数据块之间聚类中心的相互影响,提及的 IFCM(c+p)算法很好地解决了这些问题。

为了增加数据块间聚类中心的相互影响程度,本文添加了一个平衡项 $\alpha\sum_{i=1}^{c} \parallel \boldsymbol{v}_i - \boldsymbol{v}_i^o \parallel^2$,其中 α 被称为平衡因子,往往它的取值与 $J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V})$ 有关。由此,可以得到提及算法的目标函数:

$$J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\alpha) = J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V}) + \alpha\sum_{i=1}^{c} \parallel \boldsymbol{V}_i - \boldsymbol{V}_i^o \parallel^2 = \sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m} \parallel \boldsymbol{x}_k - \boldsymbol{v}_i \parallel^2 + \sum_{j=1}^{p}\sum_{k=1}^{N}\zeta_{jk}^{m} \parallel \boldsymbol{x}_k - \boldsymbol{z}_j \parallel^2 + \alpha\sum_{i=1}^{c} \parallel \boldsymbol{v}_i - \boldsymbol{v}_i^o \parallel^2 \tag{7}$$

式中:$\boldsymbol{v}_i$ 表示第 i 簇的聚类中心,μ_{ik} 表示第 k 个样本属于第 i 簇的程度,ζ_{jk} 表示第 k 个样本属于第 j 簇的程度,$\boldsymbol{z}_j$ 表示已知的第 j 簇的聚类中心,$\boldsymbol{V}_i^o$ 表示经过 FCPM 算法得到的上一个数据块的聚类中心。对所有的样本而言,都应该满足式(3)所示的关系。

下面采用拉格朗日极值法求模糊划分矩阵 $\boldsymbol{U}$、$\boldsymbol{T}$ 以及聚类中心 $\boldsymbol{V}$ 的更新公式。

$$G(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\lambda) =$$

$$J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\alpha) - \lambda\left(\sum_{i=1}^{c}\mu_{ik} + \sum_{j=1}^{p}\zeta_{jk} - 1\right) =$$

$$J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V}) + \alpha\sum_{i=1}^{c} \| \boldsymbol{v}_i - \boldsymbol{v}_i^o \|^2 -$$

$$\lambda\left(\sum_{i=1}^{c}\mu_{ik} + \sum_{j=1}^{p}\zeta_{jk} - 1\right) =$$

$$\sum_{i=1}^{c}\mu_{ik}^m \| \boldsymbol{x}_k - \boldsymbol{v}_i \|^2 + \sum_{j=1}^{p}\zeta_{jk}^m \| \boldsymbol{x}_k - \boldsymbol{z}_j \|^2 +$$

$$\alpha\sum_{i=1}^{c} \| \boldsymbol{v}_i - \boldsymbol{v}_i^o \|^2 - \lambda\left(\sum_{i=1}^{c}\mu_{ik} + \sum_{j=1}^{p}\zeta_{jk} - 1\right)$$

$$\forall k \in [1,N] \tag{8}$$

对 $G(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\lambda)$ 中的各个变量分别求偏导并令其等于零得:

$$\begin{cases} \dfrac{\partial_{J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\lambda)}}{\partial_{\mu_{ik}}} = m\sum_{i=1}^{c}\mu_{ik}^{m-1} \| \boldsymbol{x}_k - \boldsymbol{v}_i \|^2 - \lambda = 0 \\ \dfrac{\partial_{J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\lambda)}}{\partial_{\zeta_{ik}}} = m\sum_{i=1}^{c}\zeta_{jk}^{m-1} \| \boldsymbol{x}_k - \boldsymbol{z}_j \|^2 - \lambda = 0 \\ \dfrac{\partial_{J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\lambda)}}{\partial_{\lambda}} = \sum_{i=1}^{c}\mu_{ik} + \sum_{j=1}^{p}\zeta_{jk} - 1 = 0 \\ \dfrac{\partial_{J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V},\lambda)}}{\partial_{\boldsymbol{v}_i}} = -2\sum_{i=1}^{c}\mu_{ik}^m \| \boldsymbol{x}_k - \boldsymbol{v}_i \| + \\ \qquad 2\alpha\sum_{i=1}^{c} \| \boldsymbol{v}_i - \boldsymbol{v}_i^o \| = 0 \end{cases} \tag{9}$$

通过(9)可以很容易地求出模糊划分矩阵的更新公式 μ_{ik} 和 ζ_{jk},如式(4)、(5)所示。可以发现,模糊划分矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{T}$ 与平衡因子 α 无关。

由式(9)第 4 个等式可得

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N}\mu_{ik}^m \boldsymbol{x}_k + \alpha\,\boldsymbol{v}_i^o}{\sum_{k=1}^{N}\mu_{ik}^m + \alpha}, \forall i \in [1,c] \tag{10}$$

从式(10)可以看出,根据平衡因子 α 是否等于 0,又可以分为两种情况。

当 $\alpha = 0$ 即不考虑数据块间聚类中心的相互影响时,在每一个数据块的聚类过程中,将某个数据块产生的聚类中心加入下一个数据块中参与聚类,为了增大对数据块间聚类效果的影响程度,把距聚类中心最近的 n_0 个样本点也一同加入下一个数据块参与聚类,以此类推,直至计算出最后一个数据块的聚类中心,这个最终的聚类中心就是我们所要求的整个数据集的聚类中心。

$\alpha = 0$ 时的情况仅仅考虑了某一数据块的聚类中心及其周围的 n_0 个样本点对下一个数据块的聚类性能的影响,这样得出的聚类效果并不理想。为了提高聚类性能,应该考虑数据块间聚类中心的相互影响即 $\alpha \neq 0$ 时的情况,此时平衡项的加入很好地提高了聚类性能。

如下所述为 *IFCM*(*c*+*p*)算法的具体计算步骤。

输入:$\boldsymbol{X}$, c, p, m, n_0, ε ;

输出:聚类中心 $\boldsymbol{V}$。

1)把样本集 $\boldsymbol{x}$ 随机划分成大小相等的 s 个子集即 $\boldsymbol{x} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_s\}$;

2)定义一个空的集合 $\boldsymbol{X}_{\text{incre}}$ 和 $\boldsymbol{X}_{\text{near}}$;

3)遍历所有的数据块获取聚类中心:

for $l = 1,2,\cdots,s$

①初始化未知类和已知类的聚类中心 $\boldsymbol{V}$、Z;

②把从上一数据块获得的样本 $\boldsymbol{X}_{\text{incre}}$ 添加到当前数据块,即 $\boldsymbol{X}_l = \{\boldsymbol{X}_l \cup \boldsymbol{X}_{\text{incre}}\}$;

③使用式(4)、(5)和(10)计算当前数据块的聚类中心 $\boldsymbol{V}_l$;

④取出距当前数据块的聚类中心最近的 n_0 个样本点存入 $\boldsymbol{X}_{\text{near}}$ 中;

⑤把聚类中心 $\boldsymbol{V}_l$ 及其附近的 n_0 个样本点存入 $\boldsymbol{X}_{\text{incre}}$ 中,即 $\boldsymbol{X}_{\text{incre}} = \{\boldsymbol{V}_l \cup \boldsymbol{X}_{\text{near}}\}$;

end for

上述算法步骤 2)的 $\boldsymbol{X}_{\text{incre}}$ 用以存放每一个数据块产生的聚类中心及其附近的 n_0 个样本点 $\boldsymbol{X}_{\text{near}}$,3)对这 s 个数据块进行遍历,求其聚类中心。3)中的主要迭代过程在每个数据块中使用 FCPM 算法计算聚类中心,使用欧氏距离求距聚类中心最近的 n_0 个样本点,并把它们一同加入到下一个数据块中去参与聚类。注意在初始化聚类中心时,采用前面提到的 FCPM 算法的初始化方法对已知类和未知类的聚类中心 $\boldsymbol{Z}$、$\boldsymbol{V}$ 进行初始化,聚类中心 $\boldsymbol{V}$ 和模糊隶属度矩阵 $\boldsymbol{U}$ 的更新公式分别为(10)、(4), $\| \cdot \|$ 表示求欧氏距离。FCPM 算法的迭代终止于聚类中心的连续变化值的 Frobenius 范数小于 ε。整个 IFCM(c+p)算法终止于所有的数据块遍历结束并获得最终的聚类中心。

2.2　算法的可行性分析

正如传统的增量式聚类算法一样,IFCM(c+p)算法对每个数据块进行聚类。在 IFCM(c+p)算法中,没有添加平衡项时,将每个数据块的 c 个聚类中心及距其最近的 n_0 个样本点作为一次聚类结果的历史信

息加入到新增数据中,即每次都有 $c + n_0$ 个样本点加入到新增数据中参与聚类,那么这些历史信息的加入势必将影响新增数据的聚类效果。如果历史信息恰好位于新增数据附近,则其聚类效果将变好,如果历史信息远离它们,历史信息的加入反而会导致一个很差的聚类效果。对于 SPFCM 算法和 OFCM 算法而言,它们通过添加样本权值以增加聚类效果,在一定程度上比仅仅添加历史信息得到的聚类效果要好,但也存在上面所提到的一些问题。为了克服以上问题,提到的 IFCM(c+p)算法添加了平衡项,通过平衡项中的平衡因子去改变数据块间聚类中心的相互影响程度,此时即便历史信息远离新增数据,通过合理调节平衡因子 α 的取值也可以使得聚类中心吸引它周围的新增数据,从而提高聚类效果。

2.3 算法复杂度

文献[15]详细介绍了 rseFCM、SPFCM 算法的时间和空间复杂度,如表 1 所示,本文提到的 FCPM 及 IFCM(c+p)算法的时间和空间复杂度也如表 1 所示。其中 t 表示非增量式算法的迭代次数, t' 表示增量式算法中每个数据块的平均迭代次数, d 表示数据集维数, c 表示未知类的聚类个数, p 表示已知类的聚类个数, s 表示数据块的个数, n_0 表示在 IFCM(c+p)算法中距每个数据块的聚类中心最近的样本点个数。

表 1 各算法的时间、空间复杂度

Table 1 Time and space complexity of algorithms

算法	时间复杂度	空间复杂度
FCPM	$O(tnd(c+p)+tc)$	$O(n(d+c+p))$
rseFCM	$O(tc^2dn/s)$	$O((d+c)n/s)$
SPFCM	$O(ndt'c^2)$	$O((d+c)n/s)$
IFCM(c+p)	$O(t'nd(c+p)+t'c)$	$O((d+c+p+n_0)n/s)$

如表 1 所示,本文提到的算法均在相同环境下运行,都对同一数据集 X 进行处理,时间复杂度都为 $O(n)$。然而从第 3 部分的实验可以看出,各算法的运行时间存在着显著不同。对于增量式模糊聚类算法,由于它们在每个数据块的处理中能够快速收敛因而可以使得算法总的运行时间减少。

本文提到的增量式模糊聚类算法都是对数据进行分块处理,因此需要计算每个数据块所占用的空间即为 n/s。如表 1 所示,同 rseFCM 和 SPFCM 算法相比,由于 IFCM(c+p)算法需要存储聚类中心及其周围的一些样本,因此需要占用相对较多的存储空间,也就拥有相对高的空间复杂度。

3 相关实验研究

3.1 评价指标

为了公正地对各聚类算法的聚类效果做出合理的评价,本文采用如下 3 种评价指标进行算法的性能分析。

3.1.1 算法运行时间的加速比 speedup

该指标反映了聚类算法在指定数据集下运行时间的比较情况。定义加速比:

$$\text{speedup} = t_{\text{full}}/t_{\text{incremental}}$$

式中: t_{full} 表示在整个数据集下采用 FCPM 算法所运行的时间; $t_{\text{incremental}}$ 表示采用增量式算法比如 SPFCM、IFCM(c+p)等所运行的时间。

2)归一化互信息(normalized mutual information, NMI)[20-21]

$$\text{NMI} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} N_i^j \log\left(\frac{N \cdot N_i^j}{N_i \cdot N_j}\right)}{\sqrt{\sum_{i=1}^{c} N_i \log\left(\frac{N_i}{N}\right)} \cdot \sqrt{\sum_{j=1}^{c} N_j \log\left(\frac{N_j}{N}\right)}}$$

式中: N 表示样本总数, N_i 表示经本文聚类算法之后第 i 簇的样本总数, N_j 表示真实数据集的第 j 类的样本总数, N_i^j 表示第 i 簇与第 j 类的契合程度,即二者共有的样本总数。

3)芮氏指标(rand index, RI)[20-22]

$$\text{RI} = \frac{f_{00} + f_{11}}{N(N-1)/2}$$

式中: f_{00} 表示样本点具有不同的类标签并且属于不同类的配对样本数目, f_{11} 则表示样本点具有相同的类标签并且属于同一类的配对样本数目, N 表示样本总数。

以上 NMI、RI 两种指标,其取值范围均为[0, 1],且取值越靠近 1 越能反映该聚类算法在某数据集下的聚类效果越好,反之越靠近 0 则反映该聚类算法的聚类效果越差。加速比 speedup 越大反映了增量式聚类算法的运行时间越短。

3.2 实验结果

1)实验环境

本文所有的实验均在如表 2 的环境中进行。

2)实验数据集

实验所选取的数据集包括人工数据集 2D15(http://www.uef.fi/en/sipu/datasets)、UCI(http://archive.ics.uci.edu/ml/datasets.html)、标准数据集 waveform、forest 和手写数字数据集 MNIST(http://yann.lecun.com/exdb/mnist/)。各数据集的分布情况如表 3。

表 2　实验环境

Table 2　Experiment environment

结构	具体参数
操作系统	Windows 7 专业版 64 位
处理器	Intel(R)Xeon(R)E5-1620 v2@ 3.7GHz
运行内存	64G
软件及版本	MATLAB 7.11.0.584 (R2010b)

表 3　各数据集的分布情况

Table 3　Distribution of the datasets

数据集	大小	维数	类别数
2D15	5 000	2	15
waveform	5 000	21	3
forest	581 012	54	7
MNIST	70 000	784	10

MNIST 数据集是手写数字集的一个子集,包含了 70 000 张 28 × 28 像素的数字 0~9 的图像,每个像素都在整数 0~255 之间取值。为加快运算,对 MNIST 数据集中的所有样本分别除以 255 进行归一化处理[15]。为方便计算,本文随机取 forest 的 581 000 个样本进行计算。同样,对其他数据集也进行归一化处理以加快运算,即用每个特征的所有样本与该特征的最小值作差再除以该特征的最大值与最小值之差。

3)实验参数设置

本文中所有的参数都按如下取值:模糊指数 m 取 2,最大迭代次数均为 100,迭代终止参数 ε 取 1e-3,聚类中心附近的样本点个数 n_0 取 5,其中数据集 2D15、waveform 重复试验 50 次,由于数据集 MNIST 和 forest 样本过大,我们重复试验 20 次。数据块的大小应由用户指定,但在实验中,由于计算机内存受限,forest 数据集的数据块大小依次取0.1%、0.5%、1%、2.5%、5%,其余均按照整个数据集的 1%、2.5%、5%、10%、25%、50% 随机抽取。取 MNIST 数据集 70% 的样本、forest 数据集 10% 的样本参与 FCPM 算法的聚类。平衡因子 α 的具体取值也由用户指定,但是必须在给定的经验值范围内取值,本文中的所有 α 值均是在多次重复实验中,提到的聚类指标的均值达到最好的时候的取值。我们计算提到的几种算法在各个数据集上的 NMI 和 RI 的最值、均值以及标准差,其中均值反映了算法的平均聚类性能,最值和标准差反映了算法的稳定鲁棒性。

4)算法性能比较

本文采用 SPFCM 算法和 rseFCM 算法同 IFCM(c+p)算法在聚类性能和加速比上进行比较。

1)各算法在数据集上的聚类性能比较

各算法在指定数据集下的聚类性能如表 4~11 所示,其中最优均值已用黑体标出。

表 4　IFCM(c+p)、SPFCM、rseFCM 算法的 NMI 值

Table 4　NMI of IFCM(c+p), SPFCM, rseFCM

样本大小	IFCM(c+p)(α=2.1)		IFCM(c+p)(α=0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
1%	**0.410 7**	0.023 3	0.409 1	0.002 6	0.390 9	0.007 4	0.307 8	0.003 7
	0.385 5	0.439 8	0.398 0	0.417 3	0.356 7	0.424 8	0.306 1	0.333 2
2.50%	**0.432 9**	0.006 2	0.348 4	0	0.361 3	0.001 8	0.319 9	0.004 0
	0.432 0	0.475 6	0.347 2	0.348 5	0.348 8	0.361 6	0.318 5	0.347 4
5%	**0.419 4**	0.007 9	0.334 5	0.001 0	0.336 9	0	0.346 3	0
	0.387 2	0.422 0	0.334 3	0.341 5	0.336 5	0.337 1	0.346 3	0.346 4
10%	**0.341 1**	0	0.333 2	0	0.325 9	0	0.296 4	0
	0.340 9	0.341 5	0.329 9	0.333 3	0.325 8	0.326 3	0.295 8	0.296 8
25%	**0.335 0**	0.004 9	0.330 8	0	0.324 5	0	0.336 2	0
	0.302 5	0.344 2	0.330 7	0.330 9	0.323 7	0.325 1	0.336 1	0.336 3
35%	**0.365 6**	0	0.325 3	0	0.331 1	0	0.325 9	0
	0.361 7	0.366 0	0.325 1	0.325 4	0.331 1	0.331 2	0.325 7	0.326 5
50%	**0.355 6**	0	0.335 4	0	0.336 1	0	0.324 1	0
	0.355 4	0.355 8	0.335 3	0.335 4	0.334 9	0.336 5	0.323 9	0.324 6
FCPM	0.328 5				0.008 1			
	0.289 2				0.330 2			

从各表中的实验结果对比发现,增量式模糊聚类算法的聚类性能均优于 FCPM 算法。在人工数据

集 2D15 的聚类性能比较中发现数据块大小取 25%、35%和 50%时,rseFCM 算法和 SPFCM 算法的聚类性能略优于 IFCM(c+p)算法,对类似 2D15 这样的小样本数据集而言这种情况是可能的,而 IFCM(c+p)算法在大规模数据集的聚类问题上可以表现出很好的效果。在本文提到的其他数据集中,IFCM(c+p)算法均能保持最好的聚类性能。在高维大样本的手写数字集 MNIST 和大样本的 forest 数据集的实验结果中可以发现 IFCM(c+p)算法在提高了聚类性能的同时还具备很好的稳定鲁棒性,这是本文其他算法不具备的。另外还可以发现随着数据块大小的增加,所有增量式模糊聚类算法的聚类性能均呈下降趋势,这是由于本文提及的增量式算法均采用分块处理的方式,随着数据块大小的增加直至接近原数据集大小时,在某数据块中聚类就相当于在整个数据集上进行聚类,很明显这样增加算法运行时间的同时还降低了聚类性能。另外还注意到对于大样本数据,随机抽取的数据块较小时 rseFCM 算法会由于无法加载进内存而致使该算法失效,而 IFCM(c+p)算法不用担心这个问题。

表 5 IFCM(c+p)、SPFCM、rseFCM 算法在 2D15 数据集中的 NMI 值

Table 5 NMI of IFCM(c+p), SPFCM, rseFCM for 2D15 dataset

样本大小	IFCM(c+p)(α=2.1)		IFCM(c+p)(α=0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
1%	**0.939 9**	0.024 4	0.843 8	0.031 2	0.926 0	0.016 4	0.868 8	0.021 3
	0.878 5	0.987 0	0.769 9	0.905 5	0.897 2	0.949 4	0.819 5	0.913 7
2.50%	**0.934 8**	0.010 0	0.869 5	0.019 3	0.930 2	0.016 3	0.896 5	0.025 2
	0.900 0	0.955 3	0.827 8	0.923 8	0.904 2	0.951 3	0.829 8	0.941 0
5%	**0.951 9**	0.015 9	0.912 3	0.008 6	0.943	0.016 5	0.903 5	0.020 2
	0.915 5	0.971 5	0.878 3	0.926 2	0.905 5	0.960 6	0.865 2	0.943 3
10%	**0.936 9**	0.008 4	0.877 5	0.004 7	0.931 3	0.011 5	0.920 0	0.019 3
	0.928 4	0.950 7	0.871 7	0.888 2	0.890 6	0.944 9	0.877 6	0.943 2
25%	0.926 0	0	0.864 6	0.013 0	0.922 2	0.014 7	**0.929 4**	0.019 5
	0.925 5	0.926 3	0.825 9	0.871 3	0.895 8	0.940 5	0.886 6	0.950 9
35%	0.913 7	0.024 8	0.909 7	0	**0.927 0**	0.015 7	0.921 4	0.020 8
	0.895 2	0.947 4	0.908 7	0.910 4	0.899 2	0.943 1	0.865 2	0.943 8
50%	0.911 8	0	0.907 9	0	**0.921**	0.017 4	0.916 9	0.02
	0.911 1	0.912 2	0.907 6	0.908 3	0.883 9	0.943 9	0.869 6	0.942 5
FCPM	0.860 7				0			
	0.860 4				0.860 8			

表 6 IFCM(c+p)、SPFCM、rseFCM 算法在 MNIST 数据集中的 NMI 值

Table 6 NMI of IFCM(c+p), SPFCM, rseFCM for MNIST dataset

样本大小	IFCM(c+p)(α=160)		IFCM(c+p)(α=0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
1%	**0.313 9**	0	0.244 7	0	0.224 6	0.070 9	—	—
	0.313 9	0.313 9	0.244 7	0.244 7	0.133 7	0.334 5	—	—
2.50%	**0.260 6**	0	0.239 0	0	0.258 8	0.031 0	—	—
	0.260 6	0.260 6	0.239 0	0.239 0	0.191 5	0.317 3	—	—
5%	**0.297 4**	0	0.164 6	0	0.242 4	0.042 3	—	—
	0.297 4	0.297 4	0.164 6	0.164 6	0.141 7	0.298 7	—	—

续表 6

样本大小	IFCM(c+p)(α =160)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
10%	**0.321 7**	0	0.208 9	0	0.229 9	0	—	—
	0.321 7	0.321 7	0.208 9	0.208 9	0.229 6	0.230 1	—	—
25%	**0.263 4**	0	0.218 0	0	0.181 8	0	—	—
	0.263 4	0.263 4	0.218 0	0.218 0	0.180 7	0.183 2	—	—
35%	**0.330 9**	0	0.171 2	0	0.202 7	0.013 1	—	—
	0.330 9	0.330 9	0.171 2	0.171 2	0.176 3	0.232 8	—	—
50%	**0.213 5**	0	0.207 2	0	0.174 4	0.018 8	—	—
	0.213 5	0.213 5	0.207 2	0.207 2	0.163 2	0.203 7	—	—
FCPM(70%)		0.172 3				0		
		0.172 3				0.172 3		

表 7　IFCM(c+p)、SPFCM、rseFCM 算法在 forest 数据集中的 NMI 值

Table 7　NMI of IFCM(c+p), SPFCM, rseFCM for forest dataset

样本大小	IFCM(c+p)(α =21)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
0.1%	**0.159 3**	0.001 6	0.103 6	0	0.116 1	0.008 9	—	—
	0.158 3	0.163 1	0.103 6	0.103 6	0.104 2	0.137 0	—	—
0.5%	**0.112 3**	0	0.101 6	0	0.101 2	0.004 9	—	—
	0.112 3	0.112 5	0.101 6	0.101 6	0.094 3	0.114 3	—	—
1%	**0.126 4**	0	0.105 5	0	0.110 2	0.005 6	—	—
	0.126 0	0.127 3	0.105 5	0.105 5	0.103 6	0.122 5	—	—
2.50%	**0.107 7**	0	0.106 4	0	0.100 3	0.003 0	—	—
	0.107 7	0.107 7	0.106 4	0.106 4	0.096 3	0.107 5	—	—
5%	**0.109 2**	0	0.102 1	0	0.102 5	0.005 0	—	—
	0.105 3	0.109 4	0.102 1	0.102 1	0.098 8	0.111 3	—	—
FCPM(10%)		0.101 5				0		
		0.101 5				0.101 5		

表 8　IFCM(c+p)、SPFCM、rseFCM 算法在 waveform 数据集中的 RI 值

Table 8　RI of IFCM(c+p), SPFCM, rseFCM for waveform dataset

样本大小	IFCM(c+p)(α =2.1)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
0.1%	**0.702 3**	0.009 5	0.680 4	0.001 8	0.669 1	0.330 1	0.657 2	0.001 8
	0.684 1	0.717 6	0.676 7	0.687 3	0.663 7	0.688 2	0.6565 6	0.669 7
2.50%	**0.689 0**	0.001 5	0.663 8	0	0.674 3	0	0.663 3	0.002 8
	0.688 8	0.699 6	0.662 7	0.663 9	0.674 3	0.674 8	0.662 3	0.682 4
5%	**0.700 9**	0.002 4	0.665 9	0	0.662 5	0	0.665 8	0
	0.691 1	0.701 7	0.665 8	0.667 4	0.662 2	0.662 6	0.665 8	0.665 9
10%	**0.666 6**	0	0.665 1	0	0.661 9	0	0.653 1	0
	0.666 4	0.670 8	0.663 2	0.665 2	0.661 9	0.662 1	0.652 9	0.653 3

续表 8

样本大小	IFCM(c+p)(α =2.1)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
25%	**0.664 3**	0.001 8	0.663 3	0	0.661 3	0	0.663 3	0
	0.653 7	0.669 3	0.663 3	0.663 4	0.661 2	0.661 5	0.663 2	0.663 4
35%	**0.670 2**	0	0.664 0	0	0.663 2	0	0.662 3	0.002 4
	0.669 2	0.670 3	0.663 8	0.664	0.663 2	0.663 3	0.661 9	0.679 0
50%	**0.669 9**	0	0.664 4	0	0.665 1	0	0.660 7	0
	0.669 8	0.670 1	0.664 4	0.664 4	0.664 8	0.665 2	0.660 7	0.660 8
FCPM		0.662 2				0.002 7		
		0.649 2				0.662 7		

表 9 IFCM(c+p)、SPFCM、rseFCM 算法在 2D15 数据集中的 RI 值

Table 9 RI of IFCM(c+p), SPFCM, rseFCM for 2D15 dataset

样本大小	IFCM(c+p)(α =0.2)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
0.1%	**0.979**	0.009 7	0.938 3	0.012 7	0.975 5	0.006	0.969 7	0.007 9
	0.955 9	0.997 6	0.913 5	0.966 5	0.966 5	0.983 7	0.950 4	0.985 6
2.50%	**0.984 8**	0.002 4	0.960 5	0.006 4	0.981 8	0.005 4	0.977 3	0.008
	0.975 6	0.987 9	0.946 1	0.976 1	0.971 7	0.988 1	0.956 1	0.991 2
5%	**0.989 7**	0.005 0	0.976 7	0.003 0	0.986 9	0.004 9	0.978 7	0.006 4
	0.979 2	0.994 6	0.964 8	0.980 5	0.977 3	0.991 6	0.964 7	0.991 5
10%	**0.987 1**	0.003 3	0.970 1	0.001 9	0.986 4	0.003 9	0.983 6	0.006 7
	0.983 6	0.991 4	0.967 7	0.972 0	0.974 3	0.990 3	0.968 7	0.991 7
25%	0.983 5	0	0.965 1	0.002 9	0.984 9	0.005 2	**0.986 1**	0.006 1
	0.983 4	0.983 5	0.956 6	0.967 8	0.976 3	0.990 1	0.970 8	0.992 9
35%	0.983 2	0.006 6	0.980 4	0	**0.986 0**	0.005 3	0.984 4	0.006 7
	0.978 3	0.993 2	0.980 3	0.980 5	0.977 9	0.991 0	0.967 2	0.991 7
50%	0.980 1	0	0.981 2	0	**0.984 2**	0.005 6	0.982 9	0.006 6
	0.979 9	0.980 2	0.981 2	0.981 3	0.971 1	0.991 5	0.965 2	0.991 3
FCPM		0.968 3				0		
		0.968 3				0.968 4		

表 10 IFCM(c+p)、SPFCM、rseFCM 算法在 MNIST 数据集中的 RI 值

Table 10 RI of IFCM(c+p), SPFCM, rseFCM for MNSIT dataset

样本大小	IFCM(c+p)(α =160)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
0.1%	**0.786 4**	0	0.722 2	0	0.735 2	0.096 1	—	—
	0.786 4	0.786 4	0.722 2	0.722 2	0.606 3	0.840 1	—	—
2.50%	**0.793 5**	0	0.712 3	0	0.792 4	0.033 6	—	—
	0.793 5	0.793 5	0.712 3	0.712 3	0.692 5	0.830 0	—	—
5%	**0.757 6**	0	0.612 9	0	0.754 1	0.050 3	—	—
	0.757 6	0.757 6	0.612 9	0.612 9	0.644 8	0.822 1	—	—

续表 10

样本大小	IFCM(c+p)(α =160)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
10%	**0.811 7**	0	0.642 9	0	0.794 7	0	—	—
	0.811 7	0.811 7	0.642 9	0.642 9	0.794 4	0.795 3	—	—
25%	**0.737 5**	0	0.660 8	0	0.727 4	0	—	—
	0.737 5	0.737 5	0.660 8	0.660 8	0.727 0	0.728 9	—	—
35%	**0.772 9**	0	0.607 6	0	0.721 5	0.009 9	—	—
	0.772 9	0.772 9	0.607 6	0.607 6	0.703 2	0.740 8	—	—
50%	**0.663 3**	0	0.661 2	0	0.642 0	0.022 4	—	—
	0.663 3	0.663 3	0.661 2	0.661 2	0.609 4	0.691 6	—	—
FCPM(70%)			0.613 4			0		
			0.613 4			0.613 4		

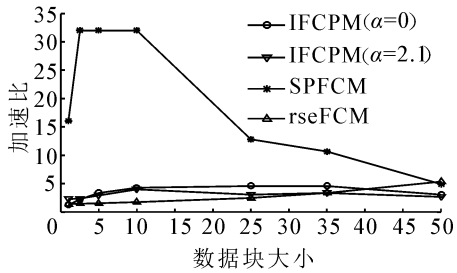
表 11　IFCM(c+p)、SPFCM、rseFCM **算法在** forest **数据集中的** RI **值**

Table 11　**RI of IFCM(c+p), SPFCM, rseFCM for forest dataset**

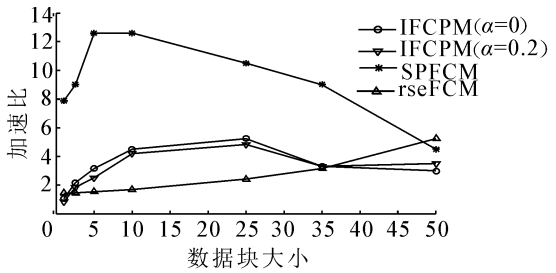
样本大小	IFCM(c+p)(α =21)		IFCM(c+p)(α =0)		SPFCM		rseFCM	
	avg.	std.	avg.	std.	avg.	std.	avg.	std.
0.1%	**0.582 8**	0	0.569 7	0	0.580 4	0.012 2	—	—
	0.582 7	0.583	0.569 7	0.569 7	0.561 6	0.600 7	—	—
0.5%	0.564 8	0	0.560 8	0	**0.567 4**	0	—	—
	0.564 8	0.564 8	0.560 8	0.560 8	0.558 3	0.587	—	—
1%	**0.573 7**	0	0.564 1	0	0.572 8	0.007 2	—	—
	0.573 7	0.573 7	0.564 1	0.564 1	0.564 3	0.588 7	—	—
2.50%	0.568 0	0	0.564 0	0	**0.572 2**	0.009 3	—	—
	0.568 0	0.568 0	0.564 0	0.564 0	0.561 5	0.590 4	—	—
5%	**0.569 6**	0	0.562 6	0	0.568 8	0.018 8	—	—
	0.569	0.569 6	0.562 6	0.562 6	0.561 5	0.594 1	—	—
FCPM(10%)			0.562 8			0		
			0.562 8			0.562 8		

2)各算法在数据集上运行时间的加速比比较

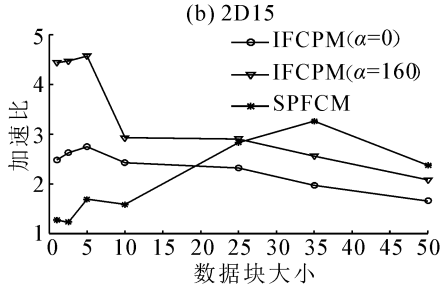
各个算法相对于 FCPM 算法在不同数据集上不同大小的数据块下运行时间的加速比的比较情况下图 1 所示。



(a) waveform



(b) 2D15



(c) MNIST

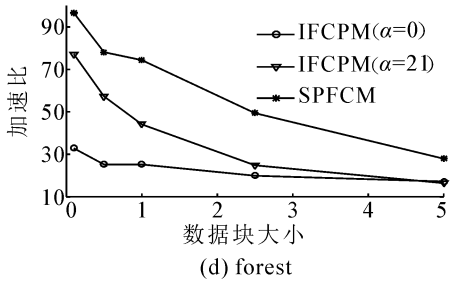


图 1 IFCM(c+p)、SPFCM、rseFCM 算法在不同数据集的不同大小数据块下的加速比

Fig.1 Speedup ratio of IFCM(c+p), SPFCM, rseFCM for different chunk size of different datasets

从图 1 中可以看出,本文提到的算法的运行时间的加速比基本上随着数据块大小的增加呈下降趋势,这是由于随着数据块大小的增加,提到的算法单次运行的样本总量在增加,因而运行时间会随之增加。在小样本数据集 waveform 和 2D15 中,IFCM(c+p)算法的运行时间高于 SPFCM 算法。由于 forest 数据集的数据块取得较小,因而 SPFCM 算法在此时的运行时间也较短。而在大样本的 MNIST 数据集中,SPFCM 算法的加速程度明显降低,IFCM(c+p)算法的加速程度明显提高。由此可见,IFCM(c+p)算法会随着数据集样本的增加而加速程度得到提高,尤其是对于一类聚类中心已知的大规模数据集,该算法的运行时间会大幅降低。

4 结束语

针对 FCPM 算法对大样本数据聚类性能较差甚至可能出现算法失效的问题,本文在该算法的基础上提出了 IFCM(c+p)算法,特别是适合处理某一类已知的大规模数据集的聚类问题。通过对每一个数据块使用 FCPM 算法获取其聚类中心,并把它们及其附近的一些样本点加入到下一个数据块中参与聚类,同时添加平衡项以提高聚类性能。通过第 3 部分的实验可以发现,平衡项的加入提高了 IFCM(c+p)算法的聚类性能和运行时间,另外还保持了很好的稳定鲁棒性。平衡项中的平衡因子的合理选择是 IFCM(c+p)算法的关键所在,本文中所采用的方法是根据经验值,保证公式(7)中的 $J(\boldsymbol{U},\boldsymbol{T},\boldsymbol{V})$ 值与平衡项尽量处于同一数量级,取在各数据集下 IFCM(c+p)算法能够达到最好的聚能性能时的 α 值作为算法的最佳平衡因子。对于如何才能选取更好的平衡因子 α ,如何既保证算法的聚类性能又提高运行时间,都是我们继续研究的方向。

参考文献:

[1] BEZDEK J C, EHRLICH R, FULL W. FCM: the fuzzy c-means clustering algorithm[J]. Computers & Geosciences, 1984, 10(2): 191-203.

[2] CAN F, DROCHAK N D II. Incremental clustering for dynamic document databases[C]//Proceedings of the 1990 Symposium on Applied Computing. Fayetteville, AR, USA, 1990: 61-67.

[3] KAUFMAN L, ROUSSEEUW P J. Finding groups in data: an introduction to cluster analysis[M]. New York: John Wiley & Sons, 2009: 830-832.

[4] GUHA S, RASTOGI R, SHIM K. Cure: an efficient clustering algorithm for large databases[J]. Information systems, 2001, 26(1): 35-58.

[5] CAN F. Incremental clustering for dynamic information processing[J]. ACM transactions on information systems, 1993, 11(2): 143-164.

[6] CAN F, FOX E A, SNAVELY C D, et al. Incremental clustering for very large document databases: Initial MARIAN experience[J]. Information sciences, 1995, 84(1/2): 101-114.

[7] ZHANG Tian, RAMAKIRSHNAN R, LIVNY M. BIRCH: An efficient data clustering method for very large databases[C]//Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. New York, USA, 1996: 103-114.

[8] NG R T, HAN Jiawei. CLARANS: A method for clustering objects for spatial data mining[J]. IEEE transactions on knowledge and data engineering, 2002, 14(5): 1003-1016.

[9] SHANKER B U, PAL N R. FFCM: An effective approach for large data sets[C]//Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing. Iizuka, Japan, 1994: 331-332.

[10] CHENG Taiwai, GOLDGOF D B, HALL L O. Fast clustering with application to fuzzy rule generation[C]//Proceedings of 1995 IEEE International Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium. Yokohama, Japan, 1995: 2289-2295.

[11] KOLEN J F, HUTCHESON T. Reducing the time complexity of the fuzzy c-means algorithm[J]. IEEE transactions on fuzzy systems, 2002, 10(2): 263-267.

[12] KOTHARI D, NARAYANAN S T, DEVI K K. Extended fuzzy c-means with random sampling techniques for clustering large data[J]. International journal of innovative research in advanced engineering (IJIRAE), 2014, 1(1):

1-4.

[13] HORE P, HALL L O, GOLDGOF D B. Single pass fuzzy c means[C]//Proceedings of IEEE International Fuzzy Systems Conference. London, UK, 2007: 1-7.

[14] HORE P, HALL L O, GOLDGOF D B, et al. Online fuzzy c means[C]//Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society. New York, USA, 2008: 1-5.

[15] HAVENS T, BEZDEK J, LECKIE C, et al. Fuzzy c-means algorithms for very large data[J]. IEEE transactions on fuzzy systems, 2012, 20(6): 1130-1146.

[16] WANG Yangtao, CHEN Lihui, MEI Jianping. Incremental fuzzy clustering with multiple medoids for large data[J]. IEEE transactions on fuzzy systems, 2014, 22(6): 1557-1568

[17] BÖHM C, PLANT C, SHAO J, et al. Clustering by synchronization[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA, 2010: 583-592.

[18] 应文豪, 许敏, 王士同, 等. 在大规模数据集上进行快速自适应同步聚类[J]. 计算机研究与发展, 2014, 51(4): 707-720.
YING Wenhao, XU Min, WANG Shitong, et al. Fast adaptive clustering by synchronization on large scale datasets [J]. Journal of computer research and development, 2014, 51(4): 707-720.

[19] LESKI J M. Fuzzy (c+p) -means clustering and its application to a fuzzy rule-based classifier: towards good generalization and good interpretability[J]. IEEE transactions on fuzzy systems, 2014, 23(4): 802-812.

[20] LIU Jun, MOHAMMED J, CARTER J, et al. Distance-Based clustering of CGH data[J]. Bioinformatics, 2006, 22(16): 1971-1978.

[21] DENG Zhaohong, CHOI K S, CHUNG Fulai, et al. Enhanced soft subspace clustering integrating within-cluster and between-cluster information[J]. Pattern recognition, 2010, 43(3): 767-781.

[22] RAND W M. Objective criteria for the evaluation of clustering methods[J]. Journal of the American statistical association, 1971, 66(336): 846-850.

作者简介:

李滔,男,1990 年生,硕士研究生,主要研究方向为人工智能与模式识别、模糊聚类算法、增量式学习。

王士同,男,1964 年生,教授,博士生导师,中国离散数学学会常务理事,中国机器学习学会常务理事。主要研究方向为人工智能/模式识别、图像处理及其应用等。发表学术论文近百篇,其中被 SCI、EI 检索 50 余篇。