自动确定聚类个数的模糊聚类算法

陈海鹏[1,2],申铉京[1,2],龙建武[3],吕颖达[4]
(1. 吉林大学计算机科学与技术学院,吉林长春 130012;2. 吉林大学符号计算与知识工程教育部重点实验室,吉林长春 130012;
3. 重庆理工大学计算机科学与工程学院,重庆 400054;4. 吉林大学公共计算机教学与研究中心,吉林长春 130012)

摘 要: 本文通过集成多次FCM(Fuzzy C-Means)聚类结果以及采用软化分方式,提出一种新的自动确定聚类个数的模糊聚类算法.本算法首先利用不同的聚类数目对数据进行FCM聚类,然后充分利用多次FCM聚类得到的隶属度信息构建一个累积邻接矩阵,最后采用迭代方式对累积邻接矩阵进行图切分以获取最终聚类结果.大量的仿真实验表明,相对现有集成聚类方法,本文方法能够有效减少FCM的聚类次数,并且在图切分过程中的迭代次数为现有方法的1/2左右.

关键词: 模糊聚类;FCM算法;图切分
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2017)03-0687-08
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.03.028

Fuzzy Clustering Algorithm for Automatic Identification of Clusters

CHEN Hai-peng[1,2], SHEN Xuan-jing[1,2], LONG Jian-wu[3], LÜ Ying-da[4]
(1. *College of Computer Science and Technology, Jilin University, Changchun, Jilin* 130012, *China*;
2. *Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin* 130012, *China*;
3. *College of Computer Science and Engineering, Chongqing University of Technology, Chongqing* 400054, *China*;
4. *Center for Computer Fundamental Education, Jilin University, Changchun, Jilin* 130012, *China*)

Abstract: To automatically determine the number of clusters, a new fuzzy clustering algorithm is proposed in this study, which is based on soft partition scheme and integrates many FCM clustering results. In this method, FCM clustering is implemented on data by the cluster number; then the membership information is used to build a cumulative adjacency matrix; finally, the graph cut method is adopted to the cumulative adjacency matrix by iterative manner to obtain clustering results. Simulation experiments show that, compared to the current integrated clustering method, our method can effectively reduce the number of FCM clustering; furthermore, its iterations in the graph cut process is about 1/2 of the existing method.

Key words: fuzzy clustering; fuzzy C-means algorithm; graph partition

1 引言

数据聚类是指将一个数据集划分成多个数据子集,并且处于同一数据子集的数据样本间具有较大的相似度,而处于不同子集数据间样本的相似度较小[1~3].聚类是进行有效数据挖掘的一项重要研究方法,在图像分割、模式识别、计算机视觉等领域中有着十分广泛的研究与应用[1~13].

根据数据在聚类中的集聚规则及应用这些规则的方法,可以将聚类分为四类[1,2]:层次化聚类算法、划分式聚类算法、基于密度的聚类算法和基于网格的聚类算法.其中划分式聚类算法因其具有简单、有效等特性而得到了广泛地研究与应用[1~17].典型的划分式聚类算法有K-Means算法、K-Medoid算法、FCM算法等[1,2].前两种算法属于硬划分聚类方法,后一种算法属于软划分聚类方法.由于引入了模糊信息,FCM算法得到了更为广泛地关注.

为完成准确聚类,传统FCM算法需要事先指定聚类个数.如在图像分割领域,由Cai等人提出的结合局部信息的快速和鲁棒的模糊聚类算法(FGFCM)[5]、由

收稿日期:2015-07-17;修回日期:2015-10-08;责任编辑:梅志强
基金项目:国家自然科学基金(No. 61305046, No. 61502065);吉林省自然科学基金(No. 20140101193JC, No. 20130522117JH, 20150101055JC);重庆市基础与前沿研究计划项目(No. cstc2015jcyjBX0127)

Krinidis 等人提出的一种鲁棒的使用局部信息的聚类算法(FLICM)[6]等,都需要人为指定聚类个数,因此很大程度上限制了 FCM 算法的灵活性. 然而,实际应用中通常事先并不知道待处理数据集的分布情况. 因此,如何确定待处理数据集的聚类个数引起了研究者的广泛关注. 如在彩色图像分割中,Yu 等人利用蚁群优化算法来确定聚类中心[7],但该方法时空开销很大. Tan 等人利用颜色信息子分量直方图中的主峰作为聚类中心[8],该方法对图像聚类非常有效,而对于其它的一些数据集,一般很难有效地统计出其直方图,因此该方法的适用范围非常有限. 另外,聚类有效性指标(Cluster Validity Index,CVI)是一种常见的聚类结果评价指标,直至目前仍有大量研究者对该类方法进行研究[14~17]. 此类方法是通过选取不同的聚类个数进行聚类,然后根据 CVI 来选取出最佳的聚类结果. 对于一些简单的、线性可分的数据集,CVI 方法比较有效,但是对于那些分布复杂、线性不可分的数据集,该方法可行性较差.

对于分布复杂、线性不可分的数据集,即使给定最佳的聚类个数,采用 FCM 算法也不能对其准确聚类[9]. 为解决该问题,基于核方法的模糊聚类算法(KFCM)[10~13]得到了广泛研究. 对于这些在当前维度下不可分的数据集,通过核方法向高维度或低维度进行投影后,可使得投影后得到的数据集变得线性可分. 然后再采用 FCM 算法即可准确地完成聚类,但代价是带来了较大的时空开销. 另外,对于聚类个数、核函数及其参数的选取都使得整个过程更为复杂,通常这些参数的选择都是凭借经验,因此其实用性相对较差[13]。最近,Mok 等人提出了一种自动确定聚类个数的聚类算法[9]. 但是,该方法在构造累积邻接矩阵时采用了硬划分方式,从而导致其需要进行更多次的 FCM 聚类,同时在后续的图切分过程中也需要更多次迭代.

基于此,本文从自动确定聚类个数的角度,借助概率论等相关知识,提出了一种新的模糊聚类算法. 该算法充分利用每次 FCM 聚类结果得到的隶属度信息,重新构造累积邻接矩阵,并且采用软划分方式,从而大大降低了后续图切分过程中的迭代次数,同时仅需较少次数的 FCM 聚类就可得到与原始方法相同的聚类结果.

2 相关工作

属于同一组的数据样本间的相关性要强于属于不同组的数据样本间的相关性,即属于同一组的数据集被划分到一类的概率要远远大于属于不同组的数据集被划分到一类的概率,根据这一特性,Mok 等人提出了一种自动确定聚类个数的聚类算法[9].

该算法首先给定最大聚类个数 K,然后采用 FCM 进行多次聚类,每次的聚类个数为 C,其中 C 的取值范围为$[2,K]$. 对于 K 值的选取并没有核聚类方法中相关参数选取那么复杂,只要选取较大的 K 值便可获得理想的聚类效果. 根据每次聚类得到隶属度矩阵 $\boldsymbol{U}_C$,该方法采用硬划分方式将每个样本点 x_i 划分到具有最大隶属度的那一聚类中,由此可以得到一个由 N 个样本所属聚类编号组成的一维向量 $\boldsymbol{L}_C=[l_1,l_2,\cdots,l_N]$,其中 $l_i=\arg\max\limits_{1\leqslant k\leqslant C}\{u_{ik}\}$,$u_{ik}$为第 i 样本隶属于第 k 个聚类的隶属度. 然后根据向量 $\boldsymbol{L}_C$ 便可构造出一个 N 个样本点间的邻接矩阵 $\boldsymbol{O}_C=[o_{ij}]_{N\times N}$,其中 o_{ij}定义如式(1)所示:

$$o_{ij}=\begin{cases}1, & \text{if } l_i=l_j(i\neq j)\\ 0, & \text{else}\end{cases} \tag{1}$$

即当第 i 个样本与第 j 个样本属于同一聚类时,$o_{ij}=1$,否则 $o_{ij}=0$,显然该邻接矩阵为一个对称矩阵. 然后,将所有聚类得到的邻接矩阵 $\boldsymbol{O}_C$ 进行累加,其中 $C\in[2,K]$,从而得到一个累积邻接矩阵 $\boldsymbol{J}$:

$$\boldsymbol{J}=\sum_{C=2}^{K}\boldsymbol{O}_C \tag{2}$$

累积邻接矩阵 $\boldsymbol{J}$ 由于集成了不同聚类个数下的 FCM 聚类结果,因此,增强了属于同一组数据间的聚合程度,同时也削弱了属于不同组数据间的聚合程度.

但是,该方法在构造邻接矩阵 $\boldsymbol{O}_C$ 时采用了硬划分方式,没有充分利用每次聚类结果所得到的隶属度信息,从而在其邻接矩阵 $\boldsymbol{O}_C$ 中无法体现出各个样本间的聚合程度.

3 自动确定聚类个数的模糊聚类算法

针对文献[9]中的问题,本文充分利用隶属度信息,提出一种自动确定聚类个数的模糊聚类算法. 该算法采用软划分方式,使得每次构造出的邻接矩阵更为合理,并且在后续图切分过程中可以有效降低迭代次数,同时可以适当扩大 K 的取值范围.

3.1 算法原理及过程

在每次完成 FCM 聚类时,本文方法不仅保存每个样本所属的聚类编号,而且还保存对应的最大隶属度信息. 因此,本文按照式(3)对相应的矩阵 $\boldsymbol{L}_C$ 进行重新定义.

$$\boldsymbol{L}_C=\begin{bmatrix}l_1,l_2,\cdots,l_N\\ u_1,u_2,\cdots,u_N\end{bmatrix} \tag{3}$$

其中,$l_i=\arg\max\limits_{1\leqslant k\leqslant C}\{u_{ik}\}$,$u_i=\max\limits_{1\leqslant k\leqslant C}\{u_{ik}\}$. 式中 $\text{argmax}\{\cdot\}$ 用于计算使得隶属度 u_{ij}最大的聚类编号 l_i;而 $\max\{\cdot\}$ 用于计算隶属度 u_{ij}的最大值.

任取数据集中两个样本点 x_i 和 x_j,相应的聚类编号分别为 l_i 和 l_j,相应的最大隶属度分别为 u_i 和 u_j. 根据概率论相关知识,若将 x_i 被划分到第 l_i 类和 x_j 被划分到第 l_j 类看作两个独立事件,则这两个事件分别发生的概率分别为 $p_i=u_i$ 和 $p_j=u_j$. 若 $l_i=l_j$,即样本点 x_i

和 x_j 被划分到同一类,则这两个独立事件同时发生的概率为:$p_{ij} = p_i \cdot p_j$. 因此,本文对邻接矩阵 $\boldsymbol{O}_C = [\overline{O}_{ij}]_{N\times N}$ 进行重新定义,其中 $\overline{O}_{ij}$ 定义如式(4)所示:

$$\overline{O}_{ij} = \begin{cases} u_i \cdot u_j, & \text{if } l_i = l_j (i \neq j) \\ 0, & \text{else} \end{cases} \tag{4}$$

在集成所有聚类结果时,累积邻接矩阵仍按公式 $\boldsymbol{J} = \sum_{C=2}^{K} \boldsymbol{O}_C$ 进行计算.

由于本文方法得到的累积邻接矩阵为浮点数矩阵,而文献[9]中的图切分过程要求为整数矩阵,因此需要对累积邻接矩阵做进一步处理. 为了完成对线性不可分的数据集进行准确聚类,通常需要选取较大的 K 值,因此可以忽略累积邻接矩阵中小于 1 的值,而对大于等于 1 的值做四舍五入取整操作,如式(5)所示:

$$\boldsymbol{J}'_{ij} = \begin{cases} \text{INT}(\boldsymbol{J}_{ij} + 0.5), & \text{if } J_{ij} \geqslant 1 \\ 0, & \text{else} \end{cases} \tag{5}$$

其中 INT(·)表示取整操作.

完成累积邻接矩阵 $\boldsymbol{J}'$ 的构造后,采用迭代图切分方法[9]对其进行处理,具体过程如算法 1.

算法 1

输入:累积邻接矩阵 $\boldsymbol{J}'$

输出:最佳聚类个数 ClusterNumber 和聚类结果 SubGraph

方法:

1. 初始化:$t = 0$,$\boldsymbol{J}^{(t)} = \boldsymbol{J}'$;
2. 采用图深度优先搜索方法查找累积矩阵 $\boldsymbol{J}^{(t)}$ 中的连通子图 SubGraph^t 和子图个数 ClusterNumber^t;
3. 累积邻接矩阵减 1:

$$\boldsymbol{J}_{ij}^{(t+1)} = \begin{cases} \boldsymbol{J}_{ij}^{(t)} - 1, & \text{if } \boldsymbol{J}_{ij}^{(t)} > 0 \\ 0, & \text{else} \end{cases};$$

4. 如果矩阵 $\boldsymbol{J}^{(t+1)}$ 为零矩阵,转步骤 5,否则令 $t = t + 1$ 并转步骤 2;
5. 统计出现次数最多的聚类个数 ClusterNumber(其中 ClusterNumber > 1)和相应的连通子图 SubGraph.

需要说明的是,由于 K 通常取值较大,从而导致初始得到的累积邻接矩阵 $\boldsymbol{J}$ 中仅含有一个连通子图,并且这个连通子图需要经过多次图切分操作才能产生新的连通子图,因此在步骤 5 中需要增加聚类个数大于 1 的限制条件.

3.2　聚类实例

假设给定包含 6 个数据点的数据集:$\boldsymbol{X} = \begin{bmatrix} 1558 & 15 & 58 & 30 & 36 \\ 18 & 20 & 23 & 26 & 39 & 39 \end{bmatrix}$,其分布图如图 1 所示:

以图 1 中的数据集为例,演示本文中聚类算法的执行过程,如下所示:

当取最大聚类个数 $K = 4$,聚类个数 C 取 2、3 和 4 时,采用 FCM 算法对数据集 $\boldsymbol{X}$ 进行聚类,得到的聚类结果向量 $\boldsymbol{L}_C$ 分别如下:

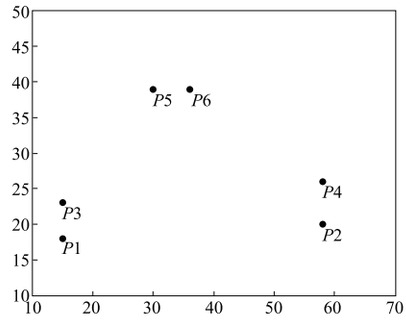


图1　数据集$\boldsymbol{X}$的样本分布

$$\boldsymbol{L}_{C=2} = \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 1 \\ 0.9283 & 0.9849 & 0.9640 & 0.9950 & 0.8161 & 0.6468 \end{bmatrix}$$

$$\boldsymbol{L}_{C=3} = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & 2 \\ 0.9887 & 0.9862 & 0.9860 & 0.9841 & 0.9760 & 0.9770 \end{bmatrix}$$

$$\boldsymbol{L}_{C=4} = \begin{bmatrix} 1 & 2 & 1 & 4 & 3 & 3 \\ 0.9855 & 1.0000 & 0.9827 & 1.0000 & 0.9681 & 0.9650 \end{bmatrix}$$

根据向量 $\boldsymbol{L}_C$,利用式(4)可以分别得到相应的邻接矩阵 $\boldsymbol{O}_C$:

$$\boldsymbol{O}_{C=2} = \begin{bmatrix} 0.000 & 0.000 & 0.895 & 0.000 & 0.758 & 0.600 \\ 0.000 & 0.0000 & .000 & 0.980 & 0.000 & 0.000 \\ 0.895 & 0.000 & 0.000 & 0.000 & 0.787 & 0.624 \\ 0.000 & 0.980 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.758 & 0.000 & 0.787 & 0.000 & 0.000 & 0.528 \\ 0.600 & 0.000 & 0.624 & 0.000 & 0.528 & 0.000 \end{bmatrix}$$

$$\boldsymbol{O}_{C=3} = \begin{bmatrix} 0.000 & 0.000 & 0.975 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.971 & 0.000 & 0.000 \\ 0.975 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.971 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.954 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.954 & 0.000 \end{bmatrix}$$

$$\boldsymbol{O}_{C=4} = \begin{bmatrix} 0.000 & 0.000 & 0.968 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.968 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.934 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.934 & 0.000 \end{bmatrix}$$

从上述邻接矩阵可以看出,在本文方法得到的邻接矩阵中各样本间的聚合程度比原始方法更为合理. 如在 $\boldsymbol{O}_{C=2}$ 中,样本点 x_1 与 x_3 的聚合程度比 x_1 与 x_5 和 x_6 的聚合程度要强. 这与样本点 x_1 与 x_3 同属一类而与 x_5 和 x_6 属于不同类相吻合,而在原始方法得到的邻接矩阵中却没有体现. 在 $\boldsymbol{O}_{C=2}$ 中虽然样本点 x_5 与 x_6 的聚合程度较弱,但在邻接矩阵 $\boldsymbol{O}_{C=3}$ 和 $\boldsymbol{O}_{C=4}$ 中却得到了相应增强. 因此,这两个点间的聚合程度仍然较高,从下面

的累积邻接矩阵可以看出.

根据上述结果,可以得到浮点数累积邻接矩阵 $\boldsymbol{J}$ 和最终的累积邻接矩阵 $\boldsymbol{J}'$,分别如下所示:

$$\boldsymbol{J}=\sum_{C=2}^{4}\boldsymbol{O}_C=\begin{bmatrix}0.000 & 0.000 & 2.838 & 0.000 & 0.758 & 0.600\\ 0.000 & 0.000 & 0.000 & 1.951 & 0.000 & 0.000\\ 2.838 & 0.000 & 0.000 & 0.000 & 0.787 & 0.624\\ 0.000 & 1.951 & 0.000 & 0.000 & 0.000 & 0.000\\ 0.758 & 0.000 & 0.787 & 0.000 & 0.000 & 2.416\\ 0.600 & 0.000 & 0.624 & 0.000 & 2.416 & 0.000\end{bmatrix}$$

$$\boldsymbol{J}'=\begin{bmatrix}0 & 0 & 3 & 0 & 0 & 0\\ 0 & 0 & 0 & 2 & 0 & 0\\ 3 & 0 & 0 & 0 & 0 & 0\\ 0 & 2 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 2\\ 0 & 0 & 0 & 0 & 2 & 0\end{bmatrix}.$$

从上述结果可知,相对于文献[9]的方法,本文方法进一步削弱了属于不同组的数据间的聚合程度.

分别对文献[9]和本文方法得到的累积邻接矩阵采用迭代图切分方法进行处理,其图切分过程分别如图 2 和图 3 所示.

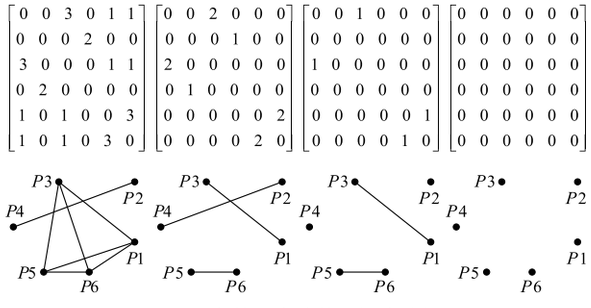


图2 文献[9]方法的图切分过程

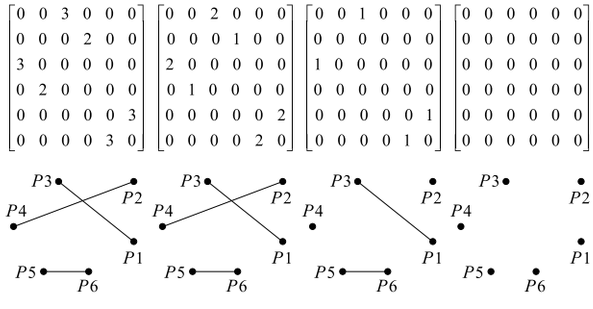


图3 本文方法的图切分过程

在完成上述迭代过程后,通过统计聚类个数发现,由文献[9]方法得到的结果中聚类个数2、3 和4 各出现一次,因此无法确定聚类个数. 而在本文方法得到的结果中,聚类个数 3 出现了 2 次,聚类个数 4 出现了 1 次,因此可以选择聚类数 3 作为最佳聚类个数,相应的连通图(图 2)作为最终的聚类结果. 这里不考虑聚类个数与样本数相同的情况,因为通常情况下聚类个数均小于样本数. 实验中,文献[9]和本文方法的聚类个数的统计结果,分别如图 4 和图 5 所示:

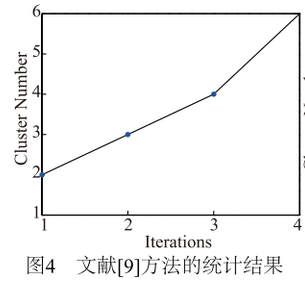


图4 文献[9]方法的统计结果

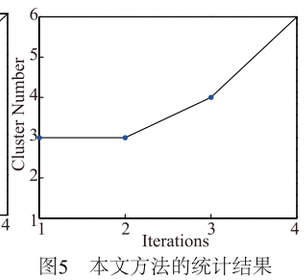


图5 本文方法的统计结果

4 实验结果与分析

4.1 实验环境与测试数据集

本仿真实验的测试环境如下:CPU 为 AMD Athlon 7750 Dual-Core 2. 70GHz,2G 内存,VS2008 编程环境,采用了 C ++编程语言. 本实验包括两个部分:人工数据集的对比实验和真实数据集的对比实验,测试对象共有 7 个人工数据集 dataset1 ~ dataset7 和 1 个真实数据集 Iris. 各数据集的样本数、聚类数以及数据维度,如表 1 所示.

表 1 测试数据集

数据集	样本数	聚类数	数据维度
dataset 1	250	2	2
dataset 2	250	2	2
dataset 3	200	2	2
dataset 4	230	3	2
dataset 5	221	4	2
dataset 6	245	4	2
dataset 7	246	4	2
Iris	150	3	4

其中 dataset3 来源于 Kuncheva 教授提供的数据集[18],其他 6 个人工数据集是由本文作者手工制作而成. 这 7 个数据集中的样本数据均是分布在二维空间中的数据点,如图 6 所示. 真实数据集 Iris 来源于 UCI 数据库[19],该数据集中的每个样本具有 4 个属性值,即每个样本点分布在 4 维空间中.

4.2 人工数据集聚类结果对比分析

如图 7 所示,为 dataset2 在图切分过程中的图划分过程,其中 k 为每次迭代过程中的聚类个数. 由图 7 可以看出,除 $k=1$ 外,在迭代过程中聚类个数 $k=2$ 的状态出现次数最多(共 3 次). 即该数据集在经过 3 次迭代后就达到了稳定状态,而经 9 次迭代后便达到收敛状态(即积累邻接矩阵成为零矩阵),可见本算法在执行过程中的收敛速度较快.

从图 6 可知,每个数据集的样本数据分布结构均比较复杂,传统的 FCM 算法已无法对其进行准确聚类[9]. 虽然这些数据集可以采用核聚类方法进行聚类,但是

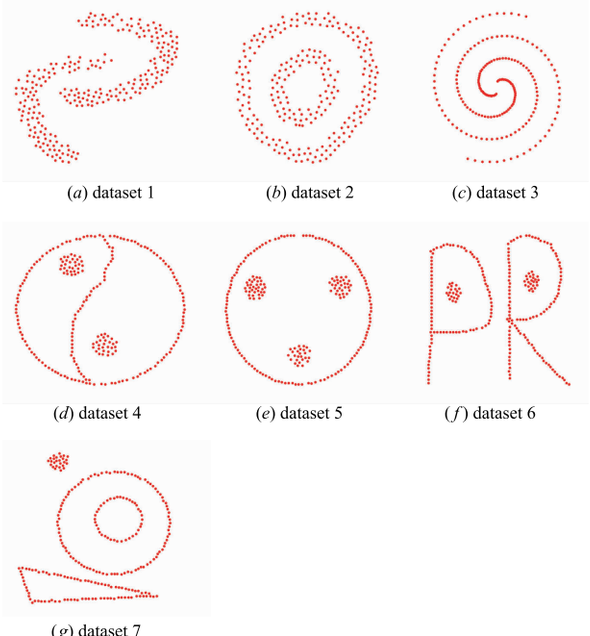


图6　测试数据集

对于核函数的选取、特定核函数中的参数的设置以及聚类个数的选取都十分困难,通常很难选择合适的核函数和相关的参数,并且聚类结果对参数的选取非常敏感. 另外,核聚类方法的计算过程相对更复杂,时空开销更大,实时性较差,因此这类方法在实际应用中并不十分有效[13]. 而通过集成最基本的 FCM 算法在不同聚类个数下的聚类结果,文献[9]方法和本文方法都能获得非常理想的聚类效果. 图 8 是采用文献[9]方法和本文方法得到的聚类结果,从聚类结果可知,即使对于分布比较复杂的数据集,这两种方法均能准确地对其进行聚类.

表 2 为文献[9]方法和本文方法的聚类结果对比,其中包括最大聚类个数 K 的选取、图切分过程中的迭代次数、聚类结果得到的聚类个数以及整个聚类过程的运行时间.

表 2 中所给的 K 值均为对相应数据集进行准确聚类的最小 K 值. 由于前 6 个数据集相对不是太复杂,因此相应的 K 值均小于 100,并且文献[9]方法和本文方法具有相同的最小 K 值. 当数据集分布结构相对比较复杂时,如 dataset7,为了完成对其准确聚类而需要选取较大的 K 值. 相对文献[9]方法,对于 dataset7,本文方法只需选取较小的 K 值即可完成数据聚类,从而在本文方法中扩大了 K 值的选择范围. 对于所有的数据集,通过实验发现,在图切分过程中,本文方法的迭代次数为文献[9]方法的 1/2 左右,如表 2 所示. 这是由于本文方法充分利用了每次 FCM 聚类结果的隶属度信息,因此,得到的累积邻接矩阵中的值相对原始方法累积邻接矩阵中的值要小很多,从而使得本文方法在图切分过程中的迭代次数得到了大幅度减少.

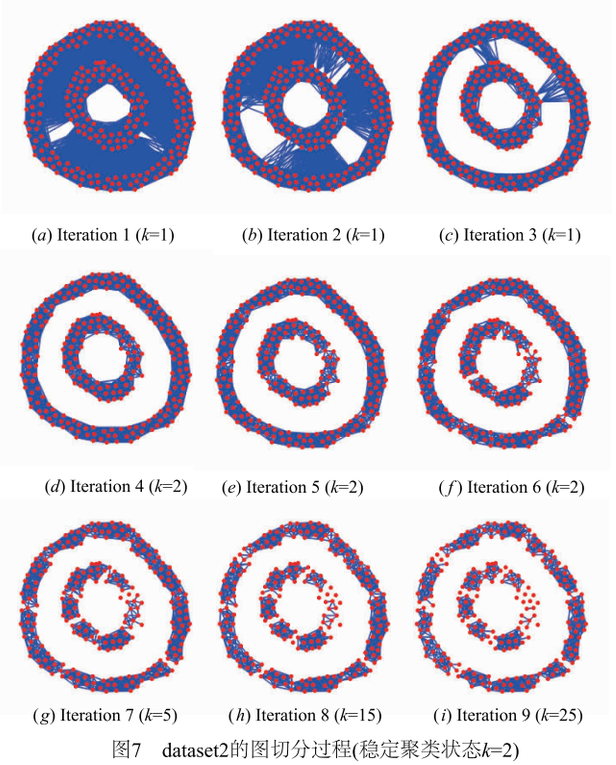


图7　dataset2的图切分过程(稳定聚类状态k=2)

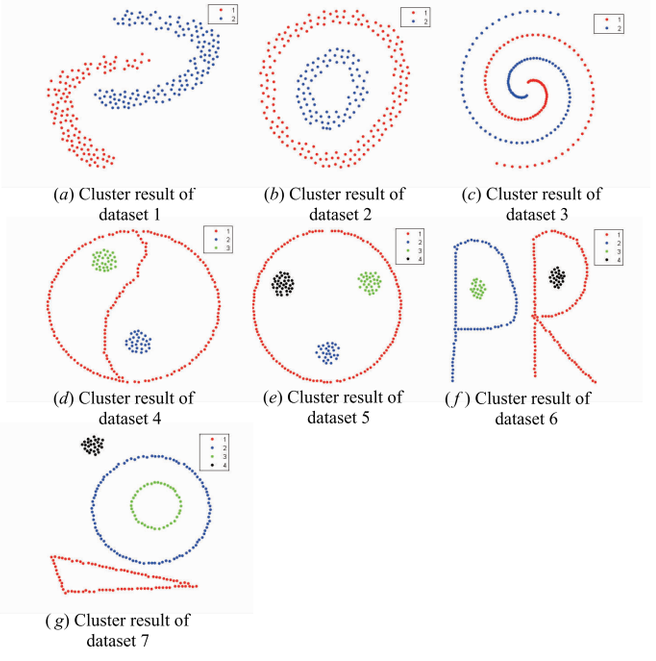
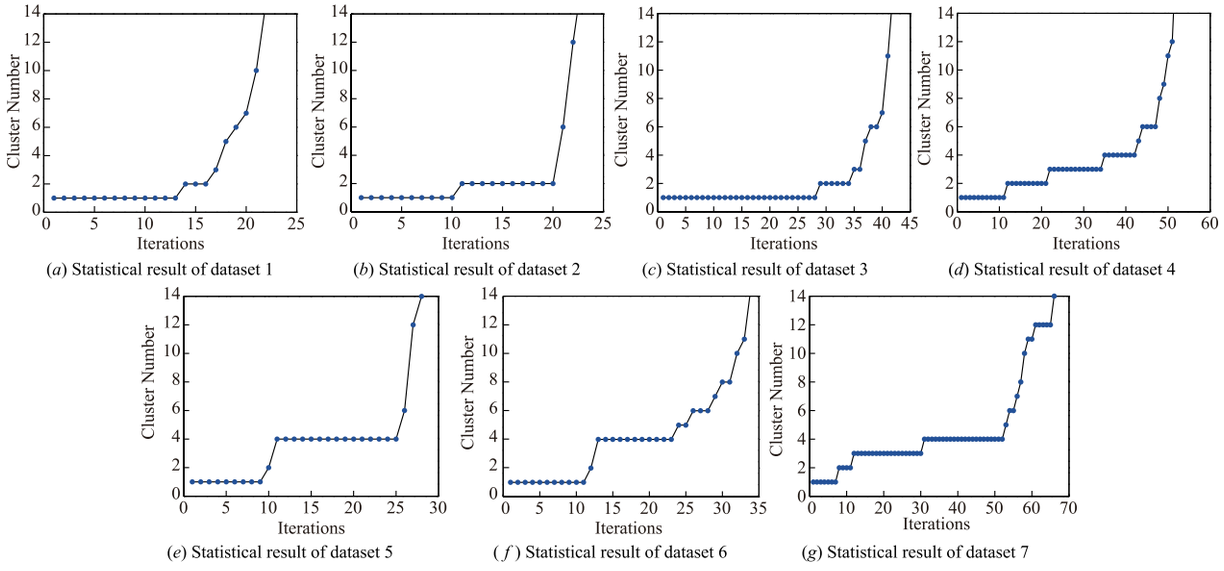


图8　文献[9]方法和本文方法得到的聚类结果

文献[9]方法和本文方法对聚类个数的统计结果见图 9 和图 10 所示.

表 2 文献[9]方法和本文方法聚类结果比较

	参数 K		图切分迭代次数		聚类个数		运行时间(s)	
	文献[9]方法	本文方法	文献[9]方法	本文方法	文献[9]方法	本文方法	文献[9]方法	本文方法
dataset 1	30	30	24	10	2	2	0.703	0.562
dataset 2	30	30	24	9	2	2	0.844	0.563
dataset 3	60	60	46	19	2	2	9.828	9.625
dataset 4	90	90	67	34	3	3	3.640	3.609
dataset 5	40	40	32	16	4	4	1.296	0.906
dataset 6	50	50	40	18	4	4	2.188	1.671
dataset 7	150	140	110	50	4	4	8.609	7.531



(*a*) Statistical result of dataset 1 (*b*) Statistical result of dataset 2 (*c*) Statistical result of dataset 3 (*d*) Statistical result of dataset 4

(*e*) Statistical result of dataset 5 (*f*) Statistical result of dataset 6 (*g*) Statistical result of dataset 7

图9 文献[9]方法的统计结果

虽然本文方法在图切分过程中迭代次数较文献[9]方法更少,但从图 10 可以看出,由本文方法得到的最佳聚类个数在迭代过程中的分布同样比较稳定. 在运行时间上,与文献[9]方法相比,本文方法的运行时间也相对要短. 但提升幅度并不十分明显,这是由于本文方法需要进行大量的浮点数乘积运算,而文献[9]方法中仅需简单的整数赋值操作.

对于 dataset7,相对文献[9]方法,本文方法只需取较小的 K 值,于是扩大了 K 值的选取范围. 如对 dataset7,本文方法在[140,246)范围内对 K 取值均可得到稳定的迭代状态,如图 11 所示,其聚类结果如图 8(g)所示.

4.3 真实数据集聚类结果对比分析

Iris 数据集共包含 150 个样本点,由 3 个聚类构成,每个聚类均由 50 个样本点组成. 为了更为有效地测试本文算法的聚类性能,在此实验中加入了聚类准确率这一评价标准. 聚类准确率定义为由测试算法得到的被正确聚类的样本数与总样本数的比值. 其中,通过比对测试聚类结果和真实数据集中理想聚类结果即可计算出被正确聚类的样本数.

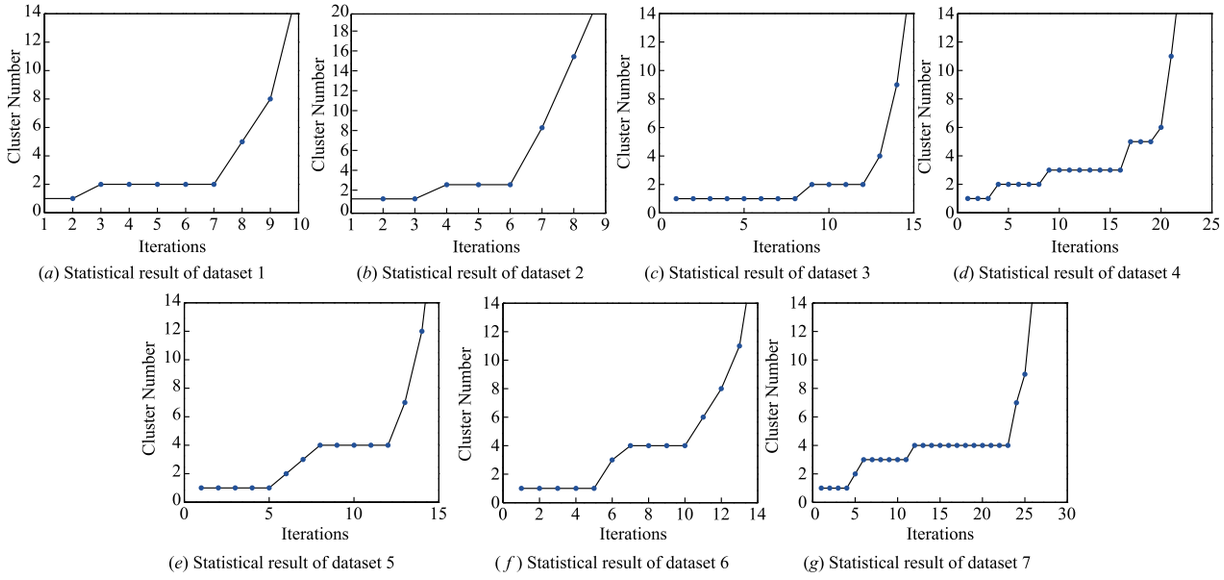
表 3 为对 Iris 数据集分别采用文献[9]方法和本文方法在不同参数 K 的取值下的聚类结果对比. 其评价标准包括图切分迭代次数、聚类个数、运行时间和聚类准确率共 4 个指标. 同人工数据集的测试结果,在相同参数 K 的取值下,相比文献[9]方法,本文方法所需图切分迭代次数更少,且运行时间也略短一些. 但在聚类性能上,本文算法比文献[9]方法更具优势. 图 12 所示为不同 K 值下对 Iris 数据集的统计结果对比.

表 3 文献[9]方法和本文方法聚类结果比较

测试方法	参数 K	图切分迭代次数	聚类个数	运行时间(s)	聚类准确率(%)
文献[9]方法	80	59	2	1.766	66.7
本文方法	80	18	3	1.556	100
文献[9]方法	140	115	3	4.156	95.3
本文方法	140	41	3	3.706	100

从图 12(a)可知,当 $K=80$ 时,由文献[9]方法得到聚类数为 2,并没有得到准确的聚类数. 通过实验发现,由文献[9]方法获得的聚类结果当中的 100 个样本点被划分到同一类,其中 50 个样本点被错误划分,即聚类准确率为 $100 \div 150 = 66.7\%$. 而采用本文方法获取

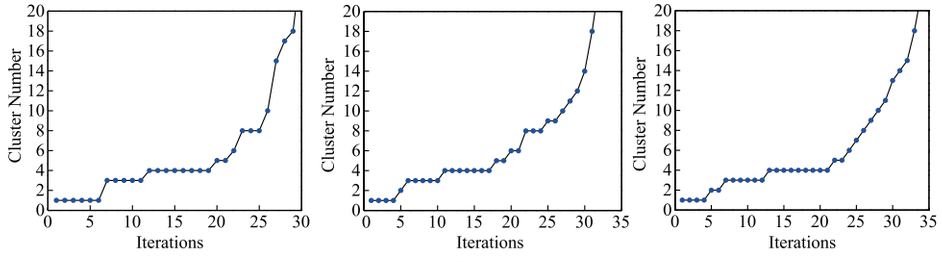
到的聚类结果其聚类个数和聚类准确率均达到理想结果. 当增大 K 值($K=140$)时,从图 12(c)可知,文献[9]方法获取到了准确的聚类个数,但在由该方法获取到的聚类结果中仍有将近 5% 的样本点被错误划分,如表 3 所示. 而采用本文方法同样能够获取到理想的聚类个数以及聚类准确率.



(a) Statistical result of dataset 1　(b) Statistical result of dataset 2　(c) Statistical result of dataset 3　(d) Statistical result of dataset 4

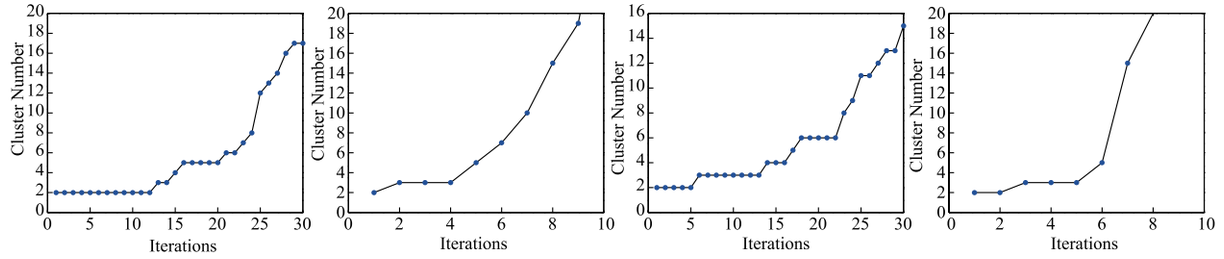
(e) Statistical result of dataset 5　(f) Statistical result of dataset 6　(g) Statistical result of dataset 7

图10　本文方法统计结果



(a) K=150　(b) K=170　(c) K=200

图11　不同K值下对dataset7的统计结果



(a) 文献[9]方法(K=80)　(b) 本文方法(K=80)　(c) 文献[9]方法(K=140)　(d) 本文方法 (K=140)

图12　不同K值下对Iris数据集的统计结果对比

5　总结

针对自动确定聚类个数的聚类算法采用的是硬划分方式,而没有充分利用每次 FCM 聚类结果的隶属度信息这一问题,本文采用软划分方式,充分利用每次 FCM 聚类结果的隶属度信息,提出了一种自动确定聚类个数的模糊聚类算法. 由于本文方法采用了软划分方式,在构造累积邻接矩阵时相对于原始方法更为准确,因此可以适当减少 FCM 的聚类次数,同时在图切分过程中迭代次数也得到了很大程度上的减少,尤其对于分布结构非常复杂的那些数据集,本文方法的优势更为明显.

当然,本文方法也存在着一些不足,由于引入了隶属度信息,从而导致了在整个聚类过程中需要进行大量的浮

点数乘积运算. 因此,相对于原始方法,本文方法在执行效率上并没有得到十分明显地提升. 在 FCM 聚类过程中,无论本文方法还是原始方法,虽然最大聚类个数 K 可以通过选取较大值来避免最佳 K 值的选取问题,但仍需要借助实际经验知识. 另外,由于传统 FCM 算法采用了随机数方法初始聚类中心,其初始状态将直接影响后续聚类结果的好坏. 因此,针对这些问题,一种自适应的自动确定聚类个数的模糊聚类方法有待进一步探讨与研究.

参考文献

[1] 田小林,焦李成,缑水平. 基于 PSO 优化空间约束聚类的 SAR 图像分割[J]. 电子学报,2008,36(3):453 – 457.
TIAN X L,JIAO L C,GOU S P. SAR image segmentation based on spatially constrained fcm optimized by particle swarm optimization[J]. Acta Electronica Sinica,2008,36(3):453 – 457. (in Chinese)

[2] 黄宇,付琨,吴一戎. 基于 Markov 随机场 K-Means 图像分割算法[J]. 电子学报,2009,37(12):2700 – 2704.
HUANG Y,FU K,WU Y R. Image segmentation method using k-means based on markov random field[J]. Acta Electronica Sinica,2009,37(12):2700 – 2704. (in Chinese)

[3] JAIN A K. Data clustering:50 years beyond K-means[J]. Pattern Recognition Letters,2010,31(8):651 – 666.

[4] WANG H S,FEI B W. A modified fuzzy C-means classification method using a multiscale diffusion filtering scheme[J]. Medical Image Analysis,2009,13(2):193 – 202.

[5] Cai W L,Chen S C,Zhang D Q. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation[J]. Pattern Recognition,2007,40(3):825 – 838.

[6] Krinidis S,Chatzis V. A robust fuzzy local information C-means clustering algorithm[J]. IEEE Transactions on Image Processing,2010,19(5):1328 – 1337.

[7] Yu Z D,Au O C,Zou R B,et al. An adaptive unsupervised approach toward pixel clustering and color image segmentation[J]. Pattern Recognition,2010,43(5):1889 – 1906.

[8] TAN K S,ISA N A M. Color image segmentation using histogram thresholding-fuzzy C-means hybrid approach[J]. Pattern Recognition,2011,44(1):1 – 15.

[9] Mok P Y,Huang H Q,Kwok Y L,et al. A robust adaptive clustering analysis method for automatic identification of clusters[J]. Pattern Recognition,2012,45(8):3017 – 3033.

[10] CHEN L,CHEN C L P,LU M Z. A multiple-kernel fuzzy c-means algorithm for image segmentation[J]. EEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,2011,41(5):1263 – 1274.

[11] Tsai D M,Lin C C. Fuzzy C-means based clustering for linearly and nonlinearly separable data[J]. Pattern Recognition,2011,44(8):1750 – 1760.

[12] Huang H C,Chuang Y Y,Chen C S. Multiple kernel fuzzy clustering[J]. IEEE Transactions on Fuzzy Systems,2012,20(1):120 – 134.

[13] Graves D,Pedrycz W. Kernel-based fuzzy clustering and fuzzy clustering:A comparative experimental study[J]. Fuzzy Sets and Systems,2010,161(4):522 – 543.

[14] Bezdek J C. Cluster validity with fuzzy sets[J]. Journal of Cybernetics,1973,3(3):58 – 73.

[15] Xie X L,Beni G. A validity measure for fuzzy clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1991,13(8):841 – 847.

[16] Wu K L,Yang M S,Hsieh J N. Robust cluster validity indexes[J]. Pattern Recognition,2009,42(11):2541 – 2550.

[17] Zalik K R. Cluster validity index for estimation of fuzzy clusters of different sizes and densities[J]. Pattern Recognition,2010,43(10):3374 – 3390.

[18] Kuncheva LI. Clustering Date[DB/OL]. http://pages.bangor.ac.uk/~mas00a/activities/artificial_data.htm,2014 – 9 – 10.

[19] Lichman M. UCI Machine Learning Repository[DB/OL]. http://archive.ics.uci.edu/ml/index.html,2014 – 9 – 10.

作者简介

陈海鹏 男,1978 年生于山东曹县. 吉林大学计算机科学与技术学院副教授. 研究方向为多媒体技术、图像处理和信息安全.
E-mail:chenhp@jlu.edu.cn

申铉京 男,1958 年生于吉林和龙. 吉林大学计算机科学与技术学院教授、博士生导师. 研究方向为多媒体技术、图像处理和智能检测系统.

龙建武 男,1984 年生于湖北恩施. 重庆理工大学计算机科学与工程学院讲师. 研究方向为图像处理和计算机视觉.

吕颖达(通讯作者) 女,1983 年生于河北文安. 吉林大学公共计算机教学与研究中心讲师. 研究方向为图像处理与模式识别.
E-mail:ydlv@jlu.edu.cn