基于模糊c-means与自适应粒子群优化的模糊聚类算法

耿宗科 王长宾 张振国
(河北师范大学数学与信息科学学院 石家庄 050024)

摘 要 已有的粒子群模糊聚类算法需要设置粒子群参数并且收敛速度较慢,对此提出一种基于改进粒子群与模糊c-means的模糊聚类算法。首先,使用模糊c-means算法生成一组起始解,提高粒子群演化的方向性;然后,使用改进的自适应粒子群优化方法对数据进行训练与优化,训练过程中自适应地调节粒子群参数;最终,采用模糊c-means算法进行模糊聚类过程。对比实验结果表明,所提方法大幅度提高了计算速度,并获得了较高的聚类性能。

关键词 粒子群优化,参数调节,模糊聚类算法,自适应调节,收敛速度
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.8.054

Fuzzy c-means and Adaptive PSO Based Fuzzy Clustering Algorithm

GENG Zong-ke WANG Chang-bin ZHANG Zhen-guo
(College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang 050024, China)

Abstract The existing PSO fuzzy clustering algorithms need to set the PSO parameters and converge very slowly, a fuzzy c-means and adaptive PSO based fuzzy clustering algorithm was proposed for that problem. Firstly, the fuzzy c-means algorithm is used to generate the initial solution, leading to a more directed search process. Then, the improved adaptive PSO is used to train and optimize the dataset, and the PSO parameters are adjusted adaptively in the training process to achieve a better optimal result. Lastly, the fuzzy c-means algorithm is used for fuzzy clustering. Compared experiments results show that the proposed method improves computational speed greatly and achieve good clustering performance.

Keywords Particle swarm optimization, Parameter adjustment, Fuzzy clustering algorithm, Aadaptively adjustment, Convergence speed

1 引言

聚类属于非监督模式识别问题,其特点是输入空间的样本没有期望输出,其目标是将样本按照某种相似性度量分为不同的类[1]。如果按照隶属度的取值范围可以将聚类分为两类[2]:硬聚类、模糊聚类。隶属度概念由模糊集理论引申而来。硬聚类算法的隶属度只有0与1两个值,即样本只可完全属于某一个类。模糊C-均值聚类(FCM)是经典的模糊聚类算法,其实现简单、计算速度较快,但其直接利用样本特征进行聚类,性能取决于样本的分布情况[3]。

已有较多的文献针对FCM的缺点提出了改进方案:文献[4]针对传统重复聚类算法精度不高、消耗资源较大的缺点,提出了一种模糊C-均值(FCM)与支持向量机(SVM)相结合的增强聚类算法。该算法先将实例数据集利用FCM粗分为C类,然后使用SVM再对每一类进行细化分类,提出了基于完全二叉树的决策级联式SVM模型,以便达到增强聚类的目的。针对使用FCM迭代聚类的过程中有可能会出现新的特征使原有的聚类失去平衡性的问题,使用划分的思想对数据集进行预处理来消除这种不利影响。文献[5]针对传统的模糊C-均值聚类算法求解隶属度公式仅仅考虑距离因素和算法对噪声数据敏感的问题,通过引入模糊熵约束,给出一种模糊C-均值聚类算法。该算法引入模糊熵作为模糊C-均值聚类算法的约束条件,重新给出了模糊C-均值聚类算法的隶属度和聚类中心求解公式。与原算法公式相比,新公式不仅考虑了距离因素,而且考虑了数据集分布特性,并对同一个数据对象隶属于所有聚类中心的隶属度进行相关性计算,使得整个隶属度求解公式具有高斯分布特性,从而可以抑制噪声数据对聚类中心的影响。文献[6]提出一种使用遗传演化算法的FCM来搜索数据属性的权重,以此建立不同类重要性的模型。文献[7]则采用约束向量机来求解低隶属度类簇的多目标问题。此类改进的FCM均具有两个缺点:1)其初始化质心随机生成;2)容易陷入局部最优。

PSO(粒子群优化)是一种实现简单、收敛速度较快的演化方法,已有一些研究将PSO引入聚类问题,以期大幅度提高模糊聚类的性能。文献[8,9]分别提出了基于PSO的硬聚类与模糊聚类算法,然而,此类基于PSO的聚类方法有两大

到稿日期:2015-09-12 返修日期:2015-11-30 本文受国家自然科学基金项目(71271067)资助。
耿宗科(1973-),男,硕士,讲师,主要研究方向为嵌入式系统、智能信息处理,E-mail:gengzk@139.com;王长宾(1984-),男,硕士,讲师,主要研究方向为数据挖掘、智能信息处理,E-mail:wang_chang_bin@163.com(通信作者);张振国(1974-),男,博士,主要研究方向为数据挖掘。

缺点:1)其计算复杂度高于传统聚类方法;2)参数设置:PSO算法需要调节3个参数,而此参数对算法的性能影响较大。

文献[10]提出了一种改进的自适应的PSO算法IDPSO,IDPSO具有两个明显的优点:1)搜索效果较好,可有效地防止早熟收敛;2)样本训练过程中动态地调节参数,因此该方法解决了上述PSO模糊聚类的两大问题。本文结合IDPSO与FCM两个算法,根据FCM与IDPSO的结合方式,提出了两个聚类方法:FCMIDPSO与EFCMIDPSO。EFCMIDPSO采用FCM产生一个起始解,以此降低FCMIDPSO起始解的随机性。实验结果表明,本方法有效地解决了已有粒子群模糊聚类的上述两大问题,无需设置粒子群的3个参数,并且收敛速度大幅度提高。

2 背景知识与相关研究

本文聚类算法包括两个部分:粒子群优化与模糊聚类算法。本文采用基本的模糊聚类算法[11]。

2.1 粒子群优化

假设粒子群的粒子数量为N_P,每个粒子均表示一个多维优化问题的完整解,且各粒子的维度D相等。将粒子P_i($1\leqslant i\leqslant N_P$)的位置表示为$X_{id}$($1\leqslant d\leqslant D$),速度设为$V_{id}$,则种群中的第$i$个粒子可表示如下:

$$P_i=[X_{i,1},X_{i,2},\cdots,X_{i,D}] \tag{1}$$

每个粒子计算其适应度值,以此判断其对应的解质量。为了获得全局最优解,粒子P_i需综合考虑其局部最优解($Pbest_i$)与全局最优解($Gbest$)来更新其位置与速度,计算方法分别如下:

$$v_i(t+1)=\omega v_i(t)+c_1r_1\times(pbest_i(t)-x_l(t))+c_2r_2\times(gbest(t)-x_i(t)) \tag{2}$$

$$x_i(t+1)=x_i(t)+v_i(t) \tag{3}$$

式中,ω表示惯性权重,c_1,c_2为两个加速因子(非负常量),r_1,r_2为均匀分布于[0,1]的随机值。

2.1.1 改进的粒子群优化算法

文献[10]提出的固定或线性递减的惯性权重值容易使得优化程序陷入局部最优,原因是PSO的搜索程序极为复杂并且为非线性过程。因此,ω线性递减无法平衡局部与全局搜索之间的关系。该研究提出了一个检测函数:$\varphi_l(t)=|(gbest-x_l(t-1))/(pbest_l-x_l(t-1))|$,其中$|gbest-x_l(t-1)|$表示第$t-1$次迭代粒子$l$于位置$x_l$与该轮迭代最优全局位置的欧氏距离,$|pbest-x_l(t-1)|$表示第$t-1$次迭代粒子$l$于位置$x_l$与该轮迭代最优局部位置的欧氏距离。

可使用函数$\varphi_l(t)$计算每个粒子l的权重w_l,c_{1l}与c_{2l},同时考虑全局与局部搜索,如式(4)—式(6)所示:

$$\omega_l(t)=\frac{\omega_{initial}-\omega_{final}}{1+e^{\varphi_l(t)(t-((1+\ln(\varphi_l(t)))K_{\max})/\mu)}}+\omega_{final} \tag{4}$$

$$c_{1l}(t)=c_{1l}(t-1)\omega_l(t)^{-1} \tag{5}$$

$$c_{2l}(t)=c_{2l}(t-1)\omega_l(t) \tag{6}$$

其中,$K_{\max}$是最大的迭代次数,$\omega_{initial}$与ω_{final}分别是惯性值的起始值与终值,μ是确保ω_l保持反向变化的调节因子。粒子l与权重ω_l则基于变参数的sigmoid函数更新,因此,权重值将随时间不规则地降低,式(7)所示为使用自适应权重的粒子l的速度函数:

$$v_l(t+1)=\omega_l(t)v_l(t)+(c_{1l}(t)r_1)*(pbest_l(t)-x_l(t))+(c_{2l}(t)r_2)*(gbest(t)-x_l(t)) \tag{7}$$

该自适应PSO算法称为改进的自适应粒子群优化算法IDPSO,该研究成果显示其适应度结果优于其他的主流粒子群优化方法。算法1示出IDPSO算法的主要步骤。

算法1

输入:种群

输出:gbest

1. 初始化含有P个粒子的种群;
2. 初始化IDPSO的参数,包括:种群P大小,$\omega_{initial}$,ω_{final},c_{1l}与c_{2l};
3. 初始化每个粒子的参数x_l,v_l,$pbest_l$,$\omega_l=\omega_{initial}$,gbest;
4. 计算每个粒子的适应度值;
5. 计算每个粒子的pbest;
6. 计算种群的gbest;
7. 更新每个粒子的速度(式(7));
8. 更新每个粒子的位置(式(2));
9. 对每个粒子,更新其ω_l,c_{1l}与c_{2l}(式(4)—式(6));
10. 如果达到结束条件,则返回步骤4.

2.2 模糊c-means

本文采用文献[11]的模糊c-means聚类方法(FCM),假设$\Omega=\{1,\cdots,k,\cdots,n\}$表示$n$个对象的集合,每个目标$k$为一个定量变量的向量$\boldsymbol{x}_k=(x_{1k},\cdots,x_{jk},\cdots,x_{pk})$,其中包含$p$个变量,$j$是序号,其中$x_{jk}\in\mathcal{R}$。设$\boldsymbol{Y}=\{1,\cdots,i,\cdots,c\}$表示包含$c$个原型的集合,其中$i$表示一个定量变量的向量$\boldsymbol{y}_i=(y_{1i},\cdots,y_{ji},\cdots,y_{pi})$,其中$y_{ji}\in\mathcal{R}$。设$\boldsymbol{U}=[u_{ik}]$表示$c\times n$的隶属度矩阵,其中$u_{ik}$是目标$k$对类$i$的隶属度,$u_{ik}$的范围是[0,1],其值越大,隶属于$i$的目标越多。

FCM算法的目标是搜索一个原型矩阵$\boldsymbol{Y}^*$与一个隶属度矩阵$\boldsymbol{U}^*$,其目标是最小化目标函数,如下所示:

$$J(\boldsymbol{Y},\boldsymbol{U})=\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m d_{ik} \tag{8}$$

其中,m是簇类的模糊度,d_{ik}表示目标k的特征向量$\boldsymbol{x}_k$与原型i的特征向量$\boldsymbol{y}_i$的欧氏距离平方,其计算方法如下:

$$d_{ik}=\sum_{j=1}^{p}(x_{jk}-y_{ji})^2 \tag{9}$$

最小化准则J的原型计算如下:

$$y_{ji}=\frac{\sum_{k=1}^{n}(u_{ik})^m x_{jk}}{\sum_{k=1}^{n}(u_{ik})^m} \tag{10}$$

最小化准则J的隶属度使用下式更新:

$$u_{ik}=[\sum_{a=1}^{c}(\frac{d_{ik}}{d_{ak}})^{\frac{1}{m-1}}]^{-1} \tag{11}$$

约束条件为:$\sum_{i=1}^{c}u_{ik}=1$。式中变量m表示模糊度。详细的FCM算法可参考文献[11]。

3 本文改进的粒子群模糊聚类算法

本文采用FPSO算法[12]的粒子编码模型,将粒子l的位置定义为x_l,粒子表示了目标与分组之间的模糊关系,每个粒子代表了隶属度矩阵$\boldsymbol{U}$的一个可行解,因此x_l可表示为:

$$x_l=\begin{bmatrix}u_{11}&\cdots&u_{c1}\\ \vdots&\cdots&\vdots\\ u_{1n}&\cdots&u_{cn}\end{bmatrix}$$

其中,u_{ik}表示目标k属于分组i的隶属度,其约束条件为

$\sum_{i=1}^{c} u_{ik}=1(k=1,\cdots,n)$。该文献基于隶属度函数计算样本的原型，因此，通过代表隶属矩阵 $\boldsymbol{U}$，X_l 具有足够的信息来生成原型矩阵 $\boldsymbol{Y}$[11]。粒子的速度 v_l 同样是一个 $c\times n$ 的矩阵。以下两式分别更新粒子的速度与位置：

$$v_l(t+1)=\omega\times v_l(t)+(c_1r_1)\times(pbest_l(t)-x_l(t))+(c_2r_2)\times(gbest(t)-x_l(t)) \quad (12)$$

$$x_l(t+1)=x_l(t)\oplus v_l(t) \quad (13)$$

其中，$pbest_l$ 是一个 $c\times n$ 的矩阵，代表了粒子 l 获得的最优解，$gbest$ 同样为一个 $c\times n$ 的矩阵，表示了全局最优解。

IDPSO 针对函数优化问题获得了优于其他同类算法的良好结果，本文将 FCM 与 IDPSO 相结合来求解模糊聚类问题，为式(12)加入自适应权重，可得：

$$v_l(t+1)=\omega_l(t)\times v_l(t)+(c_{1l}(t)r_1)\times(pbest_l(t)-x_l(t))+(c_{2l}(t)r_2)\times(gbest(t)-x_l(t)) \quad (14)$$

算法 2 示出 FCMIDPSO 的算法步骤，FCMIDPSO 采用了 IDPSO 训练程序与其自适应权重程序，代替了传统的 PSO 算法。

算法 2 FCMIDPSO 算法

输入：数据集 Ω、簇数量 c

1. 建立含 P 个粒子的种群；
2. 初始化 FCM 的参数，包括：种群大小 P，m，$\omega_{initial}$，ω_{final}，c_{1l} 与 c_{2l}；
3. 初始化每个粒子的参数 x_l，v_l，$pbest_l$，$\omega_l=\omega_{initial}$，gbest；

IDPSO 算法：

4. 计算簇的原型；//式(10)
5. 计算每个粒子的 J 值；//式(8)
6. 设置每个粒子的 pbest；
7. 设置种群的 gbest；
8. 更新每个粒子的速度；//式(12)
9. 更新每个粒子的位置；//式(13)
10. 对每个粒子，更新其 ω_l、c_{1l} 与 c_{2l}；//(式(4)－式(6)
11. 如果 IDPSO 为达到结束条件，则返回步骤 4；

FCM 算法：

12. 计算簇的原型；//式(10)
13. 计算成员的度；//式(11)
14. 计算每个粒子的速度，更新每个粒子的位置；
15. 如果 FCM 未达到结束条件，返回步骤 12；
16. 如果 FCMIDPSO 为达到结束条件，返回步骤 4.

本文提出的第二个算法是 EFCMIDPSO，使用 FCM 生成一组初始化解，作为粒子种群的一个粒子，其目标是降低 FCMIDPSO 起始解的随机性，以此提高搜索过程的指向性，以期获得一个较为快速、稳定的搜索过程。算法 3 为 EFCMIDPSO 的具体步骤。

算法 3 EFCMIDPSO 算法

输入：数据集 Ω、簇数量 c

1. 建立含 P 个粒子的种群；
2. 初始化 FCM 的参数，包括：种群大小 P，m，$\omega_{initial}$，ω_{final}，c_{1l} 与 c_{2l}；
3. 运行 FCM，使用其解初始化粒子 X_1，初始化 $pbest_1$ 与 v_1；
4. 初始化每个粒子的参数 x_l，v_l，$pbest_l$，$\omega_l=\omega_{initial}$，gbest；

IDPSO 算法：

5. 计算簇的原型；//式(10)
6. 计算每个粒子的 J 值；//式(8)
7. 设置每个粒子的 pbest；
8. 设置种群的 gbest；
9. 更新每个粒子的速度；//式(12)
10. 更新每个粒子的位置；//式(13)
11. 对每个粒子，使用式(4)－式(6)更新其 ω_l，c_{1l} 与 c_{2l}；
12. 如果 IDPSO 未达到结束条件，则返回步骤 5；

FCM 算法：

13. 计算簇的原型；//式(10)
14. 计算隶属度；//式(11)
15. 计算每个粒子的速度，更新每个粒子的位置；
16. 如果 FCM 未达到结束条件，返回步骤 13；
17. 如果 FCMIDPSO 未达到结束条件，返回步骤 5.

FCM 初始化一组起始解的结束条件为：50 次迭代或者两次连续迭代之间准则 J 的提高量小于或等于 0.00001。

4 实验结果与分析

本文选择文献[12]中的 FCM-PSO 作为对比基本性能的对象，文献[12]表明该方法优于 FCM 与 FPSO 算法(快速 PSO)，本文采用两种数据集：人工合成数据集与真实数据集。第一种数据集有利于较好地控制数据的分布并直观地评估各方法的性能；第二种数据集则采用 UCI 真实数据集。

为了量化评估各方法的性能，使用式(8)计算准则 J，并且采用较为常用的一个聚类性能指标：ARI(Adjusted Rand Index)[13]。ARI 与其他的聚类度量方法有所差异，该方法对类簇的数量不敏感，因此其值的范围为[－1，1]，1 表示两个类簇具有最好的一致性，0 值表示一个随机解，负值表示该方法无法区分各类簇。

设 $Q=\{q_1,\cdots,q_i,\cdots,q_C\}$ 表示 C 个类簇的分簇，$R=\{r_1,\cdots,r_j,\cdots,r_D\}$ 表示含有 D 个簇的先验分簇，则 ARI 定义如下：

$$ARI=\frac{\sum_{i=1}^{C}\sum_{j=1}^{D}\binom{n_{ij}}{2}-\binom{n}{2}^{-1}\sum_{i=1}^{C}\binom{n_i}{2}\sum_{j=1}^{D}\binom{n_j}{2}}{\frac{1}{2}\left[\sum_{i=1}^{C}\binom{n_i}{2}+\sum_{j=1}^{D}\binom{n_j}{2}\right]-\binom{n}{2}^{-1}\sum_{i=1}^{C}\binom{n_i}{2}\sum_{j=1}^{D}\binom{n_j}{2}} \quad (15)$$

其中，n_{ij} 表示类簇 q_i，r_j 中的总目标数量，n_i 表示类簇 q_i 中的目标数量；n_j 表示类簇 r_j 中的目标数量；n 是总目标数量。

人工合成数据集的每组数据采用二元正态分布随机生成，因此所有数据点均属于 $\mathscr{R}$。使用 Monte Carlo 仿真实验法来评估本文方法，使用相同配置随机产生相同的 30 个数据集，每个方法在某个数据集独立地运行 30 次，随机初始化种群，迭代次数上限设为 500。对于每次独立运行，保存准则 J 的结果值，30 次重复实验之后，选择最优准则值对应的分簇方案。对于每个分簇方案，计算其 ARI 结果并统计其迭代次数，最终统计了 30 个准则 J 值以及对应的 30 个 ARI 值、迭代次数。

为了统计比较方法的性能，对准则 J、ARI 与迭代次数进行 5%显著性水平的 Wilcoxon 秩和检验[14]，零假设表示值间偏差的均值为 0，使用秩和检验的优点是无需假设数据的分布情况。

UCI 真实数据集是著名的机器学习数据集，本文选择 8 组数据：abalone，glass，heart disease，image segmentation，iris，magic gamma telescope，pima，wine。

实验中，每个方法均随机初始化，独立地运行 30 次，每次运行均保存准则 J 的最终值，最终统计了 30 次运行的均值与标准偏差。参数值设置如下：

(1)FCM-PSO:10 个粒子,$c_1=c_2=2$,ω 设为从 0.9 至0.1 线性递减(式(3));

(2)FCMIDPSO:10 个粒子,初始化条件为 $t=1$,$\mu=100$,$\omega_{initial}=0.9$,$\omega_{final}=0.4$,$c_i^1(t)=c_i^2(t)=2$;

(3)EFCMIDPSO:除了采用与 FCMIDPSO 相同的参数,还采用 FCM 的 50 次迭代初始化其起始解集。

采用两个合成数据集与 8 个真实数据集进行实验分析。实验环境为 Intel Core i7-2630QM,主频 2.00GHz,内存为 6GB,Ubuntu linux 操作系统。

4.1 人工合成数据集实验

两个合成数据集大小相等,但其重叠等级与分类形状均不同。使用 3 个高斯分布生成 3 组数据,表 1、表 2 所列分别为生成图 1、图 2 数据集所对应的参数。

表 1 3 组高斯分布的参数

参数	μ_1	μ_1	σ_1^2	σ_2^2	n
第一组	0	0	100	1	100
第二组	3	0	1	100	100
第三组	7	0	1	100	100

表 2 3 组高斯分布的参数

参数	μ_1	μ_1	σ_1^2	σ_2^2	n
第一组	0	0	25	25	100
第二组	15	5	25	25	100
第三组	5	15	25	25	100

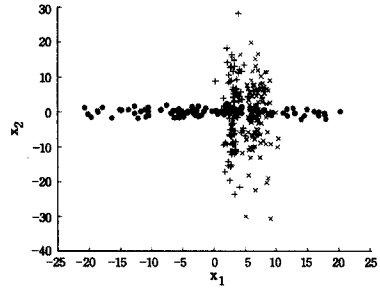

图 1 高斯分布生成的数据集 1

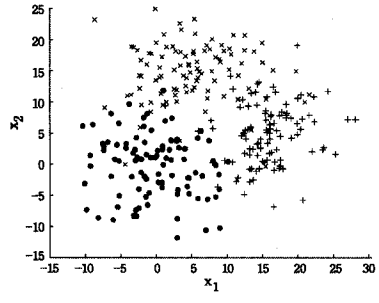

图 2 高斯分布生成的数据集 2

4.1.1 人工数据集实验结果

表 3—表 5 所列为准则 J、ARI 与迭代次数实验结果的均值与标准偏差。对于第一个数据集,本文基于 IDPSO 的方法的准则 J 均值高于 FCM-PSO 方法,并且 EFCMIDPSO 优于 FCMIDPSO。对于第二个数据集,3 个方法的均值相等,但 EFCMIDPSO 的标准偏差较小。而 EFCMIDPSO 的迭代次数最低,接近其他两个算法的一半。FCM-PSO 获得了第一个数据集的最优 ARI 结果,但对于第二个数据集,3 个算法的均值较为接近。

表 3 ARI 的均值与标准偏差

数据集		FCM-PSO	FCMIDPSO	EFCMIDPSO
数据集 1	均值	0.1566	0.1533	0.1480
	标准偏差	(0.0314)	(0.0324)	(0.0348)
数据集 2	均值	0.6929	0.6929	0.6932
	标准偏差	(0.0434)	(0.0434)	(0.0435)

表 4 准则 J 的均值与标准偏差

数据集		FCM-PSO	FCMIDPSO	EFCMIDPSO
数据集 1	均值	8.5053e+03	8.4989e+03	8.4923e+03
	标准偏差	(670.9073)	(670.5730)	(671.2449)
数据集 2	均值	8.4270e+03	8.4270e+03	8.4270e+03
	标准偏差	(451.8389)	(451.8389)	(451.8371)

表 5 迭代次数的均值与标准偏差

数据集		FCM-PSO	FCMIDPSO	EFCMIDPSO
数据集 1	均值	16.6000	16.0333	8.1000
	标准偏差	(1.5888)	(2.0083)	(0.5477)
数据集 2	均值	15.5000	15.5000	8
	标准偏差	(0.8610)	(0.8610)	(0)

对准则 J、ARI 与迭代次数分别进行 5%显著性水平的 Wilcoxon 秩和检验,表 6—表 8 为 p 值的统计结果,p 值小于 0.05 表示拒绝零假设,表中粗体表示低于 0.05。从数理统计角度亦可明显看出本文方法的优势。

表 6 ARI 的 p 值

数据集	最优算法	FCM-PSO	FCMIDPSO	EFCMIDPSO
数据集 1	FCM-PSO	*	0.0675	**4.4583e−04**
数据集 2	EFCMIDPSO	1	1	*

表 7 准则 J 的 p 值

数据集	最优算法	FCM-PSO	FCMIDPSO
数据集 1	EFCMIDPSO	**1.7344e−06**	**1.7344e−06**
数据集 2	EFCMIDPSO	**1.7344e−06**	**1.7344e−06**

表 8 迭代次数的 p 值

数据集	最优算法	FCM-PSO	FCMIDPSO
数据集 1	EFCMIDPSO	**1.3308e−06**	**1.0251e−06**
数据集 2	EFCMIDPSO	**7.4255e−07**	**7.4255e−07**

4.2 与其他聚类算法的比较

表 9 所列为本实验 UCI 数据集的简要介绍。

表 9 UCI 数据集简介

数据集	数据数量	类数	属性
abalone	4177	3	8
glass	214	6	9
heart disease	270	2	13
image segmentation	2310	7	19
iris	150	3	4
magic gamma telescope	19020	2	10
pima Indians diabetes	768	2	8
wine	178	3	13

上文实验证实本文聚类算法具有较好的性能,尤其是 EFCMIDPSO 的性能最优。为了进一步横向地评估本方法的有效性,将其与 HPSOFCM[9],CPSFC[15] 进行比较,HPSOFCM,CPSFC 均为近期性能较好的模糊粒子群聚类方法。

比较 3 种算法的 3 项指标:ARI、准则 J、运行时间。3 个算法的迭代步骤差异较大,迭代次数并非一个有效的指标,因此对比实验采用运行时间作为指标。

对比实验分为人工合成数据与真实数据。EFCMIDPSO

与 CPSFC 使用相同的结束条件：500 次迭代或直至达到稳定状态，HPSOFCM 则采用 100 次迭代，与文献[9]中作者的迭代次数相等。3 种方法的参数设置如下：

(1)EFCMIDPSO：与上文实验参数值相同；

(2)HPSOFCM：50 个粒子，$\omega_{max}=0.9$，$c_1=c_2=c_3=2$；

(3)CPSFC：50 个粒子，$\omega_{max}=0.9$，$\omega_{min}=0.4$，$c_1=c_2=2$，$I_{CLS}=1000$，$I_{GM}=3$。

表 10－表 12 所列为两个人工合成数据集准则 J、ARI 与运行时间的均值与标准偏差。

表 10　ARI 的 p 值

数据集		EFCMIDPSO	HPSOFCM	CPSFC
数据集 1	均值	**0.1480**	0.1473	0.0619
	标准偏差	**(0.0348)**	(0.0386)	(0.0496)
数据集 2	均值	**0.6932**	0.6855	0.2168
	标准偏差	**(0.0435)**	(0.0418)	(0.1467)

表 11　准则 J 的 p 值

数据集		EFCMIDPSO	HPSOFCM	CPSFC
数据集 1	均值	8.4923e+03	8.5498e+03	**1.5478e+03**
	标准偏差	(671.2449)	(675.9226)	**(770.1587)**
数据集 2	均值	8.4270e+03	8.4939e+03	**1.9391e+03**
	标准偏差	(451.8371)	(444.6099)	**(1.1820e+03)**

表 12　运行时间的均值与标准偏差(s)

数据集		EFCMIDPSO	HPSOFCM	CPSFC
数据集 1	均值	**0.6586**	17.2039	2.2031
	标准偏差	**(0.0017)**	(0.7338)	(0.1945)
数据集 2	均值	**0.6591**	17.0008	2.2384
	标准偏差	**(0.0021)**	(0.2585)	(0.2931)

CPSFC 获得了最好的准则 J，但同时其 ARI 值最低，两者含有一定的关联性：由过度拟合引起。EFCMIDPSO 获得了最优的 ARI 结果，同时计算速度最快。

为了进一步证明本方法的性能，进行 5%显著性水平的 Wilcoxon 秩和检验。表 13－表 15 所列为 p 值的统计结果，可看出本方法的性能明显优于其他两种算法。

表 13　ARI 的 p 值

数据集	最优算法	HPSOFCM	CPSFC
数据集 1	EFCMIDPSO	0.9590	**1.4936e−05**
数据集 2	EFCMIDPSO	0.0350	**1.7344e−06**

表 14　准则 J 的 p 值

数据集	最优算法	HPSOFCM	CPSFC
数据集 1	CPSFC	1.7344e−06	**1.7344e−06**
数据集 2	CPSFC	1.7344e−06	**1.7344e−06**

表 15　运行时间的 p 值

数据集	最优算法	HPSOFCM	CPSFC
数据集 1	EFCMIDPSO	**1.7344e−06**	**1.7344e−06**
数据集 2	EFCMIDPSO	**1.7344e−06**	**1.7344e−06**

表 16 所列为真实数据集下 3 个方法的准则 J 的最优值及其对应的 ARI 值与运行时间。CPSFC 方法可获得最优的 J 值与最低的 ARI 值，而 HPSOFCM 的 J 值最低，但其 ARI 值最优，EFCMIDPSO 收敛速度最快。

表 17－表 19 列出 3 个指标的均值与标准方程，HPSOFCM 并未采用 FCM 迭代来提高收敛速度，并且获得了一部分的最优 ARI 值，如表 16 所列。但该方法并不具备鲁棒性，因此，对于 7 个数据集，EFCMIDPSO 算法获得了最优的 ARI 均值。CPSFC 方法获得了 6 个数据集的最优 J 值，但由于过度拟合导致其 ARI 值较低。最终，EFCMIDPSO 的计算速度最快。

表 16　准则 J、ARI 值与运行时间的最优结果

UCI 数据集	指标	EFCMIDPSO	HPSOFCM	CPSFC
abalone	J	119.8883	251.5220	**23.8605**
	ARI	0.1361	**0.1594**	0.0689
	时间	**5.8527**	237.2834	70.1222
glass	J	7.2897	15.0260	**0.8951**
	ARI	0.1614	**0.2484**	0.0859
	时间	**0.6867**	27.0872	7.6759
heart	J	199.6642	215.4946	**29.4008**
	ARI	0.3400	**0.4131**	−1.6242e−04
	时间	**0.3642**	11.8717	4.9730
image seg	J	171.1557	458.2313	**74.0409**
	ARI	**0.5063**	0.1640	0.1063
	时间	**10.9703**	357.7596	179.3200
iris	J	5.2330	6.1538	**3.0641**
	ARI	**0.7287**	0.7141	0.6151
	时间	**0.1923**	8.8209	1.6115
magic	J	1.9960e+03	3.0218e+03	**96.4574**
	ARI	**0.0577**	0.0033	5.2898e−04
	时间	**20.3069**	726.0691	204.9379
pima	J	78.3057	88.8669	**6.3192**
	ARI	0.1069	**0.1294**	0.0041
	时间	**0.8072**	29.6668	9.4172
wine	J	28.7160	38.7284	**9.4301**
	ARI	**0.8498**	0.1671	0.0275
	时间	**0.2961**	11.9123	4.8390

表 17　ARI 的均值与标准偏差

UCI 数据集		EFCMIDPSO	HPSOFCM	CPSFC
abalone	均值	**0.1361**	0.1312	0.0963
	标准偏差	**(2.08e−04)**	(0.0472)	(0.0660)
glass	均值	0.1630	**0.1838**	0.1400
	标准偏差	(0.0012)	**(0.0400)**	(0.0511)
heart	均值	**0.3394**	0.1451	−4.57e−04
	标准偏差	**(0.0031)**	(0.1245)	(0.0035)
image seg	均值	**0.5055**	0.1726	0.2929
	标准偏差	**(0.0040)**	(0.0689)	(0.1661)
iris	均值	**0.7278**	0.6499	0.7039
	标准偏差	**(0.0035)**	(0.0860)	(0.0834)
magic	均值	**0.0577**	0.0428	0.0107
	标准偏差	**(3.25e−05)**	(0.0590)	(0.0327)
pima	均值	**0.1069**	0.0353	0.0099
	标准偏差	**(4.23e−17)**	(0.0502)	(0.0289)
wine	均值	**0.8498**	0.2970	0.1513
	标准偏差	**(4.52e−16)**	(0.1630)	(0.2495)

表 18　准则 J 的均值与标准偏差

UCI 数据集		EFCMIDPSO	HPSOFCM	CPSFC
abalone	均值	**119.8884**	367.7334	121.0329
	标准偏差	**(2.91e−06)**	(72.3766)	(24.0752)
glass	均值	7.2897	18.3994	**6.7808**
	标准偏差	(8.19e−05)	(1.9422)	**(1.5929)**
heart	均值	199.6806	231.2296	**118.8000**
	标准偏差	(0.0891)	(8.6631)	**(43.6791)**
image seg	均值	**171.2333**	522.4075	178.7635
	标准偏差	**(0.4227)**	(34.0335)	(36.8814)
iris	均值	5.2330	7.6484	**5.0038**
	标准偏差	(1.04e−06)	(1.0229)	**(0.5028)**
magic	均值	1.9960e+03	3.6193e+03	**1.3992e+03**
	标准偏差	(1.4142e−06)	(421.5930)	**(626.2046)**
pima	均值	78.3057	116.1707	**45.9576**
	标准偏差	(7.04e−07)	(21.7154)	**(20.5114)**
wine	均值	28.7160	45.6660	**19.3853**
	标准偏差	(2.75e−07)	(3.9562)	**(7.5737)**

表 19 运行时间的均值与标准偏差(s)

UCI 数据集		EFCMIDPSO	HPSOFCM	CPSFC
abalone	均值	6.4825	237.1811	60.5518
	标准偏差	(0.2209)	(0.2695)	(10.3755)
glass	均值	0.7516	27.0913	6.5454
	标准偏差	(0.0544)	(0.0231)	(1.2303)
heart	均值	0.3979	11.9177	4.8192
	标准偏差	(0.0541)	(0.0460)	(0.8819)
image seg	均值	11.0990	357.6167	146.2648
	标准偏差	(0.7583)	(0.3716)	(20.9687)
iris	均值	0.1961	8.8148	1.3451
	标准偏差	(0.0046)	(0.0249)	(0.5028)
magic	均值	20.7384	726.2841	234.4493
	标准偏差	(0.4578)	(0.8441)	(6.0954)
pima	均值	0.8450	29.6995	9.1647
	标准偏差	(0.0306)	(0.0944)	(2.6230)
wine	均值	0.2971	13.0392	4.9555
	标准偏差	(0.0084)	(0.1080)	(1.0702)

对 3 个度量参数进行 5%显著性水平的 Wilcoxon 秩和检验,表 20—表 22 所列为结果 p 值的统计结果,可明显看出本文方法的优势。

表 20 ARI 的 p 值

UCI 数据集	最优方法	EFCMIDPSO	HPSOFCM	CPSFC
abalone	EFCMIDPSO	*	0.9590	0.3820
glass	HPSOFCM	0.0041	*	0.0015
heart	EFCMIDPSO	*	4.2857e−06	1.7300e−06
image seg	EFCMIDPSO	*	1.7344e−06	1.7344e−06
iris	EFCMIDPSO	*	6.3113e−05	0.2835
magic	EFCMIDPSO	*	0.0752	3.4053e−05
pima	EFCMIDPSO	*	1.1265e−05	1.7344e−06
wine	EFCMIDPSO	*	1.7344e−06	1.7344e−06

表 21 准则 J 的 p 值

UCI 数据集	最优方法	EFCMIDPSO	HPSOFCM	CPSFC
abalone	EFCMIDPSO	*	1.7344e−06	0.0519
glass	CPSFC	0.9263	1.7344e−06	*
heart	CPSFC	1.7344e−06	1.7344e−06	*
image seg	EFCMIDPSO	*	1.7344e−06	0.2059
iris	CPSFC	0.9754	1.7344e−06	*
magic	CPSFC	1.8910e−04	1.7344e−06	*
pima	CPSFC	3.8822e−06	1.7344e−06	*
wine	CPSFC	6.3198e−05	1.7344e−06	*

表 22 运行时间的 p 值

UCI 数据集	最优方法	HPSOFCM	CPSFC
abalone	EFCMIDPSO	1.7344e−06	1.7344e−06
glass	EFCMIDPSO	1.7344e−06	1.7344e−06
heart	EFCMIDPSO	1.7344e−06	1.7344e−06
image seg	EFCMIDPSO	1.7344e−06	1.7344e−06
iris	EFCMIDPSO	1.7344e−06	1.7344e−06
magic	EFCMIDPSO	1.7344e−06	1.7344e−06
pima	EFCMIDPSO	1.7344e−06	1.7344e−06
wine	EFCMIDPSO	1.7344e−06	1.7344e−06

结束语 本文结合了 IDPSO 与 FCM 两个算法,FCM 产生一个起始解,增加收敛的方向性,以此加速收敛;此外,IDPSO 在数据训练过程中自适应地调节粒子群的 3 个参数,从而解决了粒子群模糊聚类的两大难题。实验结果表明,本文方法具有较高的计算效率、聚类质量,同时本方法由于无需预设粒子群的参数,因此实用性较高。

参 考 文 献

[1] Yao Jing, He Ju-hou. Load balance strategy of cloud computing based on fuzzy clustering analysis[J]. Journal of Computer Applications, 2012, 32(1): 213-217(in Chinese)
姚婧,何聚厚. 基于模糊聚类分析的云计算负载平衡策略[J]. 计算机应用,2012,32(1):213-217

[2] Wang Hai-liang, She Kun, Zhou Ming-tian. Shadowed Sets-based Rough Fuzzy Possibilistic C-means Clustering[J]. Computer Science, 2013, 40(1): 191-194(in Chinese)
汪海良,佘堃,周明天. 基于阴影集的粗糙模糊可能性 C 均值聚类算法[J]. 计算机科学,2013,40(1):191-194

[3] Ren Li-na, Qin Yong-bin, Xu Dao-yun. Fuzzy C-means clustering based on self-adaptive weight[J]. Application Research of Computers, 2012, 29(8): 2849-2851(in Chinese)
任丽娜,秦永彬,许道云. 基于自适应权重的模糊 C-均值聚类算法[J]. 计算机应用研究,2012,29(8):2849-2851

[4] Hu Lei, Niu Qin-zhou, Chen Yan. Enhanced clustering algorithm based on fuzzy C-means and support vector machine[J]. Journal of Computer Applications, 2013, 33(4): 991-993(in Chinese)
胡磊,牛秦洲,陈艳. 模糊 C 均值与支持向量机相结合的增强聚类算法[J]. 计算机应用,2013,33(4):991-993

[5] Liao Song-you, Zhang Ji-fu, et al. Fuzzy C Means Clustering Algorithm by Using Fuzzy Entropy Constraint[J]. Journal of Chinese Computer Systems, 2014, 35(2): 379-383(in Chinese)
廖松有,张继福,刘爱琴. 利用模糊熵约束的模糊 C 均值聚类算法[J]. 小型微型计算机系统,2014,35(2):379-383

[6] Zhang L, Pedrycz W, Lu W, et al. An interval weighed fuzzy c-means clustering by genetically guided alternating optimization [J]. Expert Systems with Applications, 2014, 41(13): 5960-5971

[7] Sabzekar M, Naghibzadeh M. Fuzzy c-means improvement using relaxed constraints support vector machines[J]. Applied Soft Computing, 2013, 13(2): 881-890

[8] Alam S, Dobbie G, Yun S K, et al. Research on particle swarm optimization based clustering: A systematic review of literature and techniques[J]. Swarm & Evolutionary Computation, 2014, 17: 1-13

[9] Chen S, Xu Z, Tang Y. A Hybrid Clustering Algorithm Based on Fuzzy c-Means and Improved Particle Swarm Optimization[J]. Arabian Journal for Science & Engineering, 2014, 39(12): 8875-8887

[10] Zhang Y C, Xiong X, Zhang Q D. An improved self-adaptive PSO algorithm with detection function for multimodal function optimization problems[J]. Mathematical Problems in Engineering, 2013, 2013(12): 657-675

[11] Bezdek J C, Ehrlich R, Full W. FCM: The fuzzy c-means clustering algorithm[J]. Computers & Geosciences, 1984, 10(2): 191-203

[12] Izakian H, Abraham A. Fuzzy C-means and fuzzy swarm for fuzzy clustering problem[J]. Expert Systems with Applications, 2011, 38(3): 1835-1838

[13] Hubert L, Arabie P. Comparing partitions[J]. Journal of classification, 1985, 2(1): 193-218

[14] Stoimenova E. Nonparametric statistical inference[J]. Journal of Applied Statistics, 2012, 39(6): 1384-1385

[15] Li C, Zhou J, Kou P, et al. A novel chaotic particle swarm optimization based fuzzy clustering algorithm[J]. Neurocomputing, 2012, 83(15): 98-109

[16] Wang Hua-qiu, Luo Jiang. Research of Fuzzy Clustering Algorithm Based on Modified Harmony Search [J]. Journal of Chongqing University of Technology(Natural Science), 2012, 26(8): 71-78(in Chinese)
王华秋,罗江. 一种改进的和声搜索模糊聚类算法[J]. 重庆理工大学学报(自然科学),2012,26(8):71-78