文章编号:1000-5862(2021)02-0137-08

随机学习萤火虫算法优化的模糊软子空间聚类算法

张 曦[1],李 璠[1,2],付雪峰[1,2],谭德坤[1,2],赵 嘉[1,2,3*]
(1.南昌工程学院信息工程学院,江西 南昌 330099;2.江西省水信息协同感知与智能处理重点实验室,江西 南昌 330099;
3.鄱阳湖流域水工程安全与资源高效利用国家地方联合工程实验室,江西 南昌 330099)

摘要: 传统软子空间聚类算法在利用局部搜索策略解决等式约束的连续非线性的变量加权问题时,易陷入局部最优导致聚类效果不佳.针对该问题,该文提出了一种随机学习萤火虫算法优化的模糊软子空间聚类算法.该算法利用具有全局搜索能力的萤火虫算法对新算法的目标函数进行优化,同时,为弥补萤火虫算法易提前收敛和寻优精度较低的缺陷,对萤火虫种群进化方式和全局最优粒子的学习方式进行了改进.新算法将权值矩阵拟化成萤火虫种群,使变量加权的等式约束变为界约束,通过萤火虫位置的更新搜索最优权重并发掘子空间中隐藏的簇类.在人工数据集、UCI标准数据集和癌症基因表达数据集上的实验结果表明:该算法具有较好的聚类效果.

关键词: 软子空间聚类;变量加权;萤火虫算法
中图分类号: TP 301.6 **文献标志码:** A **DOI:** 10.16357/j.cnki.issn1000-5862.2021.02.05

0 引言

聚类是一种无监督的学习过程,目的是将一组对象划分为簇,同一簇内对象彼此之间的相似性比不同簇的更高.聚类分析作为一种数据分析工具,已广泛应用于许多领域,如文本挖掘[1]、序列分析[2]、社区检测[3]、生物信息学[4]等.

大数据发展日新月异,全球数据呈现爆发增长、海量集聚的特点,高维聚类已成为聚类分析的一个重要研究方向[5].传统聚类方法在对高维数据进行聚类时,"维度灾难"导致聚类效果较差,如在高维数据中含有大量无关的属性,使目标簇可能只在某些维度中存在;稀疏的数据分布使数据之间的距离几乎相等,导致传统方法中的全维空间距离度量变得毫无意义.

为解决以上问题,R. Agrawal等[6]提出子空间聚类(Subspace Clustering,SC).在子空间聚类过程中,定义维度对目标簇有完全贡献或没有贡献,在维度的子空间中识别目标簇.相反,考虑到每个维度对每个目标簇都有一定贡献,软子空间聚类(Soft Subspace Clustering,SSC)为每个维度分配一个0~1之间的权值,以权重来表示该维度对目标簇的贡献程度.Jing Liping等[7]提出了熵加权k均值(Entropy Weighting k-means,EWKM)算法,它是经典的软子空间聚类算法之一.张恒巍等[8]利用数据可靠性对权重进行计算,并引入萤火虫算法搜索子空间,提出了一种基于智能优化算法的模糊软子空间聚类算法.程铃钫等[9]设计了双加权方法,定义新距离度量和目标函数,提出了一种不平衡数据的软子空间聚类算法.范虹等采用烟花算法和头脑风暴算法优化软子空间聚类算法,分别提出了烟花算法优化的软子空间MR图像聚类算法[10]和头脑风暴算法优化的乳腺MR图像软子空间聚类算法[11].

软子空间聚类算法中的变量加权问题(Variable Weighting Problem)[12]是一个等式约束的连续非线性优化问题,传统软子空间聚类算法通过局部搜索策略来解决该问题,易陷入局部最优而无法保证良好的聚类效果.针对该问题,本文采用具有全局搜索能力的萤火虫算法优化目标函数,将权值矩阵拟化成萤火虫种群,使变量加权的等式约束变为界约束,通过萤火虫位置的更新搜索最优权重并发掘子空间

收稿日期:2020-01-18
基金项目:国家自然科学基金(52069014,61762063,51669014),江西省自然科学基金(2018ACB21029)和江西省教育厅科学技术研究(GJJ190956)资助项目.
通信作者:赵 嘉(1981—),男,安徽桐城人,教授,博士,主要从事智能计算与计算智能、数据挖掘与机器学习等研究.E-mail:zhaojia925@163.com

中隐藏的簇类. 同时,为弥补萤火虫算法易提前收敛导致寻优精度较低的缺陷,将随机吸引进化方式代替全吸引进化方式,并对全局最优粒子进行随机单维贪婪学习,提出了一种随机学习萤火虫算法优化的模糊软子空间聚类算法(Fuzzy Soft Subspace Clustering Algorithm Based On Random Learning Firefly Algorithm,RLFAFSSC). 在人工数据集、UCI 标准数据集和癌症基因表达数据集上的实验结果表明,RLFAFSSC 算法具有较好的聚类效果.

1 相关工作

1.1 软子空间聚类算法

给定 D 维空间中 N 个样本的数据矩阵 $\boldsymbol{X}$,软子空间聚类的目的是将 $\boldsymbol{X}$ 划成为 C 个簇,并得到 C 个簇对应的聚类中心 $\boldsymbol{V}=(v_1,v_2,\cdots,v_C)$ 和特征权重 $\boldsymbol{W}=(\omega_1,\omega_2,\cdots,\omega_C)$,其中 $\omega_i(i=1,2,\cdots,C)$ 表示第 i 个簇的特征权重.

熵加权 k 均值(EWKM)[7] 算法是一个经典的 KM 型软子空间聚类算法,通过梯度下降法求解目标函数的最小值问题完成聚类,其目标函数为

$$J_{\mathrm{EWKM}}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}\sum_{k=1}^{D}\omega_{ik}(x_{jk}-v_{jk})^2+\gamma\sum_{i=1}^{C}\sum_{k=1}^{D}\omega_{ik}\ln(\omega_{ik}),$$

$$\text{s.t. } u_{ij}\in\{0,1\},\sum_{i=1}^{C}u_{ij}=1,\omega_{ij}\in[0,1],\sum_{k=1}^{D}\omega_{ik}=1. \quad (1)$$

其中 $\boldsymbol{U}=(u_{ij})_{C\times N}$ 是硬隶属度矩阵;$\omega_{ik}\ln(\omega_{ik})$ 为负熵项,即负权值熵,参数 γ 用来平衡负权值熵对聚类过程的影响. 负熵项的引入有效地控制了每个簇所获得的特征权重,当 γ 极大时,每个簇将被分配相等的特征权重;当 γ 极小时,每个簇在较多的维度上的特征权重被分配为 0.

EWKM 已经成为一种基准软子空间聚类算法,陆续有学者将其改进并开发出新的软子空间聚类算法.

1.2 萤火虫算法

在自然界中,萤火虫之间主要靠自身发光来相互吸引和传递信息,由于光在传递中被传播媒介吸收,所以距离越远的萤火虫传递信息的准确性越差. 受此启发,Yang Xinshe[13] 提出了萤火虫算法(Firefly Algorithm,FA). FA 采用萤火虫种群的位置表示问题的解,萤火虫的亮度表示问题的适应值,迭代中萤火虫不断被较亮萤火虫吸引并向其移动完成位置更新,当满足终止条件时输出位置最优的萤火虫.

FA 包含 2 个寻优要素,即亮度和吸引度. 亮度决定萤火虫移动方向,吸引度体现移动距离,它们的计算公式为

$$I=I_0\mathrm{e}^{-\gamma r_{ij}^2},\beta=\beta_0\mathrm{e}^{-\gamma r_{ij}^2},$$

其中 I_0 为萤火虫最大亮度,即自身位置的初始亮度;β_0 为最大吸引度,即光源处的初始吸引度,e 为自然对数. 常数 γ 为光强吸收系数. 萤火虫 i 和 j 之间的欧氏距离用 r_{ij} 表示,其计算公式为

$$r_{ij}=\|x_i-x_j\|=\left(\sum_{d=1}^{D}(x_{id}-x_{jd})^2\right)^{1/2},$$

其中 d 为萤火虫维度,x_{id} 和 x_{jd} 分别表示萤火虫 i 和萤火虫 j 的第 d 维位置.

萤火虫 i 向萤火虫 j 移动的位置更新公式为

$$x_i(t+1)=x_i(t)+\beta(x_j(t)-x_i(t))+\alpha(r_{and}-0.5), \quad (2)$$

其中 $x_i(t)$、$x_j(t)$ 分别表示萤火虫 i 和萤火虫 j 的第 t 代位置;α 为步长因子,一般取 $[0,1]$ 上的常数;r_{and} 表示在 $[0,1]$ 上服从均匀分布的一个随机因子.

2 RLFAFSSC 算法

2.1 目标函数

在处理高维数据时,聚类面临维度灾难,并且噪声数据对聚类效果产生了较大的影响. 为克服这 2 个问题,本文采用文献[14] 提出的带噪声检测的模糊软子空间聚类(Fuzzy Soft Subspace Clustering with Noise Detection,FSSC-ND) 的目标函数作为 RLFAFSSC 的目标函数,即

$$J_{RLFAFSSC}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}^m\sum_{k=1}^{D}\omega_{ik}(x_{jk}-v_{jk})^2+\sum_{j=1}^{N}\delta^2\left(1-\sum_{i=1}^{C}u_{ij}\right)^m+\rho\sum_{i=1}^{C}\sum_{k=1}^{D}\omega_{ik}\ln(\omega_{ik}),$$

$$\text{s.t. } u_{ij}\in(0,1),\sum_{i=1}^{C}u_{ij}<1,0\leqslant\omega_{ik}\leqslant1,\sum_{k=1}^{D}\omega_{ik}=1. \quad (3)$$

$\boldsymbol{X}=(x_{jk})_{N\times D}$ 是 D 维空间中 N 个样本的数据矩阵;$\boldsymbol{U}=(u_{ij})_{C\times N}$ 是模糊隶属度矩阵,表示 N 个样本分别属于 C 个簇的程度;$\boldsymbol{V}=(v_{ik})_{C\times D}$ 为聚类中心矩阵,表示 C 个簇的中心位置;$\boldsymbol{W}=(\omega_{ik})_{C\times D}$ 为权重矩阵,表示 C 个簇的特征权重;$m(m>1)$ 是模糊系数;参数 $\rho(\rho>0)$ 用来控制负权值熵对特征权重的影响. $\boldsymbol{U}$、$\boldsymbol{V}$ 的更新公式由拉格朗日乘子法得到,如下所示:

$$u_{ij} = 1/\left(\sum_{l=1}^{C}\left(d_{ij}^2/d_{lj}^2\right)^{1/(m-1)} + \left(d_{ij}^2/\delta^2\right)^{1/(m-1)}\right), \quad (4)$$

$$d_{ij}^2 = \sum_{k=1}^{D}\omega_{ik}\left(x_{jk} - v_{ik}\right)^2,$$

$$\delta^2 = \frac{\lambda}{CN}\sum_{i=1}^{C}\sum_{j=1}^{N}d_{ij}^2, \quad (5)$$

$$v_{ik} = \left(\sum_{j=1}^{N}u_{ij}^m x_{jk}\right)/\sum_{j=1}^{N}u_{ij}^m, \quad (6)$$

其中式(5)是噪声聚类中计算噪声距离的方法,λ一般设为 0.1,d_{ij}^2表示在第 k 维上第 j 个数据点到第 i 个聚类中心的距离.

2.2　萤火虫算法的改进

标准 FA 在进行种群进化时,每只萤火虫要与其他萤火虫比较,向比它更亮的萤火虫移动. 这种全吸引进化方式使萤火虫移动次数增加,获得更多搜索机会,但移动次数过多导致萤火虫搜索过程发生振荡,给深度搜索带来阻碍,增加了计算复杂度. 针对该问题,本文借鉴文献[15]中随机吸引模型的思想,将萤火虫进化方式做出调整,即每只萤火虫随机选择 2 只萤火虫比较,向比它更亮和亮度最大的萤火虫移动,随机学习 $N_{umber} = n/2$ 次,n 为萤火虫个数.

在标准 FA 的每一次迭代中,萤火虫都在向当前全局最优粒子靠近的趋势中,而当前全局最优粒子从萤火虫中产生,并没对当前全局最优粒子进行再学习. 随着迭代进行,萤火虫之间的相似性越来越大,使得全局最优粒子变化甚微,导致萤火虫进化停滞,提前收敛. 针对该问题,本文对当前全局最优粒子进行随机学习.

通过上述分析可知,若在该代萤火虫中随机选择一只作为当前全局最优粒子的学习对象,则就可能使当前全局最优粒子向该代亮度较低的萤火虫移动,导致其位置不佳; 当前全局最优粒子可能在某些维度上的位置较好,某些维度上的位置较差,因此一些维度不需要进行学习. 为使全局最优粒子每次都向该代亮度较高的萤火虫学习并同时考虑到维度上的差异,本文做出如下改进: 将萤火虫按亮度降序排列,以当前全局最优粒子的每 1 维为研究对象,将前 50% 的萤火虫视为亮度较高的萤火虫,并从中随机选择一只,利用

$$x_d^{gBest}(t+1) = x_d^{gBest}(t) + r_{and}\left(x_{rd}(t) - x_d^{gBest}(t)\right), \quad (7)$$

进行单维贪婪学习更新,其中 $x_d^{gBest}(t)$ 表示第 t 代全局最优粒子的第 d 维位置,$x_{rd}(t)$ 表示随机选择的第 r 只萤火虫第 t 代第 d 维位置,r_{and} 表示在 [0,1] 上且服从均匀分布的一个随机因子.

当前全局最优粒子随机学习完成后,将所有萤火虫与其比较,若萤火虫适应值差,则利用

$$x_i(t+1) = x_i(t) + \beta\left(x^{gBest}(t) - x_i(t)\right) \quad (8)$$

更新当前萤火虫的位置,反之,利用

$$x_i(t+1) = x^{gBest}(t) + r_{and} \quad (9)$$

突变位置,其中 $x_i(t)$ 表示第 i 只萤火虫的第 t 代位置,$x^{gBest}(t)$ 表示第 t 代全局最优粒子位置,β 表示当前全局最优粒子对当前萤火虫的吸引度.

改进的萤火虫种群进化方式减少了萤火虫的移动次数,防止了萤火虫在移动过程中出现振荡,全局最优粒子的单维贪婪学习机制避免了算法提前收敛,使萤火虫算法在求解目标函数的最小值问题时更精确,它们的结合有效地提高了聚类算法的性能.

2.3　RLFAFSSC 算法步骤

RLFAFSSC 算法以一组权重矩阵 W_0 作为萤火虫种群的初始位置,式(3)作为萤火虫适应值函数,萤火虫不断进化搜索适应值较小、亮度较大的位置,因此聚类过程就是将式(3)最小化的过程. 基于上述思路,RLFAFSSC 算法的流程如图 1 所示.

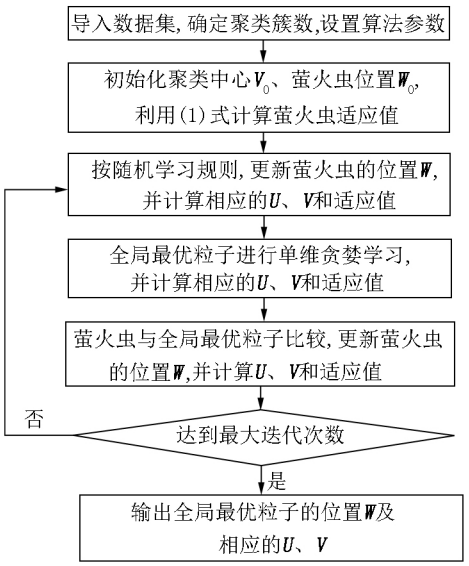


图 1　RLFAFSSC 算法流程图

由图 1 可得算法步骤如下:

(i) 导入数据集,确定聚类簇数 C,设置相关参数,如 m、λ、ρ、α、β_0、γ、萤火虫种群数 n、最大迭代次数 M;

(ii) 随机在当前数据集中选取 C 个数据作为初始聚类中心 V_0,随机初始化萤火虫位置 W_0,并利用式(3)计算每只萤火虫的适应值;

(iii) 按照本文的随机学习规则,利用式(2) 对萤火虫位置 W 进行更新,利用式(4)、式(6) 更新模糊隶属度 U 和聚类中心 V,并计算萤火虫的适应值;

(iv) 在当前迭代内排序并选出全局最优粒子 x^{gBest},并利用式(7) 进行单维贪婪学习,再计算相应的 U、V 和适应值;

(v) 将萤火虫与完成单维贪婪学习的全局最优粒子比较,若萤火虫适应值差,则利用式(8) 更新当前萤火虫的位置;否则,利用式(9) 突变位置,并计算萤火虫相应的 U、V 和适应值;

(vi) 当迭代次数达到 M 时终止算法,输出全局最优粒子的位置 W 及相应的模糊隶属度矩阵 U 和聚类中心矩阵 V,否则转到(iii).

3 实验结果与分析

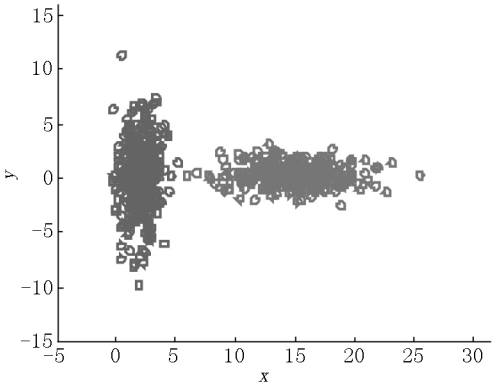
为验证 RLFAFSSC 算法的性能,本文分别对人工数据集、UCI 标准数据集和癌症基因表达数据集进行了不同的实验. 在人工数据集上,将 RLFAFSSC 算法的结果与 FSSC-ND 算法进行比较,检验其可发掘适当子空间和处理噪声数据的性能;在 UCI 标准数据集上,对比验证 RLFAFSSC 算法在低维数据聚类时具有较高的精度;在癌症基因表达数据集上,对比验证RLFAFSSC算法在处理高维数据时也能取得较好的聚类效果.

3.1 人工数据集

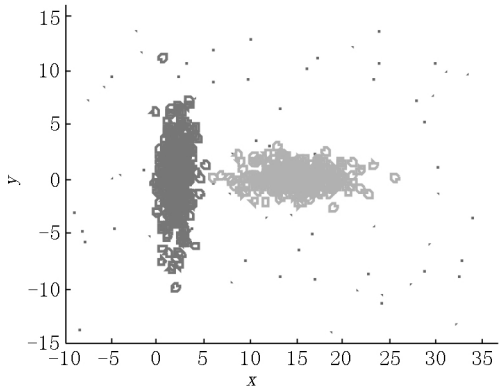
实验在人工数据集 X600 和 X700 上进行,X600 数据集是一个服从 2 元正态分布的矩阵,其中均值向量为(2,0)、(15,0),协方差矩阵为 $\begin{pmatrix}10 & 0\\ 0 & 1\end{pmatrix}$、$\begin{pmatrix}1 & 0\\ 0 & 10\end{pmatrix}$,每簇有300个点;X700数据集是在X600的基础上加入 100 个服从均匀分布的随机噪声点. X600 和 X700 的散点图如图 2 所示.

在实验中,对 2 个数据集进行归一化处理,RLFAFSSC 和 FSSC-ND 算法对其进行聚类,算法参数设置如表 1 所示. X600 和 X700 的真实簇心为 $\boldsymbol{V}_{\text{true}}=\begin{pmatrix}2.110\,5 & 0.021\,9\\ 15.083\,8 & 0.013\,8\end{pmatrix}$. 这 2 种算法在 2 个数据集上得到的簇心如表 2、表 3 所示.

由表 2 可知,在 X600 数据集上 FSSC-ND 和 RLFAFSSC 算法聚类得到的簇心与真实簇心的误差分别为 0.095 4、0.064 9,它们均接近真实簇心. 因为它们找到了各个簇合适的特征权重,区分了不同簇的子空间,但RLFAFSSC算法的误差更小,这表明它找到的子空间更为合适. 如表 3 所示,FSSCND 和 RLFAFSSC 算法在 X700 数据集上的聚类生成的簇心与真实簇心的误差分别为 0.156 6、0.126 1,后者的误差更小,因此 RLFAFSSC 算法得到的簇心更靠近真实簇心,这表明 RLFAFSSC 算法的抗噪性比 FSSCND 算法更好. 文献[4] 已通过该人工数据集实验验证了 FSSCND 算法可发掘适当子空间和处理噪声数据的优势,因此结合 2 个数据集上的结果来看,RLFAFSSC 算法继承了该优点.



(a) X600 数据集



(b) X700 数据集

图 2 X600 和 X700 的散点图

表 1 算法参数设置

算法	参数设置
RLFAFSSC	$m=2,\rho=2,\lambda=0.1,M=200,n=10,\beta_0=1,\gamma=0.9,\alpha_0=0.01$
FSSCND	$m=2,\rho=2,\lambda=0.1$

表 2 FSSCND、RLFAFSSC 在 X600 上生成的簇心及误差

算法	$\boldsymbol{V}$	$\boldsymbol{E}=\|\boldsymbol{V}_{\text{true}}-\boldsymbol{V}\|$
FSSCND	$\begin{pmatrix}2.157\,7 & 0.021\,9\\ 15.166\,7 & 0.013\,8\end{pmatrix}$	0.095 4
RLFAFSSC	$\begin{pmatrix}2.139\,0 & 0.022\,2\\ 15.142\,1 & 0.013\,5\end{pmatrix}$	0.064 9

表 3　FSSCND、RLFAFSSC 在 X700 上生成的簇心及误差

算法	V	$E = \|V_{true} - V\|$
FSSCND	$\begin{pmatrix} 2.197\,1 & 0.017\,9 \\ 15.214\,2 & 0.017\,8 \end{pmatrix}$	0.156 6
RLFAFSSC	$\begin{pmatrix} 2.176\,9 & 0.020\,6 \\ 15.191\,0 & 0.015\,1 \end{pmatrix}$	0.126 1

3.2　UCI 标准数据集

实验利用 6 个 UCI 标准数据集进行测试,检验 RLFAFSSC 算法在低维数据中的聚类性能,数据集基本信息如表 4 所示.

表 4　UCI 数据集

数据集	样本数	维数	类别数
Australian	690	14	2
Balance-scale	625	4	3
Glass	214	9	6
Heart	270	13	2
Iris	150	4	3
Wine	178	13	3

本文采用 Rand Index(RI)[10] 和 Normalized Mutual Information(NMI)[10] 指标评价聚类效果. RI 和 NMI 的计算公式分别为

$$R_I = (f_{00} + f_{11}) / (N(N-1)/2),$$

$$N_{MI} = \sum_{i=1}^{K}\sum_{j=1}^{C} n_{ij}\log(Nn_{ij}/(n_i n_j)) / \left(\left(\sum_{i=1}^{K} n_i \log(n_i/N)\right)\left(\sum_{j=1}^{C} n_j \log(n_j/N)\right)\right)^{1/2},$$

其中 K 为标签数量,C 为聚类簇数,N 为样本总数,f_{00} 表示具有不同标签的点被分配到不同簇的样本对数,f_{11} 表示具有相同标签的点被分配到同簇的样本对数,n_i 表示具有第 i 个标签的样本数,n_j 表示簇 j 中的样本数,n_{ij} 表示具有第 i 个标签的样本点被分配到簇 j 的样本对数. R_I 和 N_{MI} 在 [0,1] 区间内取值,指标值越接近 1,聚类效果越好.

对所有数据集归一化处理,参数设置如表 5 所示,在 UCI 数据集上每种算法运行 50 次,将 RLFAFSSC 算法与 FSSC-ND[14]、EWKM[7]、LAC[16]、FSC[17]、FCM[18] 和 NC[19] 算法进行比较,获得的最佳结果如表 6、表 7 所示.

表 5　算法参数设置

算法	参数设置
RLFAFSSC	$m = 1.02、2.00$; $\rho = 1、3、5、10、50、100、1\,000$; $\lambda = 0.1$; $\alpha_0 = 0.01、0.05、0.10、0.50$
FSSC-ND	$m = \min(N, D-1)/(\min(N, D-1) - 2)$; $\rho = 1、3、5、10、50、100、1\,000$; $\lambda = 0.1$

由表 6 和表 7 可知,RLFA-FSSC 算法在 6 个 UCI 数据集上的 R_I、N_{MI} 最好值均明显优于其他 6 种算法,这表明 RLFAFSSC 算法较其他算法取得了更好的聚类效果.

表 6　每种算法运行 50 次的最佳结果(RI)

数据集	评价指标	RLFAFSSC	FSSC-ND	EWKM	LAC	FSC	FCM	NC
Australian	最好值	0.770 6	0.721 7	0.687 7	0.657 5	0.557 3	0.706 6	0.708 5
	方差	0.004 6	0.011 8	0.085 0	0.092 4	0.037 0	0.076 2	0.063 2
Balance	最好值	0.747 9	0.615 5	0.594 3	0.593 5	0.594 9	0.579 3	0.595 1
	方差	0.036 0	0.029 7	0.029 6	0.015 7	0.073 7	0.054 2	0.036 3
Glass	最好值	0.742 6	0.731 8	0.661 0	0.674 2	0.653 4	0.702 5	0.726 2
	方差	0.008 7	0.036 9	0.076 0	0.015 8	0.016 6	0.081 2	0.023 2
Heart	最好值	0.721 2	0.678 9	0.673 5	0.651 0	0.652 9	0.669 9	0.670 1
	方差	0.010 2	0.025 3	0.005 8	0.057 0	0.032 1	0.051 1	0.045 5
Iris	最好值	0.934 1	0.902 3	0.878 5	0.862 3	0.837 0	0.873 7	0.892 3
	方差	0.030 8	0.004 0	0.002 7	0.033 7	0.074 0	0.080 9	0.020 0
Wine	最好值	0.962 0	0.946 7	0.931 0	0.932 6	0.882 7	0.939 8	0.940 7
	方差	0.006 7	0.004 6	0.005 4	0.008 3	0.005 5	0.012 2	0.009 6

表 7　每种算法运行 50 次的最佳结果(NMI)

数据集	评价指标	RLFAFSSC	FSSC-ND	EWKM	LAC	FSC	FCM	NC
Australian	最好值	0.448 0	0.351 7	0.310 7	0.325 4	0.427 9	0.334 3	0.345 6
	方差	0.006 0	0.086 1	0.153 7	0.145 7	6.2e-17	0.177 0	0.094 2
Balance	最好值	0.370 4	0.187 8	0.128 5	0.128 5	0.145 9	0.096 1	0.122 3
	方差	0.064 3	0.041 7	0.059 3	0.050 9	0.148 2	0.065 1	0.070 1

表 7(续)

数据集	评价指标	RLFAFSSC	FSSC-ND	EWKM	LAC	FSC	FCM	NC
Glass	最好值	0.374 4	0.350 6	0.346 0	0.347 5	0.240 4	0.296 2	0.316 0
	方差	0.022 2	0.027 0	0.024 3	0.020 5	0.113 8	0.032 1	0.028 9
Heart	最好值	0.345 9	0.290 7	0.270 9	0.237 6	0.050 1	0.264 7	0.275 0
	方差	0.016 7	0.017 4	0.008 1	0.084 9	0.047 2	0.078 8	0.050 4
Iris	最好值	0.830 8	0.742 9	0.740 7	0.718 3	0.694 4	0.730 4	0.741 5
	方差	0.035 4	0.004 5	0.005 8	0.047 6	0.105 7	0.123 3	0.008 7
Wine	最好值	0.885 5	0.862 7	0.831 7	0.834 6	0.733 4	0.846 6	0.849 4
	方差	0.013 6	0.005 8	0.042 1	0.056 6	0.040 0	0.134 6	0.029 2

3.3 癌症基因表达数据集

利用 6 个高维癌症基因表达数据集[20]进行测试,检验 RLFAFSSC 算法在高维数据中的聚类性能,数据集基本信息如表 8 所示. 实验中,对所有数据集进行归一化处理,参数设置如表 9 所示. 将 RLFAFSSC 算法与 FSSC-ND[14]、ESSC[21]、QPSOSC[20]、FWKM[22]、EWKM[7]和 LAC[16]算法进行比较,在癌症基因表达数据集上,每种算法运行 50 次并记录平均值,结果如表 10 和表 11 所示.

表 8 癌症基因表达数据集

数据集	样本数	维数	类别数
MLL	72	12 533	3
Bladder	40	5 724	3
Prostate	20	12 627	2
Lung	203	12 600	5
Breast	24	12 625	2
DLBCL	77	7 070	3

表 9 算法参数设置

算法	参数设置
RLFAFSSC	$m = 1.02、1.20、1.50、2.00$; $\rho = 1、3、5、10、50、100、1\ 000$; $\lambda = 0.1$; $\alpha_0 = 0.05、0.10、0.50$
FSSC-ND	$m = \min(N, D-1)/(\min(N, D-1)-2)$; $\rho = 1、3、5、10、50、100、1\ 000$; $\lambda = 0.1$

对比表 10、表 11 可知,RLFAFSSC 算法在 MLL、Bladder、Lung 和 Breast 数据集上的 R_I、N_{MI}平均值不仅优于 5 种具有局部搜索策略的软子空间算法,还优于具有全局搜索策略的 QPSOSC 算法;由于相同算法得到的 R_I 和 N_{MI}值不一定正相关,在 DLBCL 数据集上出现了 RLFAFSSC 算法的 R_I 平均值最优而 EWKM 算法的 N_{MI}平均值最优的情况;在 Prostate 数据集上 R_I、N_{MI}平均值最优的算法是 QPSOSC,而 RLFAFSSC 算法结果并不理想,这是因为当每个簇的大小差异较大时,RLFAFSSC 目标函数中使用单个值 δ^2 表示所有点到噪声簇的距离变得不准确.

表 10 每种算法运行 50 次的平均结果(R_I)

数据集	评价指标	RLFAFSSC	FSSC-ND	QPSOSC	ESSC	FWKM	EWKM	LAC
MLL	平均值	0.788 0	0.772 2	0.743 0	0.713 0	0.623 8	0.664 3	0.648 8
	方差	0.013 4	0.001 1	0.000 2	0.067 1	0.011 2	0.078 0	0.100 9
Bladder	平均值	0.728 1	0.716 7	0.679 5	0.636 4	0.621 4	0.636 2	0.621 5
	方差	0.036 6	2.7×10^{-5}	0.001 1	0.087 5	0.108 9	0.100 5	0.093 5
Prostate	平均值	0.523 0	0.523 0	0.605 3	0.525 8	0.590 2	0.520 1	0.505 9
	方差	1.5×10^{-14}	3.3×10^{-16}	0.002 0	0.052 0	0.056 2	0.048 9	0.043 2
Lung	平均值	0.590 0	0.560 4	0.579 6	0.544 9	0.534 8	0.562 7	0.551 9
	方差	0.019 1	0.022 9	3.7×10^{-5}	0.033 6	0.225 7	0.039 5	0.024 7
Breast	平均值	0.645 5	0.572 0	0.608 7	0.570 5	0.545 5	0.539 4	0.567 4
	方差	0.062 6	0.010 9	0.000 2	0.052 8	0.067 7	0.041 2	0.041 1
DLBCL	平均值	0.642 8	0.634 8	0.637 0	0.564 3	0.537 2	0.571 4	0.528 2
	方差	0.031 9	0.022 4	1.4×10^{-32}	7.5e-16	0	3.4×10^{-16}	4.5×10^{-16}

综合表 6、表 7、表 10 和表 11 的实验结果来看,对于低维 UCI 数据集,RLFAFSSC 算法能够有效地提高聚类精度,同时在高维癌症基因表达数据集上 RLFAFSSC 算法也取得了较好的聚类效果.

表 11　每种算法运行 50 次的平均结果(N_{MI})

数据集	评价指标		RLFAFSSC	FSSC-ND	QPSOSC	ESSC	FWKM	EWKM
MLL	平均值	0.532 8	0.526 9	0.445 1	0.397 5	0.321 1	0.344 3	0.333 7
	方差	0.035 2	0.001 1	0.000 4	0.134 4	0.157 6	0.144 7	0.160 7
Bladder	平均值	0.557 3	0.552 8	0.481 2	0.351 3	0.333 4	0.355 8	0.328 6
	方差	0.075 1	3.7×10^{-5}	0.001 4	0.141 4	0.148 9	0.144 1	0.132 9
Prostate	平均值	0.067 0	0.067 0	0.191 9	0.114 1	0.087 7	0.119 9	0.099 1
	方差	4.6×10^{-12}	1.7×10^{-17}	0.004 9	0.100 0	0.098 5	0.102 2	0.092 1
Lung	平均值	0.353 9	0.271 8	0.322 6	0.244 5	0.237 8	0.264 4	0.249 6
	方差	0.078 2	0.069 3	3.3×10^{-5}	0.072 4	0.078 5	0.079 2	0.063 5
Breast	平均值	0.297 9	0.286 3	0.240 8	0.240 5	0.179 9	0.175 4	0.181 3
	方差	0.092 5	0.046 3	0.002 7	0.084 4	0.065 3	0.076 4	0.039 8
DLBCL	平均值	0.085 7	0.098 4	0.056 3	0.279 8	0.189 9	0.333 8	0.179 7
	方差	0.013 7	0.027 2	4.3×10^{-35}	5.6×10^{-17}	2.3×10^{-16}	5.6×10^{-17}	1.1×10^{-16}

4　结束语

本文提出了一种随机学习萤火虫算法优化[23]的模糊软子空间聚类算法. RLFAFSSC 算法在 FSSC-ND 的基础上,引入了具有全局搜索能力的萤火虫算法优化目标函数,同时,为弥补 FA 提前收敛导致寻优精度较低的缺陷,对萤火虫种群进化方式和全局最优粒子的学习方式进行了改进. RLFAFSSC 算法将权值矩阵拟化成萤火虫种群,使变量加权的等式约束变为界约束,通过萤火虫位置的更新寻找最优权重并发掘子空间中隐藏的簇类. 实验结果表明: RLFAFSSC 算法在继承 FSSC-ND 算法可发掘适当子空间和处理噪声数据的优点的基础上,不仅在 UCI 标准数据集上能有效地提高聚类精度,而且对高维癌症基因表达数据集聚类也能取得较好的聚类效果. 下一步的研究方向是将其他聚类算法与 RLFAFSSC 算法进行结合,使其能处理任意簇大小的数据集.

5　参考文献

[1] 潘敏,王明文,王晓庆,等. 基于簇特征的文本增量聚类研究 [J]. 江西师范大学学报: 自然科学版,2014,38(1):95-101.

[2] 张巍,王洋,刘东宁,等. 基于随机聚类方法建模的序列分析 [J]. 江西师范大学学报: 自然科学版,2017,41(5):470-475.

[3] 程艳,解建华,谭平飞,等. 面向虚拟学习社区的学习行为特征挖掘与分组方法的研究 [J]. 江西师范大学学报: 自然科学版,2016,40(6):640-643.

[4] Tsoucas D,Yuan G C. GiniClust2: a cluster-aware,weighted ensemble clustering method for cell-type detection [J]. Genome Biology,2018,19(1):58.

[5] 张曦,赵嘉,李沛武,等. 改进萤火虫优化的软子空间聚类算法 [J]. 南昌工程学院学报,2018,37(4):61-67.

[6] Agrawal R,Gehrke J,Gunopulos D,et al. Automatic subspace clustering of high dimensional data for data mining applications [J]. Special Interest Group on Management of Data of the Association for Computing Machinery,1998,27(2):94-105.

[7] Jing L,Ng M K,Huang J Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data [J]. IEEE Transactions on Knowledge and Data Engineering,2007,19(8):1026-1041.

[8] 张恒巍,何嘉婧,韩继红,等. 基于智能优化算法的模糊软子空间聚类方法 [J]. 计算机科学,2016,43(3):256-261.

[9] 程铃钫,杨天鹏,陈黎飞. 不平衡数据的软子空间聚类算法 [J]. 计算机应用,2017,37(10):2952-2957.

[10] 范虹,侯存存,朱艳春,等. 烟花算法优化的软子空间 MR 图像聚类算法 [J]. 软件学报,2017,28(11):3080-3093.

[11] 范虹,史肖敏,姚若侠. 头脑风暴算法优化的乳腺 MR 图像软子空间聚类算法 [J]. 计算机科学与探索,2020,14(8):1348-1357.

[12] Lu Yanping,Wang Shengrui,Li Shaozi,et al. Particle swarm optimizer for variable weighting in clustering high-dimensional data [J]. Machine learning,2011,82(1):43-70.

[13] Yang Xinshe. Nature-inspired metaheuristic algorithms [M]. London: Luniver Press,2008.

[14] Chitsaz E,Jahromi M Z. A novel soft subspace clustering algorithm with noise detection for high dimensional datasets [J]. Soft Computing,2016,20(11):4463-4472.

[15] 赵嘉,谢智峰,吕莉,等. 深度学习萤火虫算法 [J]. 电子学报,2018,46(11):2633-2641.
[16] Domeniconi C,Gunopulos D,Ma S,et al. Locally adaptive metrics for clustering high dimensional data [J]. Data Mining and Know-Ledge Discovery,2007,14(1):63-97.
[17] Gan Gao,Wu Jin. A convergence theorem for the fuzzy subspace clustering algorithm [J]. Pattern Recognition,2008,41(6):1939-1947.
[18] Tao Lei,Jia Xiaohong,Zhang Yanning,et al. Significantly fast and robust fuzzy *c*-means clustering algorithm based on morphological reconstruction and membership filtering [J]. IEEE Transactions on Fuzzy Systems,2018,26(5):3027-3041.
[19] Dave R N. Characterization and detection of noise in clustering [J]. Pattern Recognition Letters,1991,12(11):657-664.
[20] Li Yangyang,Liang Xiaoxu,Lu Yujing,et al. Soft subspace clustering using QPSOSC algorithm [EB/OL]. [2019-11-17]. https://ieeexplore.ieee.org/document/8285264.
[21] Deng Zhaohong,Choi K S,Chung F L,et al. Enhanced soft subspace clustering integrating within-cluster and between-cluster information [J]. Pattern Recognition,2010,43(3):767-781.
[22] Jing Liping,Ng M K,Xu Jun,et al. Subspace clustering of text documents with feature weighting *k*-means algorithm [EB/OL]. [2019-11-17]. https://link.springer.com/chapter/10.1007/11430919_94.
[23] 谢智峰,吴润秀,吕莉. 多策略融合萤火虫算法在年径流预测中的应用 [J]. 南昌工程学院学报,2020,40(1):20-27.

The Fuzzy Soft Subspace Clustering Algorithm Optimized by Random Learning Firefly Algorithm

ZHANG Xi[1],LI Fan[1,2],FU Xuefeng[1,2],TAN Dekun[1,2],ZHAO Jia[1,2,3*]

(1. School of Information Engineering,Nanchang Institute of Technology,Nanchang Jiangxi 330099,China; 2. Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing,Nanchang Jiangxi 330099,China; 3. National-Local Engineering Laboratory of Water Engineering Safety and Effective Utilization of Resources in Poyang Lake Area,Nanchang Jiangxi 330099,China)

Abstract: The traditional soft subspace clustering algorithm uses local search strategy to solve the continuous nonlinear variable weighting problem with equality constraints,and is easy to fall into local optimum,resulting in poor clustering result. To solve this problem,a fuzzy soft subspace clustering algorithm optimized by random learning firefly algorithm is proposed. The firefly algorithm with global search ability is used to optimize the objective function of the new algorithm. At the same time,in order to make up for premature convergence and low precision of firefly algorithm,the evolution pattern of firefly population and the learning method of global optimal particle are improved. The new algorithm formulates the weight matrix into firefly population,and transforms equality constraints of variable weighting problem into bound constraints,updating the firefly position to search for optimal weight and to explore the hidden clusters in the subspace. The experimental results on artificial dataset,UCI standard dataset and cancer gene expression dataset show that the new algorithm has better clustering effect.

Key words: soft subspace clustering; variable weighting; firefly algorithm

(责任编辑:冉小晓)