基于差分演化算法的软子空间聚类

毕志升 王甲海 印 鉴
(中山大学信息科学与技术学院 广州 510006)

摘 要 软子空间聚类算法的性能主要取决于其目标函数和搜索策略.文中提出了一种基于差分演化算法的软子空间聚类算法 DESC.首先,设计了一个结合模糊加权类内相似性和界约束权值矩阵的新目标函数.然后,提出了新的隶属度计算方法.最后,引入了一种有效的全局搜索算法——复合差分演化算法,并运用该算法优化新目标函数和搜索子空间中的聚类.实验表明,新目标函数和复合差分演化算法的引入有效地提高了软子空间聚类算法的性能,新算法较已有软子空间聚类算法有明显优势.

关键词 高维数据;子空间聚类;差分演化;模糊聚类;文本分类
中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2012.02116

Subspace Clustering Based on Differential Evolution

BI Zhi-Sheng WANG Jia-Hai YIN Jian
(*School of Internation Science and Technology, Sun Yat-Sen University, Guangzhou* 510006)

Abstract The performance of soft subspace clustering largely depends on the objective function and the search strategy. This paper presents a differential evolution (DE) based algorithm for subspace clustering. In the proposed algorithm, a novel objective function is firstly designed by considering the fuzzy weighting within-cluster compactness and loosening the constraints of dimension weight matrix. Then, a novel membership between a data point and a cluster is proposed. At last, an efficient global search strategy, composite DE, is introduced to optimize the proposed objective function to search subspace clusters. The simulation results show that both the proposed objective function and the introduced DE search strategy contribute to the performance enhancement of soft subspace clustering, and thus the proposed algorithm is significantly better than existing algorithms.

Keywords high-dimensional data; subspace clustering; differential evolution; fuzzy clustering; text categorization

1 引 言

聚类分析是数据挖掘中一种常用的无监督分析方法.它根据数据样本的相似程度将数据样本划分为若干个簇,使得同一个簇中的样本相似性大,不同簇间的样本相似性小.随着数据挖掘研究的发展,聚类分析被广泛地运用在统计学、模式识别、机器学习等众多领域中[1].信息技术的发展使得数据采集和存储变得更加快捷和简单,也由此产生了大量维数多、规模大的复杂数据集.不同于低维数据集,在高维数据集中存在数据稀疏性以及"维数灾难"等问题.此外,不同

收稿日期:2012-06-30;最终修改稿收到日期:2012-08-20.本课题得到国家自然科学基金(60805026,61070076,61033010)、广东省自然科学基金(S2011020001182)、广东省科技计划项目(2009B090300450,2010A040303004,2011B040200007)、广州市珠江科技新星(2011J2200093)资助.**毕志升**,男,1983 年生,博士研究生,主要研究方向为数据挖掘、计算智能.E-mail: bivictor@gmail.com. **王甲海**(通信作者),男,1977 年生,博士,副教授,主要研究方向为计算智能和数据挖掘.E-mail: wangjiah@mail.sysu.edu.cn. **印 鉴**,男,1968 年生,博士,教授,博士生导师,主要研究领域为机器学习和数据挖掘.E-mail: issjyin@mail.sysu.edu.cn.

类别的样本往往与不同的属性子集(子空间)相关.因而,受到冗余和不相关属性的影响,传统的聚类算法无法有效解决高维数据集上的聚类问题[2].为了解决上述问题,国内外学者提出了各种特征转换和特征选择的方法[3],子空间聚类就是其中一个重要分支.

子空间聚类是一种寻找隐藏在不同低维子空间中的聚类的技术[4].它将数据样本划分成簇的同时,搜索各个簇所在的子空间.在每一个簇中,各个属性被赋予不同的权值,用于度量属性与簇的相关性.子空间就是这样一个加权的属性空间.根据加权方法的不同,子空间聚类可分为两大类:硬子空间聚类和软子空间聚类[5].在硬子空间聚类中,属性权值为0或1;而在软子空间聚类中,属性依据其与簇的相关程度被赋予[0,1]区间上的权值.权值越大,属性与簇的相关性越高.相比硬子空间聚类,软子空间聚类不仅反映了属性与簇是否相关,而且反映了各个相关属性在相关程度上的差异.

子空间聚类算法能有效减少冗余和不相关属性对聚类过程的干扰,从而提高高维数据集上的聚类效果.文献[6]对子空间聚类算法有详细介绍.本文仅关注软子空间聚类算法.

近年来,国内外出现了很多新的软子空间聚类算法[4-10].然而,这些算法仅关注于提出新的目标函数.在搜索策略上,它们普遍采用与k-均值(k-means,KM)[11]算法相似的算法结构[12],通过在KM中加入计算权值的额外步骤,迭代地优化目标函数值.因此,这类算法都保留了KM依赖于初始解以及容易陷入局部最优的缺点.为了弥补这些不足,Lu等人[12]通过引入全面学习的粒子群(Comprehensive Learning Particle Swarm Optimizer,CLPSO)[13]算法,提出了基于粒子群算法的子空间聚类算法PSOVW(Particle Swarm Optimizer for Variable Weighting).此外,PSOVW的目标函数将原本对权值矩阵的等式约束放松为界约束,简化了算法的搜索过程.实验证明,PSOVW大大提高了软子空间聚类算法的性能.

Lu等人[12]指出,合适的目标函数和有效的搜索策略是提高软子空间聚类算法性能的基础.基于这种思想,本文提出了一种基于差分演化(Differential Evolution,DE)[14]算法的软子空间聚类算法DESC(Differential Evolution for Subspace Clustering).首先,我们设计了一个结合模糊加权类内相似性和界约束权值矩阵的新目标函数.然后,通过结合模糊隶属度和硬隶属度,提出了新的隶属度计算方法.最后,引入了一种有效的全局搜索算法——复合差分演化算法(Composite DE,CoDE)[15],并运用该算法优化新目标函数和搜索子空间中的聚类.这是第一个基于DE的软子空间聚类算法.实验表明,新的目标函数和复合差分演化算法的引入有效地提高了软子空间聚类算法的性能.新算法较已有软子空间聚类算法有明显优势.

本文第2节介绍软子空间聚类算法的现状;第3节介绍新的目标函数和新的隶属度计算方法,并提出新算法DESC;第4节通过实验验证新算法的性能;第5节是本文结论.

2　相关工作

记$\boldsymbol{Z}=[z_{ik}]_{C\times D}$为聚类中心矩阵,记录每个簇的中心位置;$\boldsymbol{W}=[w_{ik}]_{C\times D}$为权值矩阵,记录每个簇的属性权值;$\boldsymbol{U}=[u_{ij}]_{C\times N}$为划分矩阵,记录每个样本对各个簇的隶属度.其中,$C$为聚类数,$N$为样本总数,$D$为样本的维数.

软子空间聚类算法的分类方法很多.例如:按加权方式可分为熵加权软子空间聚类算法和模糊加权软子空间聚类算法[6];按权值计算与聚类过程的结合方式可分为Wrapper型软子空间聚类算法和Filter型软子空间聚类算法[7].本文根据软子空间聚类算法的搜索策略将其分为基于局部搜索策略的软子空间聚类算法和基于全局搜索策略的软子空间聚类算法两大类.

2.1　基于局部搜索策略的软子空间聚类算法

一般地,基于局部搜索策略的软子空间聚类算法首先提出一个加权目标函数,然后采用基于梯度下降的技术迭代地优化目标函数值,最终收敛于局部最优解.

模糊子空间聚类(Fuzzy Subspace Clustering,FSC)算法[8]是一个经典的软子空间聚类算法.该算法模仿模糊c-均值(Fuzzy c-means,FCM)[16]算法中的模糊隶属度和模糊因子,定义了模糊权值和相应的模糊因子,并通过提出一个加权目标函数,利用与KM相似的算法结构求解软子空间聚类问题.FSC的目标函数如下:

$$J_{\mathrm{FSC}}=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}\sum_{k=1}^{D}w_{ik}^{\beta}(x_{jk}-z_{ik})^2+\varepsilon_0\sum_{i=1}^{C}\sum_{k=1}^{D}w_{ik}^{\beta}$$

$$\text{s.t. } u_{ij}\in\{0,1\},\ \sum_{i=1}^{C}u_{ij}=1,\ 0<\sum_{j=1}^{N}u_{ij}<N,\quad(1)$$

$$0\le w_{ik}\le 1\ \text{且}\sum_{k=1}^{D}w_{ik}=1$$

其中,ε_0是避免属性零散度时出现除零错误而引入的常数,β是模糊因子.w_{ik}^{β}对各个属性上样本与其所属簇中心的距离进行加权.由于w_{ik}^{β}形式上与

FCM 中模糊隶属度的加权方法相同，这种加权方法被称作模糊加权法. 使用这种加权方法的软子空间聚类算法被归类为模糊加权软子空间聚类算法[6].

FSC 的更新公式如下：

$$u_{ij}=\begin{cases}1, & i=\arg\min\limits_{q=1,\cdots,C}\sum\limits_{k=1}^{D}w_{qk}^{\beta}(x_{jk}-z_{qk})^2\\ 0, & \text{其它}\end{cases} \tag{2}$$

$$z_{ik}=\frac{\sum\limits_{j=1}^{N}u_{ij}x_{jk}}{\sum\limits_{j=1}^{N}u_{ij}} \tag{3}$$

$$w_{ik}=\frac{1}{\sum\limits_{l=1}^{D}\left[\frac{\sum\limits_{j=1}^{N}u_{ij}(x_{jk}-z_{ik})^2+\varepsilon_0}{\sum\limits_{j=1}^{N}u_{ij}(x_{jl}-z_{il})^2+\varepsilon_0}\right]^{1/(\beta-1)}} \tag{4}$$

FSC 的算法流程如下.

算法 1. FSC.

输入：聚类数 C，样本维数 D

输出：划分矩阵 $\boldsymbol{U}$，聚类中心矩阵 $\boldsymbol{Z}$，权值矩阵 $\boldsymbol{W}$

1. 在数据集中随机选择 C 个样本作为初始聚类中心，令 $\boldsymbol{W}=[w_{ik}]_{C\times D}=[1/D]_{C\times D}$；
2. 运用式(2)计算划分矩阵 $\boldsymbol{U}$；
3. REPEAT
4. 运用式(3)计算聚类中心矩阵 $\boldsymbol{Z}$；
5. 运用式(4)计算权值矩阵 $\boldsymbol{W}$；
6. 运用式(2)计算划分矩阵 $\boldsymbol{U}$；
7. UNTIL(算法收敛).

FSC 采用与 KM 相似的算法结构，通过在 KM 中加入计算权值的额外步骤(步 5)，迭代地优化目标函数值，因而保留了 KM 算法复杂度低和收敛迅速的优点. 正因为其目标函数以及算法结构与 KM 极为相似，被称为 KM 型软子空间聚类算法.

熵加权 k-均值(Entropy Weighting k-means，EWKM)算法[5]是另一个 KM 型软子空间聚类算法. 它引入了最大熵理论，通过同时优化加权类内距离和权值熵求解软子空间聚类问题. EWKM 的目标函数如下：

$$J_{\mathrm{EWKM}}=\sum_{i=1}^{C}\sum_{k=1}^{D}w_{ik}\sum_{j=1}^{N}u_{ij}(x_{jk}-z_{ik})^2+\gamma\sum_{i=1}^{C}\sum_{k=1}^{D}w_{ik}\ln w_{ik}$$

$$\text{s.t. } u_{ij}\in\{0,1\},\ \sum_{i=1}^{C}u_{ij}=1,\ 0<\sum_{j=1}^{N}u_{ij}<N,$$

$$0\le w_{ik}\le 1 \text{ 且}\sum_{k=1}^{D}w_{ik}=1 \tag{5}$$

其中 $-w_{ik}\ln w_{ik}$ 称为权值熵，表示属性 k 与簇 i 相关的确定性，熵加权 k-均值算法也因此得名. 若属性 k 与簇 i 的相关性明确，即属于簇 i 的样本在属性 k 上的分布非常集中或非常分散，则 w_{ik} 的权值熵较小；反之，其相关性不明确，权值熵较大. 参数 γ 用于平衡类内相似性和权值熵对目标函数值的影响.

如果仅要求提高类内相似性(式(5)中 $\gamma=0$)，将导致式(5)达到局部极小时各个簇所在的子空间仅仅由极少数甚至单一属性构成. 因而，EWKM 引入了权值熵，在最大化类内相似性的同时最大化权值熵，以确保权值维持在一个不为零的合理范围，从而使更多的属性包含在子空间中，为各个簇合理地保留更多信息.

EWKM 的算法流程与 FSC 一致，$\boldsymbol{Z}$ 的更新公式与 FSC 相同，$\boldsymbol{U}$ 和 $\boldsymbol{W}$ 的更新公式如下：

$$u_{ij}=\begin{cases}1, & i=\arg\min\limits_{q=1,\cdots,C}\sum\limits_{k=1}^{D}w_{qk}(x_{jk}-z_{qk})^2\\ 0, & \text{其它}\end{cases} \tag{6}$$

$$w_{ik}=\frac{\exp\left(-\frac{\sum\limits_{j=1}^{N}u_{ij}(x_{jk}-z_{ik})^2}{\gamma}\right)}{\sum\limits_{l=1}^{D}\exp\left(-\frac{\sum\limits_{j=1}^{N}u_{ij}(x_{jl}-z_{il})^2}{\gamma}\right)} \tag{7}$$

在以往的软子空间聚类算法中，类内距离作为类内相似性的度量被广泛运用，而类间相似性却一直被忽略. 针对这一现象，Deng 等人[6]提出了增强的软子空间聚类(Enhanced Soft Subspace Clustering，ESSC)算法. 该算法在 EWKM 基础上，结合用于度量类间相似性的加权类间分离度(Weighting Between-cluster Separation)，进一步提高了软子空间聚类算法的性能. ESSC 目标函数定义如下：

$$\begin{aligned}J_{\mathrm{ESSC}}=&\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}^{m}\sum_{k=1}^{D}w_{ik}(x_{jk}-z_{ik})^2+\\&\gamma\sum_{i=1}^{C}\sum_{k=1}^{D}w_{ik}\ln w_{ik}-\\&\eta\sum_{i=1}^{C}\Big(\sum_{j=1}^{N}u_{ij}^{m}\Big)\sum_{k=1}^{D}w_{ik}(z_{ik}-z_{0k})^2\end{aligned} \tag{8}$$

$$\text{s.t. } 0<u_{ij}<1,\ \sum_{i=1}^{C}u_{ij}=1,\ 0<\sum_{j=1}^{N}u_{ij}<N,$$

$$0\le w_{ik}\le 1 \text{ 且}\sum_{k=1}^{D}w_{ik}=1$$

其中，

$$z_{0k}=\frac{\sum\limits_{j=1}^{N}x_{jk}}{N} \tag{9}$$

通过引入类间相似性，ESSC 的聚类效果明显优于 FSC 等仅考虑类内相似性的软子空间聚类算

法. 然而，ESSC 因此引入了一个新的参数 η. ESSC 的算法流程与 EWKM 一致，其更新公式如下：

$$u_{ij}=\frac{(d_{ij})^{-1/(m-1)}}{\sum_{l}^{C}(d_{lj})^{-1/(m-1)}} \tag{10}$$

$$v_{ik}=\frac{\sum_{j=1}^{N}u_{ij}^{m}(x_{jk}-\eta z_{0k})}{\sum_{j=1}^{N}u_{ij}^{m}(1-\eta)} \tag{11}$$

$$w_{ik}=\exp\left(-\frac{\sigma_{ik}}{\gamma}\right)\Big/\sum_{l=1}^{D}\exp\left(-\frac{\sigma_{il}}{\gamma}\right) \tag{12}$$

其中，

$$d_{ij}=\sum_{k=1}^{D}w_{ik}(x_{jk}-z_{ik})^{2}-\eta\sum_{k=1}^{D}w_{ik}(z_{ik}-z_{0k})^{2} \tag{13}$$

$$\sigma_{ik}=\sum_{j=1}^{N}u_{ij}^{m}(x_{jk}-z_{ik})^{2}-\eta\sum_{j=1}^{N}u_{ij}^{m}(z_{ik}-z_{0k})^{2} \tag{14}$$

在上述算法中，权值的更新方法通过梯度下降法从目标函数直接推导求得，权值的计算与聚类中心矩阵紧密结合，属于 Wrapper 型软子空间聚类算法. 然而，Wrapper 型软子空间聚类算法只能是 KM 型算法[7]. 传统聚类算法中的其它优秀算法，如层次聚类算法和谱聚类算法，都无法通过这种方式运用到软子空间聚类中. 为了解决这个问题，Boongoen 等人[7]将数据可靠性引入到软子空间聚类中，提出了一种 Filter 型软子空间聚类技术，并运用该技术提出了一系列 Filter 型软子空间聚类算法.

记 N_{jk}^{α} 为样本 j 在属性 k 上 α 个近邻的集合，D_{jk}^{α} 为样本 j 在属性 k 上与其 α 个近邻的平均距离，AS_{jk}^{α} 为样本 j 在属性 k 上的“样本-属性”相关度. D_{jk}^{α} 与 AS_{jk}^{α} 的计算公式如下：

$$D_{jk}^{\alpha}=\frac{1}{\alpha}\sum_{q\in N_{jk}^{\alpha}}\sqrt{(x_{jk}-q_{k})^{2}} \tag{15}$$

$$AS_{jk}^{\alpha}=1-\frac{D_{jk}^{\alpha}}{\max_{\forall j,k}D_{jk}^{\alpha}} \tag{16}$$

D_{jk}^{α} 的大小反映了属性 k 上在样本 j 附近的样本密集程度. D_{jk}^{α} 越小，说明在属性 k 上样本 j 周围越密集，则样本 j 与属性 k 的相关性越高，AS_{jk}^{α} 越大；反之亦然. 显然，对任意给定的 k、j 和 α，AS_{jk}^{α} 仅与样本的分布有关，与聚类中心矩阵无关.

基于数据可靠性的 k-均值（Reliability-based KM，RKM）算法是其中一种运用该技术的 Filter 型软子空间聚类算法，其目标函数如下：

$$J_{\mathrm{RKM}}=\sum_{i=1}^{C}\sum_{k=1}^{D}\sum_{j=1}^{N}u_{ij}w_{ik}(x_{jk}-z_{ik})^{2}$$

$$\text{s.t. } u_{ij}\in\{0,1\},\ \sum_{i=1}^{C}u_{ij}=1,\ 0<\sum_{j=1}^{N}u_{ij}<N, \tag{17}$$

$$0\le w_{ik}\le 1\ \text{且}\ \sum_{k=1}^{D}w_{ik}=1$$

算法初始化时，随机选择 C 个样本作为初始聚类中心. 设样本 j 被选为簇 i 的中心，则簇 i 中各个属性的权值定义为

$$w_{ik}=\frac{AS_{jk}^{\alpha}}{\sum_{l}^{D}AS_{jl}^{\alpha}} \tag{18}$$

在算法的迭代过程中，$\boldsymbol{W}$ 的更新公式如下：

$$w_{ik}=\frac{MS_{ik}^{\alpha}}{\sum_{l}^{D}MS_{il}^{\alpha}} \tag{19}$$

$$MS_{ik}^{\alpha}=\min_{q\in C_{i}}AS_{qk}^{\alpha} \tag{20}$$

其中，C_i 是属于簇 i 的样本的集合. 由于权值仅通过 AS_{jk}^{α} 计算，因而不依赖于聚类中心矩阵.

RKM 的算法流程以及 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的更新公式与 EWKM 相同.

2.2　基于全局搜索策略的软子空间聚类算法

除了上述几种算法，近年国内外还出现了许多软子空间聚类算法[4,9-10]. 这些算法均采用局部搜索策略，因而算法十分依赖于初始解且容易陷入局部最优. 为了弥补这些不足，Lu 等人[12]提出了 PSOVW. 该算法引入了 CLPSO[13]求解软子空间聚类问题. 由于 CLPSO 是一种全局搜索算法，能解决以往算法依赖于初始解以及容易陷入局部最优的问题. 在 PSOVW 中，子空间聚类问题被描述为“变量加权问题（Variable Weighting Problem）”，$\boldsymbol{W}$ 被看作是问题的解. 算法为每一个簇中的每一个属性寻找最优的权值，并在加权的属性空间中寻找聚类.

PSOVW 的目标函数如下：

$$J_{P}=\sum_{i=1}^{C}\sum_{k=1}^{D}\sum_{j=1}^{N}u_{ij}\left(\frac{w_{ik}}{\sum_{l}^{D}w_{il}}\right)^{\beta}(x_{jk}-z_{ik})^{2}$$

$$\text{s.t. } u_{ij}\in\{0,1\},\ \sum_{i=1}^{C}u_{ij}=1,\ 0<\sum_{j=1}^{N}u_{ij}<N, \tag{21}$$

$$\text{且 } 0\le w_{ik}\le 1$$

PSOVW 的约束条件中不包含对 $\boldsymbol{W}$ 的等式约束 $\sum_{k=1}^{D}w_{ik}=1$，而是通过在目标函数中显式包含 $\boldsymbol{W}$ 的归一化，使得 $\sum_{k=1}^{D}\left(w_{ik}\Big/\sum_{l}^{D}w_{il}\right)=1$. 这种创新在不影响算法效果的前提下将目标函数对 $\boldsymbol{W}$ 的等式约

束放松为界约束,使得算法中 $\boldsymbol{W}$ 的更新变得更加简单[12]. PSOVW 中 $\boldsymbol{U}$ 和 $\boldsymbol{Z}$ 的更新公式如下:

$$u_{ij}=\begin{cases}1, & i=\arg\min\limits_{q=1,\cdots,C} d_{qj}\\ 0, & \text{其它}\end{cases} \tag{22}$$

$$z_{ik}=\frac{\sum\limits_{j=1}^{N} u_{ij}x_{jk}}{\sum\limits_{j=1}^{N} u_{ij}} \tag{23}$$

$$d_{qj}=\sum_{k=1}^{D}\left(\frac{w_{qk}}{\sum\limits_{l}^{D} w_{ql}}\right)^{\beta}(x_{jk}-z_{qk})^2 \tag{24}$$

不同于 EWKM 等局部搜索算法,PSOVW 仅通过梯度下降法推导出 $\boldsymbol{U}$ 和 $\boldsymbol{Z}$ 的更新公式,而 $\boldsymbol{W}$ 的更新通过 CLPSO 的搜索策略完成.

在 CLPSO 中,算法的搜索过程被形象地描述为一组粒子(particle)在解空间中运动的过程. 一个粒子 i 包含以下三方面的信息:粒子在解空间中的位置 $\boldsymbol{x}_{i,g}$,在解空间中运动的速度 $\boldsymbol{v}_{i,g}$,曾经找到的最好的位置(个体历史最优解)$\boldsymbol{pBest}_i$. 对于整个种群,CLPSO 记录一个全局最优解 $\boldsymbol{gBest}$.

在求解软子空间聚类问题时,粒子 i 在解空间中的位置 $\boldsymbol{x}_{i,g}=(x_{i,g,1},\cdots,x_{i,g,d},\cdots,x_{i,g,C\times D})$ 是长度为 $C\times D$ 的向量,对应一个权值矩阵 $\boldsymbol{W}$. 同时,算法为粒子保留其对应的聚类中心矩阵 $\boldsymbol{Z}$ 以及划分矩阵 $\boldsymbol{U}$.

在一次迭代中,算法对所有粒子完成一次演化. 其中,粒子 i 的演化过程如下.

首先构造粒子 i 的学习范例 $\boldsymbol{CpBest}_{i,g}$:

$$index(i,d)=\begin{cases}i, & \text{如果 } rand<Pc_i\\ m1, & \text{如果 } rand\geq Pc_i \text{ 且 } \boldsymbol{pBest}_{m1}\\ & \text{优于 } \boldsymbol{pBest}_{m2}\\ m2, & \text{如果 } rand\geq Pc_i \text{ 且 } \boldsymbol{pBest}_{m2}\\ & \text{优于 } \boldsymbol{pBest}_{m1}\end{cases} \tag{25}$$

$$CpBest_{i,g,d}=pBest_{index(i,d),d} \tag{26}$$

其中,$rand$ 是[0,1]区间上均匀分布的随机数,$m1$ 和 $m2$ 是[1,M]上随机选择的两个相互独立的自然数. Pc_i 是学习概率,用于平衡粒子向自身学习和向其它粒子学习的几率. 为了提高算法的搜索能力,每个粒子的学习概率并不相同. 在式(25)中,$\boldsymbol{pBest}_{m1}$ 优于 $\boldsymbol{pBest}_{m2}$ 是指 $\boldsymbol{pBest}_{m1}$ 对应的聚类结果拥有比 $\boldsymbol{pBest}_{m2}$ 更低的目标函数值. $index(i,d)$ 记录了粒子 i 第 d 维的学习对象. 在算法运行过程中,仅当粒子连续多次迭代均没有找到比个体历史最优更好的解时,$index(i,d)$ 才会重新生成. 在每一次迭代中,算法根据 $index(i,d)$ 构造 $\boldsymbol{CpBest}_{i,g}$.

得到 $\boldsymbol{CpBest}_{i,g}$ 后,粒子 i 的速度以及位置通过以下方式更新:

$$v_{i,g+1,d}=\rho v_{i,g,d}+a\cdot rand\cdot(CpBest_{i,g,d}-x_{i,g,d}) \tag{27}$$

$$x_{i,g+1,d}=x_{i,g,d}+v_{i,g+1,d} \tag{28}$$

其中,$0\leq\rho\leq1$ 是惯性因子,随着算法的运行线性递减. $rand$ 是[0,1]区间上均匀分布的随机数. a 是加速因子.

之后,根据式(22)、式(23)和式(21)计算粒子新的 $\boldsymbol{U}$、$\boldsymbol{Z}$ 以及目标函数值,并更新其个体历史最优解以及全局最优解. 至此,粒子 i 完成一次演化.

重复上述演化步骤直到算法满足终止条件.

可见,运用 CLPSO 求解软子空间聚类问题时,$\boldsymbol{W}$ 的更新不再受到 $\boldsymbol{U}$ 和 $\boldsymbol{Z}$ 的影响,有利于算法跳出局部最优和减少算法对初始解的依赖.

3 新的基于全局搜索的软子空间聚类——DESC

合适的目标函数和有效的搜索策略是提高软子空间聚类算法性能的基础. 因此,我们设计了一个结合模糊加权类内相似性和界约束权值矩阵的新目标函数. 然后,通过结合模糊隶属度和硬隶属度,提出了新的隶属度计算方法. 最后,引入了 CoDE,并运用该算法优化新目标函数和搜索子空间中的聚类.

3.1 目标函数

借鉴 FCM 和 ESSC,我们将式(21)扩展到模糊聚类. 新的目标函数如下:

$$\begin{aligned}&J_D=\sum_{i=1}^{C}\sum_{k=1}^{D}\sum_{j=1}^{N}u_{ij}^{m}\left(\frac{w_{ik}}{\sum\limits_{l}^{D}w_{il}}\right)^{\beta}(x_{jk}-z_{ik})^2\\ &\text{s.t. } 0\leq u_{ij}\leq1,\sum_{i=1}^{C}u_{ij}=1,0<\sum_{j=1}^{N}u_{ij}<N,\\ &\quad\text{且 } 0\leq w_{ik}\leq1\end{aligned} \tag{29}$$

式(29)是式(21)在模糊聚类上的扩展. 当 $m=1$ 且 $u_{ij}\in\{0,1\}$ 时,式(29)与 PSOVW 的目标函数相同;当 $\beta=0$ 时,式(29)与 FCM 的目标函数相同. 它既保留了 PSOVW 将 $\boldsymbol{W}$ 的等式约束放松为界约束的优点,又实现了 PSOVW 向模糊聚类的扩展. 参数 $m>1$ 称为模糊指标(Fuzzy Index). 当 $m\to1$ 时,DESC 退化为硬聚类;当 $m\to\infty$ 时,样本对各个聚类中心的隶属度趋向相等[17]. 参数 m 和 β 控制算法对划分矩阵 $\boldsymbol{U}$ 以及权值矩阵 $\boldsymbol{W}$ 的变化的敏感度. 在 DESC 中,我们建议两者均取较小的值(取值为 2). 这是因为在实验中我们发现,较大的取值将大大增加算法的运算时间,却并不能提高算法的效果. 另一

方面，$\boldsymbol{U}$ 和 $\boldsymbol{W}$ 都是软子空间聚类的目标且在目标函数中相互影响，所以建议 m 和 β 取值相等.

基于新的目标函数，DESC 的更新公式为

$$u_{ij}=\frac{(d_{ij})^{-1/(m-1)}}{\sum_{l=1}^{C}(d_{lj})^{-1/(m-1)}} \tag{30}$$

$$z_{ik}=\frac{\sum_{j=1}^{N}u_{ij}^{m}x_{jk}}{\sum_{j=1}^{N}u_{ij}^{m}} \tag{31}$$

其中，

$$d_{ij}=\sum_{k=1}^{D}\left(\frac{w_{ik}}{\sum_{l}^{D}w_{il}}\right)^{\beta}(x_{jk}-z_{ik})^{2} \tag{32}$$

与 PSOVW 相同，在 DESC 中权值矩阵 $\boldsymbol{W}$ 被看作是算法的解，运用 CoDE 的搜索策略进行更新.

式(30)～(32)既是式(22)～(24)在模糊聚类上的扩展，又是 FCM 在软子空间聚类上的扩展. 通常，模糊聚类描述了样本隶属于不同类别的不确定性，比硬聚类更能反映客观世界[16,18]. 然而，在高维稀疏数据集上，模糊聚类的运算时间随着维数的增加快速增加[19]. 此外，由于样本点对之间的距离差异小，一个样本到各个簇的隶属度趋同，使得算法容易陷入位于数据集中心的局部最优[20]. 因此，模糊聚类在高维数据集上不如其在低维数据集上有效. 另一方面，软子空间聚类不能直接运用在模糊聚类上. 这是因为，模糊聚类容易陷入位于数据集中心的局部最优，此时软子空间聚类算法会寻找整个数据集的相关属性而不是为每一个簇寻找与其相关的属性. 因此，结合模糊聚类和软子空间聚类的关键在于如何在算法陷入位于数据集中心的局部最优前为每一个簇寻找到足够小的子空间. 当前，硬聚类下的软子空间聚类取得了良好的效果[5-7,12]. 这是因为硬聚类能敏感地捕捉到样本点对的距离差异. 在高维稀疏数据集上，一个倾向于硬聚类的聚类算法往往比一个倾向于模糊聚类的聚类算法更有效[20]. 基于以上分析，如果在算法的初始阶段倾向于硬聚类，算法有可能快速得到较优的簇及其所在的子空间. 然后，随着不断迭代，算法逐步从硬聚类过渡到模糊聚类. 此时，模糊聚类将在一个较小的子空间中运算，有可能得到较好的聚类效果. 基于这一思想，借鉴改进的抑制式模糊 c-均值(Modified Suppressed Fuzzy c-means，MSFCM)[19]算法，我们提出了结合传统硬隶属度和模糊隶属度的新隶属度计算方法：

$$u_{ij}=\alpha(t)u_{fuzzy\_ij}+(1-\alpha(t)u_{crisp\_ij}) \tag{33}$$

$$u_{crisp\_ij}=\begin{cases}1, & i=\arg\min\limits_{q=1,\cdots,C}d_{qj}\\ 0, & \text{其它}\end{cases} \tag{34}$$

$$u_{fuzzy\_ij}=\frac{(d_{ij})^{-1/(m-1)}}{\sum_{i=1}^{C}(d_{ij})^{-1/(m-1)}} \tag{35}$$

$$d_{ij}=\sum_{k=1}^{D}\left(\frac{w_{ik}}{\sum_{k'}^{D}w_{ik'}}\right)^{\beta}(x_{jk}-z_{ik})^{2} \tag{36}$$

$$\alpha(t)=\left(\frac{t}{MaxIter}\right)^{\eta} \tag{37}$$

其中，t 是当前迭代次数，$MaxIter$ 是最大迭代次数，$\eta>0$ 是一个控制参数，用于控制算法从硬聚类过渡到模糊聚类的速度. 显然，对于式(33)有 $\sum_{i=1}^{C}u_{ij}=1$. 当 $t=0$ 时，有 $\alpha(t)=0$，此时式(33)等同于式(22). 当 $t=MaxIter$ 时，有 $\alpha(t)=1$，此时式(33)等同于式(30). 当 $0<\eta<1$ 时，u_{ij} 中模糊隶属度 u_{fuzzy_ij} 的权重将迅速增大，算法运行过程中更多地受到模糊聚类的影响；当 $\eta>1$ 时，u_{ij} 中模糊隶属度 u_{crisp_ij} 的权重增加缓慢，算法运行过程中更多地受到硬聚类的影响.

直接将算法分成硬聚类和模糊聚类两个阶段可以令算法更加简单. 例如，在算法的前 $\beta\times MaxIter$ 次迭代中令 $u_{ij}=u_{crisp\_ij}$，之后的迭代中令 $u_{ij}=u_{fuzzy\_ij}$，其中 $0<\beta<1$. 然而，通过实验我们发现，这种隶属度从硬聚类突变到模糊聚类的计算方法会影响算法的收敛，通过更为平稳的方式从硬聚类过渡到模糊聚类，令 $\boldsymbol{U}$ 和 $\boldsymbol{W}$ 在过渡中逐渐调整是一个不可缺少的过程.

式(33)中 u_{ij} 的定义与 MSFCM 中隶属度的定义相似. 两者都通过参数 α 建立了硬聚类和模糊聚类的联系. 两者的主要区别在于：式(33)中的 α 随算法的运行逐渐增大，算法在最后阶段完全过渡为模糊聚类；而 MSFCM 则始终处在硬聚类和模糊聚类之间. 这是因为本文的方法运用在子空间中，引入 α 是为了通过其变化实现算法从硬聚类到模糊聚类的过渡，提高模糊聚类在高维数据集上的性能. 而 MSFCM 始终在全空间中搜索，并未通过降低属性空间的大小避免 FCM 在高维空间的不足.

3.2　DESC 算法流程

DE 是 Storn 和 Price[14]提出的一种基于群体的全局搜索算法. 在每一次迭代中，DE 通过对个体间的差异进行加权重组实现种群的演化. 它包含变异(mutation)、交叉(crossover)和选择(selection) 3 个主要算子. DE 的性能很大程度上依赖于其试验向量的生成策略以及控制参数的选取[21]. 近年出现了大量关于试验向量生成策略和控制参数的研究. 各种生成策略和参数组合各有优势，适用于不同类型的问题. 为了充分运用这些研究成果，弥补单一参

数组合和生成策略不能全面覆盖各类问题的不足，Wang 等人[15]系统地综合了多种试验向量生成策略和参数组合，提出了复合差分演化算法(Composite Differential Evolution，CoDE)，实现了不同试验向量生成策略和参数组合的优势互补. 不同于其它 DE，在每一次迭代中，CoDE 通过随机选择的方式在控制参数池中选择参数值，并运用 3 种不同的生成策略为每一个个体生成 3 个试验向量. 因此，算法对不同类型的问题都有较高的适应性.

CoDE 是一种简单有效的全局搜索算法，因此本文运用 CoDE 求解软子空间聚类问题.

在运用 CoDE 求解软子空间聚类问题时，CoDE 中的一个个体对应于软子空间聚类问题的一个可行解. 与 PSOVW 一样，DESC 以权值矩阵 $\boldsymbol{W}$ 作为问题的解. 记 $P_g=\{\boldsymbol{x}_{1,g},\boldsymbol{x}_{2,g},\cdots,\boldsymbol{x}_{M,g}\}$ 为第 g 代种群，一个个体 $\boldsymbol{x}_{i,g}=(x_{i,g,1},\cdots,x_{i,g,d},\cdots,x_{i,g,C\times D})$ 是长度为 $C\times D$ 的向量，对应于一个权值矩阵 $\boldsymbol{W}$，其中每 D 维对应于一个簇中 D 个属性的权值. 同时，算法为每一个个体保留其对应的聚类中心矩阵 $\boldsymbol{Z}$ 以及划分矩阵 $\boldsymbol{U}$.

在算法的初始化阶段，为每一个个体在数据集中随机选择 C 个样本作为初始聚类中心，$\boldsymbol{W}=[w_{ik}]_{C\times D}=[1/D]_{C\times D}$. 然后运用式(33)和式(29)得到划分矩阵 $\boldsymbol{U}$ 以及个体的目标函数值.

种群 P_g 通过以下搜索策略进行演化：

对于每一个个体 $\boldsymbol{x}_{i,g}$，通过 3 种不同的生成策略，生成 3 个试验向量 $\boldsymbol{u}_{i\_1,g+1}$、$\boldsymbol{u}_{i\_2,g+1}$ 以及 $\boldsymbol{u}_{i\_3,g+1}$：

(1)“rand/1/bin”：

$$v_{i\_1,g+1,d}=x_{r1,g,d}+F\cdot(x_{r2,g,d}-x_{r3,g,d}),$$

$$u_{i\_1,g+1,d}=\begin{cases}v_{i\_1,g+1,d}, & rand\leq Cr \text{ 或 } d=d_{rand}\\ x_{i,g,d}, & \text{其它}\end{cases} \tag{38}$$

(2)“rand/2/bin”：

$$v_{i\_2,g+1,d}=x_{r1,g,d}+rand_1\cdot(x_{r2,g,d}-x_{r3,g,d})+F\cdot(x_{r4,g,d}-x_{r5,g,d})$$

$$u_{i\_2,g+1,d}=\begin{cases}v_{i\_2,g+1,d}, & rand\leq Cr \text{ 或 } d=d_{rand}\\ x_{i,g,d}, & \text{其它}\end{cases} \tag{39}$$

(3)“current-to-rand/1”：

$$u_{i\_3,g+1,d}=x_{i,g,d}+rand\cdot(x_{r1,g,d}-x_{i,g,d})+F\cdot(x_{r2,g,d}-x_{r3,g,d}) \tag{40}$$

其中，$rand$ 和 $rand_1$ 是[0,1]区间上均匀分布的随机数. d_{rand} 是[1,$C\times D$]上随机选择的自然数，以保证式(38)中 $\boldsymbol{u}_{i\_1,g+1}\neq\boldsymbol{x}_{i,g}$ 以及式(39)中 $\boldsymbol{u}_{i\_2,g+1}\neq\boldsymbol{x}_{i,g}$. F 是比例因子，Cr 是交叉概率. 这两个控制参数的取值从下列 3 种组合中随机选取：[$F=1.0$，$Cr=0.1$]，[$F=1.0$，$Cr=0.9$]以及[$F=0.8$，$Cr=0.2$].

得到 $\boldsymbol{u}_{i\_1,g+1}$、$\boldsymbol{u}_{i\_2,g+1}$ 以及 $\boldsymbol{u}_{i\_3,g+1}$ 后，令其对应的中心矩阵 $\boldsymbol{Z}$ 与 $\boldsymbol{x}_{i,g}$ 的中心矩阵相同. 然后，通过式(33)和式(31)更新 3 个试验向量的 $\boldsymbol{U}$ 和 $\boldsymbol{Z}$，并通过式(29)计算目标函数值. 选择 $\boldsymbol{u}_{i\_1,g+1}$、$\boldsymbol{u}_{i\_2,g+1}$、$\boldsymbol{u}_{i\_3,g+1}$ 以及 $\boldsymbol{x}_{i,g}$ 中目标函数值最小的一个作为 $\boldsymbol{x}_{i,g+1}$，添加到下一代种群 P_{g+1} 中.

重复上述演化步骤直到算法满足终止条件.

DESC 的算法流程如下.

算法 2. DESC.

输入：种群规模 M；最大评价次数 $MaxFES$；试验向量生成策略池“rand/1/bin”，“rand/2/bin”，“current-to-rand/1”；控制参数池[$F=1.0$，$Cr=0.1$]，[$F=1.0$，$Cr=0.9$]以及[$F=0.8$，$Cr=0.2$]；聚类数 C；样本维数 D

输出：最优解对应的 $\boldsymbol{U}$，$\boldsymbol{Z}$，$\boldsymbol{W}$

1. $g=0$；
2. 生成初始种群 P_0，对其中的每一个个体 $\boldsymbol{x}_{i,0}$，在数据集中随机选择 C 个样本作为初始聚类中心，令 $\boldsymbol{W}=[w_{ik}]_{C\times D}=[1/D]_{C\times D}$. 运用式(33)和式(29)得到划分矩阵 $\boldsymbol{U}$ 以及个体的目标函数值；
3. $FES=M$；
4. REPEAT
5. $P_{g+1}=\varnothing$；
6. FOR 每一个个体 $\boldsymbol{x}_{i,g}$ DO
7. 对试验向量生成策略池中的每一种策略，从控制参数池中随机选择一种参数组合，生成试验向量. 记 3 种策略生成的试验向量分别为 $\boldsymbol{u}_{i\_1,g+1}$、$\boldsymbol{u}_{i\_2,g+1}$ 以及 $\boldsymbol{u}_{i\_3,g+1}$；
8. 令 $\boldsymbol{u}_{i\_1,g+1}$、$\boldsymbol{u}_{i\_2,g+1}$ 以及 $\boldsymbol{u}_{i\_3,g+1}$ 对应的中心矩阵 $\boldsymbol{Z}$ 与 $\boldsymbol{x}_{i,g}$ 的中心矩阵相同；
9. 运用式(33)计算 3 个试验向量的划分矩阵 $\boldsymbol{U}$；
10. 运用式(31)计算 3 个试验向量的中心矩阵 $\boldsymbol{Z}$；
11. 运用式(29)计算 3 个试验向量的目标函数值；
12. 选择 $\boldsymbol{u}_{i\_1,g+1}$、$\boldsymbol{u}_{i\_2,g+1}$、$\boldsymbol{u}_{i\_3,g+1}$ 以及 $\boldsymbol{x}_{i,g}$ 中目标函数值最小的一个作为 $\boldsymbol{x}_{i,g+1}$；
13. $P_{g+1}=P_{g+1}\cup\boldsymbol{x}_{i,g+1}$；
14. $FES=FES+3$；
15. END FOR
16. $g=g+1$；
17. UNTIL $FES\geq MaxFES$.

3.3 算法复杂度

在 DESC 中，$\boldsymbol{Z}$ 和 $\boldsymbol{W}$ 是 $C\times D$ 的矩阵，$\boldsymbol{U}$ 是 $C\times N$ 的矩阵. 对于每一个个体，生成试验向量(步 7)的时间复杂度为 $O(CD)$. 之后，对所有样本进行重新划分(步 9)，时间复杂度为 $O(NCD)$. 为每一个个体重新计算聚类中心(步 10)的时间复杂度为 $O(CD)$. 因此，对每一个个体而言，算法的主循环(步 7～14)的时间复杂度是 $O(NCD)$，其中 N 是样本总数. 假设算法收敛需要的迭代次数为 T，种群规

模为 M，则 DESC 的时间复杂度为 $O(MNCDT)$. 可见，DESC 的时间复杂度随数据集的维数、聚类数以及样本总数线性增加.

3.4 DESC 的优势

DESC 是一种运用全局搜索策略的软子空间聚类算法. 与 KM 型软子空间聚类算法相比，拥有不依赖于初始解以及不易陷入局部最优的优点. 与 PSOVW 相比，DESC 基于模糊聚类对其进行了扩展. 在现实生活中，不同簇之间往往并没有明确的分界. 模糊聚类能很好地反映这一现实，因而模糊聚类往往能获得比硬聚类更好的效果. 此外，CLPSO 需要更多的控制参数以及维护额外的个体速度和个体历史最优解. 因而 CoDE 比 CLPSO 更简单，DESC 也因此比 PSOVW 更易于实现.

4 实　验

为了验证 DESC 的算法性能，我们将对 DESC 和 PSOVW[12]、KM[11]、MSFCM[19]、RKM[7]、EWKM[5] 以及 ESSC[6] 进行比较. 同时，为了验证搜索策略、目标函数以及新隶属度对算法的影响，我们对搜索策略、目标函数以及新隶属度进行了 4 种组合，如表 1 所示.

表 1　搜索策略、目标函数以及新隶属度的 4 种组合

算法	搜索策略	目标函数	更新方程
DESC-crisp	CoDE	式(21)	式(22)
DESC-fuzzy	CoDE	式(29)	式(30)
PSOVW-fuzzy	CLPSO	式(29)	式(30)
PSOVW-NF	CLPSO	式(29)	式(33)

由于 DESC 和 PSOVW 是基于群体的算法，而 EWKM 等是单点搜索算法，本文采用最大评价次数 $MaxFES$ 作为所有测试算法的终止条件. 令 $MaxFES$ 为 500，所有基于群体算法的种群规模为 20. 其余参数设置如表 2 所示.

表 2　参数设置

算法	参数设置
DESC DESC-fuzzy PSOVW-fuzzy PSOVW-NF	$m=2$ $\beta=2$ $\eta=0,0.2,0.4,0.6,0.8,1,2,3,4,5,6,7,8,9,10$
PSOVW DESC-crisp	$\beta=8$
MSFCM	$m=2$
RKM	$\alpha=3$
EWKM	$\gamma=0.5$
ESSC	$\gamma=1.0$ $\eta=0.1$ $m=\frac{\min(N,D-1)}{\min(N,D-1)-2}$

测试环境为 2.66 GHz CPU，4 GB 内存，所有算法在 WEKA 平台[22] 下开发.

4.1 性能指标

本文采用两种性能指标对聚类效果进行评价：*Rand Index* (RI)[23] 和 *Normalized Mutual Information* (NMI)[24]，其定义如下：

$$RI=\frac{f_{00}+f_{11}}{N(N-1)/2} \tag{41}$$

$$NMI=\frac{\sum_{i=1}^{K}\sum_{j=1}^{C}n_{ij}\log\frac{N\cdot n_{ij}}{n_i\cdot n_j}}{\sqrt{\left(\sum_{i=1}^{K}n_i\log\frac{n_i}{N}\right)\left(\sum_{j=1}^{C}n_j\log\frac{n_j}{N}\right)}} \tag{42}$$

其中，K 是类别数，C 是聚类数，N 是样本总数. f_{00} 是不属于同一个类并被分配到不同簇的样本点对的数量，f_{11} 是属于同一个类并被分配到同一个簇的样本点对的数量. n_i 是属于类 i 的样本数，n_j 是属于簇 j 的样本数，n_{ij} 是属于类 i 并被分配到簇 j 的样本数. 两个性能指标都是越大越好.

4.2 实验结果

本文采用来自 UCI①、生物医学②以及 NIPS2003 特征选择挑战赛③的 13 个数据集对各个算法进行测试. UCI 数据集用于考察算法对低维数据集的兼容性. 生物医学以及 NIPS2003 特征选择挑战赛的数据集用于检验算法在高维空间的聚类性能. 数据集的详细信息见表 3. 每一个算法独立运行 30 次，实验结果如表 4～8 及图 1、图 2 所示. 每一个数据集的最好结果在表中加粗表示.

表 3　数据集详细信息

数据集	样本数	维数	类别数
Iris	150	4	3
KDD synthetic control	600	60	6
Segment	2310	19	7
Sonar	208	60	2
Vehicle	846	18	4
Leukemia-ALLAML	72	7129	2
Lung harvard	203	12600	5
Leukemia-MLL	72	12582	3
Lung michigan	96	7129	2
Prostate tumor vs normal train	102	12600	2
Breast cancer	97	24481	2
Leukemia stjude	327	12558	7
Arcene	200	10000	2

依据表 4～7，我们有以下结论. 首先，对比 DESC-crisp 和 DESC-fuzzy 以及 PSOVW 和 PSOVW-fuzzy 可以看出，直接将软子空间聚类从硬聚类扩展到模糊聚类的效果并不理想. 在低维数据集中，这种

① http://www.cs.waikato.ac.nz/~ml/weka/index.html/
② http://datam.i2r.a-star.edu.sg/datasets/krbd/
③ http://www.nipsfsc.ecs.soton.ac.uk/datasets/

表 4 DESC、PSOVW 以及 4 种不同组合算法 30 次运行的实验结果(*RI*)

数据集		PSOVW	PSOVW-fuzzy	PSOVW-NF	DESC-crisp	DESC-fuzzy	DESC
Iris	Mean	0.8571	0.8747	0.9036	0.8909	0.9304	**0.9423**
	Std.	0.0634	0.0443	0.0380	0.0411	0.0213	0.0180
KDD synthetic control	Mean	0.8287	0.8181	0.8765	0.8241	0.8499	**0.8853**
	Std.	0.0413	0.0200	0.0175	0.0425	0.0129	0.0159
Segment	Mean	0.8030	0.8186	**0.8617**	0.7840	0.8562	0.8563
	Std.	0.0480	0.0343	0.0230	0.0704	0.0144	0.0158
Sonar	Mean	0.5038	0.5036	0.5055	0.5010	0.5054	**0.5075**
	Std.	0.0098	0.0103	0.0079	0.0051	0.0086	0.0115
Vehicle	Mean	0.5917	0.6121	0.6462	0.5807	0.6451	**0.6476**
	Std.	0.0420	0.0313	0.0110	0.0480	0.0122	0.0162
Leukemia ALLAML	Mean	0.5776	0.5107	0.5618	0.5671	0.5110	**0.5888**
	Std.	0.0375	0.0155	0.0344	0.0421	0.0206	0.0347
Lung harvard	Mean	0.5798	0.5063	0.5344	0.5641	0.5008	**0.5724**
	Std.	0.0145	0.0071	0.0157	0.0300	0.0056	0.0234
Leukemia MLL	Mean	0.7326	0.5390	**0.7622**	0.7208	0.7164	0.7534
	Std.	0.0144	0.0610	0.0168	0.0722	0.0586	0.0173
Lung michigan	Mean	**0.6579**	0.5341	0.5154	0.6534	0.5091	0.5476
	Std.	0.1011	0.0362	0.0265	0.1093	0.0207	0.0787
Prostate tumor vs normal train	Mean	0.5229	0.5242	0.5230	0.5211	0.5241	**0.5243**
	Std.	0.0008	0.0048	0.0000	0.0045	0.0038	0.0022
Breast cancer	Mean	0.4974	0.4983	0.5245	0.5091	0.5024	**0.5278**
	Std.	0.0000	0.0045	0.0240	0.0172	0.0108	0.0148
Leukemia stjude	Mean	0.6947	0.6686	**0.7530**	0.6827	0.7087	0.7404
	Std.	0.0557	0.0391	0.0276	0.0712	0.0127	0.0466
Arcene	Mean	0.5410	0.5431	0.5438	0.5329	0.5455	**0.5455**
	Std.	0.0056	0.0022	0.0027	0.0130	0.0052	0.0034

表 5 DESC、KM、MSFCM、RKM、EWKM 以及 ESSC 30 次运行的实验结果(*RI*)

数据集		DESC	KM	MSFCM	RKM	EWKM	ESSC
Iris	Mean	**0.9423**	0.8720	0.8421	0.8606	0.8767	0.8709
	Std.	0.0180	0.0027	0.0643	0.0612	0.0964	0.0459
KDD synthetic control	Mean	**0.8853**	0.8688	0.8809	0.8695	0.8290	0.8523
	Std.	0.0159	0.0274	0.0285	0.0158	0.0622	0.0166
Segment	Mean	0.8563	0.8592	**0.8632**	0.8605	0.4112	0.7833
	Std.	0.0158	0.0151	0.0152	0.0193	0.1806	0.0750
Sonar	Mean	0.5075	0.5022	0.5020	0.5029	0.5007	0.4989
	Std.	0.0115	0.0023	0.0029	0.0030	0.0073	0.0011
Vehicle	Mean	0.6476	0.6507	**0.6515**	0.6483	0.3886	0.5823
	Std.	0.0162	0.0074	0.0035	0.0092	0.1242	0.0828
Leukemia ALLAML	Mean	**0.5888**	0.5556	0.5067	0.5718	0.5235	0.5433
	Std.	0.0347	0.0456	0.0226	0.0325	0.0274	0.0500
Lung harvard	Mean	0.5724	0.5682	0.5077	0.5633	0.5742	**0.5764**
	Std.	0.0234	0.0236	0.0125	0.0233	0.0396	0.0289
Leukemia MLL	Mean	**0.7534**	0.7040	0.7285	0.7241	0.6196	0.7255
	Std.	0.0173	0.0735	0.0914	0.0536	0.1107	0.0576
Lung michigan	Mean	0.5476	0.6605	0.5684	0.6147	0.7051	**0.7065**
	Std.	0.0787	0.1068	0.0812	0.1141	0.1060	0.0965
Prostate tumor vs normal train	Mean	**0.5243**	0.5201	0.5223	0.5209	0.5193	0.5177
	Std.	0.0022	0.0070	0.0017	0.0051	0.0077	0.0023
Breast cancer	Mean	**0.5278**	0.5117	0.5016	0.4981	0.4978	0.4980
	Std.	0.0148	0.0138	0.0077	0.0009	0.0010	0.0008
Leukemia stjude	Mean	**0.7404**	0.6698	0.6649	0.6814	0.3212	0.2068
	Std.	0.0466	0.0717	0.1061	0.0747	0.1669	0.0040
Arcene	Mean	**0.5455**	0.5359	0.5359	0.5367	0.5338	0.5353
	Std.	0.0034	0.0103	0.0113	0.0129	0.0178	0.0181

表 6　DESC、PSOVW 以及 4 种不同组合算法 30 次运行的实验结果（*NMI*）

数据集		PSOVW	PSOVW-fuzzy	PSOVW-NF	DESC-crisp	DESC-fuzzy	DESC
Iris	Mean	0.7448	0.7450	0.7881	0.7828	0.8332	**0.8529**
	Std.	0.0803	0.0721	0.0660	0.0563	0.0356	0.0319
KDD synthetic control	Mean	0.6618	0.5776	0.7056	0.6390	0.6426	**0.7406**
	Std.	0.0542	0.0322	0.0379	0.0619	0.0321	0.0259
Segment	Mean	0.5431	0.5361	**0.6188**	0.5248	0.5960	0.6129
	Std.	0.0599	0.0541	0.0367	0.0656	0.0251	0.0338
Sonar	Mean	**0.0217**	0.0151	0.0130	0.0210	0.0133	0.0162
	Std.	0.0257	0.0184	0.0124	0.0228	0.0137	0.0190
Vehicle	Mean	0.1279	0.1276	0.1106	0.1367	0.1151	**0.1382**
	Std.	0.0322	0.0232	0.0222	0.0301	0.0177	0.0306
Leukemia ALLAML	Mean	0.1118	0.0127	0.1022	0.1051	0.0233	**0.1290**
	Std.	0.0459	0.0185	0.0491	0.0542	0.0256	0.0403
Lung harvard	Mean	0.3104	0.1405	0.2043	0.2631	0.1518	**0.2840**
	Std.	0.0364	0.0178	0.0422	0.0730	0.0129	0.0536
Leukemia MLL	Mean	0.4387	0.1040	0.4651	0.4470	0.3738	**0.4674**
	Std.	0.0290	0.0743	0.0399	0.0976	0.1286	0.0354
Lung michigan	Mean	0.0785	0.0328	0.1068	0.1278	0.0469	**0.1440**
	Std.	0.1765	0.0369	0.0585	0.1534	0.0494	0.0955
Prostate tumor vs normal train	Mean	0.0667	**0.0681**	0.0670	0.0628	0.0627	0.0677
	Std.	0.0016	0.0119	0.0000	0.0095	0.0107	0.0028
Breast cancer	Mean	0.0389	0.0079	0.0439	0.0354	0.0122	**0.0518**
	Std.	0.0000	0.0101	0.0356	0.0228	0.0162	0.0250
Leukemia stjude	Mean	0.2786	0.0543	0.2425	0.2418	0.0586	**0.3411**
	Std.	0.0794	0.0100	0.0876	0.0788	0.0137	0.0806
Arcene	Mean	0.0791	0.0813	0.0818	0.0611	0.0836	**0.0840**
	Std.	0.0122	0.0038	0.0039	0.0296	0.0070	0.0045

表 7　DESC、KM、MSFCM、RKM、EWKM 以及 ESSC 30 次运行的实验结果（*NMI*）

数据集		DESC	KM	MSFCM	RKM	EWKM	ESSC
Iris	Mean	**0.8529**	0.7337	0.7100	0.7286	0.7828	0.7604
	Std.	0.0319	0.0128	0.0649	0.0669	0.1172	0.0527
KDD synthetic control	Mean	**0.7406**	0.7377	0.6979	0.7360	0.6698	0.7016
	Std.	0.0259	0.0338	0.0684	0.0287	0.1032	0.0339
Segment	Mean	**0.6129**	0.6105	0.5966	0.6032	0.2013	0.5345
	Std.	0.0338	0.0215	0.0126	0.0160	0.1518	0.0878
Sonar	Mean	0.0162	0.0067	0.0061	0.0075	0.0422	**0.0426**
	Std.	0.0190	0.0044	0.0053	0.0048	0.0267	0.0257
Vehicle	Mean	0.1382	0.1147	0.0995	0.1310	0.1039	**0.1504**
	Std.	0.0306	0.0244	0.0100	0.0344	0.0528	0.0376
Leukemia ALLAML	Mean	**0.1290**	0.0998	0.0251	0.0965	0.0457	0.0817
	Std.	0.0403	0.0523	0.0343	0.0492	0.0408	0.0755
Lung harvard	Mean	0.2840	0.2730	0.1684	0.2696	0.2715	**0.2974**
	Std.	0.0536	0.0690	0.0293	0.0699	0.0835	0.0776
Leukemia MLL	Mean	0.4674	0.4379	0.4466	0.4520	0.3093	**0.4823**
	Std.	0.0354	0.0892	0.1232	0.1125	0.1637	0.1072
Lung michigan	Mean	**0.1440**	0.0668	0.0144	0.1283	0.1351	0.1259
	Std.	0.0955	0.0700	0.0157	0.1360	0.2007	0.2127
Prostate tumor vs normal train	Mean	**0.0677**	0.0628	0.0656	0.0636	0.0626	0.0579
	Std.	0.0028	0.0084	0.0033	0.0065	0.0091	0.0008
Breast cancer	Mean	**0.0518**	0.0373	0.0115	0.0477	0.0437	0.0464
	Std.	0.0250	0.0181	0.0126	0.0103	0.0125	0.0100
Leukemia stjude	Mean	**0.3411**	0.2843	0.2519	0.2962	0.0912	0.0632
	Std.	0.0806	0.1152	0.0997	0.1109	0.0623	0.0089
Arcene	Mean	**0.0840**	0.0680	0.0679	0.0681	0.0626	0.0682
	Std.	0.0045	0.0241	0.0259	0.0287	0.0324	0.0317

表 8 DESC、DESC-fuzzy、PSOVW、KM、MSFCM、RKM、EWKM 以及 ESSC 30 次运行的平均运行时间 (单位:s)

数据集	DESC	DESC-fuzzy	PSOVW	KM	MSFCM	RKM	EWKM	ESSC
Iris	0.0660	0.0993	0.0147	0.0107	0.0602	0.0258	0.0431	0.8014
KDD synthetic control	3.6940	4.2713	0.3627	0.2737	3.8323	0.8559	2.3594	73.2477
Segment	4.9820	7.7673	0.5503	0.4527	6.1503	1.8712	3.5180	46.2487
Sonar	0.3007	0.5073	0.0727	0.0540	0.4557	0.1791	0.6073	8.1587
Vehicle	0.9793	1.6360	0.1460	0.1230	1.2397	0.4973	0.9590	21.0380
Leukemia-ALLAML	13.8033	19.4700	7.4383	2.6217	17.3543	9.7597	26.8448	444.8639
Lung harvard	205.1150	232.0363	43.1253	21.8050	215.2240	79.3096	168.5557	5222.9100
Leukemia-MLL	53.1457	49.7217	22.0410	5.8207	46.3545	18.7152	51.6876	1203.9746
Lung michigan	25.7577	24.8160	8.0867	3.8543	23.0503	13.1464	35.6785	453.6368
Prostate tumor vs normal train	46.5177	46.7740	16.4073	8.4883	43.2416	28.8430	67.9280	742.2579
Breast cancer	58.0587	86.4857	34.4033	16.3483	49.9300	45.8059	125.9387	1304.3037
Leukemia stjude	308.5667	510.1517	74.3323	42.4197	498.2954	142.1722	345.8762	2115.6487
Arcene	59.4197	72.9287	16.5643	13.8207	67.1170	42.2878	107.1521	1059.6873

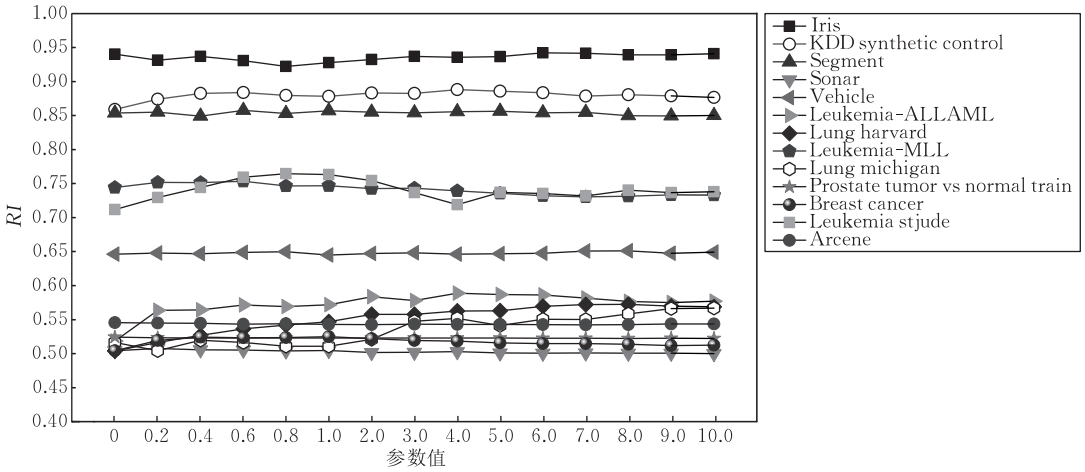


图 1 DESC 在不同的 η 取值下 *RI* 评价值

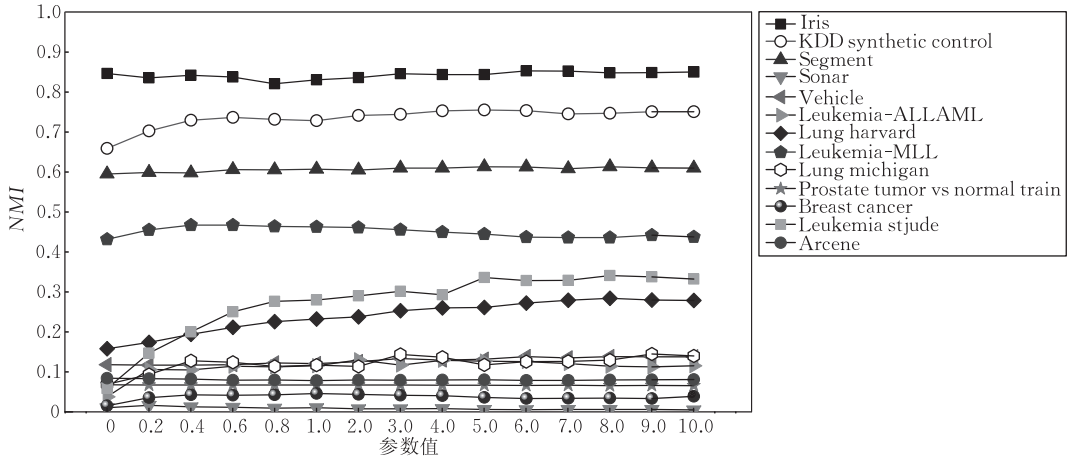


图 2 DESC 在不同的 η 取值下 *NMI* 评价值

扩展确实提高了算法的效果.但是,在高维空间中 DESC-fuzzy 和 PSOVW-fuzzy 都不如其硬聚类版本理想.在 *NMI* 指标下这种差异尤为明显.这是因为 FCM 在高维数据集上存在不足[20],DESC-fuzzy 和 PSOVW-fuzzy 通过式(30)更新隶属度,保留了 FCM 的这种不足.

其次,对比 DESC 和 DESC-fuzzy 以及 PSOVW-NF 和 PSOVW-fuzzy,可以看到新的隶属度计算方法能有效提高算法效果.在算法的初始阶段倾向于硬聚类,有利于算法快速找到较优的簇及其子空间.然后随着不断迭代,算法逐步从硬聚类过渡到模糊聚类.此时,模糊聚类将在一个较小的子空间中运算,能充分发挥模糊聚类的性能,避免其在高维空间中的不足.在这里,算法的初始阶段相当于为后续的

模糊聚类寻找合适的初始解. 而算法的后续阶段则是在这个基础上进一步寻优.

再次，对比基于 CoDE 的算法以及基于 CLPSO 的算法，前者效果优于后者.

最后，无论是 RI 还是 NMI，DESC 是所有算法中效果最好的算法. 无论是搜索策略、新的目标函数还是新的隶属度计算方法，都有效地提高了软子空间聚类算法的效果.

表 8 是 DESC、DESC-fuzzy、PSOVW、KM、MS-FCM、RKM、EWKM 以及 ESSC 的运行时间. 比较 DESC 和 DESC-fuzzy，可以看到新的隶属度算法并不会对算法的运行时间带来明显影响.

同样是基于群体的全局搜索算法，DESC 的运行时间是 PSOVW 的 2～7 倍，极个别达到 10 倍. 这是因为 DESC 需要更多的时间计算模糊隶属度. 波动较大是因为在 PSOVW 的运行过程中，由于其搜索策略自身的特点，有可能产生不满足 $0\leq w_{ik}\leq 1$ 的 $\boldsymbol{W}$. 此时，算法将不对该粒子进行评价[12]，导致算法终止时实际的迭代次数高于 DESC. 这种现象会随着数据集维数增加而加剧，因而在高维数据集上 PSOVW 与 DESC 的时间差异小于低维数据集上两者的时间差异.

在高维数据集上，DESC 的运行时间低于 EWKM. 这是因为虽然 DESC 需要额外的时间来维护自身种群，但其权值的更新方式明显比 EWKM 简单. 在高维数据集上，这种简单的更新方式有效地减少了算法的运行时间，弥补了两者运行时间上的差距.

RKM 的运行时间比 EWKM 以及 ESSC 短. 这同样是因为在 RKM 中，权值的更新方式比 EWKM 以及 ESSC 简单.

KM 作为最简单的算法，运行时间最短. MSFCM 虽然也不需要计算权值，但是其模糊隶属度的计算大大增加了其运算时间. MSFCM 的运行时间是 KM 的 7.5 倍，与 DESC 和 PSOVW 的差异相当，说明 DESC 的时间开销比 PSOVW 大的主要原因是模糊聚类和硬聚类的差异而不是 CoDE 和 CLPSO 的差异.

ESSC 的运行时间过长，一方面是因为 ESSC 的更新公式较其它对比算法复杂. 另一方面是因为 ESSC 在运行过程中经常出现某些簇样本数量为零，以致需要运行额外的修补程序，增加了运行时间.

图 1、图 2 是 DESC 在不同参数取值下的实验结果. 实验表明，在 η 取值在[0.4，10]时，DESC 有较稳定的聚类效果. 区间范围较广，意味着 DESC 对参数的设置并不敏感.

5　结　论

本文提出了一种基于差分演化算法的软子空间聚类算法 DESC. 在 DESC 中，我们首先设计了一个结合模糊加权类内相似性和界约束权值矩阵的新目标函数. 然后，通过结合模糊隶属度和硬隶属度，提出了新的隶属度计算方法. 最后，引入了复合差分演化算法，并运用复合差分演化算法优化新目标函数和搜索子空间中的聚类. 这是第一个基于 DE 的软子空间聚类算法. 实验表明，新目标函数和复合差分演化算法的引入有效地提高了软子空间聚类算法的性能. 新算法较已有软子空间聚类算法有明显优势.

后续的工作是将 DESC 运用在如文本聚类、图像分割等现实问题中. 此外，参数的自适应也是一个研究的重点.

参　考　文　献

[1] Kriegel H-P, Kröger P, Zimek A. Clustering high dimensional data: A survey on subspace clustering, pattern based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from Data, 2009, 3(1): 1-58

[2] Moise G, Zimek A, Kröger P, Kriegel H-P, Sander J. Subspace and projected clustering: Experimental evaluation and analysis. Knowledge and Information Systems, 2009, 21(3): 299-326

[3] Huang J Z, Ng M K, Rong H, Li Z. Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 657-668

[4] Chen Li-Fei, Guo Gong-De, Jiang Qing-Shan. Adaptive algorithm for soft subspace clustering. Journal of Software, 2010, 21(10): 2513-2523 (in Chinese)
(陈黎飞，郭躬德，姜青山. 自适应的软子空间聚类算法. 软件学报，2010，21(10)：2513-2523)

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

[6] Deng Z, Choi K-S, Chung F-L, Wang S. Enhanced soft subspace clustering integrating within-cluster and between-cluster information. Pattern Recognition, 2010, 43(3): 767-781

[7] Boongoen T, Shang C, Iam-On N, Shen Q. Extending data reliability measure to a filter approach for soft subspace clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2011, 41(6): 1705-1714

[8] Gan G, Wu J, Yang Z. A fuzzy subspace algorithm for clustering high dimensional data//Li Xue, Osmar Zaïane, Li Z-H eds. Advanced Data Mining and Applications, Series

Lecture Notes in Computer Science. Germany: Springer Berlin/Heidelberg, 2006, 4093: 271-278

[9] Friedman J H, Meulman J J. Clustering objects on subsets of attributes. Journal of the Royal Statistical Society, 2004, 66: 815-849

[10] Jing L, Ng M, Xu J, Huang J. Subspace clustering of text documents with feature weighting k-means algorithm// Ho T, Cheung D, Liu H. Advances in Knowledge Discovery and Data Mining, Series. Lecture Notes in Computer Science. Germany: Springer Berlin/Heidelberg, 2005, 3518: 802-812

[11] MacQueen J B. Some methods for classification and analysis of multivariate observations//Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, USA, 1967: 281-297

[12] Lu Y, Wang S, Li S, Zhou C. Particle swarm optimizer for variable weighting in clustering high-dimensional data. Machine Learning, 2011, 82(1): 43-70

[13] Liang J J, Qin A K, Suganthan P N, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295

[14] Storn R, Price K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 1997, 11(4): 341-359

[15] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 55-66

[16] Bezdek J C. Pattern Recognition with Fuzzy Objective Function Algorithms. USA: Kluwer Academic Publishers, 1981

[17] Yu J, Cheng Q, Huang H. Analysis of the weighting exponent in the FCM. IEEE Transactions on Systems, Man, and Cybernetics, 2004, 34(1): 634-639

[18] Setnes M, Kaymak U. Fuzzy modeling of client preference from large data sets: An application to target selection in direct marketing. IEEE Transactions on Fuzzy Systems, 2001, 9(1): 153-163

[19] Hung W-L, Yang M-S, Chen D-H. Parameter selection for suppressed fuzzy c-means with an application to mri segmentation. Pattern Recognition Letters, 2006, 27(5): 424-438

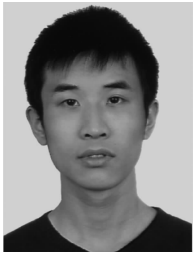
[20] Winkler R, Klawonn F, Kruse R. Fuzzy c-means in high dimensional spaces. International Journal of Fuzzy System Applications, 2011, 1(1): 1-16

[21] Das S, Suganthan P N. Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31

[22] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I H. The WEKA data mining software: An update. SIGKDD Explorations Newsletter, 2009, 11(1): 10-18

[23] Rand W M. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 1971, 66(336): 846-850

[24] Strehl A, Ghosh J. Cluster ensembles: A knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 2003, 3(3/1): 583-617



BI Zhi-Sheng, born in 1983, Ph. D. candidate. His main research interests include computational intelligence and data mining.

WANG Jia-Hai, born in 1977, Ph. D., associate professor. His main research interests include computational intelligence and data mining.

YIN Jian, born in 1968, Ph. D., professor, Ph. D. supervisor. His research interests include machine learning and data mining.

Background

Clustering analysis is a data analysis technique with numerous applications in machine learning, statistics, and pattern recognition fields in the past decades. These applications include gene expression analysis, metabolic screening, customer recommendation systems and text analysis, et al. In high-dimensional sparse data space, conventional clustering algorithms usually fail to find useful clusters. A solution for this problem is subspace clustering, which seeks to search clusters from the subspaces of the data instead of the entire data space. Recently, much effort has been made to improve the performance of subspace clustering. Most of them focus on designing an objective function and then optimizing it just by a local search strategy as in k-means algorithm.

In PSOVW, Lu et al. pointed out that the performance of soft subspace clustering largely depends on the objective function and the search strategy. Go further along with this study, this paper presents a differential evolution based algorithm for subspace clustering, called DESC. In the proposed algorithm, a novel objective function is firstly designed by considering the fuzzy weighting within-cluster compactness and loosening the constraints of dimension weight matrix. Then, a novel membership between a data point and a cluster is proposed. At last, an efficient global search strategy, composite DE, is introduced to optimize the proposed objective function to search subspace clusters. This is the first subspace clustering algorithm based on DE. The simulation results show that both the proposed objective function and the introduced DE search strategy contribute to the performance enhancement of soft subspace clustering, and thus DESC is significantly better than existing algorithms.

This work is supported by the National Natural Science Foundation of China (No. 60805026, No. 61070076, No. 61033010), Natural Science Foundation of Guangdong Province (No. S2011020001182), Research Foundation of Science and Technology Plan Project in Guangdong Province (No. 2009B090300450, No. 2010A040303004, No. 2011B040200007), and the Zhujiang New Star of Science and Technology in Guangzhou City (No. 2011J2200093).