ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2010/04(11)-1019-08
DOI: 10.3778/j.issn.1673-9418.2010.11.007

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-51616056

k-means 型软子空间聚类算法*

张燕萍，姜青山+
厦门大学 软件学院，福建 厦门 361005

A *k*-means-based Algorithm for Soft Subspace Clustering*

ZHANG Yanping, JIANG Qingshan+
School of Software, Xiamen University, Xiamen, Fujian 361005, China
+ Corresponding author: E-mail: qjiang@xmu.edu.cn

ZHANG Yanping, JIANG Qingshan. A *k*-means-based algorithm for soft subspace clustering. Journal of Frontiers of Computer Science and Technology, 2010, 4(11): 1019-1026.

Abstract: Soft subspace clustering is an important part and research hotspot in clustering research. Clustering in high dimensional space is especially difficult due to the sparse distribution of the data and the curse of dimensionality. By analyzing limitations of the existing algorithms, the concept of subspace difference is proposed. Based on these, a new objective function is given by taking into account the compactness of the subspace clusters and subspace difference of the clusters. And a subspace clustering algorithm based on *k*-means is presented. The additional parameter is not necessary in the novel algorithm. Theoretical analysis and experimental results demonstrate that the proposed algorithm significantly improves the accuracy.
Key words: high dimensional data; *k*-means; subspace clustering; subspace difference

摘 要：软子空间聚类是聚类研究领域的一个重要分支和研究热点。高维空间聚类以数据分布稀疏和“维度效应”现象等问题而成为难点。在分析现有软子空间聚类算法不足的基础上，引入子空间差异的概念；在此基础上，结合簇内紧凑度的信息来设计新的目标优化函数；提出了一种新的 *k*-means 型软子空间聚类算法，该算法在聚类过程中无需设置额外的参数。理论分析与实验结果表明，相对于其他的软子空间算法，该算法具有更好的聚类精度。

*The National Natural Science Foundation of China under Grant No.10771176 (国家自然科学基金).
Received 2010-05, Accepted 2010-07.

关键词: 高维数据; k 均值; 软子空间算法; 子空间差异
文献标识码：A 中图分类号：TP311

1 引言

在面向数据挖掘的聚类研究中, 所要处理的数据经常会有几十甚至几百个维度(也称作特征或属性)。将这些数据对象表示成高维空间中的点或者向量, 就可以把现实应用中的对象集用高维空间中点的集合来表示。典型的高维数据包括零售交易数据、文档数据、空间数据、序列数据、基因数据等。传统的算法在聚类如此高维的数据时性能大大降低, 这也成为目前数据挖掘研究的挑战性课题[1]。

高维数据与低维数据在许多方面表现出不同的特性, 如稀疏性以及“维度效应”现象[1]等。为了处理这些问题, 通常采用降维的方法, 而在实际的应用中, 一些不同的簇类往往与不同的属性子空间相关联。子空间聚类算法[2]的提出在一定程度上解决了这一问题。根据加权方式的不同, 现有子空间算法可分为硬子空间(hard subspace)聚类和软子空间(soft subspace)聚类。其中, 软子空间聚类通过赋予每聚类一个最优权值向量([0,1]区间)来识别聚类。现有的算法需要用户提供专门的领域知识来设置它们的输入参数, 比如FWKM[3]的 β 和 δ、LAC[4]的 h、FSC[5]的 α 和 ε 以及 EWKM[6]的 γ 等, 这些参数都被定义为全局的(不同的簇类使用统一的加权参数), 降低了算法的适用性。且现有软子空间算法多数关注的是数据集划分的优化, 而忽略了子空间的优化, 这是不完备的[7]。

本文提出一种基于子空间差异的软子空间算法(soft-subspace clustering based on subspace difference, SDSC)。除簇类数目 K 外, SDSC 不需要用户指定额外的参数值, 在类 k-means[7]的算法流程中, 实现了数据集和子空间两方面的优化。实验表明, 该算法有更好的聚类精度和适应性。

本文组织如下：第 2 章简要介绍软子空间算法的相关工作; 第3章给出具体的算法描述; 第4章描述实验方法, 并对算法的实验结果进行比较和分析; 第 5 章总结全文, 并指出下一步的研究方向。

2 相关工作

子空间聚类算法有两个主要任务：寻找存在簇的子空间和存在于子空间中的簇。相对于硬子空间而言, 软子空间对维度的处理具有更好的灵活性与伸缩性。因此, 软子空间聚类算法越来越为人们所关注, 成为子空间聚类算法研究的热点问题[8]。为方便阐述, 在此给出与软子空间聚类相关的一些符号表示。

给定数据集 $DB=\{X_1,X_2,...,X_N\}$, 其中 $X_i=<x_{i1},x_{i2},...,x_{iD}>$, 称 X_i 为D(D>1)维数 据空间的第 i 个数据点 i=1,2,…,N, 这里 N(N>1)表示数据点数目, 并假设数据已作规范化处理使得所有的 $x_{ij}\in[0,1]$, j=1,2,…,D。聚类算法将 DB 划分为 K 个子集的集合 $C=\{C_1,C_2,...,C_k\}$, 且 $\forall k\neq l,1\leq k,\ l\leq K$, $C_k\cap C_l=\varnothing$, C_k 称为 DB 的第 k 个簇(k=1,2,…, K), K(K>1)是给定的簇数目。

Jing[3]等提出了一种处理文本数据的软子空间聚类算法 FWKM, 该算法通过维度权值的幂函数形式对k-means目标函数进行扩展得到如下新的目标函数：

$$J(\boldsymbol{C},\boldsymbol{V},\boldsymbol{W})^{(\mathrm{FWKM})}=\sum_{k=1}^{K}\sum_{j=1}^{D}\boldsymbol{w}_{kj}^{\beta}\sum_{x_i\in C_k}(x_{ij}-\boldsymbol{v}_{kj})^2$$

其中, $\boldsymbol{w}_k=<w_{k1},w_{k2},...,w_{kD}>$ 和 $\boldsymbol{v}_i=<v_{i1},v_{i2},...,v_{iD}>$ 分别表示 C_k 的维度权值和簇中心向量, 且 $\sum_{j=1}^{D}\boldsymbol{w}_{kj}=1$, k=1,2,…,K, $\boldsymbol{V}=\{\boldsymbol{v}_{kj}\}_{k\times D}$ 和 $\boldsymbol{W}=\{\boldsymbol{w}_{kj}\}_{k\times D}$

是两个矩阵，β为用户定义的加权参数，其作用是调节权值的影响力。FWKM 算法的聚类目标是使目标函数 J 取值最小，即加权的簇内紧凑度达到最小，使用 Lagrange 乘子技术求出维度权重的计算公式：

$$\boldsymbol{w}_{kj}^{(FWKM)}=\left(\sum_{l=1}^{D}\left(\frac{\sum_{x_i\in C_k}(x_{ij}-\boldsymbol{v}_{kj})^2}{\sum_{x_i\in C_k}(x_{il}-\boldsymbol{v}_{kl})^2}\right)^{\frac{1}{\beta-1}}\right)^{-1}$$

从上式可以看出，维度权值的取值与方差成反比，这意味着数据分布越紧凑的维度将获得越大的权值。由 Jing 等提出的 EWKM[6]算法，其目标函数、权值计算公式的形式与 LAC[4]很相似，但是 EWKM 保留了簇大小差异对权值的影响。具体定义如下：

$$E(\boldsymbol{C},\boldsymbol{W})=\sum_{k=1}^{K}\sum_{j=1}^{D}(\boldsymbol{w}_{kj}\sum_{x_i\in C_k}(x_{ij}-\boldsymbol{v}_{kj})^2+ r\boldsymbol{w}_{kj}\log\boldsymbol{w}_{kj})+\sum_{k=1}^{K}\lambda_k(1-\sum_{j=1}^{D}\boldsymbol{w}_{kj})$$

$$\boldsymbol{w}_{kj}^{(\mathrm{EWKM})}=\frac{\exp\left(-\sum_{x_i\in C_k}(x_{ij}-\boldsymbol{v}_{kj})^2\Big/r\right)}{\sum_{l=1}^{D}\exp\left(-\sum_{x_i\in C_k}(x_{ij}-\boldsymbol{v}_{kj})^2\Big/r\right)}$$

其中，r 为加权参数，用来控制权值分布的离散程度。以上所述算法具有较好的扩展性，适用于高维数据的聚类分析。同时，这些算法聚类结果的质量部分依赖于目标函数的设置，由于目标函数多数关注于簇内紧凑度信息，而忽略了子空间的优化[8]，所以在一定程度上降低了算法的聚类精度。

根据上述分析，提出了一种衡量维度权值分布离散程度的新指标，用于优化子空间。此新指标与“簇内紧凑度”度量相结合来定义新的目标优化函数。新算法 SDSC 基于此目标函数，在类 *k*-means 的聚类过程中，不断地迭代优化目标函数，以获得高质量的聚类结果。

3 算法 SDSC

本章首先提出子空间差异等形式定义，在此基础上推导出 SDSC 的目标优化函数，然后给出详细的描述，并对算法的性能进行分析。

3.1 子空间差异与目标函数

软子空间加权方法体现了一个共同特点：维度权值的大小与数据点投影到该维度上的分布离散程度成“反比”[6]，即数据分布越紧凑的那个维度将获得越大的权值。因此，维度权值分布的信息反映了子空间中数据点分布的离散程度。现将每个簇的维度权值当做一个数据对象，如果簇的维度权值紧凑度越高，则其数据分布越集中，子空间越优化。因此，在算法的目标函数中考虑维度权值紧凑度的评估信息，以便优化聚类过程，从而获得更好的聚类结果。与软子空间算法不同的是，传统聚类没有涉及到维度加权的概念，故可认为各维度权重相同，于是基于 $w_{k1}+w_{k2}+...+w_{kD}=1$ 的约束条件，引入衡量子空间维度权值分布的离散程度的计算公式：

$$diff'(S_k)=\sum_{j=1}^{D}(\boldsymbol{w}_{kj}-\frac{1}{D})^2$$

其中，S_k 表示第 k 个子空间簇类。通过引入约束条件 $w_{k1}+w_{k2}+...+w_{kD}=1$ 来求解 $diff'(S_k)$ 的极值，可得出 $diff'(S_k)\in[0,D-1/D]$。基于 $diff'(S_k)$ 的取值范围，可以将 $diff'(S_k)$ 变换到[0,1]区间，并定义标准化的子空间差异大小如下：

定义 **1** (子空间差异) 记第 k 个子空间簇的子空间差异为 $diff(S_k)$：

$$diff(S_k)=\frac{D\sum_{j=1}^{D}\left(\boldsymbol{w}_{kj}-\frac{1}{D}\right)^2}{D-1}$$

定义 **2** (子空间两点距离[9]) 设 y_i 是 x_i 在 S_k 所属子空间上的投影，基于常用的加权欧式距离[9]来度量 S_k 所在的子空间中两个投影点 y_1 和 y_2 间的距离为：

$$dist_{S_k}(y_1,y_2)=\sqrt{\sum_{j=1}^{D}\boldsymbol{w}_{kj}\ (x_{1j}-x_{2j})^2} \qquad (1)$$

基于定义 1 和定义 2, 构造算法 SDSC 的目标函数如下：

$$J(\boldsymbol{C},\boldsymbol{V},\boldsymbol{W})=\sum_{k=1}^{K}\left(\sum_{j=1}^{D}\boldsymbol{w}_{kj}\sum_{x_i\in C_k}(x_{ij}-\boldsymbol{v}_{kj})^2+p_k\times diff(S_k)\right)$$

其中, 目标函数的前一项定义了加权的簇内紧凑度, 后一项定义了子空间差异。p_k (p_k>0)为平衡系数, 可参考文献[8]将其设置为：

$$p_k=\frac{1}{D}\sum_{j=1}^{D}X_{kj}$$

这里, 引入了新记号X_{kj}, 其中$v_k=<v_{k1},v_{k2},...,v_{kD}>$表示 C_k的簇中心：

$$X_{kj}=\sum_{x_i\in C_k}(x_{ij}-\boldsymbol{v}_{kj})^2$$

3.2 优化方法

目标函数 J 的最小化是一个非多项式的问题。常用的解决办法是基于 $\boldsymbol{W}$ 和 $\boldsymbol{V}$ 的局部最优化来解决目标函数 J 的最优化。为计算 $\boldsymbol{W}$ 和 $\boldsymbol{V}$ 的局部最优值, 在J的基础上引入$\boldsymbol{w}_{kj}$的约束条件($w_{k1}+\ w_{k2}+...+w_{kD}=1$)构造目标优化函数 J_1：

$$J_1(\boldsymbol{C},\boldsymbol{V},\boldsymbol{W})=\sum_{k=1}^{K}\sum_{j=1}^{D}\boldsymbol{w}_{kj}X_{kj}+\sum_{k=1}^{K}\left(\frac{1}{D}\sum_{j=1}^{D}X_{kj}\times\frac{D\sum_{j=1}^{D}\left(\boldsymbol{w}_{kj}-\frac{1}{D}\right)^2}{D-1}\right)+\sum_{k=1}^{K}\lambda_k\left(\sum_{j=1}^{D}\boldsymbol{w}_{kj}-1\right)$$

其中, λ_k(k=1,2,…,K)为拉格朗日乘子。$|C_k|$表示 C_k 的数据点数目。令$\frac{\partial J_1}{\partial \boldsymbol{v}_{kj}}=0$来求解 v。经过推导有：

$$\boldsymbol{v}_{kj}=\frac{1}{|C_k|}\sum_{x_i\in C_k}x_{ij} \tag{2}$$

令$\frac{\partial J_1}{\partial \boldsymbol{w}_{kj}}=0$来推导 $\boldsymbol{w}_{kj}$的计算公式, 经整理有：

$$\boldsymbol{w}_{kj}=\frac{(\lambda_k+X_{kj})(1-D)}{2\sum_{l=1}^{D}X_{kl}}+\frac{1}{D} \tag{3}$$

由$\frac{\partial J_1}{\partial \lambda_k}=\sum_{j=1}^{D}\boldsymbol{w}_{kj}-1=0$，将公式(3)代入可得：

$$\lambda_k=\frac{-\sum_{j=1}^{D}X_{kj}}{D} \tag{4}$$

将公式(4)代入公式(3)中得到结果如下：

$$\boldsymbol{w}_{kj}=\frac{(1-D)}{2\sum_{l=1}^{D}X_{kl}}X_{kj}+\frac{1+D}{2D} \tag{5}$$

3.3 算法实现

根据优化的目标函数, SDSC 算法是基于传统的 k-means[7]框架来实现的, 具体过程描述如下：

算法 The SDSC algorithm

输入：N 个 D 维的数据对象; 聚类数目 K; 迭代终止阀值ε

输出：K 个子集的集合$C=\{C_1,C_2,...,C_k\}$和相应的维度权值集合$\{w_1,w_2,...,w_k\}$

初始化：随机地选择 K 个初始中心点, 所有的权值初始化为 1/D, 创建簇中心矩阵$\boldsymbol{V}^{(0)}$和权值矩阵$\boldsymbol{W}^{(0)}$

设 p 为迭代步骤的序号, p=0

Repeat

1. 根据公式(1)将所有对象指派到最近的簇中
2. p=p+1
3. 根据公式(2)计算$\boldsymbol{V}^{(p)}$
4. 根据公式(5)计算$\boldsymbol{W}^{(p)}$

Until $\left\|\boldsymbol{V}^{(p)}-\boldsymbol{V}^{(p-1)}\right\|<\varepsilon$ and $\left\|\boldsymbol{W}^{(p)}-\boldsymbol{W}^{(p-1)}\right\|<\varepsilon$

其中，ε是一个很小的正实数，本文设定ε=10^{-6}, 其目的是控制算法的结束。SDSC 算法基于经典的 k-means 框架，不同之处在于增加了一个计算子空间差异的步骤和重新定义了各迭代步骤的计算公式。由于 SDSC 算法通过子空间差异优化子空间, 所以该算法不仅延续了 k-means 算法的迭代收敛条件(簇中心收敛), 还关注于维度权值(子空间)的收敛。

下面从迭代过程分析该算法的时间复杂度。首先通过公式(1)来划分数据集, 其时间复杂度等

同于在 K 个子空间簇类中计算公式(1)的复杂度，则时间复杂度为 $O(KND)$，同理，更新簇中心和计算维度权值的两个步骤分别都为 $O(KND)$。假设算法迭代次数为 P，则算法 SDSC 的时间复杂度为 $O(KNDP)$。与现有算法相比，算法 SDSC 的特点首先体现在其算法过程不需要用户输入额外的参数；其次，新算法的目标还包括子空间的优化，这是现有其他算法所不具备的。

4 实验结果与分析

实验采用多组高维数据测试集，通过与其他 3 种算法的实验结果进行比较分析，来验证 SDSC 算法的有效性。

4.1 实验数据

实验选取5个常用的数据集来检验 SDSC 的有效性和聚类精度。数据集的有关信息参见表 1。第1个实际数据集是SpamBase[1]，它来自UCI machine learning repository，常用于机器学习、模式识别领域的垃圾邮件过滤研究。第 2 个数据集 Email-1431[2] 是基于内容的垃圾邮件过滤研究中常用的一个英文电子邮件语料库，通过去除停用词，抽取词条并计算词频等预处理[2]步骤，选取2 002个出现频度最高的词(词频在28以上)，最后得到Email-1431-2002 数据集。第3个数据集 Ling-Spam[3] 由 2 412 封正常邮件与 481 封垃圾邮件组成。通过相同的预处理后，得到 48 742 个词条，去掉 44 300 个只在 14 封邮件(约 0.5%)中出现的低频词条和 7 个 1 157 封邮件(约 40%)中都出现过的高频词，最后得到 4 435 个词条。第 4 个和第 5 个数据集都抽取自专门用于中文文本分类测试的文本库 Tan-Corp[4]，Tan-Corp 包含 14 150 个中文文本，分为 12 大类，实验从体育、卫生和娱乐 3 个类别中分别抽取了 5 711 个文本和 6 301 个文本，经过相同的预处理后选择了1 191个词条，形成 Tan-5711-1191 数据集和Tan-6301-1191 数据集。

Table 1 Information about the real-world datasets
表 **1** 实际应用数据集的有关信息

数据集	数据点数目	维数	类别数目
SpamBase	4 601	54	2
Email-1431	1 431	2 002	2
Ling-Spam	2 893	4 435	2
Tan-5711-1191	5 711	1 191	3
Tan-6301-1191	6 301	1 191	4

为了更准确地分析 SDSC 算法对数据点数目和数据维度数两个数据集参数的可伸缩性。根据文献[10]所提供的合成数据集的方法生成测试数据集。数据集的有关信息参见表 2。D_{1-4} 数据集测试新算法对数据点数目的可伸缩性，合成参数设置如下：类别数目为 4，维度数为 1 000。D_{5-8} 数据集测试新算法对数据点数目的可伸缩性，合成参数设置如下：类别数目为 4，数据点数目为 5 000。

Table 2 Summarized parameters of the synthetic datasets
表 **2** 合成数据集的有关信息

D_{1-4}	(类别数目为 4，维度数为 1 000)			
数据点数目	D_1	D_2	D_3	D_4
	10 000	20 000	30 000	40 000
D_{5-8}	(类别数目为 4，数据点数目为 5 000)			
维度数	D_5	D_6	D_7	D_8
	1 000	2 000	3 000	4 000

4.2 实验设置

实验环境为 Intel 2.99 GHz CPU, 2 GB 内存，操作系统为 Microsoft Windows 7，采用 C++语言实现。

为了能有效分析 SDSC 算法的性能，实验选

[1] ftp.ics.uci.edu:pub/machine-learning-databases.
[2] http://www2.imm.dtu.dk/~rem/data/.
[3] http://www.aueb. gr/users/ion/data/.
[4] http://lcc.software.ict.ac.cn/~tansongbo/corpus1.php.

择了 EWKM[6]、FWKM[3]和 FSC[4]3 种聚类算法进行对比分析，与本文算法不同的是，其他 3 种对比算法都需要用户设置额外的(参数 K 除外，这是 4 种算法都需要的) 算法参数。对于 EWKM[6]参数γ，根据建议设定为 0.5，此外，为测试其参数的敏感性，也将γ调整为 1.0 进行实验；FWKM[3]和 FSC[5]参数β和α分别根据文献[3]和文献[5]的建议，设定为 1.5 和 2.1；同样地，实验还将β和α设定为 2.0 和 3.0 来验证对比算法的适应性。

4.3 实验结果

实验采用 Micro-F1 和 Macro-F1 这两个评价指标进行性能评估。F1[9]的定义如下：

$$\mathrm{F1}=\frac{2\times Recall\times Precision}{Recall+Precision}$$

其中，*Recall* 和 *Precision* 分别表示“召回率”和“精确度”。在 F1 基础上，使用 Micro-F1 和 Macro-F1[2]作为衡量算法性能的指标。

由于4种算法都是基于 *k*-means 框架，从不同的初始簇中心出发，*k*-means 算法可能导致有差异的聚类结果。因此，对同一个数据集，实验分别运行 4 种算法 100 次，最后报告平均的聚类精度。为公平起见，4 种算法每次运行都选择同样的初始簇中心开始。表 3 和表 4 显示 4 种算法在 5 个数据集上取得的平均聚类精度。

如表 3 和表 4 所示，算法都能较好地对实际应用的数据集进行聚类。SDSC 算法可以获得比其他算法更好的聚类精度。与其他 3 种算法相比，本文算法保持了较高的聚类正确率，在 5 种数据集的 100 次随机运算中，F1 值达到 80%以上。对于各类别样本数量分布不均匀的 Ling-Spam 数据集，聚类难度较大，在其他聚类算法精度平均为 60%左右的情况下，SDSC 仍然显示出优势，取得了 85%的聚类精度。以数据维度数较高的 Email-1431 数据集为例，其他聚类算法平均都只获得 70%左右的聚类精度，SDSC 算法则获得了 90%

Table 3 Comparison of the clustering results on the real-world datasets (Micro-F1)
表 **3** 实际应用数据集上的聚类结果比较(Micro-F1)

	SpamBase	Email-1431	Ling-Spam	Tan-5711-1191	Tan-6301-1191
SDSC	0.801 3	0.958 1	0.848 3	0.927 3	0.827 5
EWKM(γ=0.5)	0.567 3	0.582 9	0.654 3	0.578 3	0.498 4
EWKM(γ=1.0)	0.559 1	0.605 1	0.608 7	0.603 6	0.592 7
FWKM(β=1.5)	0.641 1	0.602 7	0.840 2	0.668 3	0.564 8
FWKM(β=2.0)	0.610 3	0.746 2	0.807 2	0.733 8	0.709 1
FSC(α=2.1)	0.598 4	0.654 2	0.614 6	0.642 8	0.618 4
FSC(α=3.0)	0.736 9	0.798 6	0.747 6	0.698 7	0.665 9

Table 4 Comparison of the clustering results on the real-world datasets (Macro-F1)
表 **4** 实际应用数据集上的聚类结果比较(Macro-F1)

	SpamBase	Email-1431	Ling-Spam	Tan-5711-1191	Tan-6301-1191
SDSC	0.800 9	0.930 9	0.809 4	0.916 4	0.817 3
EWKM(γ=0.5)	0.419 2	0.463 4	0.518 4	0.479 3	0.501 2
EWKM(γ=1.0)	0.407 9	0.487 4	0.496 8	0.527 4	0.496 6
FWKM(β=1.5)	0.499 6	0.487 1	0.495 7	0.583 9	0.523 7
FWKM(β=2.0)	0.520 4	0.681 1	0.514 8	0.692 1	0.671 4
FSC(α=2.1)	0.473 5	0.615 7	0.484 3	0.491 8	0.489 6
FSC(α=3.0)	0.700 8	0.746 3	0.705 9	0.583 5	0.432 8

以上的聚类精度。从表 3 和表 4 可以看出，算法参数的设置与其聚类性能密切相关。作为对比，除簇类数目 K 以外，SDSC 算法无需给定其他的参数，提高了算法的适应性。

由图 1 可见，对同一个数据集进行聚类时，EWKM 聚类时间明显多于其他 3 种算法。原因在于为 EWKM 选择合适的算法参数比较困难，因此聚类时间较长。FWEM、FSC 和 SDSC 算法的平均运行时间虽然差异不大，但是，在相同的运行条件下, SDSC 具有更高的聚类精度。

SDSC 算法继承了 *k*-means 算法在数据点数和数据维度数方面的可伸缩性。图 2 显示 SDSC 的运行时间与数据点数目和数据维度数之间的关系。如图 2 中两个图所示, SDSC 的运行时间与数据点数目、数据维度数目都呈线性关系，验证了该算法的可伸缩性。

以上实验结果表明，SDSC 算法能够获得较为稳定的聚类结果。与其他 3 种算法相比，由于 SDSC 在聚类的过程中不断优化簇类所在的子空间，能更准确地估计出每个划分所在的最佳子空间，从而获得更高的聚类精度。

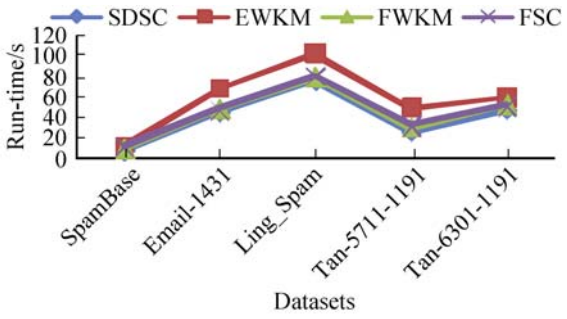

Fig.1 Comparison of the runtime of different algorithms

图 1 不同算法的运行时间对比

5 结论与展望

本文通过分析常用软子空间算法的不足，提出了一种基于子空间差异的聚类算法 SDSC。该算法在聚类过程中使用提出的聚类优化目标函数，在最小化簇内紧凑度的同时，优化了簇类所在的子空间，且不需要用户给定额外的算法参数。与其他算法相比，文中算法具有较好的聚类精度和适应性。SDSC 算法还有需要进一步改进之处，如针对高维数据进行聚类的初始簇中心选择方法，这将是下一步研究的方向。

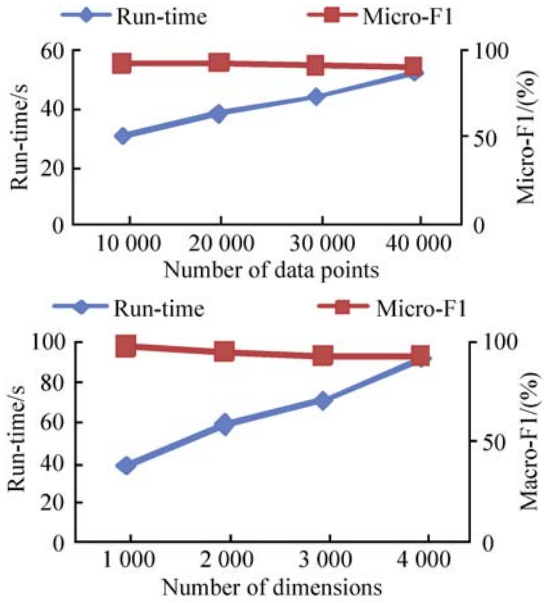

Fig.2 Relationships between the runtime of SDSC, and different numbers of data points and dimensions

图 2 SDSC 算法的运行时间与数据点数目和数据维度数之间的关系

References:

[1] Yang Q, Wu X. 10 challenging problems in data mining research[J]. International Journal of Information Technology and Decision Making, 2006, 5(4): 597–604.

[2] Tan S, Cheng X, Ghanem M, et al. A novel refinement approach for text categorization[C]//Proceedings of the ACM 14th Conference on Information and Knowledge Management, 2005: 469–476.

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

[4] Domeniconi C, Gunopulos D, Ma S, et al. Locally adaptive metrics for clustering high dimensional data[J]. Data Mining and Knowledge Discovery, 2007, 14: 63–97.

[5] Gao G, Wu J, Yang Z. A fuzzy subspace clustering algo-

rithm for clustering high dimensional data[C]//Proceedings of the 2nd International Conference on Advanced Data Mining and Applications. Berlin Heidelberg: Springer-Verlag, 2006: 271–278.

[6] Jing L, Ng M K, Huang J. 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.

[7] Chu Y, Chen Y, Yang D, et al. Reducing redundancy in subspace clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(10): 1432–1446.

[8] Chen L, Guo G, Jiang Q. An adaptive algorithm forsoft subspace clustering[J]. Journal of Software, 2010, 21(10).

[9] Chen L, Shi L, Jiang Q, et al. Supervised feature selection for DOS detection problems using a new clustering criterion[J]. Journal of Computational Information Systems, 2007, 3(5): 1983–1992.

[10] Aggarwal C, Procopiuc C, Wolf J, et al. Fast algorithm for projected clustering[C]//Proceedings of the ACM SIGMOD Conference. New York: ACM Press, 1999: 61–71.

附中文参考文献:

[8] 陈黎飞, 郭躬德, 姜青山. 自适应的软子空间聚类算法[J]. 软件学报, 2010, 21(10).

ZHANG Yanping was born in 1986. She is a M.S. candidate at Xiamen University. Her main research interest is data mining.

张燕萍(1986–), 女, 厦门大学硕士研究生, 主要研究领域为数据挖掘。

JIANG Qingshan was born in 1962. He is a professor and doctoral supervisor at Xiamen University. His main research interests include data mining, image processing and database system, etc.

姜青山(1962–), 男, 厦门大学教授、博士生导师, 主要研究领域为数据挖掘, 图像处理, 数据库系统等。