文章编号: 1001-0920 (2014) 09-1683-05 **DOI:** 10.13195/j.kzyjc.2013.0660

基于密度调整的改进自适应谱聚类算法

王雅琳, 陈 斌, 王晓丽, 桂卫华

(中南大学 信息科学与工程学院，长沙 410083)

摘 要: 针对谱聚类存在构造相似度矩阵时对尺度参数敏感以及处理多重尺度数据集效果不理想的缺陷, 提出一种基于密度调整的改进自适应谱聚类算法. 该算法将样本点所处领域的密度引入谱聚类, 利用密度差来调整样本点之间的相似度, 使其更符合实际簇类中样本点间的内在关系, 在一定程度上解决了多尺度聚类问题; 同时, 通过样本点的近邻距离自适应得到尺度参数, 使算法对尺度参数相对不敏感. 仿真实验验证了所提出算法的有效性和优越性.

关键词: 谱聚类；密度调整；自适应；尺度参数；多重尺度数据集

中图分类号: TP181 **文献标志码:** A

Improved adaptive spectral clustering algorithm based on density adjustment

WANG Ya-lin, *CHEN Bin*, *WANG Xiao-li*, *GUI Wei-hua*

(School of Information Science and Engineering，Central South University，Changsha 410083，China. Correspondent: WANG Ya-lin，E-mail：ylwang@csu.edu.cn)

Abstract: As spectral clustering is sensitive to the scaling parameter while calculating the affinity matrix and the result of clustering multi-scale dataset is not ideal, an improved adaptive spectral clustering algorithm based on density adjustment is proposed. The algorithm introduces local density of data into spectral clustering, using the density difference to adjust the similarity between sample points, which makes it more consistent with the data points' internal relations of the clusters' actual structure. So that it solves the multi-scale clustering problem to some extent. At the same time, the algorithm is relatively insensitive to the scaling parameter by using the distances between data points and their neighbor points to get the scaling parameter adaptively. Simulation experiment shows the effectiveness and superiority of the proposed algorithm.

Key words: spectral clustering；density adjustment；adaptive；scaling parameter sensitivity；multi-scale dataset

0 引 言

作为数据挖掘中进行数据处理的一种重要方法[1], 聚类分析是根据一定的相似准则将数据对象划分为由类似对象组成的多个类的过程. 由聚类得到的同一类中的对象彼此相似, 而不同类中的对象彼此相异. 现如今, 聚类算法已经有很多种, 如传统的K-means 算法[2]、FCM 算法[3]等. 这些经典的聚类算法虽然在凸样本空间上有比较好的聚类效果, 但是当样本空间不为凸时, 算法很容易陷入局部最优. 近年来出现的谱聚类算法使得这个问题得以解决. 谱聚类[4-6]是一种性能优越的聚类算法, 它不受数据集样本点簇类形状的影响, 即使样本空间非凸, 也能得到全局最优解, 因此, 谱聚类很快成为了关注焦点[7-10].

谱聚类算法具有比其他聚类算法更优越的性能, 但其本身也存在构造相似度矩阵时对尺度参数比较敏感以及处理多重尺度数据集时结果不理想等问题. 为此, 广大学者对谱聚类算法进行了研究. Gong 等[11]通过数据点的邻近点信息来构造相似图, 以此获得数据点间的相似度矩阵, 提出了一种基于局域信息的谱聚类算法; Ozertem 等[12]利用 Mean Shift 算法求得的权值来建立谱聚类中的相似图, 提出了一种 Mean Shift 谱聚类算法; 周林等[13]提出了基于谱聚类的聚类集成算法, 首先利用谱聚类算法的内在特性构造多样性聚类成员, 然后采用连接三元组方法和 Nyström 采样方法计算相似度矩阵, 扩充了数据点之间的相似性信息. 以上算法都不需要人为设定尺度

收稿日期: 2013-05-21；**修回日期:** 2013-08-28.

基金项目: 国家自然科学基金项目(61273187)；教育部博士点新教师类基金项目(20120162120022)；湖南省科技计划项目(2012CK4018).

作者简介: 王雅琳(1973−), 女, 教授, 博士, 从事复杂过程建模、优化与控制等研究；陈斌(1989−), 男, 硕士生, 从事数据挖掘与聚类分析的研究.

参数, 使得算法对尺度参数不敏感, 但并没有对多尺度聚类的问题进行研究. Yang 等[14]在对相似性度量进行分析的同时结合数据聚类特性, 提出了一种数据依赖的相似性度量——密度敏感的相似性度量, 并将其引入谱聚类得到密度敏感的谱聚类算法, 但其中密度敏感的距离定义和计算相对复杂; Zelnik-Manor 等[15]提出了 Self-Tuning 算法, 将数据点的领域信息加入相似度的计算中, 使数据点领域内的数据分布对数据点间的相似度产生作用, 更真实地反映了数据点间的内在联系, 但算法对数据点的位置分布未考虑全面, 仍存在不足之处.

本文在文献 [15] 的基础上进行改进, 提出一种基于密度调整的改进自适应谱聚类算法. 通过数据点所处领域的密度差来调整它们之间的相似度, 使其更符合实际结构中数据点间的内在联系; 另外, 该算法利用数据点的近邻距离来自适应求取尺度参数, 避免了对尺度参数的人为设定, 从而对尺度参数相对不敏感. 仿真结果表明, 本文所提算法在处理簇密度相差很大的多重尺度数据集时具有很好的簇类结果, 且提高了聚类质量.

1 相关算法

1.1 谱聚类算法

谱聚类算法是基于谱图理论[16]中图的最优划分思想提出的, 其本质是将样本点的聚类问题转化为寻求一种对图的最优分割方法的问题. 谱聚类将数据集中的每个样本点看作图中的顶点 V, 顶点之间用边 E 连接, 其权重为样本点间的相似度 W, 由此构造出了一个无向的加权图 $G=(V,E)$. 这样就把原来的聚类问题转化成了在图 G 上的最优划分问题.

虽然谱聚类算法的实现形式多种多样, 但都可以归纳为如下大体流程[6].

1) 构建样本点的相似图, 得到相似度矩阵和 Laplacian 矩阵 L;

2) 计算 L 的前 k 个特征向量, 建立特征向量空间;

3) 通过 K-means 或其他经典聚类算法对特征向量空间的特征向量进行聚类.

尽管谱聚类算法具有不受簇类形状影响和不易陷入局部最优值的优点, 但在使用高斯相似函数构造相似度矩阵时仍存在对尺度参数 σ 比较敏感, 人为设定困难等问题. 而且当处理具有多重尺度的数据集时, 谱聚类算法得到的簇类结果往往不够理想.

1.2 Self-Tuning 算法

多重尺度数据集是指各个簇中数据点密度差异较大的数据集. 由于标准的谱聚类算法没有考虑到邻近点的影响, 当它处理具有多重尺度的数据集时无法得到很好的聚类结果.

图 1 所示为一个多重尺度数据集, 它包含一个密集簇和一个稀疏簇. 其中: 样本点 a 和 c 位于稀疏簇中, 样本点 b 位于密集簇中. 假设样本点 a、b 之间与 a、c 之间的欧氏距离相等 $d(s_a,s_b)=d(s_a,s_c)$, 那么由高斯相似函数有相似度 $A_{ab}=A_{ac}$, 即样本点 a、b 间的相似度等于样本点 a、c 间的相似度. 但实际上由于样本点 a、c 同处在稀疏簇中, 它们的相似度应该要比 a、b 的大.

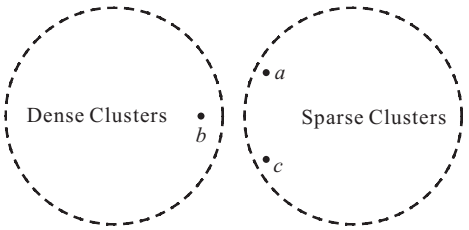

图 1 多重尺度数据集

为了解决上述问题, Zelnik-Manor 等[15]提出了 Self-Tuning 算法. Self- Tuning 算法将数据点的领域信息引入到相似度的计算中, 定义了一个能够自适应得到尺度参数 σ 的高斯相似函数

$$\hat{A}_{ij}=\exp\Big(\frac{-d^2(s_i,s_j)}{\sigma_i\sigma_j}\Big), \tag{1}$$

其中 $\sigma_i=d(s_i,s_K)$ 表示样本点 s_i 到其第 K 个最近邻居点的距离, 文献 [15] 中取 $K=7$.

利用式 (1) 计算图 1 中样本点 a、b、c 之间的相似度, 有 $\sigma_c>\sigma_b$, 从而 $\sigma_a\sigma_c>\sigma_a\sigma_b$, 可得 $\hat{A}_{ac}>\hat{A}_{ab}$, 即样本点 a 与 c 之间的相似度大于 a 与 b. 这样, 样本点 a、c 更有可能被聚类到同一簇类中, 从而与实际相符.

2 基于密度调整的改进自适应谱聚类算法 (IASCBDA)

2.1 算法提出

虽然 Self-Tuning 算法对如图 1 中样本点 a、b、c 间的相似度进行了调整, 但是当样本点 a、b、c 的位置如图 2 所示, 样本点 a、b 位于密集簇, 样本点 c 位于稀疏簇时, 可以发现, 仍有 $\sigma_c>\sigma_b$, $\sigma_a\sigma_c>\sigma_a\sigma_b$, 通过式 (1) 计算依旧为 $\hat{A}_{ac}>\hat{A}_{ab}$. 这显然与实际中同处在密集簇中的 a 与 b 的相似度应该更大是不相符的. 由此可见, Self-Tuning 算法依然存在不足之处.

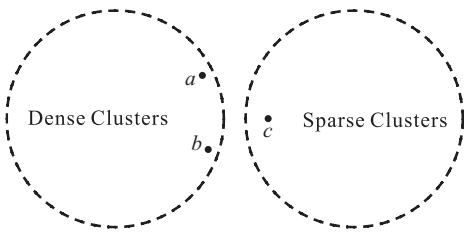

图 2 多重尺度数据集

通过观察图 1 和图 2 可以发现, 图 1 中的样本点 a、c 与图 2 中的样本点 a、b 应该被划分到同一个类中, 这是因为它们都处于密度相同或相近的区域内, 同时第 3 个样本点所处区域内的密度与它们的密度相差较大. 由此表明, 数据点所处领域内的密度对数据点之间的相似度是有影响的, 而且两数据点领域内的密度越接近时, 它们在同一簇类中的可能性越大. 因此, 本文在 Self-Tuning 算法的基础上, 将数据点所处领域内的密度引入谱聚类, 提出一种基于密度调整的改进自适应谱聚类算法.

2.2　基本思想

基于密度调整的改进自适应谱聚类算法的基本思想为: 在对具有多重尺度的数据集进行聚类分析时, 因为仅仅根据欧氏距离无法得到符合实际簇类中数据点间相似度, 所以将数据点所处领域内的密度引入到谱聚类中, 通过数据点领域内密度差的大小对数据点之间的相似度进行调整. 密度差越大, 对应的相似度越小; 反之, 相似度越大.

式 (1) 中 σ_i 的大小在一定程度上反映了样本点 s_i 周围数据点分布的紧密程度: σ_i 越大, 样本点 s_i 周围的数据点分布的越稀疏, 反之越紧密. 所以, 可以用 σ_i 表示样本点 s_i 所处领域内的密度.

为此, 本文算法定义了一个新的相似度函数

$$A'_{ij}=\exp\Big[\frac{-d^2(s_i,s_j)}{\sigma'^2}\Big(1+\frac{|\sigma_i-\sigma_j|}{\sigma_{\max}}\Big)\Big], \tag{2}$$

$$\sigma_{\max}=\max\{|\sigma_i-\sigma_j|;i=1,\cdots,n;j=1,\cdots,n\}, \tag{3}$$

$$\sigma'=\frac{1}{n}\sum_{i=1}^{n}\sigma_i, \tag{4}$$

其中 $\sigma_i=d(s_i,s_K)$ 表示样本点 s_i 到其第 K 个最近邻居点的距离 (本文取 $K=4$).

σ_i 表示样本点 s_i 所处领域内的密度, 则 $|\sigma_i-\sigma_j|$ 为样本点 s_i 与 s_j 之间的密度差. 相似函数中的 $\sigma_{\max}$ 取 $|\sigma_i-\sigma_j|$ 中的最大值, 如式 (3) 所示. 这样在用式 (2) 计算数据点间相似度的过程中, 当两样本点所处簇的密度存在差异时, 就可以通过权值 $\frac{|\sigma_i-\sigma_j|}{\sigma_{\max}}$ 对相似度进行调整, 密度相差越大, 相似度越小. 另外, 为了减小噪声数据对算法的影响, 本文用 K 近邻距离的期望值来代替 Self-Tuning 算法中的 $\sigma_i\sigma_j$, 作为尺度参数 σ 的取值, 如式 (4) 所示.

本文将新的相似度函数与 NJW 算法[4]相结合, 得到基于密度调整的改进自适应谱聚类算法 (IASCBDA), 步骤如下.

输入: n 个数据点 $S=\{s_i\}_{i=1}^n$, 聚类数目 C;

输出: 数据点的 C 个划分.

Step 1: 利用式 (2) 计算亲合矩阵 $A\in R^{n\times n}$, 其中 $A_{ii}=0$;

Step 2: 构造拉普拉斯矩阵 $L=D^{-1/2}AD^{-1/2}$, 其中 D 为对角矩阵, 对角元素为 $D_{ii}=\sum_{j=1}^{n}A_{ij}$;

Step 3: 选择矩阵 L 的前 C 个最大特征向量 x_1, $x_2\cdots,x_C$, 并构造矩阵 $X=[x_1,x_2,\cdots,x_C]\in R^{n\times C}$;

Step 4: 对矩阵 X 中的每一行进行单位化处理, 得到矩阵 Y, 其中 $Y_{ij}=X_{ij}\Big/\Big(\sum_j X_{ij}^2\Big)^{1/2}$;

Step 5: 将 Y 中的每一行视为 R^l 空间中的一个点, 对其使用 K-means 算法进行聚类;

Step 6: 当且仅当矩阵 Y 中的第 i 行属于第 c 类时, 样本点 s_i 也属于第 c 类.

2.3　算法分析

为了能够真实地反映数据集中数据点之间的内在联系, 计算数据点间相似度的相似函数必须满足如下 4 个基本性质.

1) 非负性: $A_{ij}\geqslant 0$;

2) 自反性: $A_{ij}=0$;

3) 对称性: $A_{ij}=A_{ji}$;

4) 一致性: 当 $d(s_a,s_b)>d(s_a,s_c)$ 时, $A_{ab}=A_{ac}$, 即相邻的数据点具有较高的相似性.

证明　欧氏距离与密度差 $|\sigma_i-\sigma_j|$ 满足非负性、自反性和对称性. 所以, 式 (2) 同样满足非负性、自反性和对称性. 而从式 (2) 可以看出, 当 σ'、$|\sigma_i-\sigma_j|$ 和 $\sigma_{\max}$ 一定时, A'_{ij} 随欧氏距离 $d(s_i,s_j)$ 的增大而减小. 因此, 式 (2) 也满足一致性. □

改进的自适应谱聚类算法中的相似度函数除了必须满足 4 个基本性质外, 还应满足一个条件: 两数据点领域内的密度相差越大, 它们之间的相似度越小. 同一致性的证明类似, 可得式 (2) 也满足这一条件.

利用 IABCBDA 分别计算图 1 和图 2 中 3 个样本点的相似度: 当 $d(s_a,s_b)=d(s_a,s_c)$ 时, 图 1 中, 根据式 (2), 由 $|\sigma_a-\sigma_b|>|\sigma_a-\sigma_c|$, 可得 $A'_{ab}<A'_{ac}$; 图 2 中, 有 $|\sigma_a-\sigma_b|<|\sigma_a-\sigma_c|$, 则 $A'_{ab}>A'_{ac}$. 均符合实际簇结构.

根据上述分析, 本文将数据点所处领域内的密度引入谱聚类, 得到的改进自适应谱聚类算法是可行的.

3　实验分析

3.1　有效性实验

为了验证本文所提出算法的有效性, 选择 6 个人工数据集: 数据集 (a)、(b)、(c)、(d) 分别包括 3 个线形簇类、4 个线形簇类、2 个半月形簇类和 3 个半月形簇类, 数据集 (e) 包含1个半月形簇类和 2 个球状簇类,

(f) 包含 1 个环状簇类和 2 个球状簇类. 其中数据集 (b)、(c)、(e)、(f) 取自文献 [15].

使用本文所提出的算法对这 6 个数据集进行聚类, 所得结果如图 3 所示. 不同簇类已在图中通过不同的符号和颜色进行了标示.

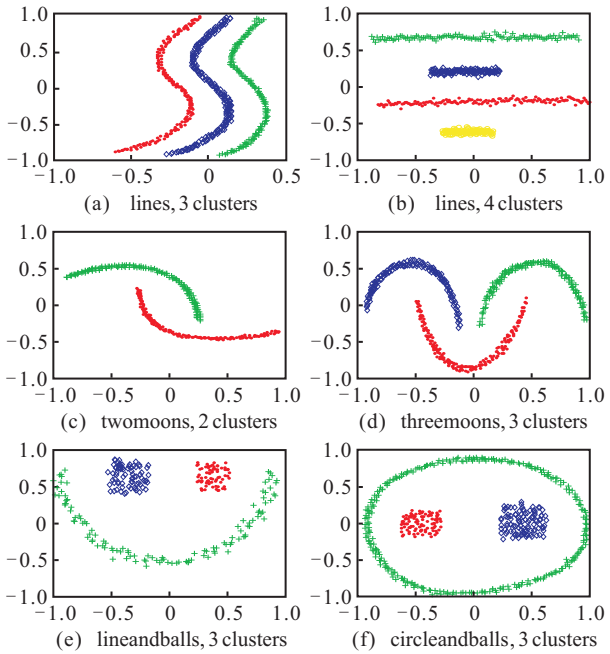

图 3 本文算法 (IASCBDA) 对多个数据集的聚类结果

从图 3 可以看出, 本文提出的改进自适应谱聚类算法能够很好地区分出各种不同的簇, 且聚类结果比较理想.

3.2 对比性实验

为了对比分析本文算法与 Self-Tuning 算法, 选择 3 个数据集进行仿真实验. 其中: (a) 是由 3 个圆环状簇构成的数据集, (b) 为包含 2 个簇类的双螺旋状数据集, (c) 为具有多重尺度数据集. 数据集 (a)、(b) 取自文献 [17].

图 4 给出了两种算法分别对 5 个数据集的聚类结果. 从图 4 可以看出: Self-Tuning 算法在处理这 5 个数据集时都不能得到有效的簇类结果, 尤其是对双螺旋数据集 (d), 聚类结果较差; 而本文提出的改进的自适应谱聚类算法在对这 5 个数据集进行聚类时都能有效地区分出各个簇, 且所得结果比较理想.

表 1 为两种算法对 5 个数据集聚类后出现的误分点个数. 可以看出, Self-Tuning 算法对 5 个数据集都存在误分点, 其中对数据集 (a)、(d) 的误分点数为 128、132, 误分率为 24%、32%, 聚类结果很差. 本文所提算法对 5 个数据集的聚类结果都比较理想, 没有出现误分点.

表 2 列出了两种算法对 5 个数据集进行聚类时所耗的计算时间. 本文算法的计算时间没有明显的增加, 两种算法耗时相当.

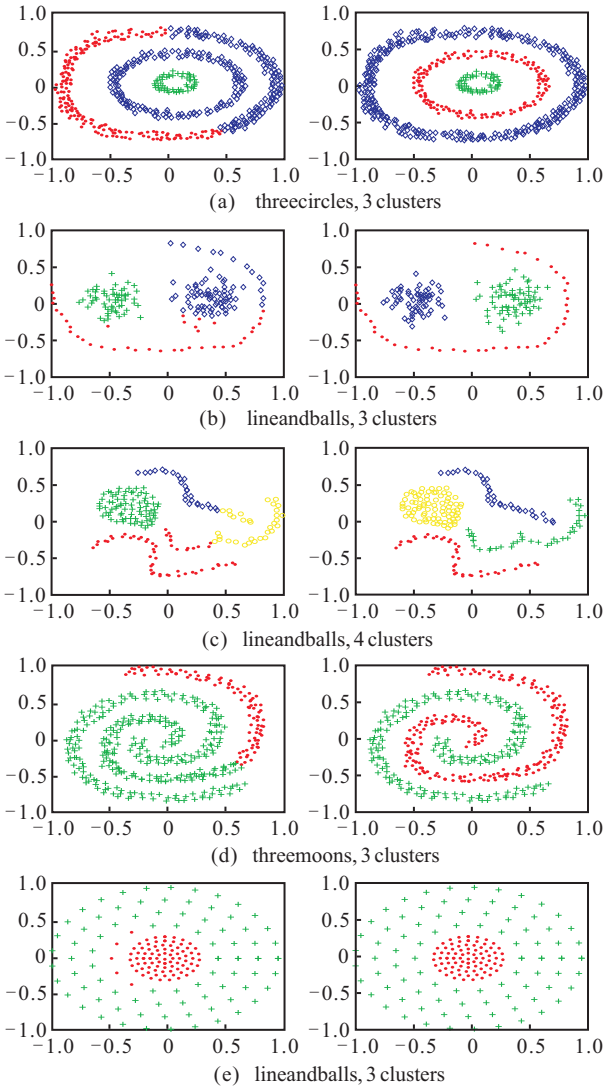

图 4 两种算法聚类结果的比较

表 1 误分点数

数据集	总点数	误分点数	
		Self-Tuning	本文算法
(a)	530	128	0
(b)	180	20	0
(c)	196	33	0
(d)	415	132	0
(e)	177	5	0

表 2 计算时间比较

数据集	计算时间/s	
	Self-Tuning	本文算法
(a)	0.731 4	0.737 22
(b)	0.298 56	0.301 35
(c)	0.377 04	0.381 31
(d)	0.394 53	0.397 46
(e)	0.238 88	0.240 11

由此看见, 相比 Self-Tuning 算法, 本文提出的改进的自适应谱聚类算法具有一定的优越性.

4 结 论

本文通过对 Self-Tuning 算法进行改进, 提出了一种基于密度调整的改进自适应谱聚类算法. 利用数据

点所处领域的密度差来调整它们间的相似度, 更真实地反映了簇类中数据点的内在联系, 从而有效解决了对多重尺度数据集聚类效果不理想的问题; 同时, 通过对尺度参数的自适应求取, 使得该算法对尺度参数相对不敏感. 对多个数据集进行聚类的仿真实验显示, 无论是在处理单一尺度数据集还是多重尺度数据集, 本文算法都能得到理想的簇, 并且比 Self-Tuning 算法有更好的聚类结果. 如何确定谱聚类算法中的聚类数目是接下来的研究工作.

参考文献(References)

[1] Witten I H, Frank E. Data Mining: Practical machine learning tools and techniques[M]. Massachusetts: Morgan Kaufmann, 2005: 81-82.

[2] Trevor Hastie, Robert Tibshirani, Friedman J J H. The elements of statistical learning[M]. New York: Springer, 2001: 460-514.

[3] Chen W, Giger M L. A fuzzy c-means(fcm) based algorithm for intensity inhomogeneity correction and segmentation of MR images[C]. 2004 IEEE Int Symposium on Biomedical Imaging: From Nano to Macro Marriott Crystal Gateway. Arlington: IEEE Press, 2004: 1307-1310.

[4] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm[J]. Advances in Neural Information Processing Systems, 2002, 2(14): 849-856.

[5] Von Luxburg U. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395-416.

[6] 蔡晓妍, 戴冠中, 杨黎斌. 谱聚类算法综述[J]. 计算机科学, 2008, 35(7): 14-18.
(Cai X Y, Dai G Z, Yang L B. Survey on spectral clustering algorithms[J]. Computer Science. 2008, 35(7): 14-18.)

[7] Xiang T, Gong S. Spectral clustering with eigenvector selection[J]. Pattern Recognition, 2008, 41(3): 1012-1029.

[8] 徐森, 卢志茂, 顾国昌. 解决文本聚类集成问题的两个谱算法[J]. 自动化学报, 2009, 35(7): 997-1002.
(Xu S, Lu Z M, Gu G C. Two spectral algorithms for ensembling document clusters[J]. Acta Automatica Sinica. 2009, 35(7): 997-1002.)

[9] 贾建华, 焦李成. 空间一致性约束谱聚类算法用于图像分割[J]. 红外与毫米波学报, 2010, 29(1): 69-74.
(Jia J H, Jiao L C. Image segmentation by spectral clustering algorithm with spatial coherence constraints[J]. J of Infrared and Millimeter Waves, 2010, 29(1): 69-74.)

[10] Zhao F, Jiao L, Liu H, et al. Spectral clustering with eigenvector selection based on entropy ranking[J]. Neurocomputing, 2010, 73(10): 1704-1717.

[11] Gong Y C, Chen C. Locality spectral clustering[M]. AI 2008: Advances in Artificial Intelligence. Berlin Heidelberg: Springer, 2008: 348-354.

[12] Ozertem U, Erdogmus D, Jenssen R. Mean Shift spectral clustering[J]. Pattern Recognition, 2008, 41(6): 1924-1938.

[13] 周林, 平西建, 徐森, 等. 基于谱聚类的聚类集成算法[J]. 自动化学报, 2012, 38(8): 1335-1342.
(Zhou L, Ping X J, Xu S, et al. Cluster ensemble based on spectral clustering[J]. Acta Automatica Sinica, 2012, 38(8): 1335-1342.)

[14] Yang P, Zhu Q, Huang B. Spectral clustering with density sensitive similarity function[J]. Knowledge-Based Systems, 2011, 24(5): 621-628.

[15] Zelnik-Manor L, Perona P. Self-tuning spectral clustering[J]. Advances in Neural Information Processing Systems, 2004, 17: 1601-1608.

[16] Chung F R K. Spectral graph teory[M]. Rhode Island: Amer Mathematical Society. 1997: 1-2.

[17] Liu X, Zong L, Zhang X, et al. Active semi-supervised spectral clustering[C]. The 4th Int Symposium on Parallel Architectures, Algorithms and Programming(PAAP). Tianjin: IEEE Press, 2011: 95-99.

(责任编辑: 齐　雳)