文章编号:1672-3961(2009)05-0022-05

# 自适应谱聚类算法研究

卜德云,张道强*
(南京航空航天大学计算机科学与工程系,江苏 南京 210016)

**摘要**:谱聚类能识别出在原空间中线性不可分的聚类,且其效果优于传统聚类算法.谱聚类要想获得好的效果必须选择一个合适的尺度参数,本文在传统谱聚类算法的基础上引入类似核选取的技巧,提出了一个能自动选取该尺度参数的自适应谱聚类算法.将该算法和现有的谱聚类参数选择算法作了比较,在人工数据集和UCI数据集上的实验表明,自适应谱聚类算法在很多情况下优于其它参数选择算法.

**关键词**:自适应;谱聚类;参数选取
**中图分类号**:TP391 **文献标志码**:A

# Adaptive spectral clustering algorithm

BU De-yun, ZHANG Dao-qiang*
(Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

**Abstract**: Spectral clustering has been used to identify clusters that are non-linearly separable in input space, and usually outperforms traditional clustering algorithms. However, the performances of spectral clustering are severely dependent on values of the scaling parameter. In this paper, an adaptive spectral clustering (ASC) algorithm was proposed based on traditional spectral clustering, which can choose the scaling parameter automatically by using techniques similar to kernel selection. The new algorithm was compared to existing parameter selection based spectral clustering algorithms on both synthetic and UCI data sets, and the experimental results validate the effectiveness of the proposed algorithm.

**Key words**: adaptive; spectral clustering; parameter selecuion

## 0 引言

所谓聚类,就是将数据对象划分成几个不同的类别(cluster),使在同一类中的数据对象尽可能相似,在不同类中的数据对象差别尽可能大.聚类分析是一种广泛应用于数据挖掘和数据分析的有效方法[1],已在统计学、计算机科学、生物学、甚至社会心理学等领域得到了广泛应用.聚类分析通常以无监督方式处理数据并找出其内在结构.目前有众多聚类算法,如K-means及其模糊化的模糊C均值(FCM)等[2].但是这些传统的聚类算法均建立在凸球形的样本空间之上,而当样本空间非凸时,算法易于陷入局部最优点.近年来,谱聚类算法已成了机器学习研究领域的热点之一,其主要基于谱图划分理论[3],并可在任意形状的样本空间上聚类,且具有全局最优性.谱聚类算法首先利用数据样本构造出一个相似性矩阵(affinity matrix),进而计算得出拉普拉斯矩阵(Laplace matrix),然后利用该拉普拉斯矩阵的特征向量来找出数据样本间的内在联系[4].对特征向量的使用方法不同,会产生不同的谱聚类算法[5-7],较经典的有Shi和Malik在2000年提出的

**收稿日期**:2009-06-16
**基金项目**:国家自然科学基金资助项目(60875030);南京航空航天大学创新基金资助项目(Y0804-042)
**作者简介**:卜德云(1985-),男,山东东平人,硕士研究生,研究方向为机器学习及图像处理. E-mail: bdy1985@nuaa.edu.cn
*通讯作者:张道强(1978-),男,山东滕州人,博士,教授,研究方向为机器学习、模式识别与数据挖掘. E-mail: dqzhang@nuaa.edu.cn

Normalized Cuts 算法[6] 和 Ng 等人在 2002 年提出的 Ng-Jordan-Weiss (NJW)算法[5].上述谱聚类算法的基本原理是类似的,本文则主要考虑 NJW 算法.NJW 算法主要利用拉普拉斯矩阵的最大特征值所对应的特征向量,相应的相似性矩阵根据不同数据点间的距离度量来构造.在相似性矩阵的构造中涉及一个关键参数,而它的选择对最后的聚类效果影响很大,如果选择不当,聚类效果将很不理想.

近几年,众多学者都在研究如何提升谱聚类的性能,一些学者在特定条件下优化谱聚类算法使之在特定的场合获得优越的效果[8-9],另一些学者则希望获取一个最佳参数从而构造出合适的相似性矩阵[10],本文主要研究的是后一种方法. Ng 等人[5] 在提出 NJW 算法时给出了一种参数选择方法:首先计算出相似性矩阵,进而构造出拉普拉斯矩阵,然后根据其最大特征值对应的特征向量来进行 K-means 聚类,根据 K-means 的聚类效果 (即能否给出最紧凑的聚类) 来决定是否采用本次实验的参数. 因此,如果想要通过这种方法选择一个合适的参数,在找到合适的参数前不得不反复做特征值分解和 K-means,计算开销很大.Zelnik-Manor 和 Perona[10] 同样也提出了一种称为 Self-Tuning 的算法来获取最佳参数,该算法主要通过所谓的"local scale"思想,利用数据点间的"local scale"来构造相似性矩阵.

另一方面,Dhillon 等人[11] 已证明了 Normalized Cuts 问题等价于一个迹的最大化问题.同样,核 K-means 也可解释为迹的最大化问题[11].因此,可给 NJW 算法做一个全新的解释:首先构造核矩阵 $\boldsymbol{A}$ (在原算法中解释为相似性矩阵,这里采用高斯核),然后计算度矩阵(degree matrix)$\boldsymbol{D}$,根据 $\boldsymbol{D}$ 计算出归一化的拉普拉斯矩阵 $\boldsymbol{L}=\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$,然后计算 $\boldsymbol{L}$ 的特征值和特征向量,最后选取其合适的特征向量(即前若干个最大特征值所对应的特征向量)来进行聚类.这样,谱聚类参数选取问题就转化为对核矩阵的选取.张等人在文献[12] 中提出了一种用于核选取的算法 A-KPCA. A-KPCA 基于一种可以同时进行非线性特征选取和无监督核选择的新准则,即可在无样本标号情形下进行有效的核选取.受张等人的方法的启发,本文提出了自适应谱聚类算法(adaptive spectral clustering,ASC).

## 1　谱聚类算法

给定一批样本数量为 $n$,样本维数为 $l$ 的样本集 $S=\{s_1,s_2,\cdots,s_n\}\in\mathbf{R}^l$,NJW 算法的主要步骤如下所示[5]:

(1) 构造相似性矩阵 $\boldsymbol{A}\in\mathbf{R}^{n\times n}$,矩阵中元素 $A_{ij}=\exp(-\|s_i-s_j\|^2/2\sigma^2)$,且当 $i=j$ 时,$A_{ii}=0$;

(2) 构造矩阵 $\boldsymbol{D}$ 为度矩阵,度矩阵主对角线上的元素 $D(i,i)$为相似性矩阵 $\boldsymbol{A}$ 的第 $i$ 行元素之和,其它元素均为 0,然后构造拉普拉斯矩阵 $\boldsymbol{L}=\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$;

(3) 对拉普拉斯矩阵 $\boldsymbol{L}$ 进行特征值分解,找出其前 $k$ 个最大特征值所对应的特征向量 $x_1,x_2,\cdots,x_k$,然后构造矩阵 $\boldsymbol{X}=[x_1,x_2,\cdots,x_k]\in\mathbf{R}^{n\times k}$,其中特征向量按列存储;

(4) 对 $\boldsymbol{X}$ 的行向量进行再归一化,记归一化后的矩阵为 $\boldsymbol{Y}$,$Y_{ij}=X_{ij}/(\sum_j X_{ij}^2)^{1/2}$;

(5) 把 $\boldsymbol{Y}$ 的每一行看作为空间 $\mathbf{R}^k$ 中的样本(样本数量为 $n$,样本维数为 $k$),然后将这些样本用 K-means 算法进行聚类;

(6) 最后,把最初的样本点 $s_i$ 划分为第 $j$ 聚类当且仅当矩阵 $Y$ 的第 $i$ 行被划分为第 $j$ 聚类;

NJW 算法中,在构造相似性矩阵的时候需要给出一个尺度参数 $\sigma$ 来控制两个样本点 $s_i$ 和 $s_j$ 之间的距离对相似性矩阵 $A_{ij}$ 的影响.本文主要解决的问题就是设计一个算法能自动的选取最优参数.

Ng 等人[5] 在提出 NJW 算法的同时也给出了一种选取尺度参数 $\sigma$ 的方法:他们认为对于一个合适的 $\sigma$,谱聚类算法在第 5 步给出矩阵 $\boldsymbol{Y}$ 的各行应该能给出 $k$ 个很"紧凑"的聚类.然而这种方法需要人工调整才能最终收敛到最优解,计算量比较大.

最近,Zelnik-Manor 和 Perona 提出了 Self-Tuning 谱聚类算法[10],他们利用一种称为"Local Scaling"的思想,跟据此思想,算法并不是为整个样本集选择一个尺度参数 $\sigma$,而是为每一个样本点 $s_i$ 选择一个 $\sigma_i$,而且从样本点 $s_i$ 到 $s_j$ 的距离表示为 $d(s_i,s_j)/s_i$,反之为 $d(s_j,s_i)/s_j$,据此构造的相似性矩阵为

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

其中,$\sigma_i=d(s_i,s_K)$,$s_K$ 是样本点 $s_i$ 的第 $K$ 个近邻.在实验中,一般选取 $K$ 取 7,这是一个经验值,取此值时算法的效果最佳,而且在高维数据上的实验表现突出[10].

## 2　自适应谱聚类算法

### 2.1　尺度参数的选取

尺度参数的选取等价于对相似性矩阵的选取,同样等价于对拉普拉斯矩阵的选取,而根据谱聚类与核方法之间的联系[11],问题转化为对核(高斯核)的选取.

核的选取一直是基于核的学习算法的一个研究重点,ZHANG 等人在文献[12]中给出一种方法可以在一种无监督的状态下进行核的选取,据此思想提出一种选取谱聚类尺度参数的方法.

尺度参数选取思想如下:首先给出 $n$ 个备选参数(备选参数对算法的影响不大,实验中随机选取若干备选参数,不需人工指定),根据这些备选参数可以构造出 $n$ 个拉普拉斯矩阵 $\boldsymbol{L}_k, k=1,2,\cdots,n$.根据谱聚类与核方法之间的等价性,此处的 $n$ 个拉普拉斯矩阵 $\boldsymbol{L}_k$ 可以看作 $n$ 个核矩阵,因而对拉普拉斯矩阵的选取问题转化为了对核矩阵的选取问题,从而可以把核选取的方法用在拉普拉斯矩阵的选取上.

### 2.2 自适应谱聚类算法 ASC

给定一组样本数量为 $n$,样本维数为 $l$ 的样本集 $S=\{s_1,s_2,\cdots,s_n\}\in\mathbf{R}^l$,将其划分为 $c$ 个聚类中,并给定一个大小为 $h$ 的尺度参数集 $\Sigma=\{\sigma_1,\sigma_2,\cdots,\sigma_h\}$,则本文提出的 ASC 算法的主要思想为:首先根据尺度参数集 $\Sigma$ 中的不同尺度参数构造出不同的相似性矩阵,进而得到不同的拉普拉斯矩阵,然后用迭代的方法(核选取的方法)求解出一个新的拉普拉斯矩阵,利用这个新得到的拉普拉斯矩阵进行特征值分解以及后续的 K-means 聚类,具体步骤如下:

(1) 构造一组相似性矩阵 $\boldsymbol{A}^{(k)}\in\mathbf{R}^{n\times n}$,定义为 $A_{ij}^{(k)}=\exp(-\|s_i-s_j\|^2/2\sigma_k^2)$,当 $i=j$ 时,$A_{ij}^{(k)}=0$;

(2) 根据不同的相似性矩阵构造一组度对角矩阵 $\boldsymbol{D}^{(k)}$,$\boldsymbol{D}^{(k)}$ 主对角线上元素 $D_{ii}^{(k)}$ 为相应的相似性矩阵 $A^{(k)}$ 的第 $i$ 行所有元素之和,进而构造出一组拉普拉斯矩阵 $\boldsymbol{L}^{(k)}=\boldsymbol{D}^{(k)^{-1/2}}\boldsymbol{A}^{(k)}\boldsymbol{D}^{(k)^{-1/2}}$;

(3) 初始化 $\boldsymbol{N}$ 并赋 $t\leftarrow 1$;

(4) 对于给定的 $\boldsymbol{N}_{t-1}$,计算矩阵 $\boldsymbol{F}_M=\sum_{k=1}^{n}\boldsymbol{L}_k\boldsymbol{N}\boldsymbol{N}^{\mathrm{T}}\boldsymbol{L}_k^{\mathrm{T}}$ 的前 $d$ 个最大特征值所对应的特征向量 $\boldsymbol{M}_t=(m_1,m_2,\cdots,m_d)$;

(5) 对于给定的 $\boldsymbol{M}_t$,计算矩阵 $\boldsymbol{F}_N=\sum_{k=1}^{n}\boldsymbol{L}_k\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{L}_k^{\mathrm{T}}$ 前 $g$ 个最大特征值所对应的特征向量 $\boldsymbol{N}_t=(n_1,n_2,\cdots,n_g)$;

(6) 置 $t\leftarrow t+1$,然后返回第 4 步直至收敛;

(7) 将得到的 $\boldsymbol{M}$ 进行归一化,并记归一化后的矩阵为 $\boldsymbol{Y}$,$Y_{ij}=M_{ij}/(\sum_j M_{ij}^2)^{1/2}$;

(8) 把 $\boldsymbol{Y}$ 的每一行看作为空间 $\mathbf{R}^l$ 中的样本,然后将这些样本用 K-means 算法进行聚类;

(9) 最后,把最初的样本点 $s_i$ 划分为第 $j$ 聚类当且仅当矩阵 $\boldsymbol{Y}$ 的第 $i$ 行被划分为第 $j$ 聚类.

## 3 实验结果及分析

在本节中将通过一系列实验来验证 ASC 算法的有效性.首先在两个 Toy 数据集上测试一下 ASC 算法并直观的观察聚类效果,然后在 UCI 数据集[13]上作进一步的实验,并将 ASC 算法和谱聚类算法及现有参数选择的方法进行对比.

### 3.1 Toy problem 实验

Toy 样本数据集 Toy1 和 Toy2 的详细信息如表 1 所示,图 1 与图 2 分别画出了两个数据集.

表 1 Toy1 和 Toy2 数据集

Table 1 Toy1 and Toy2 datasets

| 名称 | 维数 | 样本数 | 类别数 | 噪声 |
|---|---|---|---|---|
| Toy1 | 2 | 98 | 2 | 高斯 |
| Toy2 | 2 | 147 | 3 | 高斯 |

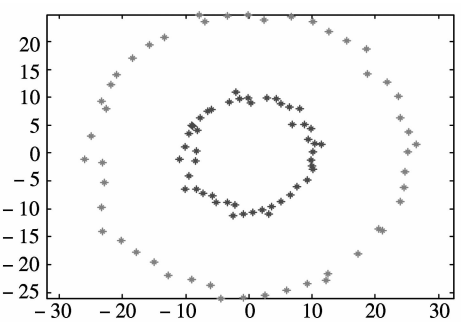


图 1 2D Toy1 数据集,分为内外 2 个环形类

Fig.1 2D Toy1 dataset with 2 classes corresponding to the outer/inner circles respectively

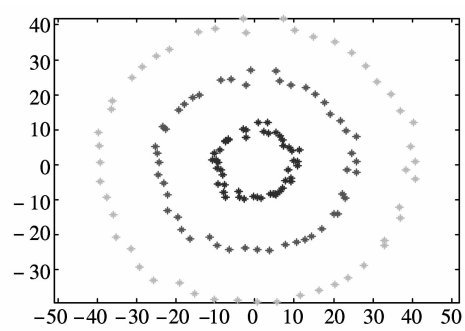


图 2 2D Toy2 数据集,分为内外共 3 个环形类

Fig.2 2D Toy2 dataset with 3 classes corresponding to the outer/inner circles respectively

实验设置如下:相似性矩阵定义为 $A_{ij}^{(k)}=\exp(-\|s_i-s_j\|^2/2\sigma_k^2)$,其中,$\sigma_k=\sigma_0\times i, i=1,2,\cdots,5$,$\sigma_0$ 是样本数据的标准差.算法中的参数取值:

$d=1$, $g=3$.在 Self-Tuning 谱聚类算法中,根据本文中的说明,选取 $K=7$[10].

在下面的实验中,取 $\sigma_k=\sigma_0\times r$,其中 $r$ 是一个范围从 0.5 到 10 的随机数(如果 $r$ 小于 0.5,得出的拉普拉斯矩阵可能无法计算出特征值).重复试验 50 次,同样计算了 ASC 算法的聚类正确率的均值(*mu*)和方差(*sigma*).算法效果如图 3 所示.

表 2　不同算法在 Toy 数据集上的比较

Table 2　Comparisons among different algorithms on Toy datasets

| 数据集 | Spectral | | | | | Self-Tuning | A-Spectral |
|---|---|---|---|---|---|---|---|
| | $s_0$ | $2\times s_0$ | $3\times s_0$ | $4\times s_0$ | $5\times s_0$ | | |
| Toy1 | 0.5 | 0.510 2 | 0.510 2 | 0.510 2 | 0.510 2 | 1 | 1 |
| Toy2 | 0.340 1 | 0.346 9 | 0.346 9 | 0.346 9 | 0.346 9 | 0.666 7 | 0.938 8 |

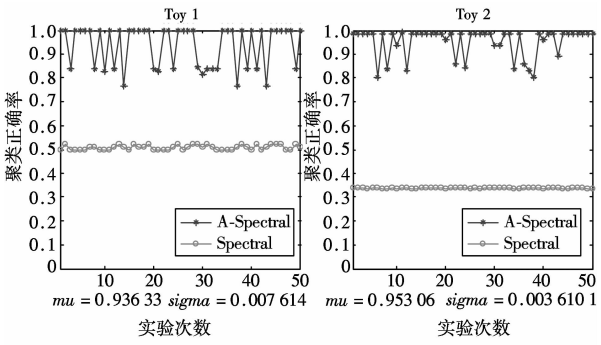


图 3　算法在 Toy1 和 Toy2 上的效果

Fig.3　Comparisons between ASC and spectral algorithm on Toy datasets

由表 2 可知, ASC 算法在人工数据集上的聚类效果明显优于 NJW 算法和 Self-Tuning 算法.如图 3 所示,在重复 50 次的实验中,ASC 算法的聚类效果仍然优于谱聚类算法.在图 3 中没有给出 Self-Tuning 的聚类效果曲线,原因是 Self-Tuning 算法的聚类效果是固定不变的,其聚类精度与表 2 给出的一致.

### 3.2　UCI 数据集上的实验

表 3 介绍了实验中用到的 UCI 数据集.

表 3　实验中用到的 UCI 数据集

Table 3　UCI datasets used in our experiments

| 名称 | 维数 | 样本数 | 类别数 |
|---|---|---|---|
| ionosphere | 34 | 351 | 2 |
| Tea | 5 | 151 | 3 |
| SPECT | 22 | 267 | 2 |
| Haberman | 3 | 306 | 2 |
| Parkinson | 22 | 195 | 2 |
| Wpbc | 33 | 194 | 2 |

在 UCI 数据集上采用了和在 Toy 数据集上相同的实验配置,首先仍然进行固定参数的组合实验,即 $\sigma_k=\sigma_0\times i, i=1,2,\cdots,5$, $\sigma_0$ 是样本数据的标准差,实验结果见表 4.进一步,不再固定尺度参数 $\sigma_k$,取 $\sigma_k=\sigma_0\times r$,其中 $r$ 是一个范围从 0.5 到 10 的随机数.在每个数据集上重复实验 50 次,并计算了均值 *mu* 和方差 *sigma*,实验结果见图 4.

表 4　不同算法在 UCI 数据集上的比较

Table 4　Comparisons among different algorithms on UCI datasets

| 数据集 | Spectral | | | | | Self-Tuning | A-Spectral |
|---|---|---|---|---|---|---|---|
| | $s_0$ | $2\times s_0$ | $3\times s_0$ | $4\times s_0$ | $5\times s_0$ | | |
| ionosphere | 0.695 2 | 0.703 7 | 0.703 7 | 0.703 7 | 0.703 7 | 0.695 2 | 0.814 8 |
| Tea | 0.357 6 | 0.377 5 | 0.344 4 | 0.344 4 | 0.344 4 | 0.423 8 | 0.410 6 |
| SPECT | 0.554 3 | 0.546 8 | 0.546 8 | 0.546 8 | 0.554 3 | 0.550 6 | 0.775 3 |
| Haberman | 0.522 9 | 0.529 4 | 0.532 7 | 0.532 7 | 0.539 2 | 0.562 1 | 0.653 6 |
| Parkinson | 0.676 9 | 0.666 7 | 0.671 8 | 0.687 2 | 0.661 5 | 0.533 3 | 0.784 6 |
| Wpbc | 0.567 0 | 0.536 1 | 0.525 8 | 0.536 1 | 0.603 1 | 0.515 5 | 0.613 4 |

由表 4 可知,自适应谱聚类算法 ASC 在 UCI 数据集上的聚类效果明显优于 NJW 算法和 Self-Tuning 算法.如图 4 所示,在重复 50 次的实验中 ASC 算法的聚类效果同样优于谱聚类算法.同样是因为 Self-Tuning 算法的聚类效果是固定不变的,在图 4 中没有给出 Self-Tuning 的聚类效果曲线,其聚类精度同表 4 给出的一致.

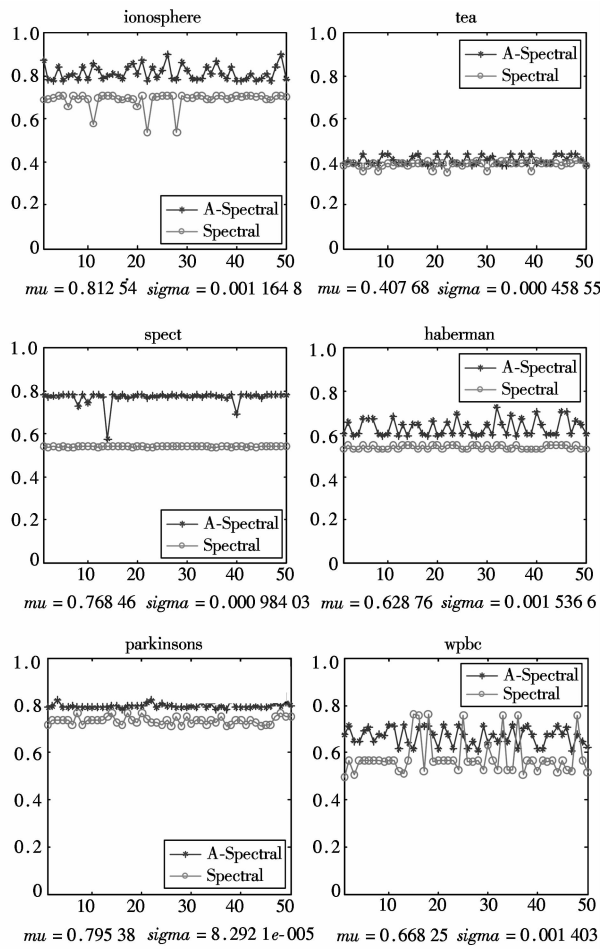


图 4 算法在 UCI 数据集上的比较

Fig.4 Comparisons among different algorithms on UCI datasets

## 4 总结与展望

在传统谱聚类算法的基础上,结合核选取的技巧,提出了自适应谱聚类算法.该算法可以自动选取谱聚类算法中的尺度参数,并且在人工数据集和 UCI 数据集上验证了所提算法的性能.目前该算法的缺点是在数据样本数量较多的时候比较耗时,其原因是谱聚类在样本数量较大的时候本身效率就不高,而且是通过迭代来求解尺度参数.在未来的工作中,将进一步研究算法的优化问题,提高算法运行效率.

**致谢**:感谢陈松灿教授对本文工作所提的宝贵意见.

**参考文献**:

[1] JAIN A, MURTY M, FLYNN P. Data clustering: a review [J]. ACM Computing Survey, 1999, 31(3):264-323.

[2] BEZDEK C. Pattern recognition with fuzzy objective function algorithms[M]. Norwell, MA: Kluwer Academic Publishers, 1981.

[3] FiEDLER M. Algebraic connectivity of graphs[J]. Czechoslovak Mathematical Journal, 1973, 23(98):298-305.

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

[5] NG A, JORDAN M, WEISS Y. On spectral clustering: analysis and an algorithm[C]// Advances in Neural Information Processing Systems (NIPS). Cambridge, MA: MIT Press, 2002.

[6] SHI J, MALIK J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8):888-905.

[7] KANNAN R, VEMPALA S, VETTA A. On clusterings-good, bad, and spectral[C]// Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science. [S.l.]: IEEE Press, 2000.

[8] HUANG L, YAN D, JORDAN M. Spectral clustering with perturbed data[C]// Advances in Neural Information Processing Systems (NIPS). Cambridge, MA: MIT Press, 2008: 705-712.

[9] CHI Y, SONG X, ZHOU D. Evolutionary spectral clustering by incorporating temporal smoothness[C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose, California: ACM Press, 2007: 153-162.

[10] ZELNIK MANOR L, PERONA P. Self-tuning spectral clustering[C]// Advances in Neural Information Processing Systems (NIPS). Cambridge, MA: MIT Press, 2004.

[11] DHILLON I, GUAN Y, KULIS B. Kernel k-means, spectral clustering and normalized cuts[C]// Proceedings of the 10th International Conference on Knowledge Discovery and Data mining. Seattle, WA, USA: ACM Press, 2004.

[12] ZHANG D, ZHOU Z H, CHEN S. Adaptive kernel principal component analysis with unsupervised learning of kernels [C]// Proceedings of the 6th IEEE International Conference on Data Mining (ICDM'06). Hong Kong, China: IEEE Press, 2006.

[13] BLAKE C, KEOGH E, MERZ C J. UCI repository of machine learning databases [DB/OL]. [2009-05-25]. http://www.ics.uci.edu/~mlearn/MLRepository.html.

(编辑:许力琴)