http://qks.cqu.edu.cn

doi:10.11835/j.issn.1000-582X.2015.06.020

K 近邻的自适应谱聚类快速算法

范 敏[1],王 芬[1],李泽明[1],李志勇[2],张晓波[2]
(1.重庆大学 自动化学院,重庆 400030; 2.国网重庆市电力公司 江北供电分公司,重庆 401147)

摘 要:谱聚类算法建立在谱图划分理论基础上,与传统的聚类算法相比,它具有能在任意形状的样本空间上聚类且收敛于全局最优解的优点。然而,谱聚类算法涉及如何选取合适的尺度参数 σ 构造相似度矩阵的问题。并且,在处理大规模数据集时,聚类的过程需要较大的时间和内存开销。研究从构造相似度矩阵入手,以传统 NJW 算法为基础,提出一种基于 K 近邻的自适应谱聚类快速算法 FA-SC。该算法能自动确定尺度参数 σ;同时,对输入数据集分块处理,并用基于 K 近邻的稀疏相似度矩阵保存样本信息,减少计算的内存开销,提高了运行速度。通过实验,与传统谱聚类算法比较,FA-SC 算法在人工数据集和 UCI 数据集上能够取得更好的聚类效果。

关键词:谱聚类;K 近邻;稀疏矩阵;自适应;快速算法
中图分类号:TP301.6 **文献标志码**:A **文章编号**:1000-582X(2015)06-147-06

A fast algorithm for adaptive spectral clustering based on K-nearest neighbors

FAN Min[1], *WANG Fen*[1], *LI Zeming*[1], *LI Zhiyong*[2], *ZHANG Xiaobo*[2]
(1. School of Automation, Chongqing University, Chongqing 400030, P.R.China;
2. Chongqing Jiangbei Branch of State Grid Corporation of China, Chongqing 401147, P.R.China)

Abstract: Based on spectral partition theory, spectral clustering algorithms are effective to solve the clustering of arbitrary sphere of sample spaces, and they can converge to global optimal solution. However, spectral clustering algorithms have to adopt the appropriate scaling parameter to calculate the whole similarity matrix, which may have a great impact on the clustering results. Moreover, when the number of data instances is large, computational complexity and memory use of the algorithm will greatly increase. So, we propose a fast algorithm for adaptive spectral clustering based on K-nearest neighbors, which can choose the scaling parameter automatically. Meantime, we divide the data set into different blocks and compute it separately. We also construct sparse matrix via retaining nearest neighbors to overcome the computational and the memory difficulties. Compared with traditional spectral clustering algorithms, experimental results show this algorithm can achieve better clustering effect on artificial datasets and UCI public databases.

Key words: spectral clustering; K-nearest neighbors; sparse matrix; adaptive; fast algorithm

聚类分析是数据挖掘领域的研究热点,是人们认识和探索事物之间内在联系的有效手段。聚类分析就

收稿日期:2015-07-12
基金项目:国家电网公司科技资助项目(SGZQJB00FZJS1400341),重庆市科技攻关资助项目(CSTC2012GG-YYJS40008)。
Supported by Science and Technology Project Funding(SGCQJB00FZJS1400341)of State Grid Corporation and Science and Technology Research Projects of Chongqing(CSTC2012GG-YYJS40008).
作者简介:范敏(1975-),女,副教授,主要从事数据挖掘、智能控制与智能管理、人工智能及应用方向研究,(E-mail)fanmin@cqu.edu.cn。

http://qks.cqu.edu.cn

是把对象按照性质上的亲疏程度分成多个类或簇，使得类或簇内的数据具有较高相似度，类或簇间的数据具有较高的相异度[1]。它不需要先验知识或假设，因此是一种无监督的学习过程。传统的聚类算法有k-means算法、EM算法、模糊C均值(FCM)等。这些算法仅在具有凸形结构的样本空间上有较好的效果，而当样本空间为非凸时，算法易陷入局部最优解。

谱聚类算法是近年来广受关注的一种高性能计算方法，它建立在谱图划分理论基础上，将聚类问题转化为图的最优划分问题，使得子图内部的相似度最大，子图之间的相似度最小[2]。谱聚类算法克服了传统聚类算法的缺点，具有明显的优势。比较典型的谱聚类算法有Perona和Freeman提出的PF算法[3]，Shi和Malik提出的SM算法[4]，Ng、Jordan和Weiss等[5]提出的NJW算法。其中NJW算法要首先根据样本空间构建相似度矩阵$\boldsymbol{W}$，这会涉及尺度参数σ的选取问题，σ的取值对聚类结果影响较大，往往依赖于领域知识和个人经验。Ng等[5]同时也给出了一种选择σ的方法，通过反复运行NJW算法来自动确定σ的大小，这消除了人为因素，却增加了运算时间。文献[6]提出了一种自调整的谱聚类算法，该算法为每个样本点指定一个σ_i，以此取代全局σ构建相似度矩阵，然而，σ_i的确定也依赖于一定的经验值。此外，当处理大规模数据集时，构造相似度矩阵和求取拉普拉斯矩阵的特征向量都需要很大的内存开销和计算时间。对此问题，Fowlkes等[7]提出使用Nyström逼近方法减少求解特征问题的计算复杂度；Yan等[8]提出利用K均值算法或者RP-tree将数据分成若干微簇，然后对每个微簇选择的代表点进行聚类，最后对应得到所有数据的类标。

以NJW谱聚类算法为基础，提出了一种基于K近邻的自适应谱聚类快速算法(Fast algorithm for adaptive spectral clustering based on K- nearest neighbors，FA-SC)。该算法能够根据输入数据集的空间分布，自动地确定自适应尺度参数σ_i，用于取代全局值，消除了人为选取参数的不确定性，使聚类结果更符合实际；同时，对输入数据集分块计算，构造基于K近邻的相似度矩阵$\boldsymbol{W}$，并对$\boldsymbol{W}$进行稀疏化处理，大大减少了内存开销和计算复杂度。实验结果表明，提出的FA-SC算法简化了输入参数的选取，减少了运行时间，能够更有效地处理大数据集聚类问题。

1 谱聚类算法

谱聚类的思想来源于谱图划分理论[9]。假定将每个数据样本看作图G中的顶点V，根据样本间的相似度将顶点间的边E赋权重值，就得到一个基于样本相似度的无向加权图G(V，E)，那么聚类问题就转化为图G的最优划分问题，划分准则就是使划分成的子图内部相似度最大，子图之间的相似度最小。考虑问题的连续放松形式，可将图划分问题转换成求解相似矩阵或拉普拉斯矩阵的谱分解，可以认为谱聚类是对图划分准则的逼近[10]。

1.1 图的矩阵表示

谱聚类首先构造样本空间的相似度矩阵，用$\boldsymbol{W}(\boldsymbol{W}\in\boldsymbol{R}^{n\times n})$表示。相似矩阵中包含了聚类所需的全部信息，如果相似矩阵具有优良的性质，可以预期谱聚类算法的表现也会令人满意。通常用高斯核函数计算$\boldsymbol{W}$，由公式(1)给出。其中，x_i，x_j表示不同的样本点，$\|x_i-x_j\|$取欧氏距离，$\sigma(\sigma>0)$是人为指定的尺度参数，决定w_{ij}随样本点之间距离的衰减速度。显然，$\boldsymbol{W}$为n阶(n表示样本容量)对角矩阵，且对角元素为0。

$$\boldsymbol{w}_{ij}=\begin{cases}\exp\left(-\dfrac{x_i-x_j^{\,2}}{2\sigma^2}\right), & i\neq j\ ;\\ 0, & i=j\ 。\end{cases}\tag{1}$$

将相似度矩阵$\boldsymbol{W}$的每行元素相加，得该顶点$v_i(v_i(V)$的度d_i，由公式(2)定义，以所有度值为对角元素构成的对角矩阵称为度矩阵，通常用$\boldsymbol{D}$表示。

$$\boldsymbol{d}_i=\sum_{j=1}^{n}\boldsymbol{w}_{ij},\tag{2}$$

图的拉普拉斯矩阵分为非规范型和规范型2种。非规范型拉普拉斯矩阵表示为：$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$。规范型拉普拉斯矩阵有2种形式，用公式(3)和公式(4)表示。选用不同的拉普拉斯矩阵所得的聚类结果也会有差别。如何根据具体环境选择合适的拉普拉斯矩阵，还需进行大量的理论研究和实验工作。

$$\boldsymbol{L}_{\mathrm{sys}}=\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}},\tag{3}$$

$$\boldsymbol{L}_{rw}=\boldsymbol{D}^{-1}\boldsymbol{W}。\tag{4}$$

http://qks.cqu.edu.cn

1.2 NJW 谱聚类算法描述

研究提出的基于 K 近邻的自适应谱聚类快速算法(FA-SC)是以 NJW 算法为基础,因此,给出 NJW 算法的处理过程[5]为

1)根据公式(1)构造样本空间的相似度矩阵 $\boldsymbol{W}(\boldsymbol{W}\in\boldsymbol{R}^{n\times n})$,取欧氏距离,尺度参数 σ 由人为指定。

2)根据公式(3)计算规范化的拉普拉斯矩阵 $\boldsymbol{L}_{\text{sys}}$,求解 $\boldsymbol{L}_{\text{sys}}$ 的前 k 个最大特征值对应的特征向量 $\boldsymbol{x}_1$,$\boldsymbol{x}_2$,$\cdots\boldsymbol{x}_k$(必要时正交化处理),建立矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots\boldsymbol{x}_k]$,其中 $\boldsymbol{x}_i$ 为列向量。

3)对 $\boldsymbol{X}$ 的行向量归一化处理,处理后得到矩阵 $\boldsymbol{Y}$,其中 $\boldsymbol{Y}_{ij}=\dfrac{\boldsymbol{x}_{ij}}{\sqrt{\sum_{j=1}^{k}\boldsymbol{x}_{ij}^2}}\in\boldsymbol{R}^{n\times k}$。

4)将矩阵 $\boldsymbol{Y}$ 的每一行看成是 $\boldsymbol{R}^k$ 空间中的一个数据点,使用 k-means 算法把 n 行数据分为 k 个聚类 A_1,$A_2\cdots A_k$。

5)当矩阵 $\boldsymbol{Y}$ 的第 i 行在类 A_j 中时,划分原样本空间中样本点 $\boldsymbol{x}_i$ 到 C_j 类。

从 NJW 算法的实现过程可知,使用高斯核函数构造相似度矩阵 $\boldsymbol{W}$,其中,尺度参数 σ 要求人工设置,而 σ 的取值依赖于领域知识和经验,没有一定的规律可循,其取值将直接影响聚类结果的好坏。并且,相似度矩阵 $\boldsymbol{W}$ 中保存了所有样本点之间的信息,然而在实际中,并不是每 2 个点的 w_{ij} 都是有意义的。尤其对于大规模数据集来说,计算、存储相似度矩阵和特征向量都需要较大的时间和内存开销,非常不利于算法的扩展。

2 基于 K 近邻的自适应谱聚类快速算法(FA-SC)

如何自动的确定尺度参数 σ,加快算法运行速度,并作用于大规模数据集,是研究工作的关键。因此,提出了一种基于 K 近邻的自适应谱聚类快速算法(FA-SC 算法),该算法的处理过程与 NJW 算法相似,区别在于第一步中构造相似度矩阵和加快算法运行速度。具体的研究工作分为以下几个部分:

1)根据样本分布自动确定合适的尺度参数 σ;

2)对输入数据集分块处理,将构造相似度矩阵 $\boldsymbol{W}$ 的过程分成多步进行;

3)采取保存样本点 K 个最近邻的距离值 $\boldsymbol{w}_{ij}$ 的策略,对相似度矩阵 $\boldsymbol{W}$ 进行稀疏化处理。

2.1 尺度参数 $\boldsymbol{\sigma}$ 的选取

通过手动设置尺度参数 σ 的值是很困难的,σ 的取值应适应样本的具体分布,全局值的 σ 难以反映出这种分布情况[11]。考虑到采用全局 σ 的局限性,不妨对每个样本点定义一个自适应尺度 σ_i,使样本点具备"自适应尺度"的属性[12]。σ_i 的取值由样本点 $\boldsymbol{x}_i$ 的 K 个最近邻决定,其计算方法由公式(5)给出。

$$\sigma_i=\frac{1}{K}\sum_{m=1}^{K}\boldsymbol{x}_i-\boldsymbol{x}_m\ 。\tag{5}$$

由公式(5)可知,σ_i 表示样本点 x_i 和其 K 个最近邻距离的平均值,因此,σ_i 称为近邻自适应尺度,相应的相似度矩阵 $\boldsymbol{W}$ 可由公式(6)定义。

$$\boldsymbol{w}_{ij}=\begin{cases}\exp\left(-\dfrac{\boldsymbol{x}_i-\boldsymbol{x}_j{}^2}{\sigma_i\cdot\sigma_j}\right), & i\neq j\ ;\\ 0, & i=j\ 。\end{cases}\tag{6}$$

σ_i 反映了近邻分布的变化,能够自适应于局部结构,使簇内点的相似度增大,簇间点的相似度减小。以图 1 为例,图中分布着 2 个密度相差较大的簇 C_1 和 C_2,且 x_t、$x_q\in C_1$,$x_p\in C_2$,x_p 和 x_q 均在 x_t 的 K 邻域内。为便于说明问题,这里假设 $x_t-x_p=x_t-x_q$,由于簇 C_1 密度小于簇 C_2,由公式(5)可得近邻自适应尺度 $\sigma_p<\sigma_q$,再由公式(6)可得 $w_{tp}<w_{tq}$,若采用全局尺度参数,由公式(1)可得 $w_{tp}=w_{tq}$。显然,采用近邻自适应尺度可使同一簇中样本间 $\boldsymbol{w}_{ij}$ 值大于不同簇中的相应值,这样更利于聚类。

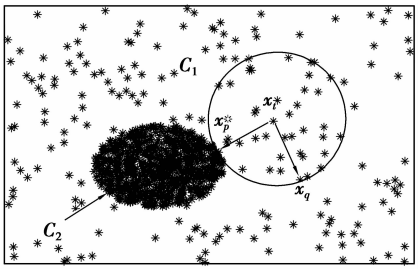


图 1　近邻自适应尺度原理

Fig.1　Principle of nearest-neighbor scaling

2.2 数据集分块计算与构造稀疏矩阵

谱聚类的运行过程中不可避免的要计算拉普拉斯矩阵的特征值与特征向量,非稀疏矩阵的计算复杂度为 $O(n^3)$。同时考虑对海量数据聚类的情况,假设输入数据集的规模

http://qks.cqu.edu.cn

为 n，则要根据公式(7)计算 C_n^2 次欧氏距离。此时计算的相似度矩阵 $\boldsymbol{W}$ 中共保存了 n^2 个数据值，计算这些数据也要考虑时间和内存的开销。

$$\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2=\|\boldsymbol{x}_i\|^2+\|\boldsymbol{x}_j\|^2-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j。\tag{7}$$

为了减小一次计算的内存消耗，在计算样本欧氏距离的过程中，采取多步计算的方法，将输入数据集分成若干块，分别计算每一块与整个样本的距离值，最后将每步计算的结果合并，得到距离矩阵。由于每一块的数据规模远小于 n，这样就大大降低了内存负担。数据集分块计算的过程由图 2 所示。

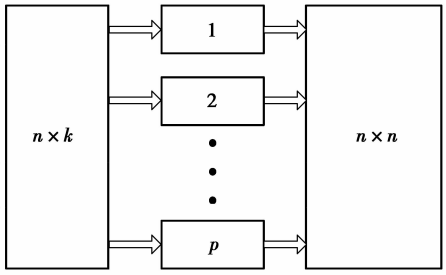


图 2 数据集分块计算过程

Fig.2 The calculating procedure of data set blocks

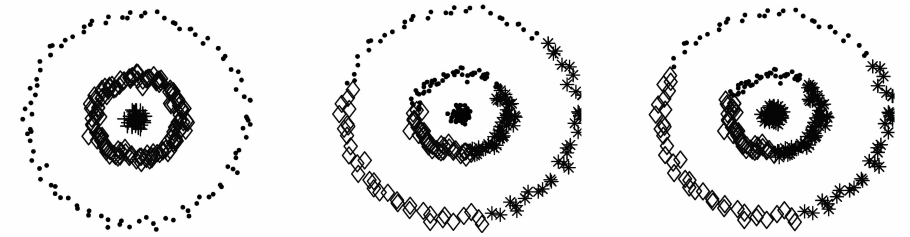
数据集分块计算虽然减轻了一次计算的内存开销，但要保存距离矩阵依然要占用较大的系统资源，甚至会超出计算机的内存。一个好的方法是对距离矩阵进行稀疏化处理，考虑到对每个样本点都要计算 n 次距离，这 n 个距离值中大部分属于无意义的信息，其中离样本点越近的距离值对聚类结果的影响往往越大。因此，为了得到稀疏距离矩阵，只保存样本点的 K 个最近邻的距离值，其他值归零，这样便可获得稀疏化的距离矩阵，再由稀疏距离矩阵利用公式(6)、(3)可进一步构造稀疏化的相似度矩阵 $\boldsymbol{W}$ 和拉普拉斯矩阵 $\boldsymbol{L}_{\mathrm{sys}}$，根据稀疏化的拉普拉斯矩阵 $\boldsymbol{L}_{\mathrm{sys}}$ 计算特征值和特征向量将变得简单[13]。

3 实验与分析

为验证提出的 FA-SC 算法的有效性，作者将该算法与 NJW 算法作对比，在人工数据集和 UCI 数据集上做了 3 组实验。实验环境为 Intel Core 2 E7400 处理器，2GB 内存，Windows XP SP3 操作系统，matlab 平台。

3.1 尺度参数自适应实验

NJW 算法在运行之前需要手动设置全局尺度参数 σ，且 σ 的取值会对聚类结果产生影响，这使它在应用上具有极大的局限性。FA-SC 算法能够很好的克服 NJW 算法的不足，根据输入样本的分布自动地选取尺度参数以获取最佳的聚类效果。作者在 3 个人工数据集 D_1、D_2、D_3 上进行了验证，对比实验的结果如图 3～图 5 所示。



(a) FA-SC (b) NJW(σ=0.48) (c) NJW(σ=0.16)

图 3 数据集 D_1 上聚类结果

Fig.3 Clustering results of data set D_1

(a) FA-SC (b) NJW(σ=0.21) (c) NJW(σ=0.05)

图 4 数据集 D_2 上聚类结果

Fig.4 Clustering results of data set D_2

http://qks.cqu.edu.cn

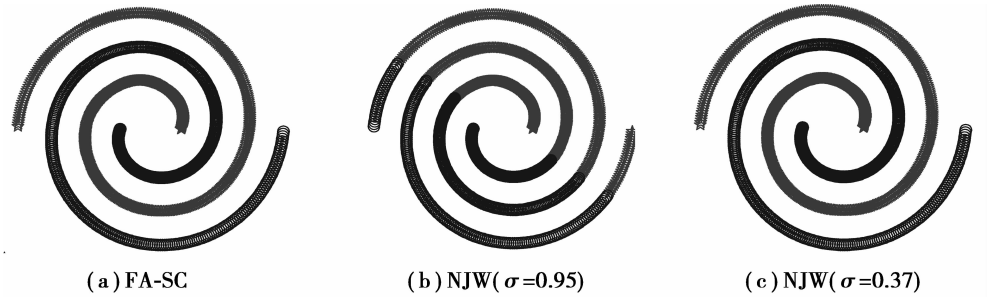


图 5　数据集 D_3 上聚类结果

Fig.5　Clustering results of data set D_3

作者在 3 个数据集上选取了多组参数进行实验。图 3、4 中的数据集都有 3 个簇,分别改变 NJW 算法中尺度参数 σ 的值,均得到了错误的聚类结果,只有 FA-SC 算法发现了正确的簇。图 5 中的数据集有 2 个簇,当尺度参数 $\sigma=0.37$ 时,NJW 算法得到了正确的聚类结果,改变 σ 的取值会导致错误的划分。对比试验说明,NJW 算法的聚类结果易受尺度参数 σ 的影响,而 σ 的取值很难确定,提出的 FA-SC 算法能够自适应地选取 σ 值,不用手动设置,实现了较好的聚类效果。

3.2　算法复杂度分析

假设输入的样本空间为 $n\times d$(n 表示样本容量,d 表示维度)。FA-SC 算法第一步计算样本点间欧氏距离并获得基于 K 近邻的稀疏距离矩阵,其时间复杂度为 $O(n^2d)+O(n^2\log K)$,空间复杂度为 $O(nK)$;第二步计算各样本点的近邻自适应尺度参数 σ_i,时间复杂度为 $O(nK)$;第三步求解规范化拉普拉斯矩阵 $\boldsymbol{L}_{sys}$ 的 k 个最大特征值和特征向量,采用文献[12]使用的 ARPACK 计算方法[14],其时间复杂度为$(O(m^3)+(O(mn)+O(nK))\times O(m-k))\times I$,空间复杂度为 $O(nK)+O(mn)$,其中,m 通常取值约为 k 几十倍,I 为迭代次数,最后调用一次 k-means 算法,时间复杂度为 $O(Ik^2n)$,其中,I 为迭代次数,且 $I<<n$。

图 6 中将 FA-SC 算法与 NJW 算法对比,测试 2 种算法在不同规模样本集上的时间开销。从图 6 中可以看到,FA-SC 算法执行速度比 NJW 算法快,而且随着样本规模的增加,这种执行速度的差距也越来越明显。这表明,FA-SC 算法降低了聚类时间开销,对数据集的规模大小具有很好的扩展性。

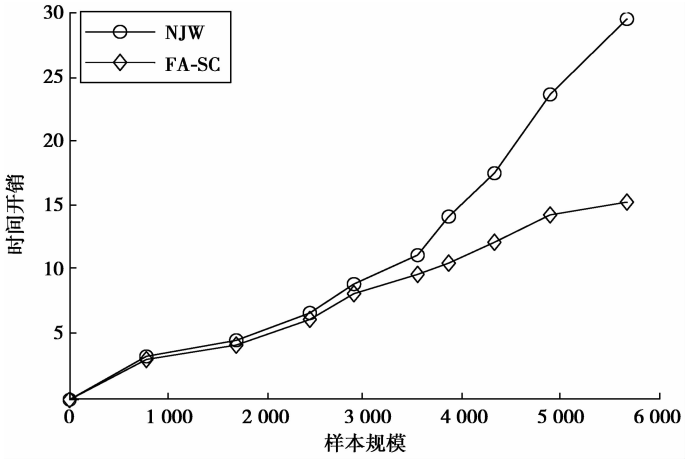


图 6　两种算法聚类时间对比

Fig.6　Contrastive time complexity of the two algorithms

3.3　聚类结果的评价

为了更客观的评价聚类结果,笔者在几个 UCI 真实数据集上作对比试验,评价聚类结果的准确率。采用文献[15]使用的 Rand 指标计算准确率,假设 C 为参考聚类结果,P 为实际聚类结果,则准确率的计算方法由公式(8)给出(其中,a:属于 C、P 中同一类的样本对数目;b:属于 C 中同一类,P 中不同类的样本对数目;c:属于 C 中不同类,P 中同一类的样本对数目;d:属于 C 中不同类,P 中不同类的样本对数目)。

$$RI=\frac{a+d}{a+b+c+d}。\tag{8}$$

http://qks.cqu.edu.cn

表1为FA-SC算法与NJW算法的准确率对比，由对比结果可知，FA-SC算法比NJW算法的准确性较高。

表1 聚类结果准确率对比

Table 1 Contrastive accuracy rate of the clustering results

Dataset	Algorithm	Clusters	Attributes	Parameters	Accuracy
Iris	FA-SC	3	4	$K=8$	0.886 1
	NJW			$\sigma=1.1$	0.846 5
Wine	FA-SC	3	13	$K=10$	0.972 1
	NJW			$\sigma=0.9$	0.952 3
Glass	FA-SC	7	9	$K=11$	0.843 1
	NJW			$\sigma=0.76$	0.835 2

4 结 语

分析了传统NJW谱聚类算法的原理和不足，提出了一种基于K近邻的自适应快速谱聚类算法(FA-SC)，能够根据输入数据集的分布自动确定尺度参数，简化了NJW算法的参数选取；同时，针对大数据集的情况，采用数据集分块计算策略构造稀疏化的相似度矩阵，减小了计算特征值和特征向量的内存开销和时间复杂度。在多个数据集上的实验结果表明，该算法可以产生较好的聚类结果并提高了聚类速度。然而，FA-SC算法依然需要事先给出样本的聚类数目，事实上，这也是谱聚类算法的难题之一。下一步，还要在如何自动确定给定数据集的最佳聚类数目上做进一步研究。

参考文献：

[1] Jain A K, Murty M N, Flynn P J. Data clustering: a review[J]. ACM Computing Surveys, 1999,31(3): 264-323.

[2] Filippone M, Camastra F, Masulli F, et al. A survey of kernel and spectral methods for clustering[J]. Pattern Recognition, 2008, 41(1): 176-190.

[3] Perona P, Freeman W T. A factorization approach to grouping[C]// 5th European Conference on Computer Vision Freiburg, June, 2-6, 1998, Germany. Springer-Verlag: Springer Berlin Heidelberg, 1998, 1406: 655-670.

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

[5] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm[J]. Advances in Neural Information Processing Systems, 2001, 14: 849-856.

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

[7] Fowlkes C, Belongie S, Chung F, et al.Spectral grouping using the Nystrom method[J], IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(2): 214-225.

[8] Yan D, Huang L, Jordan M I. Fast approximate spectral clustering[C]// Proceedings of the 15th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, New York,USA: IEEE, 2009:907-916.

[9] Fiedler M. Algebraic connectivity of graphs[J]. Czechoslovak Mathematical Journal, 1973, 23(2): 298-305.

[10] Malik J, Belongie S, Leung T, et al. Contour and texture analysis for image segmentation[C]// Perceptual Organization for Artificial Vision Systems. US: Springer, 2000, 546: 139-172.

[11] 卜德云，张道强.自适应谱聚类算法研究[J].山东大学学报：工学版，2009,39(5):22-26.
BU Deyun, ZHANG Daoqiang. Adaptive spectral clustering algorithm research[J]. Journal of Shandong University: Engineering Science, 2009, 39(5): 22-26. (in Chinese)

[12] 谷瑞军，叶宾，须文波.一种改进的谱聚类算法[J].计算机研究与发展，2007,44(z2.):145-149.
GU Ruijun, YE Bin, XU Wenbo. An improved spectral clustering algorithm[J]. Journal of the Computer Research and Development, 2007, 44(z2): 145-149. (in Chinese)

[13] Chen W Y, Song Y Q, Bai H J, et al. Parallel spectral clustering in distributed systems[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(3): 568-586.

[14] Lehoucg R B, Sorensen D C, Yang C. ARPACK users' guide : solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods[M]. [s. n.]:SIAM, 1998.

[15] Halkidi M, Batistakis Y, Vazirgiannis M. On clustering validation techniques[J]. Journal of Intelligent Information Systems, 2001, 17(2-3): 107-145.

(编辑 侯 湘)