第46卷第9期 中国科学技术大学学报 Vol. 46, No. 9
2016年9月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Sep. 2016

文章编号:0253-2778(2016)9-0757-07

基于核心点的大数据谱聚类算法

杨 艺,马儒宁

(南京航空航天大学理学院,江苏南京 210016)

摘要:针对谱聚类性能优异但因计算复杂度太高而无法应用于大数据的问题,提出一种将谱聚类应用于大数据的新算法.首先,基于数据相似性与随机抽样选取核心点集,并利用核心集对大数据分组;然后在核心集上应用谱聚类;最后综合核心集的聚类结果和数据的分组信息完成大数据聚类.该算法既将谱聚类推广到大数据,又通过核心点选取降低了噪声及异常数据的影响.实验充分验证了推广后的谱聚类应用于大数据的高效性.

关键词:大数据;谱聚类;核心点;数据分组

中图分类号:TP181 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2016.09.007

引用格式:杨艺,马儒宁. 基于核心点的大数据谱聚类算法[J]. 中国科学技术大学学报,2016,46(9):757-763.
YANG Yi, MA Runing. Core-points based spectral clustering for big data analysis[J]. Journal of University of Science and Technology of China, 2016,46(9):757-763.

Core-points based spectral clustering for big data analysis

YANG Yi, MA Runing

(*College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing* 210016, *China*)

Abstract: With regard to failures in applying spectral clustering to big data due to its computation complexity, a new spectral clustering algorithm for big data was proposed. Firstly, core-points based on random sampling and data similarity were selected, with which, the big data were grouped. Secondly, spectral clustering was applied to the core-points. Finally, the clustering of whole data was completed by combining the clustering result of the core-points and the grouped big data information. The algorithm both promotes the spectral clustering to big data and reduces the influence of noise or abnormal data by the core-points. A large number of experiments fully verify the effectiveness of the method proposed in this paper.

Key words: big data; spectral clustering; core-points; data group

0 引言

聚类是一种强有力的数据分析工具,它将相似的数据划分到同一类,无监督地完成数据的分组处理,对数据集的进一步分析和处理具有重要意义[1-3].作为目前聚类算法的研究热点之一,谱聚类算法[4]相比传统的聚类算法(如 K-means[5-6]和层次聚类[2]等)具有明显优势,它可以处理任意形状的数据且聚类质量高.谱聚类将聚类问题转化为图的划分问题,使子图内部相似而子图之间相异.它在数据相似性矩阵的基础上求拉普拉斯矩阵并进行特征分解,然后利用特征向量将原数据映射到低维的特征

收稿日期:2016-03-01;**修回日期**:2016-09-17

作者简介:杨艺,女,1993年生,硕士生.研究方向:聚类算法. E-mail:530123025@qq.com

通讯作者:马儒宁,博士/副教授. E-mail:mrning@nuaa.edu.cn

空间中,进而聚类.

谱聚类求解特征向量的计算复杂度为 $O(n^3)$,n 是数据点规模,因此无法直接处理大数据.如何将谱聚类应用于大数据成为亟需解决的难题,为此,研究者相继提出了一系列快速谱聚类方法.这些方法主要分为两类:一种是使用矩阵低秩近似,降低拉普拉斯矩阵特征向量的计算量[7-10];另一种则通过 K-means 等算法对大数据预处理,得到代表点子集,在小规模子集上应用谱聚类并将聚类结果推广到大数据集[11-12].

本文主要从大数据预处理的角度出发,提出一种新的基于核心点的大数据谱聚类算法(core-points based spectral clustering,CBSC).

1 研究基础

1.1 谱聚类回顾

谱聚类是一种基于图的聚类算法.它将数据点看成图的顶点,点之间的相似性作为连接顶点的边的权值,把聚类问题转化为带权无向图的划分问题.通过最优目标函数(如 normalized cut),使子图内部比较接近而子图之间距离较远.该方法的基本思路是在数据集相似性矩阵的基础上求拉普拉斯矩阵并进行特征分解,在得到的特征向量空间完成聚类.

谱聚类主要由以下3个步骤构成:

(Ⅰ)生成数据的相似矩阵 $\boldsymbol{S}$,并求出拉普拉斯矩阵 $\boldsymbol{A}=f(\boldsymbol{S})$;(在 normalized cut 中 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{S}$,标准化后为 $\boldsymbol{A}=\boldsymbol{D}^{-1/2}(\boldsymbol{D}-\boldsymbol{S})\boldsymbol{D}^{-1/2}$,其中 $\boldsymbol{D}$ 为对角阵,且对角元是 $\boldsymbol{S}$ 相应的行和);

(Ⅱ)计算 $\boldsymbol{A}$ 的前 k 个特征值和对应的特征向量,得到特征向量空间 $\boldsymbol{A}=(A_1,A_2,\cdots A_k)$;

(Ⅲ)单位化 $\boldsymbol{A}$ 的行向量并将其看成 R^k 空间中的 n 个点,利用 K-means 对这 n 个点进行聚类,再将聚类结果返回原数据(若 $\boldsymbol{A}$ 的第 i 行被分到第 j 类中,则原数据点 x_i 属于第 j 类).

谱聚类可以处理任意形状的数据,与数据输入顺序无关且得到全局的近似最优解.求解特征向量的计算复杂度为 $O(n^3)$,因此无法将其直接应用于大数据.

1.2 大数据预处理的常用方法

数据预处理就是对复杂且规模大的数据集约减压缩得到具代表性的小规模样本子集,经典的大数据聚类算法中涉及预处理的有 Rough-DBSCAN[13] 和 BIRCH[14] 算法等.在 Rough-DBSCAN 中的 Leaders 算法,首先从数据集中任取一点作为起始 leader,然后依次计算剩余数据与已有 leader 之间的距离:若大于给定阈值,则将数据点作为一个新的 leader,否则将该点分配给阈值内的某个 leader,遍历完数据得到 leader 集合以及剩余点关于 leader 的归属.它的优点是简单易操作,运行时间快存储空间小,但它受数据点输入顺序影响大,而且不能保证隶属于同一 leader 内的点之间的相似性大于不同 leader 内的点之间的相似性.BIRCH 算法则通过引入聚类特征(CF)和聚类特征树(CF 树)两个特有的概念,将大数据用树形结构压缩存储.新颖的数据结构不仅使算法节省了存储空间和运算量,而且也增强了对噪声的鲁棒性,但它对数据输入顺序十分敏感,且不能很好地处理非凸数据集.

1.3 近似谱聚类

近些年,研究者提出了一系列应用于大数据的近似谱聚类算法(approximate spectral clustering,ASC).一类是通过矩阵低秩近似,降低拉普拉斯矩阵特征分解的计算量.Fowlkes 等提出了基于 Nyström 方法的 ASC 算法[10],其原理是:利用小规模的样本集对空间中的卷积算子逼近求出近似特征向量,将该方法扩展并应用于谱聚类可以得到处理大数据的 ASC 算法,但该方法性能取决于样本集的质量,随机抽样简单快速,但其不确定性大大影响了聚类结果,多次抽样又增加算法复杂度.为了提高样本集的质量,基于 Nyström 方法扩展的 ASC 算法相继被提出.Zhang 等[8]使用 K-means 聚类的质心点作为样本集,Ding 等[9]则考虑多次遍历并更新抽样概率的自适应抽样方法,获得相对满意的聚类结果,但这类方法占用内存大,易丢失小簇且数值稳定性不高.另一类则通过预处理降低数据规模,Yan 等[11]提出了 KASC(K-means-based approximate spectral clustering),首先使用 K-means 聚类大数据得到质心点集,在小规模质心点集上应用谱聚类,剩余点类别则与最近质心的类别一致.Shinnou 等[12]也是先用 K-means 聚类大数据得到质心点集,但去掉质心点附近的点(给定阈值范围内),对剩余点应用谱聚类,去掉的点的类别与最近的质心类别一致.这类方法比基于 Nyström 方法运行时间快且所需内存小;主要缺点是由于 K-means 的局部收敛,若某一质心点聚类错误,则该质心附近点均同样聚类错误.

本文从数据预处理的角度出发,对复杂的大数据集约减压缩得到小规模样本集,然后对样本集使用谱聚类得出结果并推广到大数据集.显然,聚类结果的有效性取决于样本点选取的质量.因此本文从提高样本集的质量出发,提出新的应用于大数据的

谱聚类算法 CBSC.

2　基于核心点的大数据谱聚类(CBSC)

2.1　核心点选取与大数据分组

随机抽样简单便于操作,但其不确定性影响聚类结果,而 1.2 节中选取代表点方法对数据输入顺序敏感而且仅考虑数据的空间局部信息,容易选取过多地噪声成为代表点.针对这些问题,本文提出一种新的基于相似性的代表点选取方法,并将该方法与随机抽样结合,不但对噪声鲁棒对数据输入顺序不敏感,而且运行时间快.

2.1.1　基于相似性的核心点选取

2013 年,文献[15]引入了核心点的概念,所谓核心点是指与其他数据点相似性最高的点,但文献[15]中核心点选取的计算量太大,需求出所有数据对间的相似性.本文以其为基础并改进,不但继承了原方法对噪声鲁棒且能体现数据的全局分布的优点,而且大大降低了计算复杂度.下面是具体步骤.

核心点的选取大致可分为三步,其中 $u(x,y)$ 表示 x 和 y 之间的相似性.

(Ⅰ)对大数据 X(点数为 n)随机抽样得到样本集 X'(点数为 m);

(Ⅱ)计算 X 与 X' 中所有数据对间的相似性,并将相似性和最大的选为核心点;

(Ⅲ)若 x_{1^*},x_{2^*},…x_{K-1^*} 为已选的核心点,则第 K 个核心点按以下等式选取

$$x_K^* = \operatorname{argmax}\{\sum_{y\in X} u(x,y) - \lambda\sum_{i=1}^{K-1} u(x,x_i^*)\} \quad (1)$$

上述选取公式基于数据间的相似性,故核心点应该分布在密度较大的区域(往往位于聚类中心附近),对噪声鲁棒且能体现数据的全局性质.式(1)表明,在计算第 K 个核心点时,需减去与已选 $K-1$ 个核心点的相似性(排除已选核心点附近的点,而趋向于分散的选取其他点).松弛参数 λ 越大,减去的那项越大(与已选核心点的相似性),则选取的核心点更加分散,通过调节 λ 可使核心点体现数据的整体分布,通常可取为 1.与文献[15]中核心点的选取过程相比,本文不需要计算所有数据对间的相似性,将计算复杂度由 $O(n^2)$ 减少到 $O(nm)$.当数据集规模 n 很大时,m 一般取一定值,这时复杂度与 n 呈线性关系.

图 1 中(1～3)分别给出了文献[13]中 Leaders 算法、文献[14]中 BIRCH 算法及本文方法选取的核心(代表)点的图.实验数据集为以(0,0)、(0,2)和(2,0)为中心,标准差为 0.45 的正态分布,每部分均包含 20 个数据点,并加入整个样本空间的 30 个均匀噪声.图中圆圈标记的表示核心点.其中(1)中参数 r 取 0.8,(2)中 r 取 0.8,k 取 8,(3)中随机样本点数取 90,选取的核心点个数分别为 22,35,22 个.由图 1 看出,文献[13]和文献[14]中方法选取的核心点中均包含了较多的噪声,而本文基于相似性选取的核心点明显对噪声更加鲁棒(选取了较少的噪声或离群点).

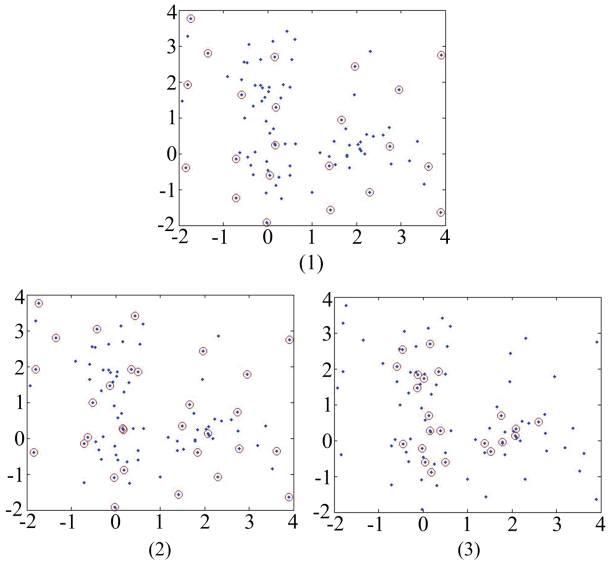

图 1　不同方法选取的核心点对比

Fig. 1　Comparison of core-points with different methods

2.1.2　基于核心点的大数据分组

文献[13]中,把与 leader 距离小于 r 的点分到同一组且以 leader 为该组中心点,即按距离实现了数据分组.文献[14]也是按距离分组但限制了每组内点的个数,若超出给定阈值,则该组分裂为两个组,即相比于文献[13],它组中元素个数相对均匀一些,但这两种分组都是将数据按照相对固定的半径分配给代表点,造成很多噪声或离群点会被单独地分为一组.

本文基于相似性选取出核心集 X^*,因此想到利用相似性完成数据的分组.对于核心集 $X^* = \{x_{1*}, x_{2*}, \cdots x_{K^*}\}$,它将 X 划分成 K 个分组,每个 x_i^* ($i=1、2\cdots K$)代表一个分组 X_i 并作为该组中心点.对于 X 中剩余非核心点 $x \in X\backslash\{X^*\}$,按式(2)分配到与之最相似的核心点所在的分组 X_{K_0}.

$$x_{k_0}^* = \operatorname*{argmax}_{1\leqslant i\leqslant k} u(x,x_i^*) \quad (2)$$

遍历完数据集,得到了大数据 X 的 K 个分组 $X_1, X_2, \cdots, X_K$,即 $X = \bigcup_{i=1}^{K} X_i$,其中 X_i 满足 $x_i^* \in X_i$,即每个分组 X_i 均由一个核心点 x_i^* 代表.基于图 1 中选出的核心点,图 2 是利用上述三种方法对于原数据的分组情况.

此处参数取值与图 1 均相同,图中黑色线段相

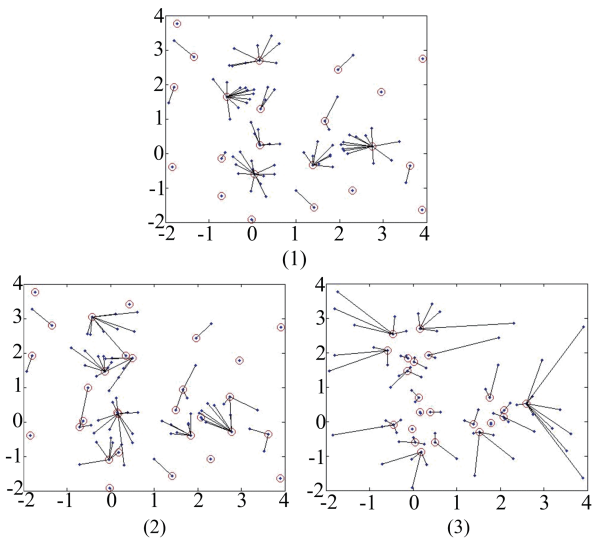

图 2 基于图 1 中核心点的数据分组

Fig. 2 Group data based on core-points in Fig. 1

连的点表示分到同一组中的点. 可以看出,由于前两种算法选取了较多的噪声点,而噪声点本身就偏离数据,因此以噪声为中心点所在的分组几乎不存在其他点,导致过多的噪声点单独作为一个分组,显然这种分组是不合理的. 而本文选取的核心点多集中在类中心附近(图 1(3)),使噪声点归入了核心点所在的分组(图 2(3)),不会单独地作为一个分组. 由此进一步说明,相比于前两种分组,本文中的数据分组更加合理.

2.2 核心集谱聚类及其推广到大数据

首先,对于 2.1.1 中选取的小规模核心集 X^*,应用谱聚类将核心集聚成 C 类,得到核心集的聚类结果 $X=\bigcup_{i=1}^{C}X^{(i)}$,其中 $X^{(i)}$ 表示核心集的第 i 类;然后,将 X^* 的聚类结果推广到大数据集 X,具体做法如下:对于非核心点 x,当 $x\in X_{K_0}$(其中 X_{K_0} 由 2.1.2 节得到,满足核心点 $x_{k_0}^*\in X_{K_0}$)时,若 $x_{k_0}^*\in X^{(i)}(1\leqslant i\leqslant C)$,则 $X^{(i)}=X^{(i)}\bigcup\{x\}$,即分组中剩余点与组中核心点的聚类一致. 遍历完非核心点集,即可得到大数据的聚类结果 $X=\bigcup_{i=1}^{C}X^{(i)}$.

2.3 CBSC 的算法实现及复杂度分析

CBSC 的算法实现如下:

输入:数据集 X,随机抽样集数目 m,核心点数目 K,聚类数目 C;

输出:数据集 X 的聚类结果

1. 利用 2.1.1 节(1)—(3)得到核心集 X^*;

2. 对非核心点,利用 2.1.2 的(2)式得到数据集 X 的分组 $X=\bigcup_{i=1}^{K}X_i$;

3. 利用谱聚类得到核心集的聚类结果 $X^*=\bigcup_{i=1}^{C}X^{(i)}$;

4. 利用 2.2 节,将每个非核心点分配到其分组中的核心点所在类中,得到最终聚类结果 $X=\bigcup_{i=1}^{C}X^{(i)}$.

CBSC 算法的计算量包括上述四个步骤. 其中步骤 1 的复杂度为 $O(nm)$,步骤 2 的复杂度为 $O(nK)$,步骤 3 的复杂度为 $O(K^3)$,步骤 4 的复杂度为 $O(n)$,注意到当数据集规模 n 很大时,m 和 K 一般取一定值,且远远小于 n,故上述四个步骤的计算量之和与 n 近似呈线性关系.

3 实验比较和分析

本节将讨论 2.1.1 节核心点选取过程中参数对选取结果的影响,并将 CBSC 与原谱聚类(简称 SC)进行比较. 其中 3.1 节是对 2.1.1 中参数讨论的实验;3.2 节是 CBSC 与 SC 的对比实验. 实验数据集为不同形状并加入噪声的人工数据集. 实验在配置为 Windows XP Intel 3.4GHz i3 处理器、4G 内存的电脑上进行.

3.1 实验数据集描述

实验数据集选取由不同参数的正态分布随机生成的 DataSet A, DataSet B 和 DataSet C,其中每个数据集均在整体样本空间中加入数据量的 20%个噪声. DataSet A 以(0,0)、(0,2)和(2,0)为中心,每个 circle 包含相等数目的数据点,属于凸状数据集;DataSet B 由两个 ring 组成,半径均值分别为 1.8 和 3 (宽度都是 0.3),且两个 ring 上数据点个数之比为 2:3,属于非凸状数据集;DataSet C 由一个 ring 和三个 circle 组成,其中外环和内部三个团的数据量之比为 2:1:1:1,属于凸状与非凸状的混合. 三个数据集所在的平面区域分别为$[-2,4]\times[-2,4]$, $[-4,4]\times[-4,4]$以及 $[-4,4]\times[-4,4]$. 图 3 是三个数据集分别包含 3 000、2 500 和 2 500 个数据的显示图.

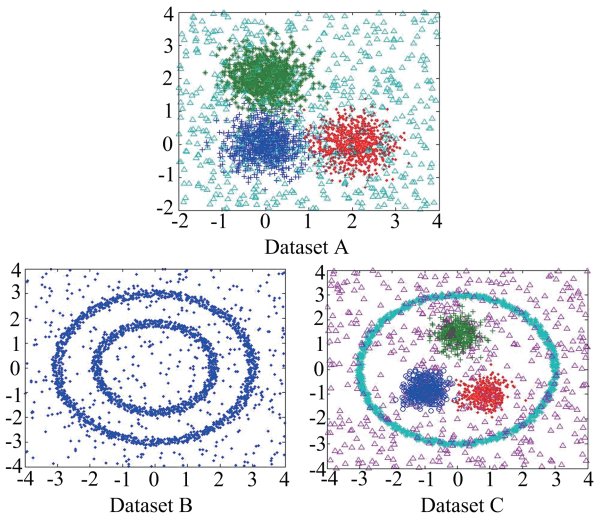

图 3 人工数据集

Fig. 3 Synthetic datasets

3.2　CBSC 算法参数分析

本节中，将随机样本点占数据集总数的比例记作 r_s，核心点与随机样本点集数量的比值记作 r_c（这样，核心点集占数据集总数的比例为 $r_s \cdot r_c$）. 并记 r_n 为核心点集中噪声点占据的比例，显然，r_n 越小说明选取的核心点集中包含的噪声数据越少.

3.2.1　r_c 对核心点分布的影响

在 r_s 基本固定的情况下（在本文的实验中，一般随机样本点的数量小于 10^3），r_c 反映了选取核心点的数量，其对核心点分布的影响较大，r_c 越小，选出的核心点就大多集中在聚类中心附近，噪声点占据的比例 r_n 也就越低；但若 r_c 太小，核心点个数过少则无法体现数据集的全局分布. 本节对上述三种人工数据集 DataSets A，B，C 在 r_c 取不同值时，统计了 r_n 值的变化. 表 1 列出了 r_c 值从 1 减少到 1/10 时，其相应的 r_n 值.

表 1　r_c 取不同值时，r_n 的变化

Tab. 1　Variation of r_n with different r_c

r_n	r_c					
	1	1/2	1/4	1/6	1/8	1/10
DataSet A	0.374	0.085	0.035	0.031	0.029	0.030
DataSet B	0.398	0.098	0.073	0.071	0.087	0.076
DataSet C	0.386	0.072	0.041	0.036	0.032	0.032

由表 1 看出，当 r_c 不超过 1/2 时，r_n 均不超出 0.1，即核心点集中噪声所占比例相比原噪声比例（20%）显著降低. 因此，通常 r_c 取值不超过 1/2，便可以使核心点中的噪声占据较小的比例. 图 4(1-4) 分别对应了 r_c 分别取值 1/10、1/5、1/2 和 1 时，数据集 DataSet A 中核心点的分布情况.

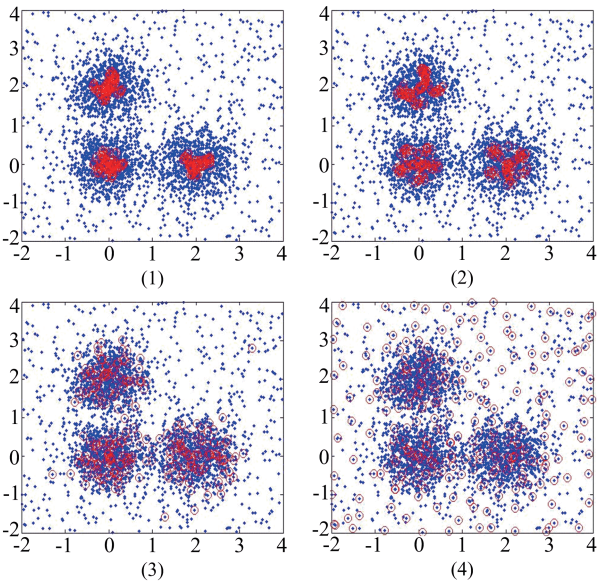

图 4　r_c 取不同值时，核心点的分布

Fig. 4　Distribution of core-points with different r_c

由图 4 可以看出，r_c 不超过 1/2（图 4(1-3)）时，选取的核心点对噪声鲁棒. 因此，综合表 1 和图 4 可知，r_c 取值不超过 1/2 可以使选取的核心点比较理想（本文由于随机样本点的数量小于 10^3，r_c 一般取为 1/2）.

3.2.2　参数 λ 对核心点分布的影响

由核心点的选取过程知（2.1.1 节），λ 越大，则选取的核心点越分散，更加体现数据的整体分布. 但若 λ 太大，太过分散又会选取较多的噪声点. 本节对于人工数据集 DataSets A，B，C，针对 λ 的不同取值，统计了 r_n 值的变化. 表 2 列出了 λ 从 0 取到 2.5 时，其相应的 r_n 的值.

表 2　λ 取值不同时，r_n 的变化

Tab. 2　Variation of r_n with different λ

r_n	λ					
	0	0.5	1	1.5	2	2.5
DataSet A	0.024	0.030	0.049	0.086	0.196	0.290
DataSet B	0.059	0.061	0.063	0.099	0.200	0.283
DataSet C	0.032	0.034	0.031	0.065	0.200	0.750

由表 2 可以看出，当 λ 值不超过 1.5 时，r_n 均不超过 0.1，核心点集中噪声比例相比原噪声比例（20%）显著降低. 同样地，还需要讨论 λ 取值不同时，核心点的分布情况. 对于 DataSet B（包含 2 500 个数据点），取 λ 分别为 0、1、1.5、2 时其核心点的分布分别对应图 5 中的(1-4).

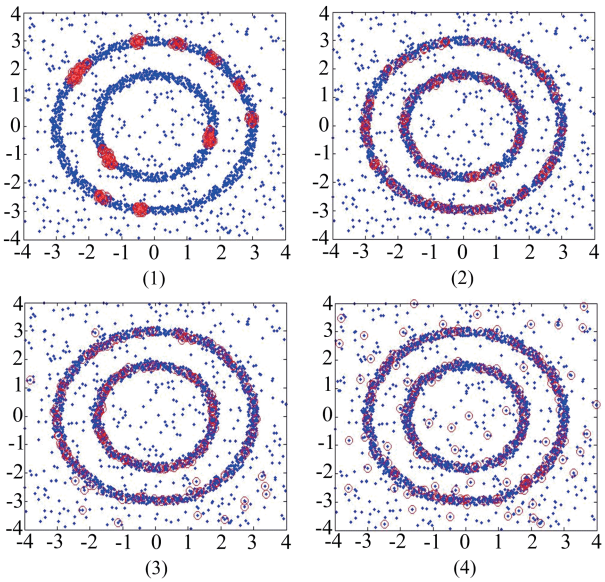

图 5　λ 取值不同时，核心点的分布

Fig. 5 Distribution of core-points with different λ

由图 5 可以看出，通常 λ 取值小于 1.5（图 5(1-3)）可以使选取的核心点对噪声鲁棒. 但是 λ 取值太小（例如 λ＝0），核心点过于集中在某些局部密度

较大的区域，无法体现数据集的形状信息（图5(1)）. 在本文实验中，如不特殊说明，λ 取为1.

3.3 CBSC的有效性

本节将说明CBSC应用到大数据上的有效性，并同时对比CBSC与SC的聚类时间与聚类结果. 实验数据集为3.1中的DataSets A，B. 评价指标为聚类准确率（pure value），记作 P，以及衡量两种聚类结果接近程度的指标，记作 r.

P 值定义为：

$$P = \mathrm{Pure}(\Omega,\Phi) = \frac{1}{n}\max_{\{j_k\}\subset\{1,\cdots N\}}\sum_{k=1}^{K} w_k \cap \varphi_{j_k} \tag{3}$$

式中，$\Omega = \{\omega_1,\cdots\omega_K\}$ 表示分类的集合，$\Phi = [\varphi_1,\cdots,\varphi_N]$ 表示聚类集合，$\{j_k, k = 1,\cdots,K\}$ 表示 $\{1,\cdots,N\}$ 中任意的 K 个自然数，n 表示数据集总数.（由于实验中聚类结果的好坏取决于算法对数据集中非噪声点的分类情况，因此将聚类的评价指标定义为非噪声点的分类准确率）.

r 指标定义为：

$$r = r(I_1, I_2) = \frac{a+b}{C_n^2} \tag{4}$$

式中，I_1 与 I_2 是对数据集 X（含 n 个点）使用两种不同聚类算法得到的聚类结果，a 为在 I_1 与 I_2 中均聚成相同类的数据对的数量，b 为在 I_1 与 I_2 中均聚成不同类的数据对的数量. 显然 $r\in[0,1]$，当 r 取0时表示两种算法的聚类结果完全不同，r 取1时表示两种算法的聚类结果完全相同. 即 r 指标值越大，说明两种算法法的聚类结果接近程度越高.

3.3.1 数据集的规模

SC可以处理任意形状的数据集，是一种相对理想的聚类算法，但由于计算复杂度较高，它仅能有效地聚类规模不超过5 000的数据集. 而本文中的CBSC则可以将数据集的规模提升到几十万，且聚类准确率较高（P 值较大）. 本节对数据集规模不断增加的DataSet A 和DataSet B，测试了相应的聚类时间和聚类准确率 P 值（λ 取1，r_c 取1/2）.

表3 不同规模下CBSC的聚类时间与聚类准确率 *P*

Tab. 3 Time and precision of CBSC with different dataset size

(1) DataSet *A*

DataSet size	750	1 500	7 500	15 000	75 000	150 000
CoreSet size	200	200	400	400	400	400
time	0.306	0.456	2.358	4.415	24.873	67.385
P-value	0.973 ±0.004	0.979 ±0.001	0.982 ±0.003	0.981 ±0.003	0.981 ±0.003	0.981 ±0.002

(2) DataSet *B*

DataSet size	1 250	5 000	12 500	50 000	125 000	500 000
CoreSet size	200	400	400	400	400	500
time	0.413	1.616	3.771	16.678	52.852	538.510
P-value	1±0	1±0	1±0	1±0	1±0	1±0

由表3可以看出，CBSC可以有效处理大规模的数据集. 表中的 P 值是10次实验的均值和标准差，可以看出，聚类准确率 P 不但很高而且很稳定（标准差不超过0.004）. 这进一步说明了CBSC的稳定性，虽然每次随机抽取的样本具有不确定性，但最后聚类的准确率都很高很稳定. 表3说明，随机抽样并不会影响聚类结果的有效性，故CBSC可以高效地处理大数据.

3.3.2 CBSC与SC的对比实验

对于规模小于5 000的数据集DataSets A，B，同时使用CBSC与SC进行聚类. 图6为两种方法在时间上的对比曲线，图7是两个数据集 r 指标值的变化趋势. 其中(1)是DataSet A 的实验结果，(2)是DataSet B 的实验结果.

图6分别给出了CBSC与SC在不同规模下的时间曲线，可以看出，CBSC的时间曲线与数据个数基本呈线性关系，而SC在数据规模持续增大时，时间有增加越来越快的趋势. 即随着数据规模的增大，CBSC时间缩短的会更加明显. 图7是相应的 r 指标值，可以看出，CBSC与SC的聚类结果相当接近（r 指标值接近于1），说明CBSC在降低时间复杂度的情况下，并没有改变SC的有效性.

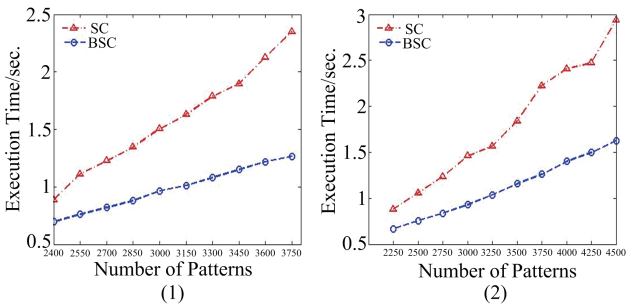

图 6　CBSC 与 SC 的时间对比

Fig. 6　Time comparison between BSC and SC

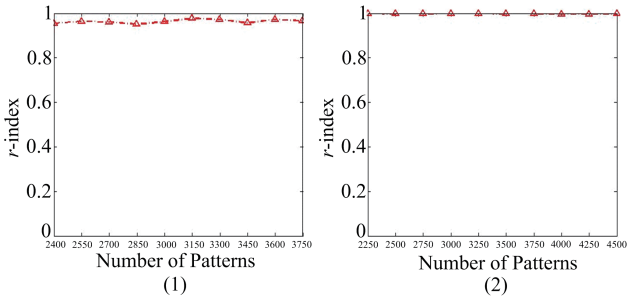

图 7　随着数据量的增加，r 指标值的变化

Fig. 7　Changes of r-index with increased patterns

4　结论

针对谱聚类性能优异却无法处理大数据的问题，本文从优化样本集选取的这两个方面出发，提出一种新的算法 CBSC.

本文的创新性和高效性在于：首先，基于相似性选取核心点克服了随机抽样和 K-means 聚类的质心点不稳定及无法体现原数据整体分布的缺点；其次，结合随机抽样以较低的计算代价得到核心集；最后综合核心集的谱聚类结果及数据的分组信息完成大数据的聚类. 该框架以较小的精度损失换来算法效率的大幅提升，将性能优异的谱聚类推广到了大数据. 实验结果表明，推广后的 CBSC 不但提升了聚类数据集的规模，降低了聚类时间，而且继承了原谱聚类的有效性. 后面进一步探索将其应用于其他性能优异的传统聚类算法上的有效性.

参考文献(References)

[1] 刘冰. Web 数据挖掘[M]. 北京：清华大学出版社，2011.

[2] KAUFMAN L, ROUSSEEUW P J. Finding Groups in Data: An Introduction to Cluster Analysis[M]. New York: Wiley, 1990.

[3] XU R, WUNSCH D. Survey of clustering algorithms [J]. IEEE Transactions on Neural Networks, 2005, 16(3): 645-678.

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

[5] MACQUEEN J. Some methods for classification and analysis of multivariate observations[C]// Proceedings of the 5th Berkeley Symposium Mathematical Statistics Probability. Berkeley: 1967: 281-297.

[6] WILLIAMS P K, SOARES C V, GILBERT J E. A clustering rule based approach for classification problems [J]. International Journal of Data Warehousing and Mining, 2010, 8(1): 1-23.

[7] FOWLKES C, BELONGIE S, FAN C, et al. Spectral grouping using the Nyström method [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2004, 26(2): 214-225.

[8] ZHANG K, KWOK J T. Clustered Nyström method for large scale manifold learning and dimension reduction [J]. IEEE Transactions on Neural Networks, 2010, 21(10):1576-1587.

[9] DING S F, JIA H J, SHI Z Z. Spectral clustering algorithm based on adaptive Nyström sampling for big data analysis [J]. Journal of Software, 2014, 25(9): 2037-2049.

[10] CHEN X L, DENG C. Large scale spectral clustering with landmark-based representation[C]// Proceedings of the 25th AAAI Conference on Artificial Intelligence. San Francisco: AAAI Press, 2011: 313-318.

[11] YAN D H, HUANG L, JORDAN M I. Fast approximate spectral clustering[C]// Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France: ACM Press, 2009: 907-916.

[12] SHINNOU H, SASAKI M. Spectral clustering for a large data set by reducing the similarity matrix size [C]// Proceedings of the 6th International Language Resources and Evaluation. 2008.

[13] VISWANATH P, BABU V S. Rough-DBSCAN: A fast hybrid density based clustering method for large data sets [J]. Pattern Recognition Letters, 2009, 30 (16): 1477-1488.

[14] ZHANG T, RAMAKRISHNAN R, LIVNY M. BIRCH: An efficient data clustering method for very large databases [J]. ACM SIGMOD Record, 1999, 25 (2): 103-114.

[15] 马儒宁，王秀丽，丁军娣. 多层核心集凝聚算法[J]. 软件学报，2013，24(3)：490-506.
MA R N, WANG X L, DING J D. Multilevel core-sets based aggregation clustering algorithm [J]. Journal of Software, 2013, 24(3): 490-506.