软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(11):2836−2846 [doi: 10.13328/j.cnki.jos.004888]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

求解大规模谱聚类的近似加权核 *k*-means 算法*

贾洪杰[1,2], 丁世飞[1,2], 史忠植[2]

[1](中国矿业大学 计算机科学与技术学院,江苏 徐州 221116)

[2](中国科学院 计算技术研究所 智能信息处理重点实验室,北京 100190)

通讯作者: 丁世飞, E-mail: dingsf@cumt.edu.cn

摘 要: 谱聚类将聚类问题转化成图划分问题,是一种基于代数图论的聚类方法.在求解图划分目标函数时,一般利用 Rayleigh 熵的性质,通过计算 Laplacian 矩阵的特征向量将原始数据点映射到一个低维的特征空间中,再进行聚类.然而在谱聚类过程中,存储相似矩阵的空间复杂度是 $O(n^2)$,对 Laplacian 矩阵特征分解的时间复杂度一般为 $O(n^3)$,这样的复杂度在处理大规模数据时是无法接受的.理论证明,Normalized Cut 图聚类与加权核 *k*-means 都等价于矩阵迹的最大化问题.因此,可以用加权核 *k*-means 算法来优化 Normalized Cut 的目标函数,这就避免了对 Laplacian 矩阵特征分解.不过,加权核 *k*-means 算法需要计算核矩阵,其空间复杂度依然是 $O(n^2)$.为了应对这一挑战,提出近似加权核 *k*-means 算法,仅使用核矩阵的一部分来求解大数据的谱聚类问题.理论分析和实验对比表明,近似加权核 *k*-means 的聚类表现与加权核 *k*-means 算法是相似的,但是极大地减小了时间和空间复杂性.

关键词: 谱聚类;迹最大化;加权核 *k*-means;近似核矩阵;大数据

中图法分类号: TP181

中文引用格式: 贾洪杰,丁世飞,史忠植.求解大规模谱聚类的近似加权核 *k*-means 算法.软件学报,2015,26(11):2836−2846. http://www.jos.org.cn/1000-9825/4888.htm

英文引用格式: Jia HJ, Ding SF, Shi ZZ. Approximate weighted kernel *k*-means for large-scale spectral clustering. Ruan Jian Xue Bao/Journal of Software, 2015,26(11):2836−2846 (in Chinese). http://www.jos.org.cn/1000-9825/4888.htm

Approximate Weighted Kernel *k*-means for Large-Scale Spectral Clustering

JIA Hong-Jie[1,2], DING Shi-Fei[1,2], SHI Zhong-Zhi[2]

[1](School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

[2](Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Spectral clustering is based on algebraic graph theory. It turns the clustering problem into the graph partitioning problem. To solve the graph cut objective function, the properties of the Rayleigh quotient are usually utilized to map the original data points into a lower dimensional eigen-space by calculating the eigenvectors of Laplacian matrix and then conducting the clustering in the new space. However, during the process of spectral clustering, the space complexity of storing similarity matrix is $O(n^2)$, and the time complexity of the eigen-decomposition of Laplacian matrix is usually $O(n^3)$. Such complexity is unacceptable when dealing with large-scale data sets. It can be proved that both normalized cut graph clustering and weighted kernel *k*-means are equivalent to the matrix trace maximization problem, which suggests that weighted kernel *k*-means algorithm can be used to optimize the objective function of normalized cut without the eigen-decomposition of Laplacian matrix. Nonetheless, weighted kernel *k*-means algorithm needs to calculate the kernel matrix, and its space complexity is still $O(n^2)$. To address this challenge, this study proposes an approximate weighted kernel *k*-means algorithm in which only part of the kernel matrix is used to solve big data spectral clustering problem. Theoretical analysis and experimental

* 基金项目: 国家重点基础研究发展计划(973)(2013CB329502); 国家自然科学基金(61379101); 江苏省普通高校研究生科研创新计划(KYLX15_1442)

收稿时间: 2015-02-15; 修改时间: 2015-05-11, 2015-07-14; 定稿时间: 2015-08-26

comparison show that approximate weighted kernel *k*-means has similar clustering performance with weighted kernel *k*-means algorithm, but its time and space complexity is greatly reduced.

Key words: spectral clustering; trace maximization; weighted kernel *k*-means; approximate kernel matrix; big data

科技的发展以及互联网的普及加快了人们生活的节奏,同时也产生了海量的数据和信息.我们正处在一个“数据丰富”而“知识贫乏”的时代,如何从浩瀚的数据中获取有价值的知识成为当务之急.大规模聚类是数据挖掘的主要工具之一,它能高效地组织大量数据,便于用户访问.聚类可以应用在各种场景中,例如网页搜索、图像检索、基因表达分析、推荐系统,以及基于交易数据的市场调研等[1].

很多文献给出的大规模聚类技术是基于欧氏距离的,并且隐含假设所有数据点位于欧氏几何结构中.基于核的聚类方法突破了这一限制,它将数据点嵌入到一个高维非线性流形中,并且使用非线性的核距离函数度量这些点的相似性[2].谱聚类就是一种典型的基于核的聚类方法,它把数据点当作无向加权图的节点,这样,聚类问题就变成了如何寻找最佳的图划分,使子图内部的连接权值最大,而子图之间的连接权值最小[3].但是,图聚类的很多目标函数,如 Ratio cut[4],Normalized cut[5],都无法在多项式时间内找到最优解,属于 NP 难问题.传统的求解方法是,将它们写成 Rayleigh 熵的形式,因为 Rayleigh 熵的最小值,第二小值,...,最大值分别对应 Laplacian 矩阵的最小特征值,第二小特征值,...,最大特征值,且极值在相应的特征向量处取得.于是.可以通过求解 Laplacian 矩阵的特征值和特征向量,将图划分的组合优化问题转化为数值优化问题,使其可以在多项式时间内解决.谱聚类的基本思想是:构造数据点的相似矩阵和对应的 Laplacian 矩阵,然后对 Laplacian 矩阵特征分解,再利用得到的特征向量进行聚类[6].由于用到了矩阵的特征值和特征向量,所以这种算法称为谱聚类.

尽管谱聚类在规模较小的数据集上表现很好,可是如果数据点的个数 n 很大,谱聚类在计算 n 个点的成对相似性和存储大的相似矩阵时,会遇到 $O(n^2)$级资源占用的瓶颈.此外,求解 Laplacian 矩阵的前 k 个特征向量也需要相当大的时间和内存,这些问题都限制了谱聚类算法在大数据中的应用.

解决计算和内存难题最常用的方法是:使相似矩阵的一些元素归零,将矩阵稀疏化;从获得的稀疏相似矩阵,找到对应的 Laplacian 矩阵,然后调用稀疏的特征求解方法计算特征向量[7].稀疏表示可以有效解决内存瓶颈,但是一些稀疏化方法仍然需要计算相似矩阵的全部元素.另一个加速谱聚类的重要方法是:利用 Nyström 近似技术进行特征分解,仅需相似矩阵的一部分,避免使用整个相似矩阵[8].Kumar 等人[9]分析了 Nyström 方法的误差界,并指出,集成的 Nyström 方法比标准的 Nyström 方法具有更快的收敛速度.这种方法牺牲了准确的相似性值,但换来了更短的计算时间,提高了算法的效率.在不同尺度参数下,平移不变核会从低秩结构变化到块对角结构.基于此,Si 等人[10]提出 MEKA 算法来逼近平移不变核矩阵.该算法同时考虑了核矩阵的低秩和块结构,在速度、逼近误差和内存使用方面都有良好的表现.此外,Chen 等人[11]研究了并行的谱聚类方法,通过将 n 个数据点分配到 p 个分布的机器节点上,实现大规模数据的聚类.

因为对 Laplacian 矩阵特征分解的时间复杂度也很高,一般为 $O(n^3)$,Dhillon 等人[12]提出了一种不使用特征向量的聚类方法.该方法源于核 *k*-means 与谱聚类的内在联系.核 *k*-means 算法是标准 *k*-means 算法的一个泛化.与标准 *k*-means 相比,核 *k*-means 的优势在于:通过将数据隐式映射到一个高维空间,可以发现输入空间中非线性分布的簇[13].研究结果表明,加权形式的核 *k*-means 目标函数与 Normalized Cut 的目标函数都能写成矩阵迹的形式,可以认为两者在数学上是等价的[14].这种等价性意味着:我们可以使用加权核 *k*-means 算法来优化 Normalized Cut 的目标函数;反过来,也可以将谱方法应用到加权核 *k*-means 中.在计算特征向量很困难的情况下,例如要对一个非常大的矩阵特征分解,加权核 *k*-means 算法可能比谱方法更合适.

虽然核距离函数有助于捕捉数据中的非线性结构,但是它要求计算并在内存中存储一个 $n\times n$ 的核矩阵,所以加权核 *k*-means 算法的空间复杂度也是 $O(n^2)$,仍然不适合处理大数据问题.本文中,我们尝试解决这个由大的核矩阵构成的挑战.注意到,加权核 *k*-means 之所以需要全部核矩阵,是因为依据 Representer 定理,类中心是由所有数据点的线性组合表示的[15].换句话说,类中心位于所有待聚类数据点的生成子空间中.我们可以通过把类中心限制在一个较小的子空间中,避免计算全部核矩阵.基于此,本文提出了求解大规模谱聚类的近似加权核 *k*-means 算法,通过随机选取一部分数据点,然后使用由这些数据点生成的子空间中的向量来逼近类中心.该近

似仅需要计算和存储整个核矩阵的一部分.理论分析和实验对比均表明,近似加权核 k-means 与使用整个核矩阵的加权核 k-means 具有相似的聚类表现.

本文第 1 节介绍 Normalized Cut 的基本原理,并将其归结为矩阵迹的最大化问题.第 2 节分析加权核 k-means 算法,然后给出加权核 k-means 与 Normalized Cut 目标函数之间统一的数学关系.第 3 节提出近似加权核 k-means 算法,用于解决大数据的谱聚类问题.第 4 节从理论层面讨论近似加权核 k-means 算法的计算复杂度和误差范围.第 5 节在不同的数据集上验证所提出的算法有效性,并与其他算法的聚类表现进行对比.最后归纳本文的主要贡献,以及下一步要做的工作.

1 归一化图划分

一个无向加权图可以表示为 $G=(V,E,A)$,其中,V 是所有节点的集合;E 是连接节点的边的集合,每条边都有一个权值,衡量两点属于同一类的可能性的大小;这些权值构成了亲和矩阵 A,通常 A 是非负的和对称的.

给定数据集 $X=\{x_1,x_2,\ldots,x_n\}$,X 中包含 n 个数据点.把这些数据点看作图 G 的节点,令节点集合 $V=[n]$表示所有待聚类的元素.要把 n 个点聚成 k 类,就是将 V 划分成 k 个不相交的子集,即,$V=\bigcup_{i=1}^{k}V_i$,且 $V_i\cap V_j=\varnothing,i\neq j$.

假设 $V_1,V_2\subset V$,定义 $links(V_1,V_2)$为 V_1 和 V_2 之间总的连接权值:

$$links(V_1,V_2)=\sum_{i\in V_1,j\in V_2}A_{ij} \tag{1}$$

一个子集 V_1 的度(degree)就是 V_1 中的点与所有数据点的连接权值之和:

$$degree(V_1)=links(V_1,V) \tag{2}$$

使用度作为归一项,定义 $linkratio(V_1,V_2)$表示 V_1 与 V_2 的连接在 V_1 与全集的连接中所占的比例:

$$linkratio(V_1,V_2)=\frac{links(V_1,V_2)}{degree(V_1)} \tag{3}$$

有两种特殊的 $linkratio$:一种是 $linkratio(V_1,V_1)$,用来衡量 V_1 内部的连接;另一种是它的补集 $linkratio(V_1,V\backslash V_1)$,用来衡量 V_1 外部的连接.一个好的聚类划分希望类内部的连接比较紧密,同时,类之间的连接比较松散.由这两个目标得到两种归一化的 k-way 划分准则:Normalized Association 和 Normalized Cut.它们的目标函数如下:

$$NAssoc(V_1,\ldots,V_k)=\sum_{i=1}^{k}linkratio(V_i,V_i) \tag{4}$$

$$NCut(V_1,\ldots,V_k)=\sum_{i=1}^{k}linkratio(V_i,V\backslash V_i) \tag{5}$$

根据 $links(V_2,V\backslash V_1)=degree(V_1)-links(V_1,V_1)$,可以得到 $NAssoc(V_1,\ldots,V_k)+NCut(V_1,\ldots,V_k)=k$.所以在最大化 Normalized Association 的同时,也能最小化 Normalized Cut.于是,图的 k-way 划分可以表示成下面的优化问题:

$$\max\ NAssoc(V_1,\ldots,V_k).$$

为了便于计算,定义一个隶属矩阵 $U\in\mathbb{R}^{k\times n}$ 来描述划分结果$\{V_1,\ldots,V_k\}$.设 $U=(u_1,\ldots,u_k)^T$,其中,$u_i^T\in\mathbb{R}^{1\times n}$ 是 U 的第 i 行,表示第 i 类 V_i 中包含哪些元素:对于数据集中的第 j 个点 x_j,若 $x_j\in V_i$,则 $U_{ij}=1$;若 $x_j\notin V_i$,则 $U_{ij}=0$.由于每个节点只能归到 V_1 到 V_k 中的一个类,U 的每列只包含一个 1,故 $U^T\mathbf{1}_k=\mathbf{1}_n$,其中,$\mathbf{1}_n\in\mathbb{R}^{n\times 1}$ 表示元素全为 1 的向量.根据对称的亲和矩阵 A,定义度矩阵 D:

$$D=diag(A\mathbf{1}_n) \tag{6}$$

其中,$diag(\cdot)$表示由它的矢量参数构成的对角矩阵.然后,可以重写 links 和 degree 如下:

$$links(V_i,V_i)=u_i^TAu_i \tag{7}$$

$$degree(V_i)=u_i^TDu_i \tag{8}$$

定义矩阵 $Z\in\mathbb{R}^{k\times n}$,令 $Z=(z_1,\ldots,z_k)^T=(UDU^T)^{-1/2}U$,其中,$z_i=u_i(u_i^TDu_i)^{-1/2}$,用 $tr(\cdot)$表示矩阵的迹,根据公式(7)、公式(8),则有:

$$NAssoc(V_1,...,V_k)=\sum_{i=1}^{k}\frac{u_i^TAu_i}{u_i^TDu_i}=\sum_{i=1}^{k}\frac{u_i^T}{(u_i^TDu_i)^{1/2}}A\frac{u_i}{(u_i^TDu_i)^{1/2}}=\sum_{i=1}^{k}z_i^TAz_i=tr(ZAZ^T).$$

注意到 $ZDZ^T=(UDU^T)^{-1/2}U\cdot D\cdot U^T(UDU^T)^{-1/2}=I_k$,其中,$I_k\in\mathbb{R}^{k\times k}$ 是单位矩阵.

令 $\tilde{Z}=ZD^{1/2}$,则 $\tilde{Z}^T=D^{1/2}Z^T,\tilde{Z}\tilde{Z}^T=ZD^{1/2}\cdot D^{1/2}Z^T=ZDZ^T=I_k$,那么,

$$tr(ZAZ^T)=tr(ZD^{1/2}\cdot D^{-1/2}AD^{-1/2}\cdot D^{1/2}Z^T)=tr(\tilde{Z}D^{-1/2}AD^{-1/2}\tilde{Z}^T).$$

所以,最大化 $NAssoc(V_1,...,V_k)$等价于矩阵迹的最大化问题,即

$$\max_{\tilde{Z}}tr(\tilde{Z}D^{-1/2}AD^{-1/2}\tilde{Z}^T) \tag{9}$$

注意到,该问题需要满足约束条件 $\tilde{Z}\tilde{Z}^T=I_k$.一个可行的求解方法是,根据谱图理论,定义 Laplacian 矩阵 $L=D^{-1/2}AD^{-1/2}$,通过对 L 特征分解,利用其前 k 个特征向量构造矩阵 $\tilde{Z}$.由于特征向量中包含了数据点的类属信息,因此可以在由特征向量组成的低维特征空间中对数据点进行划分.

2 加权核 *k*-means 算法

核 *k*-means 是经典 *k*-means 算法的一个非线性扩展.它把 *k*-means 算法中使用的欧氏距离函数 $d^2(x_a,x_b)=||x_a-x_b||^2$ 替换成了一个非线性核距离,定义为

$$d_\kappa^2(x_a,x_b)=\kappa(x_a,x_a)+\kappa(x_b,x_b)-2\kappa(x_a,x_b),$$

其中,$x_a\in\mathbb{R}^d$ 和 $x_b\in\mathbb{R}^d$ 是两个数据点,$\kappa(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$表示核函数.核函数建立了从原(输入)空间到高维核空间的非线性映射,有助于识别输入空间中非线性分布的簇.

令 $X=\{x_1,x_2,...,x_n\}$表示包含 n 个点的输入数据集,k 是设定的类数,$K\in\mathbb{R}^{n\times n}$ 是核矩阵,$K_{ij}=\kappa(x_i,x_j)$.由核函数 $\kappa(\cdot,\cdot)$诱导出的线性空间称为再生核希尔伯特空间(reproducing kernel Hilbert space,简称 RKHS),用$\mathcal{H}_\kappa$表示.将数据点划分成 k 个不相交的类$\{V_1,...,V_k\}$,每个类中数据点到类中心的距离的平方和称为聚类误差,核 *k*-means 的目标是寻找合理的划分,使总的聚类误差最小.

为了将核 *k*-means 算法与图的 Normalized Cut 联系起来,需要假设每个点 x_i 都有一个权值 w_i,引入权值后,核 *k*-means 就变成了加权核 *k*-means,其目标函数可以表示为

$$\min J(V_1,...,V_k)=\sum_{i=1}^{k}\sum_{x_j\in V_i}w_j\,|\,\kappa(x_j,\cdot)-c_i(\cdot)\,|_{\mathcal{H}_\kappa}^2=\sum_{i=1}^{k}\sum_{j=1}^{n}w_jU_{ij}\,|\,\kappa(x_j,\cdot)-c_i(\cdot)\,|_{\mathcal{H}_\kappa}^2 \tag{10}$$

其中,$|...|_{\mathcal{H}_\kappa}$ 表示$\mathcal{H}_\kappa$的泛函范数,$c_i(\cdot)\in\mathcal{H}_\kappa$表示类中心,$U\in\{0,1\}^{k\times n}$ 是数据点的隶属矩阵.

定义加权的隶属矩阵 $Y\in\mathbb{R}^{k\times n}$:$Y=(y_1,...,y_k)^T=UW$,其中,$W\in\mathbb{R}^{n\times n}$ 是由数据点权值构成的对角矩阵,即,$W=diag(w_1,...,w_n)$.令 $s_i=y_i^T\mathbf{1}_n$ 表示 Y 的第 i 行之和,s_i的倒数组成另一个对角矩阵 $L\in\mathbb{R}^{k\times k}$:$L=[diag(s_1,...,s_k)]^{-1}$.利用 L 将 Y 归一化,得到 $\hat{Y}\in\mathbb{R}^{k\times n}$ 和 $\tilde{Y}\in\mathbb{R}^{k\times n}$:

$$\hat{Y}=(\hat{y}_1,...,\hat{y}_k)^T=[diag(s_1,...,s_k)]^{-1}UW=LUW=LY,$$

$$\tilde{Y}=(\tilde{y}_1,...,\tilde{y}_k)^T=\left[diag\left(\sqrt{s_1},...,\sqrt{s_k}\right)\right]^{-1}UW^{1/2}=L^{1/2}UW^{1/2}.$$

那么,加权核 *k*-means 最佳的类中心可以表示为

$$c_i(\cdot)=\frac{1}{s_i}\sum_{x_j\in V_i}w_j\kappa(x_j,\cdot)=\sum_{j=1}^{n}\hat{Y}_{ij}\kappa(x_j,\cdot),i\in[k] \tag{11}$$

定理 1. 设核矩阵 $K=(\varphi_1,...,\varphi_n)^T$, $\varphi_i^T\in\mathbb{R}^{1\times n}$ 表示 K 的第 i 行.结合权值矩阵 W,加权核 *k*-means 的目标函数也可以由矩阵的迹来表示:

$$J(V_1,...,V_k)=tr(W^{1/2}KW^{1/2})-tr(\tilde{Y}W^{1/2}KW^{1/2}\tilde{Y}^T).$$

注意到,核矩阵 K、权值矩阵 W 都是给定的,故 $tr(W^{1/2}KW^{1/2})$是一个常量,所以最小化加权核 *k*-means 的目标函数等价于:

$$\max_{\tilde{Y}} tr(\tilde{Y}W^{1/2}KW^{1/2}\tilde{Y}^T) \tag{12}$$

与 $NAssoc(V_1,...,V_k)=tr(\tilde{Z}D^{-1/2}AD^{-1/2}\tilde{Z}^T)$ 对比可以发现:若令加权核 k-means 的权值矩阵 $W=D$,核矩阵 $K=D^{-1}AD^{-1}$,则

$$\tilde{Y}=L^{1/2}UW^{1/2}=(UWU^T)^{-1/2}UW^{1/2}=(UDU^T)^{-1/2}UD^{1/2}=\tilde{Z}.$$

于是,
$$tr(\tilde{Y}W^{1/2}KW^{1/2}\tilde{Y}^T)=tr(\tilde{Z}D^{1/2}\cdot D^{-1}AD^{-1}\cdot D^{1/2}\tilde{Z}^T)=tr(\tilde{Z}D^{-1/2}AD^{-1/2}\tilde{Z}^T).$$

可见,Normalized Cut 问题中 $tr(\tilde{Z}D^{-1/2}AD^{-1/2}\tilde{Z}^T)$ 的最大化与加权核 k-means 问题中 $tr(\tilde{Y}W^{1/2}KW^{1/2}\tilde{Y}^T)$ 的最大化是等价的,因此可以使用加权核 k-means 算法来求解 Normalized Cut 的目标函数,进而得到最佳的划分子图.为了保证加权核 k-means 算法最后收敛,一般要求核矩阵 K 是正定的.加权核 k-means 算法通过多次迭代,不断更新聚类中心和隶属矩阵,使聚类误差逐渐减小.与传统的谱聚类方法相比,利用迭代算法解决图划分问题,避免了对拉普拉斯矩阵特征分解,降低了计算复杂度,同时便于使用局部搜索等技术来提高聚类质量.

由定理 1 可知:加权核 k-means 的实现需要计算和存储全部 $n\times n$ 的核矩阵 K,使它不适合处理大规模的数据集.Zhang 和 Rudnicky[16]通过把核矩阵划分成小块,每次只用其中的一块来降低空间需求.尽管该技术考虑了空间复杂度,它仍然需要计算整个核矩阵.为了同时降低加权核 k-means 的时间和空间复杂度,本文考虑利用数据集中的部分抽样点计算近似的核矩阵,并设计了近似加权核 k-means 算法,用于处理大数据的谱聚类问题.

3 近似加权核 k-means 算法

降低加权核 k-means 的复杂度的一个简单而直观的方法是:从待聚类的数据集中随机选取 m 个点,在这些采样点中,使用标准加权 k-means 算法确定 k 个类中心;然后对于剩余的每个点,分别计算 k 个类中心与该点的距离,将该点与最近的类中心归为一类.采用这种方法尽管可以减少运行时间和空间需求,但是由于采样的任意性和局限性,它的表现无法和使用全部核矩阵的原算法相比,除非抽样点的数量足够大.

仔细观察公式(10)可以发现,由于类中心 $\{c_i(\cdot),i\in[k]\}$ 是全部待聚类数据点的线性组合,所以加权核 k-means 需要计算全部核矩阵 K.换句话说,类中心位于所有数据点生成的子空间中,即,$c_i(\cdot)\in\mathcal{H}_\kappa=span(\kappa(x_1,\cdot),\ldots,\kappa(x_n,\cdot))$, $i\in[k]$.倘若把类中心的解限制在一个较小的子空间 $\hat{\mathcal{H}}_\kappa\subset\mathcal{H}_\kappa$ 里,就可以避免计算整个核矩阵.构造的 $\hat{\mathcal{H}}_\kappa$ 应该具有如下性质:(1) $\hat{\mathcal{H}}_\kappa$ 应该足够小,以便高效计算;(2) $\hat{\mathcal{H}}_\kappa$ 的覆盖面应该足够大,以产生类似于使用 $\mathcal{H}_\kappa$ 的聚类结果.基于这个简单而重要的发现,本文提出了一个构造 $\hat{\mathcal{H}}_\kappa$ 的有效方法,可以显著降低加权核 k-means 的复杂度.首先,随机选取 m 个数据点($m<<n$),表示为 $\hat{X}=\{\hat{x}_1,...,\hat{x}_m\}$,用来构造子空间 $\hat{\mathcal{H}}_\kappa=span(\kappa(\hat{x}_1,\cdot),...,\kappa(\hat{x}_m,\cdot))$.给定子空间 $\hat{\mathcal{H}}_\kappa$,将加权核 k-means 的目标函数写成:

$$\min J(V_1,...,V_k)=\sum_{i=1}^{k}\sum_{j=1}^{n}w_jU_{ij}\,|\,\kappa(x_j,\cdot)-c_i(\cdot)|^2_{\hat{\mathcal{H}}_\kappa},c_i(\cdot)\in\hat{\mathcal{H}}_\kappa \tag{13}$$

定义两种核矩阵 $\hat{K}\in\mathbb{R}^{m\times m}$ 和 $\tilde{K}\in\mathbb{R}^{n\times m}$,$\hat{K}$ 由 $\hat{X}$ 中样本点之间的核相似性构成,$\tilde{K}$ 由 X 中的数据点与 $\hat{X}$ 中的样本点之间的核相似性构成.下面的引理 1 给出了限制在子空间 $\hat{\mathcal{H}}_\kappa$ 中的类中心 $c_i(\cdot)$ 的最优解:

引理 1. 给定隶属矩阵 U 和权值矩阵 W,根据采样点得到的类中心设为 $c_i(\cdot)=\sum_{j=1}^{m}\alpha_{ij}\kappa(\hat{x}_j,\cdot)$, 其中,$\alpha\in\mathbb{R}^{k\times m}$ 是未知参数.为使目标函数 $J(V_1,\ldots,V_k)$ 最小,最佳的 α 应满足 $\alpha=\hat{Y}\tilde{K}\hat{K}^{-1}$, 其中,矩阵 $\hat{Y}=LUW$.

定理 2. 若加权核 k-means 的类中心 $c_i(\cdot)\in\hat{\mathcal{H}}_\kappa$, $i\in[k]$,其目标函数可以归结为矩阵迹的最优化问题:

$$J(V_1,...,V_k)=tr(W^{1/2}KW^{1/2})-tr(\tilde{Y}W^{1/2}\tilde{K}\hat{K}^{-1}\tilde{K}^TW^{1/2}\tilde{Y}^T).$$

定理 2 表明,数据点的隶属矩阵 U 可以通过矩阵 $\tilde{K}$ 和 $\hat{K}$ 来确定.由于 $\hat{K}$ 是 $\tilde{K}$ 的一部分,事实上只需要计算 $\tilde{K}$.当 $m<<n$ 时,计算 $\tilde{K}$ 的开销远小于计算整个核矩阵 K.观察定理 1 和定理 2,定理 2 中的方法也可以视为用 $\tilde{K}\hat{K}^{-1}\tilde{K}^T$ 来逼近定理 1 中的核矩阵 K,这与矩阵低秩逼近的 Nyström 方法是类似的[17].由定理 2 得到的算法称为近似加权核 k-means,利用该算法来求解大规模谱聚类问题,可以显著降低算法的时间和空间复杂度,提高聚类

的效率.详细过程见算法 1.

算法 1. 近似加权核 *k*-means 算法.

输入:

- $X=\{x_1,\ldots,x_n\}$:待聚类的 n 个数据点的集合;
- m:随机采样的数据点个数($m<<n$);
- k:聚类个数;
- MAX*ITER*:最大迭代次数.

输出:包含最终聚类信息的隶属矩阵 U.

Step 1. 从 X 中随机选取 m 个点,表示为 $\hat{X}=(\hat{x}_1,...,\hat{x}_m)$.

Step 2. 根据核函数 $K=D^{-1}AD^{-1}$,计算 $\tilde{K}=[\kappa(x_i,\hat{x}_j)]_{n\times m}$ 和 $\hat{K}=[\kappa(\hat{x}_i,\hat{x}_j)]_{m\times m}$.

Step 3. 计算 $T=\tilde{K}\hat{K}^{-1}$.

Step 4. 随机初始化隶属矩阵 U.

Step 5. 令迭代次数 $t=0$.

Step 6. **repeat**

Step 7. 令 $t=t+1$.

Step 8. 计算加权隶属矩阵 $Y=UW=UD$ 和对角矩阵 $L=[diag(Y\mathbf{1}_n)]^{-1}$,并将 Y 归一化: $\hat{Y}=LY$.

Step 9. 计算 $\alpha=\hat{Y}T=\hat{Y}\tilde{K}\hat{K}^{-1}$.

Step 10. **for** $j=1,\ldots,n$ **do**

Step 11. 为每个点 x_j 找到最近的类中心 i^*:

$$i^*=\arg\min_{i\in[k]}|\kappa(x_j,\cdot)-c_i(\cdot)|^2_{\mathcal{H}_\kappa}=\arg\min_{i\in[k]}(\alpha_i^T\hat{K}\alpha_i-2\tilde{\varphi}_j^T\alpha_i),$$

其中,α_i^T 是α的第 i 行,$\tilde{\varphi}_j^T$ 是 $\tilde{K}$ 的第 j 行.

Step 12. 更新 U 的第 j 列,令其第 $i=i^*$行元素 $U_{ij}=1$,而其余元素为 0.

Step 13. **end for**

Step 14. **until** 隶属矩阵 U 不再变化或 $t>$MAX*ITER*.

4 算法分析

本节首先分析所提出的近似加权核 *k*-means 算法的计算复杂度和收敛性,然后与标准加权核 *k*-means 算法进行对比,从理论上研究了抽样近似方法对算法 1 聚类误差的影响.

4.1 计算复杂度

近似加权核 *k*-means 算法最费时的操作是矩阵求逆 $\hat{K}^{-1}$ 和计算 $T=\tilde{K}\hat{K}^{-1}$, 该过程的时间复杂度是 $O(m^3+m^2n)$.计算α和更新隶属矩阵 U 的计算开销是 $O(mnkt)$,其中,t 是算法收敛需要迭代的次数.所以,算法 1 总的计算复杂度为 $O(m^3+m^2n+mnkt)$.为了避免矩阵求逆 $\hat{K}^{-1}$, 进一步提高算法的运行效率,可以把计算 $\alpha=\hat{Y}\tilde{K}\hat{K}^{-1}$ 转化成下面的优化问题:

$$\alpha^*=\arg\min_{\alpha\in\mathbb{R}^{k\times m}}\left[\sum_{i=1}^{k}(s_i\alpha_i^T\hat{K}\alpha_i-2y_i^T\tilde{K}\alpha_i)\right]=\arg\min_{\alpha\in\mathbb{R}^{k\times m}}[tr(L^{-1}\alpha\hat{K}\alpha^T)-2tr(Y\tilde{K}\alpha^T)]=\arg\min_{\alpha\in\mathbb{R}^{k\times m}}\left[\frac{1}{2}tr(\alpha\hat{K}\alpha^T)-tr(\hat{Y}\tilde{K}\alpha^T)\right].$$

如果 $\hat{K}$ 是正定的,即,其最小的特征值显著大于 0,该优化问题可以用梯度下降法来求解,收敛速度是 $O(\log(1/\varepsilon))$,其中,ε是期望的精度.由于梯度下降法的每一步需要花费 $O(m^2k)$的代价,所以整个求解过程的计算开销是 $O(m^2k\log(1/\varepsilon))$.当 $k\log(1/\varepsilon)<<m$ 时,$O(m^2k\log(1/\varepsilon))<<O(m^3)$.使用这一技巧,可以将算法 1 的时间复杂度降低到 $O(m^2k\log(1/\varepsilon)+m^2n+mnkt)$.因为算法 1 中需要存储的最大矩阵是 $\tilde{K}$,所以其空间复杂度为 $O(mn)$.与标准加权核 *k*-means 算法 $O(n^2)$的复杂度相比,该方法大幅度降低了大数据聚类过程中的时间和空间需求.

4.2 算法收敛性

文献[14]中指出:如果核矩阵是正定的,就可以保证加权核 *k*-means 算法收敛.利用加权核 *k*-means 优化 Normalized Cut 的目标函数时,需要定义核矩阵 $K=D^{-1}AD^{-1}$,权值矩阵 $W=D$.但是,如果 A 是一个任意的邻接矩阵,$D^{-1}AD^{-1}$ 不一定是正定的,所以加权核 *k*-means 也不一定会收敛.本节通过为核矩阵 K 增加恰当的对角偏移量来避免这个问题.该解决方法可以看作对 Roth 等人[18]工作的推广,他们将对角偏移方法用在了非加权的情况中.

给定亲和矩阵 A,定义 $K'=\sigma D^{-1}+D^{-1}AD^{-1}$,其中,$\sigma$是一个正的足够大的常数,以保证 K'是正定的.

因为 D^{-1} 是一个正的对角矩阵,加上σD^{-1} 就为 $D^{-1}AD^{-1}$ 的对角线元素增加了正的偏移量.用 K'代替公式(12)中的 K,可以得到:

$$tr(\tilde{Y}W^{1/2}K'W^{1/2}\tilde{Y}^T)=tr(\tilde{Z}D^{1/2}\sigma D^{-1}D^{1/2}\tilde{Z}^T)+tr(\tilde{Z}D^{-1/2}AD^{-1/2}\tilde{Z}^T)=\sigma k+tr(\tilde{Z}D^{-1/2}AD^{-1/2}\tilde{Z}^T).$$

因此,使用 K'最大化 $\tilde{Y}$ 与公式(9)中的 Normalized Association 问题是等价的,只是 K'已经被构造成一个正定矩阵.运行基于 K'的近似加权核 *k*-means,可以单调优化 Normalized Association 的目标函数,保证算法的收敛性.

4.3 误差范围

与使用全部核矩阵的加权核 *k*-means 相比,近似加权核 *k*-means 的不同之处仅在于:其类中心限制在一个较小的子空间 $\hat{\mathcal{H}}_\kappa$ 中,而 $\hat{\mathcal{H}}_\kappa$ 是基于采样点构成的.下面就从该约束出发,探究使用这种近似的方法后,算法 1 的期望误差的范围.

命题 1. 给定隶属矩阵 U 和权值矩阵 W,设加权隶属矩阵 $Y=(y_1,\ldots,y_k)^T=UW$.定义二值随机变量$\xi=(\xi_1,\xi_2,\ldots,\xi_n)^T\in\{0,1\}^{n\times 1}$ 来描述随机采样过程:如果数据点 x_i 被选中构造子空间,则$\xi_i=1$;否则,$\xi_i=0$.于是,近似加权核 *k*-means 的聚类误差可以表示为

$$\mathcal{L}(Y,\xi)=tr(W^{1/2}KW^{1/2})+\sum_{i=1}^{k}\mathcal{L}_i(Y,\xi) \tag{14}$$

其中,$\mathcal{L}_i(Y,\xi)=\min_{\alpha_i\in\mathbb{R}^{n\times 1}}[s_i(\alpha_i\circ\xi)^TK(\alpha_i\circ\xi)-2y_i^TK(\alpha_i\circ\xi)]$.

注意到,$\xi=\mathbf{1}_n$ 时($\mathbf{1}_n$ 是元素全为 1 的向量),说明选择了所有数据点来构造子空间 $\hat{\mathcal{H}}_\kappa$,等价于使用全部核矩阵的加权核 *k*-means 算法,因此,$\mathcal{L}(Y,\mathbf{1}_n)$是标准加权核 *k*-means 算法的聚类误差.下面的引理 2 给出了$\mathcal{L}(Y,\xi)$期望的范围.

引理 2. 已知加权隶属矩阵 Y,$\mathcal{L}(Y,\xi)$的期望满足下面的不等式:

$$E_\xi[\mathcal{L}(Y,\xi)]\leqslant\mathcal{L}(Y,\mathbf{1}_n)+tr\left(\tilde{Y}\left[(W^{1/2}KW^{1/2})^{-1}+\frac{m}{n}[diag(W^{1/2}KW^{1/2})]^{-1}\right]^{-1}\tilde{Y}^T\right),$$

其中,$\mathcal{L}(Y,\mathbf{1}_n)=tr(W^{1/2}KW^{1/2})-tr(\tilde{Y}W^{1/2}KW^{1/2}\tilde{Y}^T)$.

推论 1. 假设对于任意 x 都有$\kappa(x,x)\leqslant 1$,权值 $w(x)\leqslant 1$,令$\lambda_1\geqslant\lambda_2\geqslant\ldots\geqslant\lambda_n\geqslant 0$ 为加权核矩阵 $W^{1/2}KW^{1/2}$ 的特征值.给定加权隶属矩阵 Y,有下面的不等式成立:

$$\frac{E_\xi[\mathcal{L}(Y,\xi)]-\mathcal{L}(Y,\mathbf{1}_n)}{\mathcal{L}(Y,\mathbf{1}_n)}\leqslant\frac{k/m}{\sum_{i=k+1}^{n}\lambda_i/n}.$$

为了说明推论 1 的结果,考虑一个特殊的加权核矩阵 $W^{1/2}KW^{1/2}$,它的前 a 个特征值等于 n/a,剩余的特征值为 0,即$\lambda_1=\ldots=\lambda_a=n/a,\lambda_{a+1}=\ldots=\lambda_n=0$.进一步假设 $a\geqslant 2k$,即,$W^{1/2}KW^{1/2}$ 的非 0 特征值的个数大于类数的 2 倍,那么根据推论 1,有下式成立:

$$\frac{E_\xi[\mathcal{L}(Y,\xi)]-\mathcal{L}(Y,\mathbf{1}_n)}{\mathcal{L}(Y,\mathbf{1}_n)}\leqslant\frac{k/m}{\sum_{i=k+1}^{n}\lambda_i/n}=\frac{k/m}{a^{-1}(a-k)}\leqslant\frac{k/m}{a^{-1}(a-a/2)}=\frac{2k}{m}.$$

上式表明:当 $W^{1/2}KW^{1/2}$ 的非 0 特征值的个数显著的大于类数时,使用逼近策略的近似加权核 *k*-means 的聚类误差与采样个数 m 是息息相关的,其与标准加权核 *k*-means 聚类误差的差别会随着样本点的增加以 $O(1/m)$

的速率减小.

5 实验与分析

将基于近似加权核 *k*-means 的谱聚算法记作 AWKK-SC,为了测试 AWKK-SC 算法的有效性,本节对比了该算法与另外 4 种典型的聚类算法在基准数据集上的聚类性能.对照算法分别是:标准谱聚类算法(NJW-SC)[19]、基于标准加权核 *k*-means 的谱聚类算法(WKK-SC)[12]、基于 Nyström 低秩近似的谱聚类算法(Nyström-SC)[8]、基于 MEKA 低秩近似的核聚类算法(MEKA-KC)[10].所有实验都是在一台高性能惠普工作站上进行的,其配置为:Intel Xeon E5-1620 3.60GHz 处理器,18G 内存,Windows 7 64 位操作系统,开发工具是 MATLAB 2012b.表 1 给出了实验中使用的基准数据集.

Table 1 Data characteristics of benchmark datasets

表 1 基准数据集的数据特征

数据集	数据点个数	维数	类数
Waveform	5 000	40	3
Ringnorm	7 400	20	2
USPS	9 298	256	10
MNIST	70 000	784	10
Forest cover type	581 012	54	7

Waveform 数据集[20]包含 3 类波形,每类各占 33%,其 40 维属性中的后 19 维都是噪声数据,噪声的均值为 0,方差是 1.Ringnorm 数据集[20]中的两类样本分别呈现两种不同的正态分布,但是这两种分布也有相互重叠的地方,不易区分.USPS 和 MNIST 都是手写数字数据集[21],它们各自含有 10 种不同类型的手写数字图片,每幅图片由一个 256 维或 784 维的特征向量表示.Forest Cover Type 数据集[22]来自美国地质调查局(USGS)和美国林务局(USFS),可分成 7 类数据,每类代表一种森林植被类型.

得到聚类结果后,可以依据簇标签与真实的类标签之间的归一化互信息(normalized mutual information,简称 NMI)来衡量算法的聚类准确度[23].

令 $U_c=(u_1^c,...,u_k^c)^T$ 表示与聚类得到的簇标签对应的隶属矩阵,$U_t=(u_1^t,...,u_k^t)^T$ 表示与真实的类标签对应的隶属矩阵,这两种变量之间的归一化互信息定义为

$$NMI(U_c,U_t)=\frac{I(U_c,U_t)}{\sqrt{H(U_c)\cdot H(U_t)}},$$

其中,$I(U_c,U_t)$是 U_c 和 U_t 之间的互信息,$H(U_c)$和 $H(U_t)$是信息熵,用于对互信息归一化,使其位于区间[0,1]内.实践中,常使用下面的公式来估计 NMI 的值:

$$NMI(U_c,U_t)=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}n_{i,j}^{c,t}\log\left(\frac{n\cdot n_{i,j}^{c,t}}{n_i^c\cdot n_j^t}\right)}{\sqrt{\left(\sum_{i=1}^{k}n_i^c\log\frac{n_i^c}{n}\right)\left(\sum_{j=1}^{k}n_j^t\log\frac{n_j^t}{n}\right)}} \tag{15}$$

其中,$n_i^c=(u_i^c)^T\mathbf{1}_n$ 表示簇 i 中数据点的个数,$n_j^t=(u_j^t)^T\mathbf{1}_n$ 表示类 j 中数据点的个数,$n_{i,j}^{c,t}=(u_i^c)^Tu_j^t$ 表示属于类 j 但是被划分到簇 i 中的数据点的个数.如果聚类结果与真实的类标签完全吻合,则 NMI 值为 1;如果数据被随意划分,则 NMI 值趋近于 0.NMI 的值越高,表示聚类的质量越好.为了客观地对比实验结果,各算法统一采用高斯核函数计算数据点之间的相似性,最大迭代次数设为 100.由于是随机初始化,算法的每次聚类结果会有小幅波动,因此,将 4 种算法在每个数据集上都运行 20 次,并计算其 NMI 指标的平均值,统计结果见表 2("–"表示内存不足,实验无法进行).

Table 2 NMI index of algorithms on different datasets

表 2 算法在不同数据集上的 NMI 指标

算法	采样个数	数据集				
		Waveform	Ringnorm	USPS	MNIST	Forest cover type
NJW-SC	100%	0.3658 (±0.0066)	0.6874 (±0.0029)	0.6095 (±0.0027)	–	–
WKK-SC	100%	0.3595 (±0.0039)	0.7632 (±0.0071)	0.6455 (±0.0049)	–	–
Nyström-SC	50	0.3668 (±0.0032)	0.6283 (±0.0073)	0.6178 (±0.0019)	0.4623 (±0.0076)	0.1066 (±0.0012)
	100	0.3687 (±0.0028)	0.6568 (±0.0060)	0.6305 (±0.0052)	0.4746 (±0.0047)	0.1107 (±0.0029)
	200	0.3716 (±0.0010)	0.6634 (±0.0026)	0.6416 (±0.0045)	0.4790 (±0.0049)	0.1125 (±0.0033)
	500	0.3742 (±0.0057)	0.6689 (±0.0014)	0.6485 (±0.0038)	0.4833 (±0.0036)	0.1210 (±0.0017)
	1 000	0.3769 (±0.0017)	0.6727 (±0.0009)	0.6514 (±0.0035)	0.4865 (±0.0025)	0.1235 (±0.0006)
	2 000	**0.3847** (±0.0053)	0.6841 (±0.0012)	**0.6564** (±0.0023)	0.4901 (±0.0047)	0.1243 (±0.0014)
MEKA-KC	50	0.3587 (±0.0035)	0.8887 (±0.0021)	0.4559 (±0.0043)	0.2709 (±0.0054)	0.0814 (±0.0015)
	100	0.3605 (±0.0012)	0.8919 (±0.0033)	0.5207 (±0.0054)	0.3326 (±0.0042)	0.0818 (±0.0009)
	200	0.3613 (±0.0023)	0.8926 (±0.0018)	0.5649 (±0.0037)	0.3514 (±0.0044)	0.0824 (±0.0005)
	500	0.3619 (±0.0014)	0.8933 (±0.0025)	0.5819 (±0.0021)	0.3776 (±0.0031)	0.0832 (±0.0011)
	1 000	0.3627 (±0.0017)	0.8941 (±0.0028)	0.5936 (±0.0065)	0.4989 (±0.0023)	0.0845 (±0.0008)
	2 000	0.3634 (±0.0015)	**0.8959** (±0.0019)	0.6118 (±0.0016)	0.5152 (±0.0027)	0.0858 (±0.0012)
AWKK-SC	50	0.3568 (±0.0011)	0.6764 (±0.0037)	0.5845 (±0.0061)	0.4754 (±0.0028)	0.0806 (±0.0034)
	100	0.3585 (±0.0021)	0.6957 (±0.0078)	0.6124 (±0.0039)	0.4929 (±0.0064)	0.1073 (±0.0015)
	200	0.3596 (±0.0008)	0.7226 (±0.0031)	0.6264 (±0.0041)	0.5089 (±0.0038)	0.1194 (±0.0022)
	500	0.3604 (±0.0016)	0.7327 (±0.0005)	0.6309 (±0.0033)	0.5144 (±0.0017)	0.1245 (±0.0014)
	1 000	0.3611 (±0.0005)	0.7345 (±0.0024)	0.6375 (±0.0026)	0.5246 (±0.0021)	0.1321 (±0.0041)
	2 000	0.3617 (±0.0013)	0.7360 (±0.0018)	0.6461 (±0.0063)	**0.5378** (±0.0035)	**0.1386** (±0.0027)

从表 2 中可以看出,NJW-SC 算法和 WKK-SC 算法在 Waveform,Ringnorm,USPS 这些较小的数据集上可以正常运行;但是当处理 MNIST,Forest Cover Type 这些大规模的数据集时,会提示内存不足而无法聚类.因为这两种算法都要使用完整的核相似矩阵,空间复杂度都是 $O(n^2)$,当数据量很大时,需要很大的内存空间来存储相似矩阵.假设每对数据点的相似性都由一个双精度浮点数表示,占 8 个字节,MNIST 数据集有 70 000 个数据点,这些数据点构成 $n\times n$ 的相似矩阵,大约需要 36.5GB 的内存;而 Forest Cover Type 数据集有 581 012 个数据点,整个相似矩阵占用的内存约为 2515.1GB.Nyström-SC 算法、MEKA-KC 算法和 AWKK-SC 算法由于采用了近似计算的策略,仅需要使用相似矩阵的一部分,空间复杂度大幅度降低,因而可以在有限的内存里对 MNIST,Forest Cover Type 进行聚类.而且随着采样点个数的增多,这 3 种算法的 NMI 值也逐渐提高.总体来看,Nyström-SC 算法在 Waveform 和 USPS 数据集上表现较好,MEKA-KC 算法对 Ringnorm 数据集的聚类精度最高,AWKK-SC 算法在 MNIST 和 Forest Cover Type 数据集上优势明显,这在一定程度上说明了 AWKK-SC 算法能够较好地处理大规模的数据集.为了进一步对比算法的运行效率,表 3 给出了每种算法在各个数据集上 20 次聚类的平均时间("–"表示内存不足,实验无法进行).

表 3 中,NJW-SC 算法的运行时间最长.因为该算法需要构造 Laplacian 矩阵,并对其特征分解,整个过程的时间复杂度很高.WKK-SC 算法利用加权核 k-means 来优化 Normalized Cut 的目标函数,不必计算特征向量,所以聚类效率比 NJW-SC 算法提高不少.但是 WKK-SC 算法要求使用全部核矩阵进行运算,如果待处理的数据点很多,依然需要花费大量时间.相比之下,仅用部分核矩阵进行近似计算的 Nyström-SC,MEKA-KC 和 AWKK-SC 算法在各个数据集上都能很快得到聚类结果,尤其是当数据点的规模达到几十万时,仍然可以流畅运行.Nyström-SC 算法与 AWKK-SC 算法都采用随机抽样策略,虽然提高抽样比例可以改善聚类准确率,但是也会增加程序的计算量,延长聚类时间.仔细观察可以发现:Nyström-SC 算法由于需要计算近似的特征向量,在样本点逐渐增多的过程中,其聚类时间的增幅较大;MEKA-KC 算法需要花费较长时间来求解最佳的 k 秩近似核矩阵,当数据集的规模很大时,算法的运行效率较低;而 AWKK-SC 算法的聚类时间主要与样本数和迭代次数有关,其变化趋势相对平缓,而且大多数情况下能够在更短的时间内完成聚类任务.这也说明 AWKK-SC 算法的聚类效率更高,适合大数据环境下的数据挖掘工作.

Table 3 Clustering time of algorithms on different datasets (s)

表 3 算法在不同数据集上的聚类时间(s)

算法	采样个数	数据集				
		Waveform	Ringnorm	USPS	MNIST	Forest cover type
NJW-SC	100%	247.484 2	707.356 8	1 835.890 9	–	–
WKK-SC	100%	157.670 1	428.077 7	1 164.494 3	–	–
Nyström-SC	50	0.118 4	0.122 1	0.185 7	1.734 8	6.232 4
	100	0.151 3	0.173 1	0.248 8	1.945 0	7.223 2
	200	0.272 7	0.324 5	0.477 5	2.691 0	10.012 6
	500	1.484 5	2.017 2	2.238 7	5.388 1	25.551 7
	1 000	19.315 1	13.028 1	13.220 3	18.341 2	59.966 1
	2 000	97.806 0	101.804 9	99.259 3	120.730 4	221.592 6
MEKA-KC	50	1.419 6	2.745 6	4.742 4	24.367 3	78.998 9
	100	1.794 0	2.995 2	5.210 4	26.130 1	96.829 8
	200	2.168 4	3.478 8	5.929 6	30.966 1	131.883 2
	500	2.698 8	4.929 6	7.503 6	54.288 3	248.681 1
	1 000	5.304 0	8.595 6	12.667 2	85.379 3	389.145 9
	2 000	16.723 3	24.819 7	30.295 3	171.211 0	556.147 3
AWKK-SC	50	0.079 4	0.092 3	0.365 9	4.909 6	16.206 4
	100	0.113 6	0.156 2	0.482 1	5.265 8	19.316 6
	200	0.200 9	0.219 1	0.648 1	6.096 2	25.567 9
	500	0.443 6	0.516 2	1.287 7	9.081 7	46.516 8
	1 000	1.030 4	1.299 6	2.353 6	14.792 9	84.936 1
	2 000	5.613 8	6.206 8	8.976 9	47.011 2	193.804 0

6 结束语

求解谱聚类目标函数的传统方法依赖于计算特征向量,时间和空间复杂性较高,无法处理非常大的数据集.本文从数学上讨论了谱聚类、核 *k*-means 与加权核 *k*-means 三者之间的联系,证明了 Normalized Cut 的目标函数与加权核 *k*-means 的目标函数是等价的.利用这一等价性,本文设计了一种适用于大数据谱聚类问题的近似加权核 *k*-means 算法.该算法一方面采用迭代的方式优化谱聚类的目标函数,避免了对 Laplacian 矩阵特征分解;另一方面,通过把类中心限制在一个由随机抽样点生成的较小的子空间中,也不用计算整个核矩阵,因此同时降低了谱聚类的计算复杂性和空间需求.理论分析表明:与使用全部核矩阵的加权核 *k*-means 相比,近似加权核 *k*-means 的期望误差会随着采样点个数的增加而逐渐减小.最后,通过在真实的数据集上进行测试,验证了所提出算法的有效性.虽然采用近似的策略会损失部分精度,但是可以大幅度提高聚类的效率.下一步考虑通过启发式采样或半监督技术来进一步改善所提出算法的性能.

致谢 在此,我们向对本文的工作给予支持和建议的同行表示感谢.

References:

[1] Sun JG, Liu J, Zhao LY. Clustering algorithms research. Ruan Jian Xue Bao/Journal of Software, 2008,19(1): 48−61 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[2] Schleif FM, Zhu XB, Gisbrecht A, Hammer B. Fast approximated relational and kernel clustering. In: Proc. of the 21st Int'l Conf. on Pattern Recognition. 2012. 1229−1232.

[3] Jia HJ, Ding SF, Xu XZ, Nie R. The latest research progress on spectral clustering. Neural Computing and Applications, 2014, 24(7-8):1477−1486. [doi: 10.1007/s00521-013-1439-2]

[4] Chan PK, Schlag MDF, Zien JY. Spectral *k*-way ratio-cut partitioning and clustering. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 1994,13(9):1088−1096. [doi: 10.1109/43.310898]

[5] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000,22(8): 888−905. [doi: 10.1109/34.868688]

[6] Rebagliati N, Verri A. Spectral clustering with more than *k* eigenvectors. Neurocomputing, 2011,74(9):1391−1401. [doi: 10.1016/j.neucom.2010.12.008]

[7] Von Luxburg U. A tutorial on spectral clustering. Statistics and Computing, 2007,17(4):395−416. [doi: 10.1007/s11222-007-9033-z]

[8] Fowlkes C, Belongie S, Chung F, Malik J. Spectral grouping using the Nyström method. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2004,26(2):214−225. [doi: 10.1109/TPAMI.2004.1262185]

[9] Kumar S, Mohri M, Talwalkar A. Sampling methods for the Nyström method. Journal of Machine Learning Research, 2012,13(1): 981−1006.

[10] Si S, Hsieh CJ, Dhillon I. Memory efficient kernel approximation. In: Proc. of the 31st Int'l Conf. on Machine Learning. 2014. 701−709.

[11] Chen WY, Song YQ, Bai HJ, Lin CJ, Chang EY. Parallel spectral clustering in distributed systems. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011,33(3):568−586. [doi: 10.1109/TPAMI.2010.88]

[12] Dhillon IS, Guan Y, Kulis B. Weighted graph cuts without eigenvectors: A multilevel approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2007,29(11):1944−1957. [doi: 10.1109/TPAMI.2007.1115]

[13] Yu S, Tranchevent LC, Liu XH, Glänzel W, Suykens JAK, De Moor B, Moreau Y. Optimized data fusion for kernel *k*-means clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2012,34(5):1031−1039. [doi: 10.1109/TPAMI.2011.255]

[14] Dhillon IS, Guan Y, Kulis B. Kernel *k*-means, spectral clustering and normalized cuts. In: Proc. of the 10th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2004. 551−556. [doi: 10.1145/1014052.1014118]

[15] Schölkopf B, Herbrich R, Smola AJ. A generalized representer theorem. In: Proc. of the Computational Learning Theory. 2001. 416−426. [doi: 10.1007/3-540-44581-1_27]

[16] Zhang R, Rudnicky AI. A large scale clustering scheme for kernel *k*-means. In: Proc. of the 16th Int'l Conf. on Pattern Recognition. 2002. 289−292. [doi: 10.1109/ICPR.2002.1047453]

[17] Ding SF, Jia HJ, Shi ZZ. Spectral clustering algorithm based on adaptive Nyström sampling for big data analysis. Ruan Jian Xue Bao/Journal of Software, 2014,25(9):2037−2049 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4643.htm [doi: 10.13328/j.cnki.jos.004643]

[18] Roth V, Laub J, Kawanabe M, Optimal cluster preserving embedding of nonmetric proximity data. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003,25(12):1540−1551. [doi: 10.1109/TPAMI.2003.1251147]

[19] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. In: Proc. of the Advances in Neural Information Processing Systems. 2002. 849−856.

[20] Bühler T, Hein M. Spectral clustering based on the graph p-Laplacian. In: Proc. of the 26th Annual Int'l Conf. on Machine Learning. 2009. 81−88. [doi: 10.1145/1553374.1553385]

[21] Cai D, He XF, Han JW, Huang TS. Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011,33(8):1548−1560. [doi: 10.1109/TPAMI.2010.231]

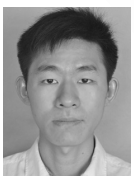
[22] Havens TC, Bezdek JC, Leckie C, Hall LO, Palaniswami M. Fuzzy c-means algorithms for very large data. IEEE Trans. on Fuzzy Systems, 2012,20(6):1130−1146. [doi: 10.1109/TFUZZ.2012.2201485]

[23] Kvalseth TO. Entropy and correlation: Some comments. IEEE Trans. on Systems, Man and Cybernetics, 1987,17(3):517−519. [doi: 10.1109/TSMC.1987.4309069]

附中文参考文献:

[1] 孙吉贵,刘杰,赵连宇.聚类算法研究.软件学报,2008,19(1):48−61. http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[17] 丁世飞,贾洪杰,史忠植.基于自适应 Nyström 采样的大数据谱聚类算法.软件学报,2014,25(9):2037−2049. http://www.jos.org.cn/1000-9825/4643.htm [doi: 10.13328/j.cnki.jos.004643]



贾洪杰(1988－),男,河北衡水人,博士生,CCF 学生会员,主要研究领域为感知计算,谱聚类,机器学习.

史忠植(1941－),男,研究员,博士生导师,CCF 会士,主要研究领域为智能科学,人工智能,机器学习.

丁世飞(1963－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为人工智能,机器学习,数据挖掘,粒度计算.