软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(12):3764−3785 [doi: 10.13328/j.cnki.jos.005327]

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

基于随机 *k*NN 图的批量边删除聚类算法*

雷小锋[1], 陈 皎[1], 毛善君[2], 谢昆青[3]

[1](中国矿业大学 计算机科学与技术学院,江苏 徐州 221116)

[2](北京大学 遥感与地理信息系统研究所,北京 100871)

[3](北京大学 信息科学技术学院 智能科学系,北京 100871)

通讯作者: 雷小锋, E-mail: lxf@cumt.edu.cn

摘 要: 建立邻接图上的批量边删除聚类算法通用框架,提出基于高斯平滑模型的批量边删除判定准则,定义了适于聚类的邻接图的一般性质,提出并证明在*k*NN 图基础上引入随机因子构造的随机*k*NN 图,可以增强顶点之间的局部连通性,使聚类结果不再强烈依赖于某条边或某些边的保留或删除.R*k*NNClus 算法简洁高效,依赖参数少,无需指定类簇数目,模拟和真实数据上的实验均有证明.

关键词: 邻接图;批量边删除聚类;随机 *k*NN 图;边删除准则;局部高斯平滑

中图法分类号: TP181

中文引用格式: 雷小锋,陈皎,毛善君,谢昆青.基于随机 *k*NN 图的批量边删除聚类算法.软件学报,2018,29(12):3764−3785. http://www.jos.org.cn/1000-9825/5327.htm

英文引用格式: Lei XF, Chen J, Mao SJ, Xie KQ. Batch edge-removal clustering based on Random *k*NN graph. Ruan Jian Xue Bao/Journal of Software, 2018,29(12):3764−3785 (in Chinese). http://www.jos.org.cn/1000-9825/5327.htm

Batch Edge-Removal Clustering Based on Random *k*NN Graph

LEI Xiao-Feng[1], CHEN Jiao[1], MAO Shan-Jun[2], XIE Kun-Qing[3]

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

[2](Institute of Remote Sensing and Geographic Information System, Peking University, Beijing 100871, China)

[3](Department of Intelligence Science, School of Electronice Engineering and Computer Science, Peking University, Beijing 100871, China)

Abstract: By generalizing batch edge-removal clustering algorithm, the clustering problem can be separated into the deterministic problem of edge-remove and the construction problem of adjacent graph. Firstly, in this paper, an edge-removal criterion is proposed according to the shifting under the local Gaussian smoothing of data objects. Secondly, the properties of adjacent graph suitable for clustering are studied, and a random *k*NN (R*k*NN) graph is suggested by introducing the random factor into the *k*NN graph. A proof is given to show that R*k*NN graph can lead to the enhancement of the local connectivity of graph and less dependency between clustering results and the removal of certain edges. The R*k*NNClus is simple and efficient without specifying the number of clusters. The experiments on synthetic datasets and real datasets demonstrate the effectiveness of the method.

Key words: adjacent graph; batched edge-remove clustering; random *k*NN graph; edge-remove criterion; local Gaussian smoothing

聚类问题经历了长期的研究和发展,产生出一系列聚类算法,如划分方法、层次方法、基于密度、基于模型以及谱聚类方法等.其中,划分方法认为,聚类是导致类内总误差平方和最小的数据划分,故其通过迭代重定位策略优化一个误差平方和准则函数,求得包围数据集的一组超球体或者椭球体.层次方法认为,聚类是数据以

* 基金项目: 国家科技重大专项(2016YFC0801800); 国家自然科学基金(41471315)

Foundation item: National Key Technologies Research & Development Program of China (2016YFC0801800); National Natural Science Foundation of China (41471315)

收稿时间: 2016-10-26; 修改时间: 2017-01-18; 采用时间: 2017-07-12

自顶向下或者自底向上的方式按照约定的类簇相异度(距离)进行分裂或者凝聚的结果,因此,层次算法可以建立数据对象的划分层次树.基于密度的方法则认为,聚类就是寻求数据集中数据分布比较稠密的一组连通区域,因此,密度算法需要规定密度的含义并求解连通的稠密子区域,基于模型的算法假定每个类簇符合特定分布模型,然后寻找数据对模型的最佳拟合.此外,目前学界普遍关注的是谱聚类方法,它认为,聚类是图划分问题,图的顶点就是数据样本点,图的边权表示样本点之间的相似度,聚类就是寻求子图内部相似度最大而子图之间相似度最小的划分.

传统的单连接分裂聚类算法以数据对象的最小生成树(minimum spanning trees,简称 MST)为邻接图,从最小生成树中按边权值从大到小的顺序删除边,将产生一系列连通分量作为聚类结果.该方法的问题有两个:(1) 在最小生成树中,删除任意一条边必然会将类簇一分为二,这种边删除分裂方式导致一条边的删除与否对聚类结果的影响过于敏感;(2) 聚类结果强烈依赖于边删除的顺序,按简单的距离权值从大到小的边删除方式往往会产生不合理的类簇分裂.针对上述问题,很多研究尝试改造最小生成树或组合多个最小生成树来构造邻接图[1−6],并基于最小生成树定义新的边权值,或通过谱分解等将数据对象映射到新的坐标系统.谱聚类方法则采用ε邻域图、k 最近邻图或完全连接图作为邻接图.

本文通过建立在邻接图上进行批量边删除的通用聚类算法框架,将问题归结为:(1) 建立合理的批量边删除准则;(2) 构建适于聚类的邻接图.提出一种基于局部高斯平滑模型的边删除准则,根据邻接图上数据对象在高斯平滑模型下的局部位移进行边删除判定;总结概括适于聚类处理的邻接图的一般性质,提出在 *k*NN 图中引入随机因子可以构建适于聚类的邻接图(随机 *k*NN 图),证明了随机 *k*NN 图可以更好地表达数据对象的局部类簇结构,且增强数据对象间的连通性,使聚类结果不再强烈依赖于某条边的保留或删除.R*k*NNClus 算法简洁高效,只有随机性因子一个自由参数,无需指定类簇数目,模拟和真实数据上均有实证.

本文第 1 节对现有聚类算法进行概述.第 2 节说明批量边删除聚类算法通用框架,重点阐述批量边删除判定准则和适于聚类的随机 *k*NN 图,给出 R*k*NNClus 算法及其分析.第 3 节提供实验,以证明 R*k*NNClus 算法的聚类效果.最后是结论和下一步工作.

1 常用聚类算法综述

(1) 基于划分的聚类算法

基于划分的聚类算法,首先根据给定类簇数目 k 创建一个初始划分,然后通过迭代重定位策略尝试在不同类簇之间移动对象来优化特定的目标函数,从而改进划分.*K*-Means 算法采用类簇中对象的均值中心来代表类簇,采用误差平方和为目标函数,因而对类球形且大小差别不大的类簇有很好的表现,但不能发现形状任意和大小差别很大的类簇,且聚类结果易受噪声数据影响.此外,*K*-Means 算法仅保证快速收敛到局部最优结果,从而导致聚类结果对初始代表点的选择非常敏感.*K*-MEDOIDS 算法用接近聚类中心的对象来表示类簇.对于存在噪声和孤立点的数据,*K*-MEDOIDS 算法比 *K*-Means 健壮,因为中心对象不像均值那样易于被极端数据影响.但 *K*-MEDOIDS 算法的执行代价比 *K*-Means 高.

一般地,基于划分的聚类算法实现简单且执行效率高,但是要求提前给定类簇数目 k,且不适于发现非凸形状或者大小差别很大的类簇,通常对噪声数据和离群点比较敏感.

2014 年,文献[7]提出一种划分聚类算法,算法假设类簇中心会被具有较低局部密度的邻近居点包围且与具有更高密度的其他点有相对较大的距离,算法通过构造决策图(点的局部密度和该点到具有更高局部密度的点的最小距离的函数)来寻找类簇中心.剩余的数据点归属到它的有更高密度的最近邻所属类簇.算法的鲁棒性非常好,且在很大程度上可以帮助使用者方便地确定类簇数目.

(2) 基于层次的聚类算法

基于层次的聚类算法以自顶向下(分裂)或自底向上(凝聚)的方式将数据对象划分成一个层次树结构,称为类簇的谱系树.层次聚类算法的聚类效果很大程度上依赖于度量类簇之间相异度的距离函数;此外,一般层次聚类算法的伸缩性不强,其时间复杂度通常为 $O(n^2)$.

BIRCH(balanced iterative reducing and clustering using hierarchies)算法[8]通过动态增量地构建 CF 树(clustering feature tree,聚类特征树)结构对样本数据点进行预聚类,然后在预聚类基础上完成最终聚类来改进层次聚类算法的性能.CF 树的节点代表类簇,表示为聚类特征向量(*N*,*LS*,*SS*),*N* 是簇样本点数,*LS* 是样本点的线性和,*SS* 是样本点的平方和.BIRCH 算法具有 $O(n)$的计算复杂度,在有限内存下可以很好地工作,但是算法使用的类簇相异度度量导致 BIRCH 只能发现球形类簇.

CURE(clustering using representative)算法[9]使用多个代表性样本点来表示类簇,类簇间的相异度距离是两个类簇代表点之间的最小距离.由于使用多个代表点来表示类簇,CURE 可以捕获复杂形状和大小差别很大的类簇,对噪声具有很好的免疫能力.此外,CURE 算法通过随机采样和预划分来加速聚类过程,处理大型的数据集.测试结果表明,CURE 较 BIRCH 性能更优.然而,CURE 算法的类簇收缩方式隐含地依赖于球形类簇假设,故在处理特殊形状的类簇时仍然比较困难.

Chameleon 算法[10]首先利用图划分算法将数据对象预先聚类为大量较小的子类簇,然后通过凝聚的层次聚类算法反复合并子类簇,形成最终的聚类结果.算法通过接近度和互连度的概念以及簇局部建模,确定最相似的两个类簇进行合并,因此能够很好地处理低维空间中任意形状、大小和密度的类簇,对噪声和离群点不敏感.但是对于高维数据,在最坏情况下的处理代价较高且效果常常有问题.此外,Chameleon 算法不会丢弃噪声数据,而是将其分配给某个类簇.

(3) 基于密度的聚类算法

基于密度的聚类算法将类簇定义为一系列连通的高密度子区域的并集.算法能够发现任意形状的类簇,并对异常点和噪声有自然的免疫能力.常见的基于密度的聚类算法有 DBSCAN,OPTICS,DENCLUE.

DBSCAN(density-based spatial clustering of application with noise)[11]是典型的面向低维空间数据的基于高密度连通子区域合并的聚类算法.如果一个点 *p* 的ε-邻域包含多于 *MinPts* 个点,则创建一个以 *p* 作为核心对象的类簇;然后,算法反复寻找从核心对象直接密度可达的对象,并将其加入类簇,其间可能需要合并一些密度可达的类簇,直到没有点被加入任意类簇则算法结束.DBSCAN 算法本质上只是提供了一个根据密度阈值参数进行聚类结果搜索的过程,聚类结果在用户指定密度阈值那一刻已经唯一地确定,算法本身并不对聚类的结果负责.其主要缺陷包括:(1) 算法对参数值设置敏感,聚类结果的质量依赖于密度阈值参数的合理选取;(2) 真实的数据集合经常分布不均匀,导致全局性的密度参数通常不能刻画数据内在的聚类结构;(3) 由于密度连通关系的传递性,有时会使绝大多数的样本点聚集到少数几个类簇中.

OPTICS(ordering points to identify the clustering structure)算法[12]针对 DBSCAN 算法的缺陷进行了改进.算法在聚类之前先将基于密度计算类簇所需的信息记录下来,这些信息反映了基于密度的聚类结构,从这些信息可以发现类簇.为此,OPTICS 算法:(1) 定义了对象的核心距离和可达距离,以反映对象附近的密度大小;(2) 在迭代查找可达对象时,对候选的种子对象按可达距离排序,在类簇扩展时,优先扩展密度值较大区域的对象.这样,算法就实现了数据库中所有对象的排序,序列中的对象从一个簇到另一个簇基本上遵循簇边缘到簇核心再到簇边缘的顺序,反映了基于密度的聚类结构.基于这些信息,用户可以比较容易地确定合适的密度参数阈值,从而克服 DBSCAN 的参数敏感性.

DENCLUE(density based clustering)算法[13,14]是一种基于核密度估计的聚类算法,它使用核密度函数来建模每个样本点对总体密度的影响,总体密度函数是所有核密度函数的总和,类簇就是围绕总体密度函数局部峰值(称为局部吸引子)的所有样本点.为了降低核密度函数计算的代价,DENCLUE 通过将空间网格化来定义近邻,并借此限制定义密度所需要考虑的样本点数量.DENCLUE 算法具有良好的数学基础,可以处理不同形状和不同大小的类簇,尤其擅长处理噪声和离群点.但是算法必需精心选择网格单元的尺寸参数以及密度参数和噪声阈值,否则可能造成聚类效果显著下降.

(4) 基于网格的聚类算法

基于网格的聚类算法的基本思想是:把数据空间划分成网格单元,形成网格结构,聚类操作就在网格结构上进行.常见的基于网格的聚类算法有 STING,WaveCluster,CLIQUE 等.

STING(statistical information grid-based method)算法[15]将数据空间划分成不同分辨率的多层网格结构,然后针对每个网格单元预先计算并存储其中样本点的统计信息.STING 算法扫描一遍数据计算网格单元的统计信息并建立层次结构,之后,基于层次网格结构进行聚类查询的效率极高;但是对于高维数据,网格单元个数会指数增长,算法性能降低.STING 算法的聚类结果质量有限,表现在类簇有水平或者垂直的马赛克效果,网格分辨率会影响聚类结构的质量;此外,对于高维数据的聚类效果很差.

WaveCluster(clustering with wavelets)算法[16]是一种采用小波变换的多分辨率网格聚类算法,它首先通过在数据空间上加一个多维网格结构来汇总数据,然后利用小波变换把多维数据从空域变换到频域,然后在频域空间中寻找高密度区域(即类簇).WaveCluster 的计算复杂度为 $O(n)$,能有效地处理大数据集;发现任意形状的类簇,处理孤立点;无需指定诸如类簇数目或邻域半径等输入参数.

CLIQUE(clustering in quest)算法[17]融合了基于密度和网格两种思路,实现了子空间聚类.算法基于网格单元在所有维的子空间中搜索稠密区域(类簇),搜索过程依赖于密度簇的单调性,即:如果一个点集在 k 维属性上形成一个稠密类簇,则相同的点集在这些维的所有可能子集上也是稠密类簇的一部分.CLIQUE 算法能够发现子空间中的类簇,且当维数增加时具有良好的扩展性.

(5) 基于模型的聚类算法

基于模型的聚类算法其基本思路是:为每个类簇假定一个模型,然后寻找数据对模型的最佳拟合.在假设适用的情况下,算法可以构建出样本数据的空间分布密度函数并定位类簇,也可以通过特定统计量自动确定类簇数目,同时能够处理噪声和离群点数据,产生鲁棒的聚类结果.常见的基于模型的聚类算法有:EM(expectation maximization)算法、COBWEB 算法、SOM(self-organizing feature map)算法.

EM 算法[18]假设样本数据是由多个同类型概率分布加权混合形成的混合模型产生的,然后通过期望最大化算法反复迭代最大化样本数据和混合分布模型之间的拟合程度.最常用的混合分布是高斯混合模型.通过 EM 算法求解混合模型,可以发现大小不同及椭球形类簇.但是 EM 算法只能收敛到局部最优解,对初始参数的选择有一定依赖性;此外,如何选择合适的类簇数目及混合模型形式也是一个问题.

COBWEB 算法[19]是一种用于概念聚类的层次聚类方法,它通过一种称为概念的概率描述来表示类簇,并通过增量地构建类簇或概念的层次结构来形成分类树.分类树中,每个节点对应一个概念,通过概率描述来总结该节点或类簇下所有样本对象的特征.分类树在某个层次上的兄弟节点就形成了一个类簇划分.COBWEB 算法适用于概念聚类,可以自动调整类簇数目而无需用户指定.问题是算法假设每个属性上的概率分布彼此独立,有时候并不成立,造成聚类效果不佳.

SOM 算法[20]是一种神经网络的聚类方法,它将高维空间的样本点映射到低维空间,并尽可能保留样本点之间的距离和邻近关系.算法首先初始化一组质心,然后迭代选择一个对象并确定该对象最近的质心,更新该质心及其附近质心,直到质心不再变化或者变化不大.最后将所有对象分配到最近的质心.SOM 算法将邻近关系映射在低维的质心之上,非常有利于高维数据的可视化和解释.缺点是用户需要选择邻域函数、网格类型、质心个数等参数;另外,对于形状复杂或者尺寸差异很大的类簇,SOM 的质心并不自然对应于类簇,可能是簇的分裂或者合并;SOM 算法不保证收敛,虽然通常收敛.

(6) 基于谱图的聚类算法

基于谱图的聚类算法[21]近年来备受关注,算法建立在谱图理论基础之上,通过构造以样本点为顶点、样本相似性为边的相似矩阵,将聚类问题转化为图最优划分问题.算法首先构造相似性矩阵 $\boldsymbol{W}$,并计算相似性矩阵或拉普拉斯矩阵的 k 个最小特征向量,然后将特征向量视为 k 维空间的 N 个向量,利用 K-Means 或其他聚类算法对 N 个向量进行聚类处理,得到聚类结果.谱图聚类方法通过特征分解可以获得连续松弛域中的全局最优解,具有坚实的理论基础,能够高质量地识别任意形状的类簇.根据图划分准则的不同,谱图聚类有 Min Cut,Average Cut,Normalized Cut,Min-Max Cut,Ratio Cut 等算法.

谱聚类有效的前提是构建能够真实反映数据对象之间相似关系的相似度矩阵,目前还没有很好的解决办法.此外,类簇数目和特征向量个数 k 都是有待解决的关键问题.

2 基于 R*k*NN 图的边删除聚类算法

2.1 批量边删除聚类算法通用框架

传统的单连接分裂聚类算法按简单的距离权值从大到小的边删除方式存在两个问题:(1) 从最小生成树中删除任意一条边必然会将类簇一分为二,导致一条边的删除与否对聚类结果的影响过于敏感;(2) 聚类结果强烈依赖于边删除的顺序,按简单的距离权值从大到小的边删除方式往往会产生不合理的类簇分裂.

针对上述问题,本文提出以邻接图为基础进行批量的边删除操作,使得聚类结果不会强烈依赖于某条边的删除与否,批量边删除聚类算法通用框架如图 1 所示.可以看出,此时聚类算法归结为两个子问题:其一是建立合理的边删除准则,尽可能删除簇间数据对象的连通边,保留簇内数据对象间的连通边,且允许一定程度的误删除操作;其二是构造适于聚类的邻接图(简称聚类邻接图),以表达数据对象的局部邻近关系,增强数据对象间的连通性,使聚类结果不再依赖于某条或某些边的保留或删除.

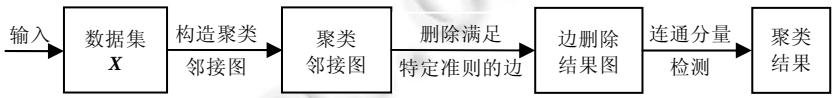


Fig.1 Framework of the batched edge-remove clustering algorithm

图 1 批量边删除聚类算法的通用框架

2.2 局部高斯平滑模型与边删除判定准则

通过数据对象在局部高斯平滑模型下的移动距离,来构造一种动态的批量边删除判定准则.

2.2.1 局部高斯平滑模型

高斯平滑(Gaussian smoothing),又称高斯模糊(Gaussian blur),是图像处理中广泛用于来减小噪声以及降低细节层次的技术[22].从数学视角看,图像进行高斯模糊平滑的结果,就是图像 $f(x,y)$与高斯函数 $g(x,y,\sigma)$做卷积运算的结果 $\boldsymbol{G}(x,y)$:

$$G(x,y)=g(x,y;\sigma)\times f(x,y),$$

其中, $g(x,y;\sigma)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$是二维高斯函数.参数$\sigma$是标准差,又称平滑尺度,大的$\sigma$平滑能力强,对应图像的概貌特征;小的$\sigma$平滑能力弱,对应图像的细节特征.

高斯平滑可以模糊掉图像细节层次,形成概念层次更高的宏观图像和概念.同理,对于聚类处理而言,从单个的数据对象到集聚的类簇,可以视为是观察尺度提升使得数据细节模糊形成的自然结果.因此,本文将高斯平滑从图像信号域推广到数据对象域,对邻接图中的数据对象进行局部高斯平滑.

定义 1(局部高斯平滑模型). 在邻接图中,假设每个数据对象都会受到相连接数据对象的吸引而发生移动,移动的目标位置由数据对象的局部高斯平滑结果决定.对于 d 维空间中的数据对象 $\boldsymbol{x}_i$,假设集合$\{\boldsymbol{x}_j|j=1\sim n\}$表示邻接图中与 $\boldsymbol{x}_i$ 相连的 n 个数据对象,则数据对象 $\boldsymbol{x}_i$ 的局部高斯平滑结果为 $g(\boldsymbol{x}_i)$:

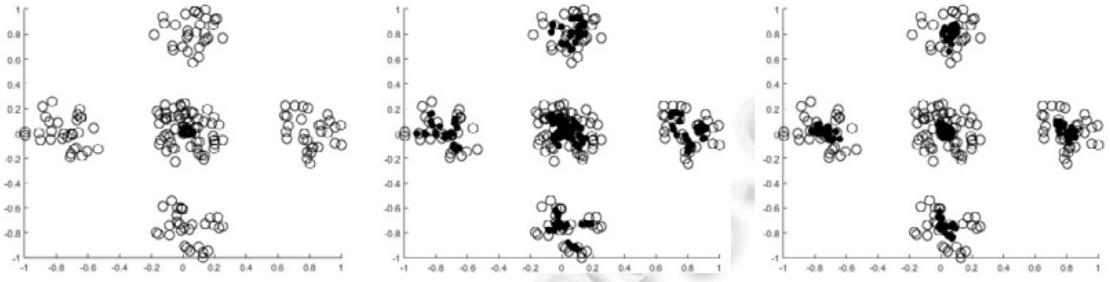
$$g(\boldsymbol{x}_i)=\frac{\sum_{j=1\sim n}g_{ij}\boldsymbol{x}_j}{\sum_{j=1\sim n}g_{ij}}.$$

即,数据对象 $\boldsymbol{x}_i$ 在相连数据对象的吸引下移动到 $g(\boldsymbol{x}_i)$.公式中,g_{ij} 表示在邻接图中与 $\boldsymbol{x}_i$ 相连接的 n 个局部数据对象 $\boldsymbol{x}_j$ 对移动的贡献权值,是以 $\boldsymbol{x}_i$ 为均值的高斯核函数在 $\boldsymbol{x}_j$ 处的函数值.

$$g_{ij}=g(\boldsymbol{x}_i,\boldsymbol{x}_j;\sigma)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right).$$

与图像的高斯平滑类似,经过高斯平滑处理的数据对象向局部类簇中心收缩移动,在视觉上类似于模糊掉一些细节特征,呈现出数据的概貌特征.例如图 2(a)所示的含 5 个球形类簇的 sph5 数据集,在不同尺度下的高斯平滑结果(如图 2(b)和图 2(c)所示,图中空心圆点为原始数据对象,实心圆点为高斯平滑后的数据对象),可以看出

数据对象向局部密度中心移动,区分数据对象的细节特征逐渐模糊,呈现宏观的类簇结构.在文献[23]一文中,将数据空间建模为图像,数据对象建模为图像中的光点,提出通过尺度空间滤波理论进行图像模糊的聚类方法,与本文的局部高斯平滑模型具有类似的思想.



(a) sph4 数据集 (b) σ^2=0.008 (c) σ^2=0.031

Fig.2 Gaussian smoothing results of sph4 under different scales parameter σ (solid dots)

图 2 数据集 sph4 在不同尺度参数σ下的高斯平滑结果(实心圆点)

假设数据对象 $\boldsymbol{x}$ 经过高斯平滑移动到目标位置 $\boldsymbol{x}'$,将位移向量记做 $\boldsymbol{x}'-\boldsymbol{x}$,则可以建立高斯平滑模型和 mean shift 算法的内在联系.高斯平滑导致的位移向量 $\boldsymbol{x}'-\boldsymbol{x}$ 实际上就是 mean shift 算法中的均值漂移向量[24].mean shift 的均值漂移向量定义为

$$\boldsymbol{m}_h(\boldsymbol{x})=\frac{\sum_{i=1}^{N}g(\boldsymbol{x},\boldsymbol{x}_i;h)\boldsymbol{x}_i}{\sum_{i=1}^{N}g(\boldsymbol{x},\boldsymbol{x}_i;h)}-\boldsymbol{x}.$$

假设数据集 $\boldsymbol{X}$ 的概率密度函数为 $f(\boldsymbol{x})$,记其核函数密度估计为 $\hat{f}(\boldsymbol{x})$,则已经证明如下结论:

$$\boldsymbol{m}_h(\boldsymbol{x})=\boldsymbol{x}'-\boldsymbol{x}=\frac{1}{2}h^2\frac{\Delta\hat{f}_K(\boldsymbol{x})}{\hat{f}_G(\boldsymbol{x})},$$

其中,$\Delta\hat{f}_K(\boldsymbol{x})$ 表示用核函数 K 估计的密度函数 $\hat{f}_K(\boldsymbol{x})$ 的梯度,$\hat{f}_G(\boldsymbol{x})$ 表示用核函数 G 估计的概率密度函数.结论是:(1) 均值漂移向量 $\boldsymbol{m}_h(\boldsymbol{x})$正比于密度函数梯度,指向概率密度增大最快的方向;(2) 均值漂移向量反比于 $\boldsymbol{x}$ 处的概率密度函数值,即,在数据对象密集的区域位移较小,在稀疏区域位移较大.

2.2.2 边删除判定准则

根据均值漂移向量的结论,对于邻接图中的一条边,作为顶点的两个数据对象在高斯平滑下移动且距离越来越远,则说明两个数据对象分属不同类簇,边应当切除.

定义 2(邻接图的边删除准则). 给定邻接图中的一条边$(\boldsymbol{x}_i,\boldsymbol{x}_j)$,假设数据对象 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的高斯平滑结果分别为 $\boldsymbol{y}_i$ 和 $\boldsymbol{y}_j$,则如果满足如下公式的拉伸条件,即判定边$(\boldsymbol{x}_i,\boldsymbol{x}_j)$应该删除:

$$\|\boldsymbol{y}_i-\boldsymbol{y}_j\|>\|\boldsymbol{x}_i-\boldsymbol{x}_j\|.$$

公式的直观含义是,数据对象 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的高斯平滑结果 $\boldsymbol{y}_i$ 和 $\boldsymbol{y}_j$ 的距离较原始距离越来越远.需要注意:满足边删除准则是删除一条边的充分条件,根据该准则并不能一次将所有不必要的边都删除掉.因此,算法并不期冀一次完成所有不必要边的删除,而是在图上多次迭代进行边删除,每次迭代删除掉部分不必要的边,导致图结构发生变化,然后在变化的图结构上重新计算数据对象的高斯平滑结果,并根据高斯平滑结果进行下一次边删除准则判定和边删除操作,直到图结构收敛不再发生变化.直观地,通过边删除准则,每次迭代都可以作出正确充分的边删除决策,通过多次正确的边删除决策,来逐步逼近最终稳定的图结构.

事实上,由于数据的局部分布可能不均一,通过边删除准则会误删除一部分边.因此,在每次边删除操作之后,可以对产生的图结构进行聚类质量评估,选择合适的聚类结果.

2.3 构造聚类邻接图

聚类邻接图是邻接图中适用于批量边删除聚类处理的邻接图,满足全局稀疏和局部稠密的性质.其中:全局稀疏性是指相对于完全连接图而言,邻接图的边数较少;而局部稠密性是指对于局部邻近的一组数据对象,其邻接图中相应的连接边应比较多,满足局部稠密性,自然会导致不同类簇之间的边比较稀疏且边的距离权值较大,同一类簇内的边比较稠密且边的距离权值较小.

定义 3(邻接图(adjacent graph,简称 AG)). 给定数据集 $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^n$ 及其距离度量,邻接图 AG 是以数据对象为顶点的简单图,图中每条边表示连接的两个数据对象以给定的距离相邻接.

常用的邻接图有 ε邻域图(ε-neighborhood graph,简称 εNG)、k 最近邻图(k-nearest neighbor graph,简称 kNNG)、完全连接图(fully connected graph,简称 FCG)、最小生成树(minimum spanning trees,简称 MST)及组合的 k 最小生成树(kMST)、相对最近邻图(relative neighborhood graph,简称 RNG)、Gabriel 图(Gabriel graph,简称 GG)、Delaunay 三角网(delauney triangulation)等,见表 1.

Table 1 Common adjacency graphs and graph clustering algorithms

表 1 常用的邻接图及图聚类算法

聚类算法	邻接子图类型	邻接子图说明	边的权值度量
单连接聚类[6]	欧氏最小生成树(EMST)	最小生成树	欧氏距离边权
ZEMST[5]	欧氏最小生成树	最小生成树	Inconsistency 度量
SDPRT[2]	双根树(dual-rooted tree)	两个最小生成树	碰撞时间(hitting time)
邻接图聚类[1]	Perturbed MSTs Disjoint MSTs	多个噪声扰动的最小生成树 或多个不相交的最小生成树	高斯核函数距离
基于有限邻近集的图论聚类[25]	Gabriel 图或相对最近邻图	Gabriel 图或相对最近邻图	Inconsistency 度量
谱聚类算法[20]	kNN 图或εN 图	kNN 图或εN 图	坐标系统映射

定义 3.1(k 最小生成树(kMST)). 假设图的最小生成树记做 MST(G).对于给定数据集 $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^n$ 及其完全连接图 FCG($\boldsymbol{X}$,$\boldsymbol{E}$),$\boldsymbol{E}$ 是带距离权值的边集,可以定义其 k 最小生成树为

$$k\text{MST}=\text{MST}_k(\text{FCG})=\begin{cases}\text{MST}(\text{FCG}), & k=1\\ \text{MST}_{k-1}(\text{FCG})\cup\text{MST}(\text{FCG}-\text{MST}_{k-1}(\text{FCG})), & k>1\end{cases}.$$

显然,完全连接图和最小生成树都不是合理的聚类邻接图.除此之外,其他邻接图乃至完全连接图的任意子图是否适于用作聚类邻接图,目前很少有研究涉猎.为此,引入聚类邻接图的全局稀疏性和局部稠密性度量,用于评估一个邻接图是否适于聚类分析处理.

定义 4(全局稀疏系数(global sparsity coefficient,简称 GSC)). 给定邻接图 AG($\boldsymbol{V}$,$\boldsymbol{E}_{\text{AG}}$),记其完全连接图为 FCG($\boldsymbol{V}$,$\boldsymbol{E}_{\text{FCG}}$),定义邻接图 AG 的全局稀疏性为

$$\text{GSC}(\text{AG})=\frac{2|E_{\text{AG}}|}{|V|(|V|-1)}.$$

邻接图的全局稀疏性是邻接图的边数与其完全连接图边数的比值,取值从 0 到 1,取值接近 0,表明邻接图越稀疏;取值 1,表明邻接图是完全连接图.用于聚类的邻接图的全局稀疏性应该比较小.

在小世界网络研究中,Watts 和 Strogatz 于 1998 年引入局部聚类系数的概念[26],可以用于度量一个邻接图图中顶点之间的聚类倾向或聚集程度,其定义如下.

定义 5(局部聚类系数(local clustering coefficient,简称 LCC)). 给定邻接图 AG($\boldsymbol{V}$,$\boldsymbol{E}_{\text{AG}}$),其局部聚类系数定义为每个顶点的局部聚类系数的加权平均:

$$\text{LCC}(\text{AG})=\frac{1}{|V|}\sum_{v_i\in V}\text{LCC}(v_i)=\frac{1}{|V|}\sum_{v_i\in V}\frac{2|\text{AG}[V_i]|}{|V_i|(|V_i|-1)},$$

其中,V_i 表示与顶点 v_i 直接相连的顶点集;AG[V_i]表示邻接图在顶点集 V_i 上的导出子图.对于顶点 v_i,其局部聚类系数是导出子图 AG[V_i]的边数与 V_i 所有顶点间可能边数的比值.对于孤立顶点和度数为 1 的顶点,$|V_i|=0$ 或 $|V_i|=0$,约定顶点的局部聚类系数为 0.局部聚类系数的取值范围[0,1],取值 0,表示每个顶点的直接相连顶点之间

都互不连接;取值为 1,表示每个顶点的直接相连顶点之间都互相连接构成完全图.

注意:局部聚类系数与顶点度数呈反比关系,度数小的顶点对局部聚类系数具有更大的贡献.为了克服这一缺陷,本文改变以顶点为局部单元的局部聚类系数计算方式,提出以边为局部单元进行计算的局部稠密系数.

定义 6(局部稠密系数(local denseness coefficient,简称 LDC)). 给定邻接图 AG($\boldsymbol{V}$,$\boldsymbol{E}_{\mathrm{AG}}$),AG 的局部稠密系数定义为每条边的局部稠密系数的加权平均:

$$\mathrm{LDC}(\mathrm{AG})=\frac{1}{|E_{\mathrm{AG}}|}\sum_{e_{ij}\in E_{\mathrm{AG}}}\mathrm{LDC}(e_{ij})=\frac{1}{|E_{\mathrm{AG}}|}\sum_{e_{ij}\in E_{\mathrm{AG}}}\frac{|\mathrm{AG}[V_i\cup V_j]|}{|V_i\cup V_j|(|V_i\cup V_j|-1)},$$

其中,e_{ij} 是顶点 v_i 与顶点 v_j 相连的边,V_i 表示与顶点 v_i 直接相连的顶点集,V_j 表示与顶点 v_j 直接相连的顶点集,$V_i\cup V_j$ 表示与顶点 v_i 和 v_j 相连的所有顶点.对于边 e_{ij} 而言,其局部稠密系数是导出子图 AG[$V_i\cup V_j$]的边数与 $V_i\cup V_j$ 所有顶点间可能边数的比值,对于$|V_i\cup V_j|$=0 或$|V_i\cup V_j|$=1 的边,约定局部稠密系数为 0.类似地,局部稠密系数取值范围[0,1].但与局部聚类系数不同,局部稠密系数以边为计算单元,将局部的范围从顶点周边邻近顶点扩大到边的周边邻近顶点,一定程度上降低了小度数顶点的贡献.

2.3.1 *k*NN 图和 *k*MST 的全局稀疏性和局部稠密性

定理 1. 给定数据集 $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^{n}$ 及其距离度量,在相同的 k 值下 *k*NN 图是 *k*MST 的生成子图.

证明:

(1) k=1,*k*NN 图包含于 *k*MST.

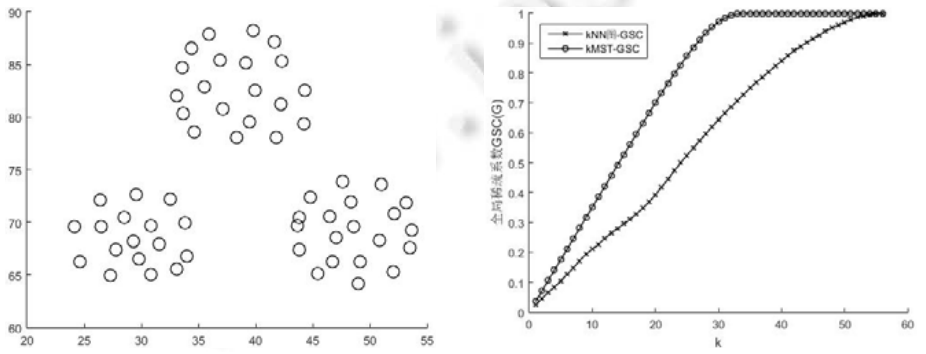
k=1 时,*k*NN 图中每个顶点连接其最近顶点,*k*MST 就是最小生成树 MST,假设 *k*NN 图中边 e_{ij} 是顶点 v_i 到 v_j 的最短边,但 e_{ij} 不属于 MST,则将边 e_{ij} 加入 MST 会产生一个环 C:$e_{ij}\rightarrow e_{jr}\rightarrow\ldots\rightarrow e_{ti}$.显然,移除环 C 中的边 e_{ti} 会产生另一棵生成树 ST=MST$-e_{ti}+e_{ij}$,由于 $e_{ti}>e_{ij}$,所以 ST 的距离权值会比 MST 的距离权值小,这与 MST 是最小生成树矛盾.故得证.

(2) 假设 $k=m-1$ 时 *k*NN 图包含于 *k*MST,证明 $k=m$ 时 *k*NN 包含于 *k*MST.

根据已知,$k\mathrm{NN}_{k=m-1}\subseteq k\mathrm{MST}_{k=m-1}\subseteq k\mathrm{MST}_{k=m}$,$k\mathrm{NN}_{k=m-1}\subseteq k\mathrm{NN}_{k=m}$.故只需证明 $k\mathrm{NN}_{k=m}-k\mathrm{NN}_{k=m-1}\subseteq k\mathrm{MST}_{k=m}$ 即可.

假设边 $e_{ij}\in k\mathrm{NN}_{k=m}-k\mathrm{NN}_{k=m-1}$ 且满足 $e_{ij}\notin k\mathrm{MST}_{k=m}$,因为 $k\mathrm{MST}_{k=m-1}\subseteq k\mathrm{MST}_{k=m}$,所以 $e_{ij}\notin k\mathrm{MST}_{k\le m}$,则将 e_{ij} 加入最小生成树 MST=$k\mathrm{NN}_{k=m}-k\mathrm{NN}_{k=m-1}$ 中会产生一个环 C:$e_{ij}\rightarrow e_{jr}\rightarrow\ldots\rightarrow e_{ti}$,,移除环 C 中的边 e_{ti} 会产生另一棵生成树,且 ST 的距离权值更小,这与 MST 是最小生成树矛盾.故得证.

相同 k 值条件下,*k*NN 图比 *k*MST 边数更少更稀疏.以图 3(a)所示的二维空间数据集 sph3 为例,数据集包含 57 个数据对象,采用欧氏距离度量.对于 k=1~56,分别构造 sph3 数据集的 *k*NN 图和 *k*MST,计算每个 *k*NN 图和 *k*MST 的全局稀疏系数 GSC、局部聚类系数 LCC 和局部稠密系数 LDC.



(a) sph3 数据集 (b) sph3 数据的 *k*NN 图和 *k*MST 的 GSC-*k* 曲线

Fig.3 Global sparse coefficients of *k*NN graphs and *k*MST for sph3 datasets (k=1~56)

图 3 sph3 数据集及其 *k*NN 图和 *k*MST 的全局稀疏系数(k=1~56)

在图 3(b)中,横轴表示 *k*NN 图和 *k*MST 的 k 值(k=1~56),纵轴表示 *k*NN 图和 *k*MST 的全局稀疏系数.可以看

出:随着 k 值的增加,全局稀疏系数从 0 增加到 1,但 kMST 每次以 $n-1$ 条边的速度线性增加,kNN 图则增加的比较缓慢,即达到相同的全局稀疏系数,kNN 图比 kMST 需要更大的 k 值.

推论 1. 给定数据集 $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^n$ 及其距离度量,则达到相同全局稀疏性的 kNNG 比 kMST 具有更大的 k.即 GSC(kNNG)=GSC(kMST)时,有结论:kNNG[k]>kMST[k](根据定理 1 可证).

但是,在全局稀疏系数较小时,相同边数的 kNN 图较 kMST 具有更好的局部稠密性.在图 4 中,横轴表示全局稀疏系数 GSC,纵轴表示局部聚类系数 LCC 和局部稠密系数 LDC.可以看出:在稀疏性较小区间约[0,0.4]内,kNN 图的局部聚类系数 LCC 和局部稠密系数 LDC 总是大于 kMST.

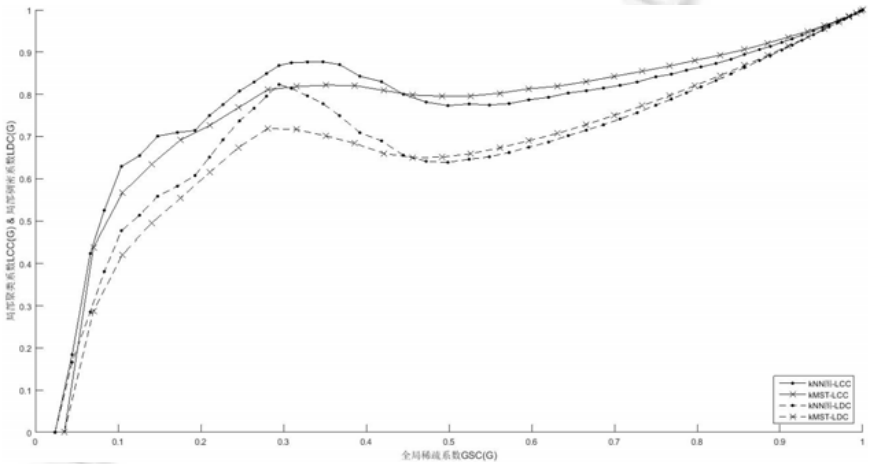


Fig.4 Local sparse coefficients of kNN graphs and kMST for sph3 datasets (k=1~56)

图 4 sph3 数据集的 kNN 图和 kMST 的局部聚类系数和局部稠密系数(k=1~56)

此外,全局稀疏系数随着 k 值的变化从 0 增大到 1,kNN 图和 kMST 的局部聚类系数和局部稠密系数也随之从 0 增大到 1.在稀疏系数较小的区间内,局部聚类系数和局部稠密系数急速增大,然后变化平缓,之后,k 值及全局稀疏系数的变化对局部聚类系数和局部稠密系数影响很小.

推论 2. 给定数据集 $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^n$ 及其距离度量,在全局稀疏性较小的情况下,相同稀疏性的 kNNG 比 kMST 具有更大的局部聚类系数和局部稠密系数.即 GSC(kNNG)=GSC(kMST)<<1 时,有结论:

LCC(kNNG)>LCC(kMST),LDC(kNNG)>LDC(kMST).

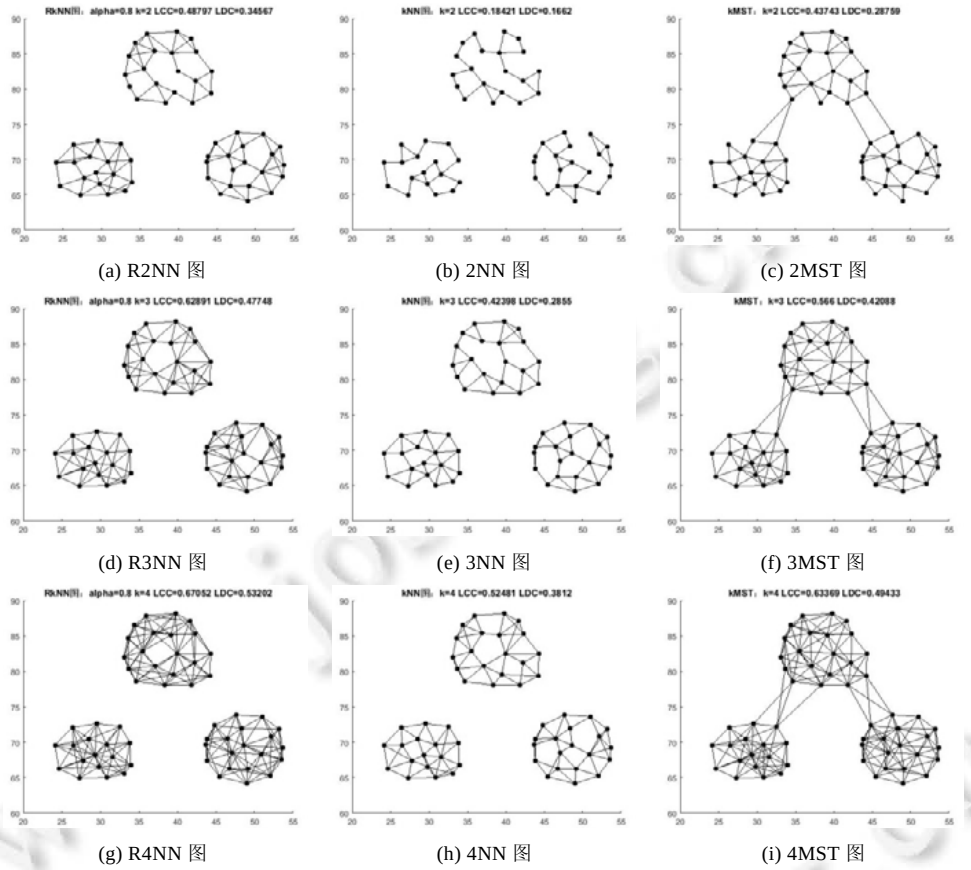
2.3.2 随机 kNN 图及其全局稀疏性和局部稠密性

根据上文的结论,在全局稀疏的条件下考虑局部稠密性,则 kNN 图较 kMST 更适用于图聚类算法.但无论是 kNN 图还是 kMST 都需要确定参数 k 值,而 k 值与局部稠密性的关系并不直观.因此,本文提出引入随机因子来构造一种随机化的 kNN 图,在 k 值固定的情况下,可以使产生的 kNN 图快速达到满足聚类要求的全局稀疏性和局部稠密性,而随机因子与局部稠密性的关系比较容易理解.

- 构造随机 kNN 图.

从给定的数据集 $\boldsymbol{X}$ 中随机抽取 m 个数据对象作为锚点($\alpha=m/n$ 是随机因子),将剩余 $n-m$ 个数据对象连接到距离其最近的 k 个锚点,生成一次 RkNN 图.执行该步骤 t=30 次,将生成的 RkNN 图合并得到最终的随机 kNN 图,记做 RkNN($\boldsymbol{X}$).

随机因子α是锚点占样本总数的比例,理解起来比较直观.以图 3(a)所示的数据集 sph3 为例,进行 30 次随机抽样(随机因子α=0.8)产生 R2NN,R3NN,R4NN 以及相应的 2NN,3NN,4NN 图和 2MST,3MST,4MST,如图 5(a)~图 5(i)所示.可以看出:引入随机性的 RkNN 图比相对应的 kNN 图和 kMST 具有更好的局部稠密性,更适用于图聚类算法.



(a) R2NN 图 (b) 2NN 图 (c) 2MST 图

(d) R3NN 图 (e) 3NN 图 (f) 3MST 图

(g) R4NN 图 (h) 4NN 图 (i) 4MST 图

Fig.5 Local sparse coefficients of R*k*NN, *k*NN graphs and *k*MST for sph3 datasets (*k*=2~4)

图 5 数据集 sph3 在 *k*=2~4 时的 R*k*NN,*k*NN 和 *k*MST 图及其局部稠密性

在 *k* 值相同的情况下,从局部聚类系数和局部稠密系数的角度观察,如图 6 所示:相同 *k* 值的 R*k*NN 图比 *k*NN 图、*k*MST 具有更大的局部稠密性,且在 *k* 值较小时即达到较大的局部稠密性.

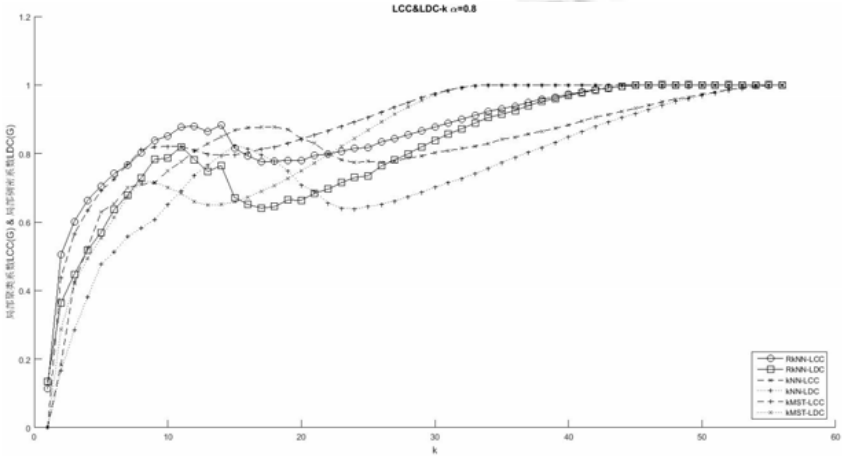
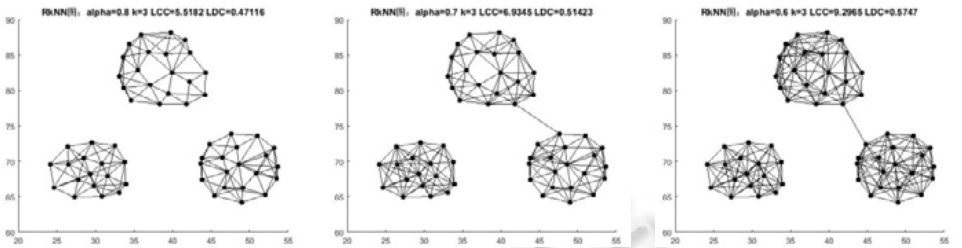


Fig.6 LCC and LDC of *k*NNG, R*k*NNG and *k*MST for sph3 datasets (*k*=1~56)

图 6 数据集 sph3 的 *k*NNG,R*k*NNG 和 *k*MST 的 LCC 和 LDC(*k*=1~56)

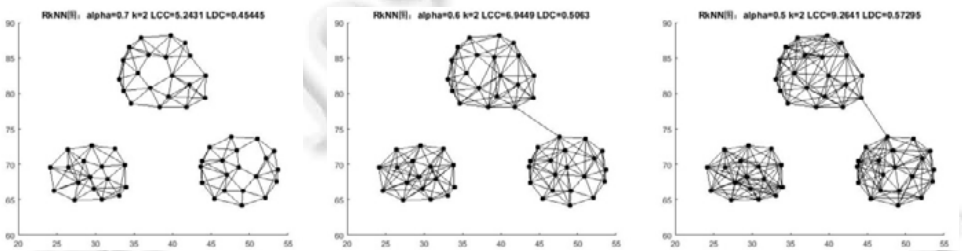
此外,在固定 k 值时减小随机因子,同样可以得到局部稠密性更好的 RkNN 图.以数据集 sph3 为例,固定 k=3,随机因子α=0.8~0.6,产生一系列 R3NN 图,如图 7 所示.同理,对于更小的 k 值,可以通过减小随机因子进行补偿,获得局部稠密性相近的 RkNN 图,如图 8 和图 9 所示.



(a) α=0.8 (b) α=0.7 (c) α=0.6

Fig.7 R3NN graph and its local density of dataset sph3 (α=0.8~0.6)

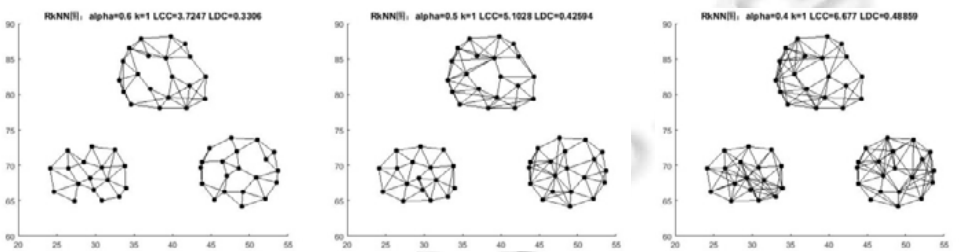
图 7 数据集 sph3 的 R3NN 图及其局部稠密性(α=0.8~0.6)



(a) α=0.7 (b) α=0.6 (c) α=0.5

Fig.8 R2NN graph and its local density of dataset sph3 (α=0.7~0.5)

图 8 数据集 sph3 的 R2NN 图及其局部稠密性(α=0.7~0.5)



(a) α=0.6 (b) α=0.5 (c) α=0.4

Fig.9 R1NN graph and its local density of dataset sph3 (α=0.6~0.4)

图 9 数据集 sph3 的 R1NN 图及其局部稠密性(α=0.6~0.4)

随机 kNN 图在固定 k 值的情况下(k=3),通过随机因子可以产生满足全局稀疏性和局部稠密性要求的邻接图,适于聚类处理,是本文构造的聚类邻接图.

2.4 基于参考邻接图的聚类质量评估

聚类结果需要考察其聚类质量,常见的聚类质量评估指标有 CH 指数、SD 有效性指数、DBI 指数、Dunn 指数等,主要思想是,通过簇内距离与簇间距离的某种形式的比值来度量聚类有效性.这些常见的评估指标需要在欧氏度量空间中统计簇内和簇间数据对象的均值方差等,指标局限性大且不稳定.

考虑到聚类邻接图已经编码了数据的局部连通性,本文提出以参考邻接图为背景和基准,建立更为稳定一

致的聚类有效性指标 *VI*.给定邻接图 AG,其每个连通分量对应一个类簇 C_i,则整个聚类结果记做$\{C_i|i=1\sim k\}$.直观地,如果每个类簇内部数据对象之间的距离都很小,则类簇质量越高,即类内边长的上界尽可能小;如果两两类簇之间的分隔距离越大,则类簇质量越高,即类间边长的下界尽可能大.

定义 7(参考邻接图). 参考邻接图记做 RAG(V,E_{RAG}),是所有待度量邻接图的超集,是度量其他邻接图的基准.通常采用初始邻接图作为参考邻接图,度量经过边删除操作产生的邻接图.

以参考邻接图 RAG(V,E_{RAG})为基准,提出簇内密集性、簇间散布性以及聚类有效性 3 种度量定义.

定义 8(簇内密集性). 给定邻接图 AG(V,E_{AG}),记其所对应的聚类结果类簇为$\{C_i|i=1\sim k\}$.对于任意一个类簇,其簇内密集性定义为以参考邻接图 RAG(V,E_{RAG})为背景的簇内数据对象间的最长边距离,则整个聚类结果的簇内密集性定义为所有类簇的簇内密集性的最大值:

$$\begin{aligned}INTRA\_COMPACTNESS(\{C_i\}_{i=1}^{k}) &= \max_{i=1\sim k}\{INTRA\_COMPACTNESS(C_i)\}\\ &= \max_{i=1\sim k}\{\max\{e_{ij}\mid e_{js}\in E_{\mathrm{RAG}}\wedge x_j\in C_i\wedge x_s\in C_i\}\}.\end{aligned}$$

推理可知:给定邻接图所表示的聚类结果的簇内密集性是以参考邻接图为基准背景下簇内数据对象之间边权距离的上界,上界越小,簇内密集性越小,聚类结果质量越高.

定义 9(簇间距离与簇间散布性). 以参考邻接图 RAG(V,E_{RAG})为基准背景,两个类簇 C_1 和 C_2 的簇间距离定义为其所属数据对象间的最小距离:

$$inter\_d(C_1,C_2)=\min\{e_{ij}|e_{ij}\in E_{\mathrm{RAG}}\wedge x_i\in C_1\wedge x_j\in C_2\}.$$

基于簇间距离定义簇间散布性.给定邻接图 AG(V,E_{AG})所表示的聚类结果$\{C_i|i=1\sim k\}$的簇间散布性,定义为以参考邻接图 RAG(V,E_{RAG})为背景的两两类簇的簇间距离的最小值:

$$INTER\_DISPERSITY(\{C_i\}_{i=1}^{k}) = \min_{i,j=1\sim k}\{inter\_d(C_1,C_2)\}.$$

不难看出:簇间散布性是最小的簇间距离,是簇间距离的下界.最小簇间距离越大,说明类簇之间分隔的越远.故簇间散布性越大,表示聚类结果质量越好.

定义 10(聚类有效性). 给定邻接图 AG(V,E_{AG}),记其相应的聚类结果类簇为$\{C_i|i=1\sim k\}$.聚类结果的聚类有效性定义为簇内密集性与簇间散布性的比值:

$$VI(\{C_i\}_{i=1}^{k}) = \frac{INTRA\_COMPACTNESS(\{C_i\}_{i=1}^{k})}{INTER\_DISPERSITY(\{C_i\}_{i=1}^{k})}.$$

聚类有效性指标越小,说明簇内密集性小且簇间散布性大,聚类结果质量更好.

2.5 基于R*k*NN图的批量边删除聚类算法

利用随机 *k*NN 图、局部高斯平滑模型、边删除准则、聚类有效性指标等概念,提出 R*k*NNClus 聚类算法.

算法 1. 基于 R*k*NN 图的边删除迭代聚类算法 R*k*NNClus

输入:数据集 $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^{n}$;

输出:一组类簇,每个类簇是数据对象的集合.

过程:

(1) 构造 R*k*NN 图.从 $\boldsymbol{X}$ 中随机抽取 $m=\alpha\times n$ 个数据对象做为锚点,将其余数据对象按距离连接到最近的 *k*(=3)个锚点,重复该步骤 *t* 次并融合结果,得到 R*k*NN 图;

(2) 设 *i*=0,记 AG(*i*)=R*k*NN,然后迭代如下步骤:

(2.1) 在 AG(*i*)图上,利用局部高斯平滑模型计算数据对象的平滑位置 $\boldsymbol{Y}$;
高斯平滑的尺度:$\sigma^2=median\{\mathrm{AG}(i)\}$;

(2.2) 假设 AG(*i*)图是聚类结果,计算 AG(*i*)的聚类有效性指标 *VI*(*i*);

(2.3) 利用边删除判定准则在 AG(*i*)图上执行边删除,得到 AG(*i*+1)图;

(2.4) 如果 AG(*i*+1)图的边数不再减少或连通分量数目大于 *sqrt*(*n*),终止迭代,跳转到步骤(3);
否则,*i*=*i*+1,跳转到步骤(2.1);

(3) 对于 AG(0)~AG(*t*)的邻接图,选择聚类有效性指标 *VI*(*i*)最小的邻接图 FAG 作为最终聚类结果,检测 FAG 的连通分量,每个连通分量作为一个类簇输出.

算法首先构造 R*k*NN 图,然后在 R*k*NN 图上根据数据对象的高斯平滑结果进行迭代边删除操作,直至图结构收敛;在迭代过程中,通过聚类有效性指标评估最合理的聚类结果,如图 10 所示.

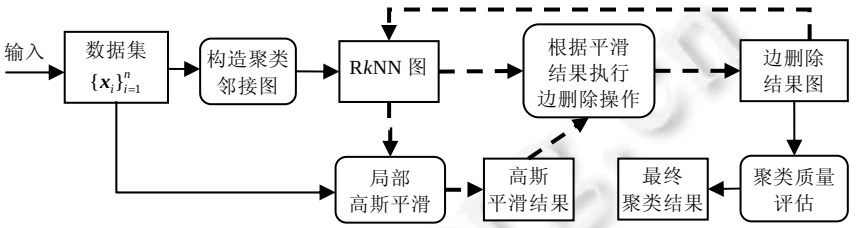


Fig.10 R*k*NNClus-the batched edge-remove clustering algorithm

图 10 R*k*NNClus 批量边删除聚类算法

2.5.1 算法复杂性分析

与经典的 *K*-Means 算法相比,R*k*NNClus 聚类算法开销较大.不过,R*k*NNClus 算法可以迭代得出类簇的数目,考虑到计算类簇数目的代价,R*k*NNClus 聚类算法的性能可以接受.

算法的主要开销由 R*k*NN 图构造、*t* 次迭代过程及图的连通分量检测这 3 个步骤组成.其中,R*k*NN 图构造的时间代价为 $O(dN(N-1)+\alpha tNN)$,图连通分量检测的时间代价为 $O(N^2)$.

此外,每次迭代需要进行局部高斯平滑计算、聚类有效性指标计算、边删除操作.其中,局部高斯平滑计算的时间代价为 $O(dN(N-1))$,聚类有效性指标计算的时间代价为 $O(dN(N/2-1))+O(dN^2+dk(k-1)N(N/2-1))$,边删除操作的时间代价为 $O(dN^2)$.综合起来,R*k*NNClus 聚类算法的总体计算复杂度为 $O(dk^2tN^2)$.

当然,时间代价还有一些优化余地,例如多次局部高斯平滑计算可以重复利用之前的结果,也可以将诸多距离矩阵缓存起来提高性能.不过,本文关注的焦点是如何构建适于聚类处理的高质量的邻接图以及如何不依赖于过多参数比较自动地获得高质量的聚类结果,需要付出更多时间代价.

2.5.2 参数设置分析

R*k*NNClus 聚类算法是在满足全局稀疏和局部稠密的 R*k*NN 邻接图上通过多次执行边删除操作及聚类有效性评估获得最佳聚类结果的迭代过程.算法涉及到两个参数设置问题:其一是构造 R*k*NN 图时的随机因子 α,其二是局部高斯平滑模型的尺度参数 σ^2.

R*k*NN 图构造的随机因子 α,其物理含义表示作为固定锚点的数据对象比例,随机抽取的锚点数据对象必须能够表达数据分布情况,同时还要能够通过随机性模糊掉局部边长的细微差距,前者要求抽样比例大,后者要求抽样比例小,前者分布代表性必须满足,故选择 α=0.5~0.8,推荐 α=0.618.

对于局部高斯平滑模型的尺度参数 σ,可以根据高斯模糊平滑与核密度函数估计问题的一致性,借鉴核密度估计中带宽参数的选择方法来估计平滑尺度.考虑密度估计的渐进积分均方误差最小化准则,有最优平滑尺度 $\sigma=O(n^{-1/5})$,表明了随着样本数量的增加平滑尺度减小的速度量级.但是对具体数据集,还是无法给出具体的参数取值.实践中可以采用代入法(plug-in)和平滑交叉验证(smoothed cross-validation)来估计最优平滑尺度.比较简便的方法是根据 3σ原则,在给定区间内手动设定并调整平滑尺度参数值:

$$\sigma\in\left[\frac{var(\boldsymbol{X})}{\sqrt{n}},\frac{3^2\times 2\pi\times var(\boldsymbol{X})}{\sqrt{n}}\right].$$

但是上述平滑尺度参数设置方法效果不佳,原因在于 R*k*NN 图已经对数据对象的局部连通性做了表达,参与高斯平滑模型计算的只是局部连通的数据对象.因此,R*k*NNClus 聚类算法采用编码了局部性的 R*k*NN 图来估计平滑尺度参数.对于给定的邻接图 AG,考虑到统计稳定性,高斯平滑尺度参数估计为邻接图边长的中值 $\sigma^2=\text{median}\{E_{AG}\}$,实验结果表明了平滑尺度参数设置的有效性.

3 实验评估

本节通过实验比较来研究 R*k*NNClus 聚类算法的性能.实验涵盖了 5 种现有的聚类算法:*K*-Means(简称 KM)、核化 *K*-Means(简称 KKM)、Chameleon(简称 CHM)、MeanShift(简称 MSF)、规范化谱聚类算法(简称 NSC),通过在一系列数据集上执行这 5 种聚类算法,来比较评估聚类算法性能.

实验数据是在各种聚类算法评估中常用的模拟数据以及网络公开的基准数据集的测试数据,表 2 中说明了实验数据的主要规格.实验首先在 7 组模拟数据上进行 5 种聚类算法评估,然后在 BSDS500,MNIST,Isolet 基准数据集上比较各类进行聚类算法评估.BSDS500 数据集包含大量人工分段标注的自然图像,MNIST 数据集包含数字 0~9 的分辨率有 16×16 和 28×28 的手写体 8 位灰度图像,Isolet 数据集包含字母 A~Z 独立发音的声学特征.实验的硬件环境:Intel Core i5-3210M 2.5GHz 的 CPU 处理器、8.0GB 的内存,软件系统环境:64 位 Microsoft Windows 7.0 操作系统、Matlab7.0.4.

Table 2 Simulation and benchmark datasets for algorithm evaluation

表 2 算法评估用的模拟数据和基准数据集

编号	数据集	类簇数目	特征数量	对象数量	数据来源
1	chained_sph2	2	2	70	模拟
2	sph4	4	2	784	模拟
3	fork	4/3	2	180	模拟
4	snake	3	2	688	模拟
5	circle3	3	2	1 500	模拟
6	semicircle3	3	2	903	模拟
7	longtail	2	2	187	模拟
8	BSDS500	人工标注	像素	500	https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html#bsds500
9	MNIST	10	784	70 000	http://yann.lecun.com/exdb/mnist/
10	Isolet	26	617	7 797	http://archive.ics.uci.edu/ml/datasets/ISOLET

3.1 模拟数据实验

对于表 2 中的 1~7 组二维的模拟数据,分别给出 K-Means、核化 K-Means、Chameleon、MeanShift、规范化谱聚类算法以及 R*k*NNClus 算法的聚类结果,根据视觉直观判断聚类结果的质量.从各模拟数据的实验中可以观察到:

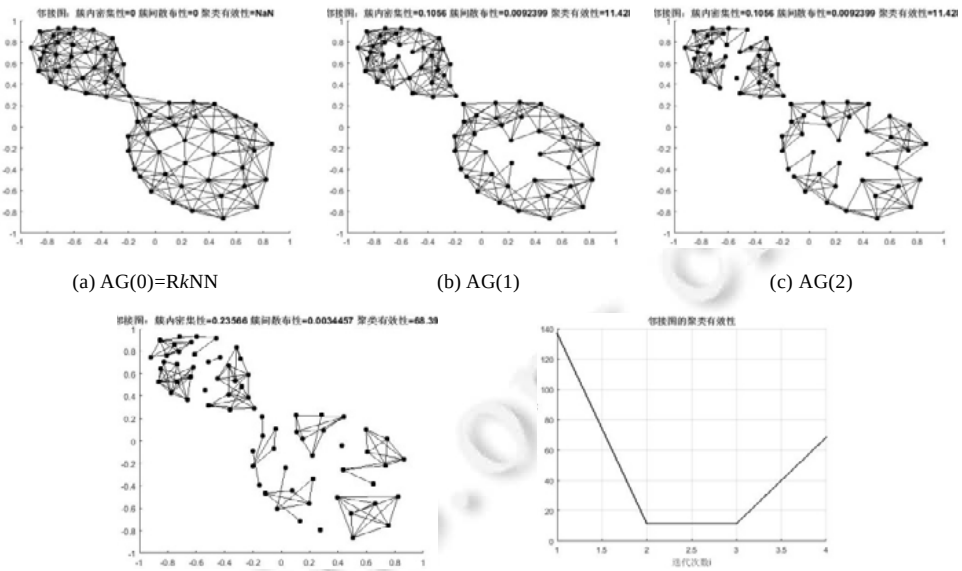
- R*k*NNClus 算法和基于 R*k*NN 图的规范化谱聚类算法的聚类质量较高,能够处理类簇形状大小变化、类簇分离和粘连等情况.不同的是:规范化谱聚类算法要求设定类簇数目 *k*;而 R*k*NNClus 算法将所有参数归结为随机化因子 α,其物理含义是用作锚点的数据对象采样比例,理解起来非常直观,一般设定 α=0.618,此外无需指定类簇数目、核宽等参数;
- *K*-Means、核化 *K*-Means、Chameleon、MeanShift 等算法需要在合适的使用场合才能发挥作用,且往往依赖于特定参数的精细设置.

3.1.1 R*k*NNClus 实验结果

图 11 给出了 chained_sph2 数据集的 R*k*NNClus 算法每次迭代过程的聚类结果,如图 11(a)~图 11(d)所示.每个聚类结果计算相应的簇内密集性、簇间散布性、聚类有效性指数,并绘制出图 11(e)的聚类有效性曲线.可以看出:第 2 次迭代的聚类有效性指数最小,聚类结果如图 11(b)所示,数据被分割为两个类簇.

图 12 给出了 sph4 数据集的 R*k*NNClus 算法每次迭代过程的聚类结果及其聚类有效性曲线.可以看出:第 1 次迭代的聚类有效性指数最小,聚类结果如图 12(a)所示,数据被分割为 4 个类簇.

图 13 给出了 fork 数据集的 R*k*NNClus 算法每次迭代过程的聚类结果及其聚类有效性曲线.可以看出:第 1 次迭代的聚类有效性指数最小,聚类结果如图 13(a)所示,数据被分割为 3 个类簇.不过,详细观察图 13 的聚类有效性曲线会发现:第 2 次的聚类结果其聚类有效性指数也很小,实际上也可以作为合理的聚类结果,此时数据被分割为 4 个类簇,如图 13(a)和图 13(b)所示.

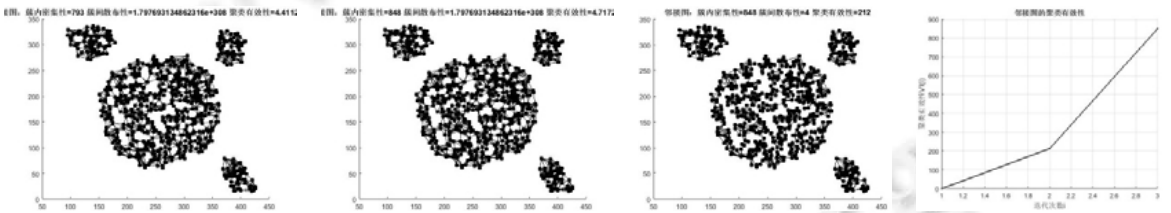


(a) AG(0)=R*k*NN (b) AG(1) (c) AG(2)

(d) AG(3) (e) 聚类有效性曲线

Fig.11 Clustering results of R*k*NNClus algorithm for dataset chained_sph2

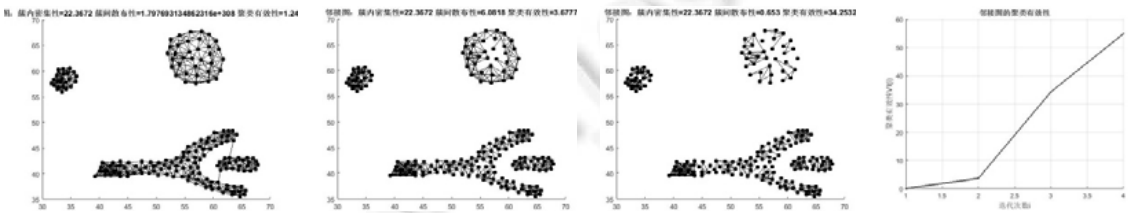
图 11 数据集 chained_sph2 的 R*k*NNClus 算法的聚类结果



(a) AG(0)=R*k*NN (b) AG(1) (c) AG(2) (d) 聚类有效性曲线

Fig.12 Clustering results of R*k*NNClus algorithm for dataset sph4

图 12 数据集 sph4 的 R*k*NNClus 算法的聚类结果



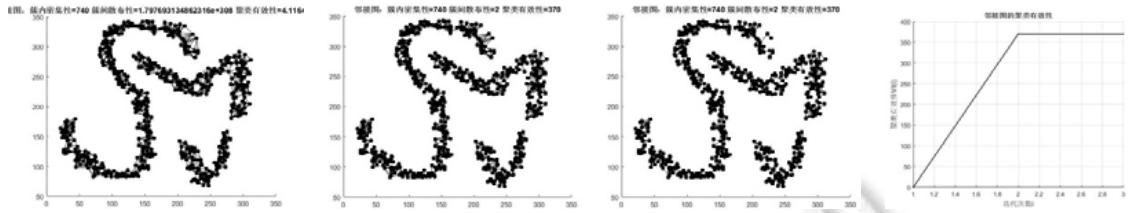
(a) AG(0)=R*k*NN (b) AG(1) (c) AG(2) (d) 聚类有效性曲线

Fig.13 Clustering results of R*k*NNClus algorithm for dataset fork

图 13 数据集 fork 的 R*k*NNClus 算法的聚类结果

图 14 给出了 snake 数据集的 R*k*NNClus 算法每次迭代过程的聚类结果及其聚类有效性曲线.可以看出:第 1 次迭代的聚类有效性指数最小,聚类结果如图 14(a)所示,数据被分割为 3 个类簇.

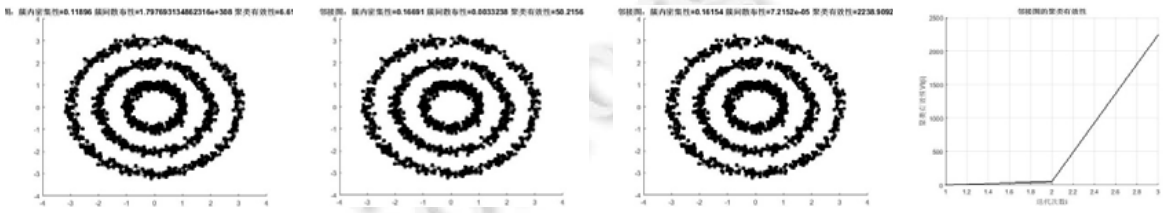
图 15 给出了 circle3 数据集的 R*k*NNClus 算法每次迭代过程的聚类结果及其聚类有效性曲线.可以看出:第 1 次迭代的聚类有效性指数最小,聚类结果如图 15(a)所示,数据被分割为 3 个类簇.



(a) AG(0)=R*k*NN (b) AG(1) (c) AG(2) (d) 聚类有效性曲线

Fig.14 Clustering results of R*k*NNClus algorithm for dataset snake

图 14 数据集 snake 的 R*k*NNClus 算法的聚类结果

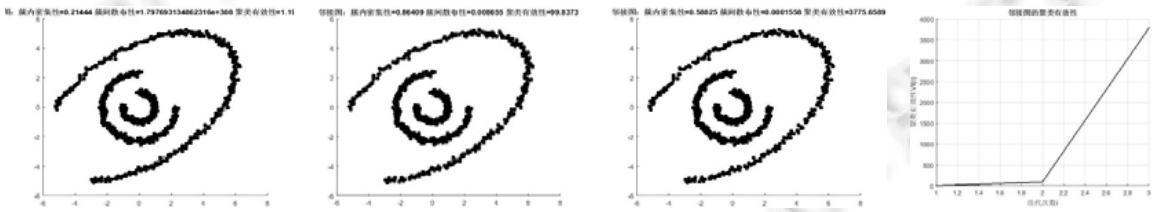


(a) AG(0)=R*k*NN (b) AG(1) (c) AG(2) (d) 聚类有效性曲线

Fig.15 Clustering results of R*k*NNClus algorithm for dataset circle3

图 15 数据集 circle3 的 R*k*NNClus 算法的聚类结果

图 16 给出了 semicircle3 数据集的 R*k*NNClus 算法每次迭代过程的聚类结果及其聚类有效性曲线.可以看出:第 1 次迭代的聚类有效性指数最小,聚类结果如图 16(a)所示,数据被分割为 3 个类簇.

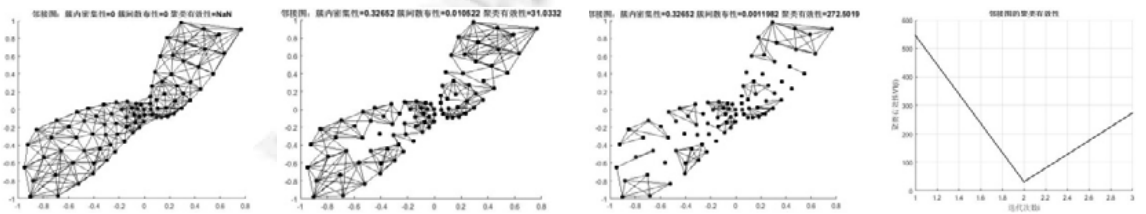


(a) AG(0)=R*k*NN (b) AG(1) (c) AG(2) (d) 聚类有效性曲线

Fig.16 Clustering results of R*k*NNClus algorithm for dataset semicircle3

图 16 数据集 semicircle3 的 R*k*NNClus 算法的聚类结果

图 17 给出了 longtail 数据集的 R*k*NNClus 算法每次迭代过程的聚类结果及其聚类有效性曲线.可以看出:第 2 次迭代的聚类有效性指数最小,聚类结果如图 17(b)所示,数据被分割为两个类簇.



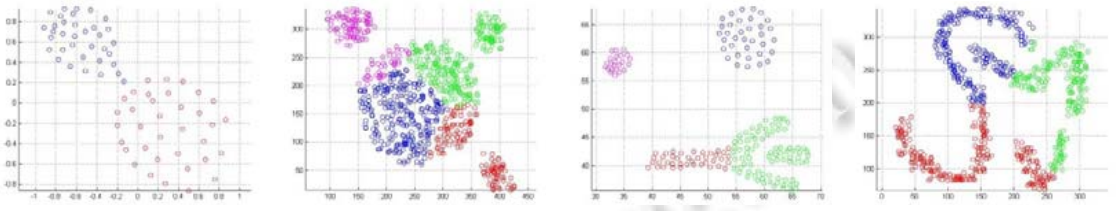
(a) AG(0)=R*k*NN (b) AG(1) (c) AG(2) (d) 聚类有效性曲线

Fig.17 Clustering results of R*k*NNClus algorithm for dataset longtail

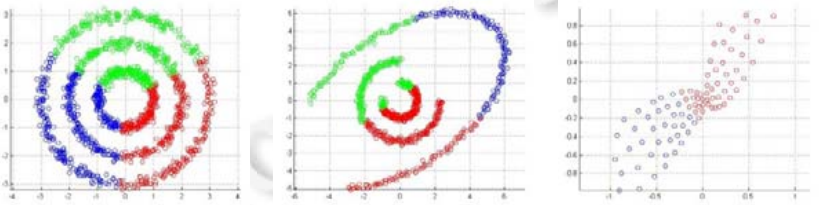
图 17 数据集 longtail 的 R*k*NNClus 算法的聚类结果

3.1.2 其他聚类算法实验结果

图 18 给出了 1~7 组模拟数据的 *K*-Means 算法的聚类结果,手动设置了类簇数目 *k*.可以想象:除了数据集 chained_sph2 和 longtail 之外,其他数据的聚类效果均不佳.



(a) chained_sph2 (*k*=2) (b) sph4 (*k*=4) (c) fork (*k*=4) (d) snake (*k*=3)

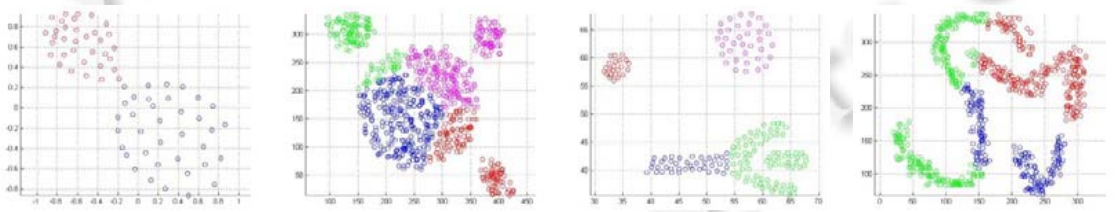


(e) circle3 (*k*=3) (f) semicircle3 (*k*=3) (g) longtail (*k*=2)

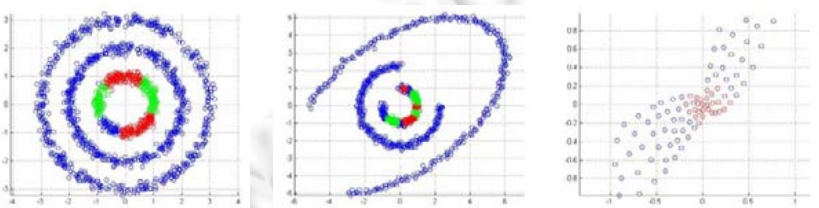
Fig.18 Clustering results of *K*-Means algorithm for datasets 1~7

图 18 模拟数据 1~7 组的 *K*-Means 算法聚类结果

图 19 给出了 1~7 组模拟数据的核化 *K*-Means 算法的聚类结果,手动设置类簇数目 *k* 以及核半径σ.经过多次调试核半径σ,聚类效果也不是很好,这应该是核化 *K*-Means 难以应用的重要原因.



(a) chained_sph2 (σ=0.318) (b) sph4 (σ=8116) (c) fork (σ=325) (d) snake (σ=1170)

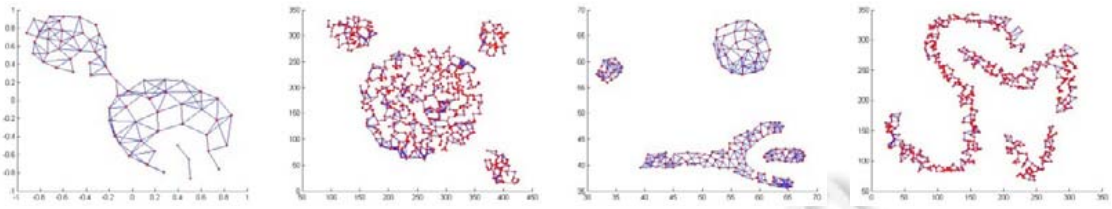


(e) circle3 (σ=0.108) (f) semicircle3 (σ=0.014) (g) longtail (σ=0.025)

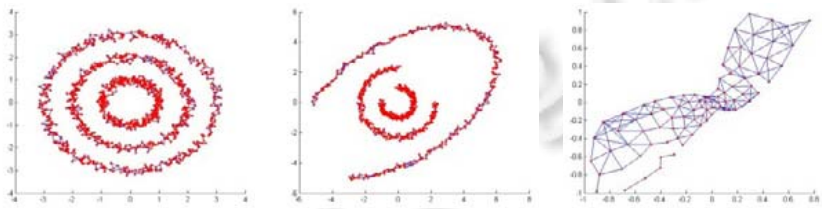
Fig.19 Clustering results of Kernel *K*-Means algorithm for datasets 1~7(σ is kernel radius)

图 19 模拟数据 1~7 组的核化 *K*-Means 算法聚类结果(σ为核半径)

图 20 给出了 1~7 组模拟数据的核化 Chameleon 算法的聚类结果,需要手动设置类簇数目 *k*.可以看出:对于分离程度较好的数据集 Chameleon,算法聚类效果很好,如图 20(b)~图 20(f)所示;但是对于类簇粘连的数据集,Chameleon 算法缺乏很好的处理手段,如图 20(a)、图 20(g)所示.



(a) chained_sph2 (*k*=2) (b) sph4 (*k*=4) (c) fork (*k*=4) (d) snake (*k*=3)

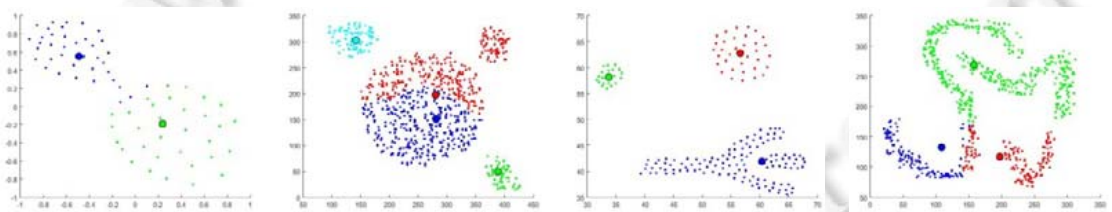


(e) circle3 (*k*=3) (f) semicircle3 (*k*=3) (g) longtail (*k*=2)

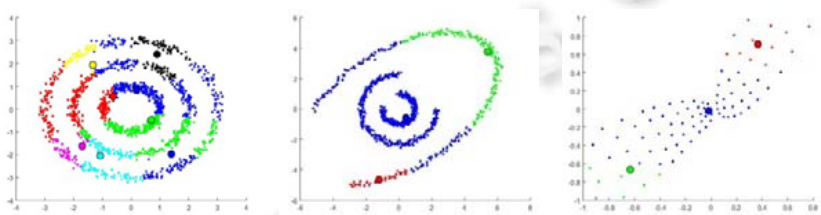
Fig.20 Clustering results of Chameleon algorithm for datasets 1~7

图 20 模拟数据 1~7 组的 Chameleon 算法聚类结果

图 21 给出了 1~7 组模拟数据的 MeanShift 算法的聚类结果,无需手动设置类簇数目 *k*,但需要仔细调整核半径σ.可以看出:MeanShift 算法适于处理类球形且分离较好的数据,如图 21(a)~图 21(c)所示.对于形状不规则数据,聚类效果不好,如图 21(d)~图 21(g)所示.此外,核半径估计比较困难.



(a) chained_sph2 (σ=0.8) (b) sph4 (σ=80) (c) fork (σ=8) (d) snake (σ=100)

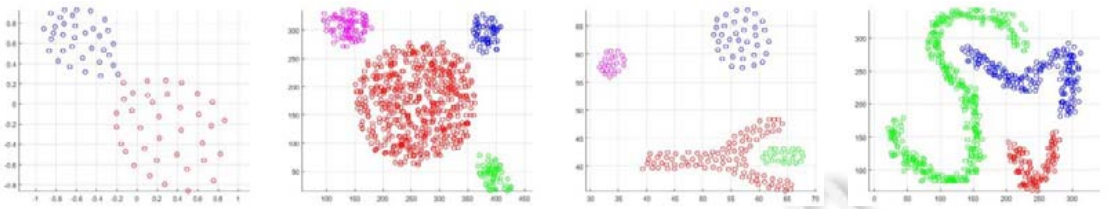


(e) circle3 (σ=0.9) (f) semicircle3 (σ=2.0) (g) longtail (σ=0.34)

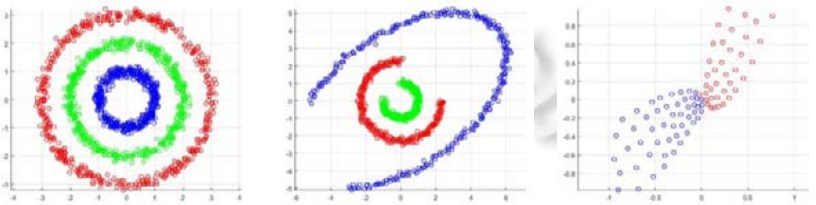
Fig.21 Clustering results of MeanShift algorithm for datasets 1~7

图 21 模拟数据 1~7 组的 MeanShift 算法聚类结果

规范化谱聚类算法通常采用全连接图、*k*NN 图或ε邻域图作为相似性图,在实践中,全连接图的核宽参数、*k*NN 图的邻居参数 *k*、ε邻域图的ε参数设定比较困难,聚类效果也不稳定.因此,本文采用随机化的 *k*NN 图(随机因子α=0.8)作为规范化谱聚类算法的相似性图,聚类质量很好且稳定性显著提高,也从另一个侧面证明随机 *k*NN 图已经编码了数据的局部连通结构,对于谱聚类处理效果有明显提升.图 22 给出了 1~7 组模拟数据的 R*k*NN 图规范化谱聚类算法的聚类结果.



(a) chained_sph2 (b) sph4 (c) fork (d) snake



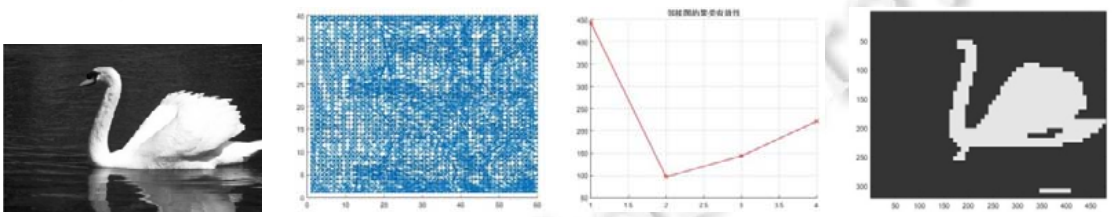
(e) circle3 (f) semicircle3 (g) longtail

Fig.22 Clustering results of spectral clustering algorithm normalized on R*k*NN graph for datasets 1~7

图 22 模拟数据 1~7 组的 R*k*NN 图规范化谱聚类算法聚类结果

3.2 基准数据实验

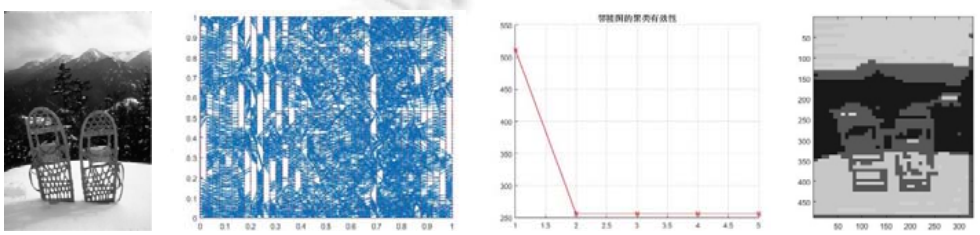
BSDS500 是 Berkeley 计算视觉研究组构建的包含大量人工分段标注的自然图像的基准数据[27],本实验从 BSDS500 中抽取部分图像利用 R*k*NNClus 聚类算法进行分段处理.在进行聚类分段之前,首先对图像进行灰度化和 8×8 分块处理,然后将分块灰度数据转换为(*x*,*y*,灰度值)的形式导入 R*k*NNClus 聚类算法,算法中随机 *k*NN 图采用随机因子 α=0.8,*k*=3 的参数进行构建.部分实验结果如图 23~图 25 所示,篇幅所限,其他结果不予赘述.可以看出,R*k*NNClus 聚类算法可以用于图像的初步分段处理.



(a) 8086 图像 (b) R*k*NN 图 (c) 聚类有效性曲线 (d) 图像分段结果

Fig.23 Segmentation results of R*k*NNClus algorithm for datasets BSD500-8086

图 23 BSD500-8086 图像数据的 R*k*NNClus 算法聚类分段结果

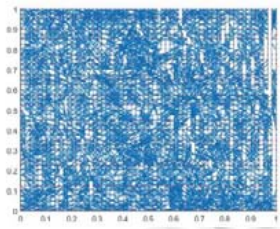


(a) 2018 图像 (b) R*k*NN 图 (c) 聚类有效性曲线 (d) 图像分段结果

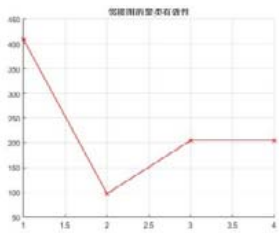
Fig.24 Segmentation results of R*k*NNClus algorithm for datasets BSD500-2018

图 24 BSD500-2018 图像数据的 R*k*NNClus 算法聚类分段结果

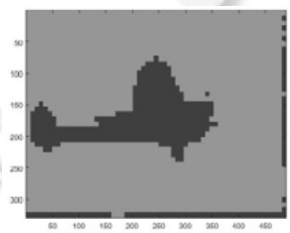
(a) 3063 图像



(b) R*k*NN 图



(c) 聚类有效性曲线



(d) 图像分段结果

Fig.25 Segmentation results of R*k*NNClus algorithm for datasets BSD500-3063

图 25 BSD500-3063 图像数据的 R*k*NNClus 算法聚类分段结果

实验从 BSD500 基准数据集中随机抽取出 20 幅图像,用各类聚类算法进行分段处理(类簇数目设置为 k=3).

- MNIST 数据集包含数字 0~9 的分辨率有 16×16 和 28×28 的手写体 8 位灰度图像[28],实验采用 28×28 数字图像,每个数字从数据集中随机抽取 50 个样本,共 50×10 个样本;
- Isolet 数据集包含 150 个说话人的字母 A~Z 独立发音的声学特征[29],每个人每个字母发声两次,说话人分成 5 组,每组 30 人,标记为 Isolets1~Isolets5,实验每次从数据集中随机抽取每个字母的 50 个样本,共 50×26 个样本.

样本导入 R*k*NNClus 算法进行聚类处理,处理结果与基准数据集中的标记结果进行比较计算出平均 *F*-measure 度量指标,表 3 给出了 *K*-Means(KM)、核化 *K*-Means(KKM)、Chameleon(CHM)、MeanShift(MSF)、规范化谱聚类算法(NSC)以及 R*k*NNClus 在基准数据集上的聚类测试评估结果.

Table 3 *F*-measure of each clustering algorithm on benchmark datasets

表 3 基准数据集上各聚类算法的 *F*-measure 评估

	KM	KKM	CHM	MSF	NSC	R*k*NNClus
BSD500	0.55	0.55	0.67	0.62	0.67	0.66
MNIST	0.48	0.53	0.58	0.54	0.58	0.62
Isolet	0.64	0.57	0.77	0.74	0.82	0.80

可以看出:图聚类算法效果普遍高于 *K*-Means 和核化 *K*-Means,R*k*NNClus 算法无需类簇数目即可获得不错的效果.

4 结论和下一步工作

本文总结概括了批量边删除聚类算法的通用框架,提出了基于局部高斯平滑模型建立批量边删除准则,通过高斯平滑位移进行边删除判定;定义了适于聚类的邻接图的一般性质和度量,提出并证明了引入随机因子的随机 *k*NN 图是一种适合聚类处理的邻接图,它增强了顶点之间的局部连通性,使聚类结果不再强烈依赖于某条边的保留或删除.R*k*NNClus 算法简洁高效,依赖参数少,无需指定类簇数目,模拟数据和基准数据集实验均有证明.算法依赖于高斯平滑模型,因此适用于欧氏空间且时间空间开销较大,下一步工作是针对算法效率进行研究改进.

References:

[1] Zemel RS, Carreira-Perpinán MA. Proximity graphs for clustering and manifold learning. In: Proc. of the Advances in Neural Information Processing Systems. 2004. 225−232.

[2] Galluccio L, Michel O, Comon P. Clustering with a new distance measure based on a dual-rooted tree. Information Sciences, 2013, 96−113. [doi: 10.1016/j.ins.2013.05.040]

[3] Galluccio L, Michel O, Comon P. Graph based k-means clustering. Signal Processing, 2012,92(9):1970−1984.

[4] Müller AC, Nowozin S, Lampert CH. Information Theoretic Clustering Using Minimum Spanning Trees. Berlin, Heidelberg: Springer-Verlag, 2012. [doi: 10.1007/978-3-642-32717-9_21]

[5] Grygorash O, Zhou Y, Jorgensen Z. Minimum spanning tree based clustering algorithms. In: Proc. of the Int'l Conf. on Tools with Artificial Intelligence. 2006. 73−81.

[6] Gower J, Ross J. Minimum spanning trees and single linkage cluster analysis. Applied Statistics, 1969,18:54−64.

[7] Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014,344(6191):1492−1496. [doi: 10.1126/science.1242072]

[8] Zhang T, Ramakrishnan R, Linvy M. BIRCH: An efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM Press, 1996. 103−114. [doi: 10.1145/233269.233324]

[9] Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. In: Haas LM, Tiwary A, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 1998. 73−84. [doi: 10.1145/276304.276312]

[10] Karypis G, Han EH, Kumar V. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer, 1999, 32(8):68−75.

[11] Ester M., Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial database with noise. In: Simoudis E, Han J, Fayyad UM, eds. Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. Oregon: AAAI Press, 1996. 226−231.

[12] Ankerst M, Breuning M, Kriegel HP, Sander J. OPTICS: Ordering points to identify the clustering structure. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM Press, 1999. 49−60.

[13] Hinneburg A, Keim DA. An efficient approach to clustering in large multimedia databases with noise. In: Proc. of the Knowledge Discovery and Data Mining. 1998. 58−65.

[14] Hinneburg A, Gabriel H. DENCLUE 2.0: Fast clustering based on kernel density estimation. In: Proc. of the Intelligent Data Analysis. 2007. 70−80.

[15] Wang W, Yang J, Muntz RR. STING: A statistical information grid approach to spatial data mining. The VLDB Journal, 1997, 186−195.

[16] Sheikholeslami G, Chatterjee S, Zhang A. Wavecluster: A multi-resolution clustering approach for very large spatial databases. The VLDB Journal, 1998,98:428−439.

[17] Kailing K, Kriegel H, Kroger P. Density-Connected subspace clustering for high-dimensional data. In: Proc. of the SIAM Int'l Conf. on Data Mining. 2004. 246−256.

[18] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 1977,39(1):1−38.

[19] Fisher DH. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 1987,2(2):139−172.

[20] Vesanto J, Alhoniemi E. Clustering of the self-organizing map. IEEE Trans. on Neural Networks, 2000,11(3):586−600.

[21] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems, 2002,2:849−856.

[22] Nixon MS, Aguado AS. Feature Extraction and Image Processing. 2nd ed., Academic Press, 2008. 88−89.

[23] Leung Y, Zhang J, Xu Z. Clustering by scale-space filtering. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000, 22(12):1396−1410.

[24] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002,24:603−619.

[25] Urquhart R. Graph theoretical clustering based on limited neighbourhood sets. Pattern Recognition, 1982,15(3):173−187.

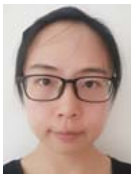
[26] Watts DJ, Strogatz SH. Collective dynamics of ‘small-world’ networks. Nature, 1998,393(6684):440−442. [doi: 10.1038/30918]

[27] Arbelaez P, Maire M, Fowlkes CC. Contour detection and hierarchical image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011,33(5):898−916.

[28] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based learning applied to document recognition. Proc. of the IEEE, 1998,86(11): 2278−2324.

[29] Fanty MA, Cole RA. Spoken letter recognition. In: Proc. of the Neural Information Processing Systems. 1990. 220−226.

雷小锋(1975－),男,陕西合阳人,博士,副教授,主要研究领域为数据库与数据挖掘,机器学习.



陈皎(1987－),女,硕士生,主要研究领域为数据挖掘.

毛善君(1964－),男,博士,教授,博士生导师,主要研究领域为数字矿山.

谢昆青(1957－),男,博士,教授,博士生导师,主要研究领域为复杂时空系统建模,智能交通系统,智能数据分析与知识发现,时空数据库与数据仓库,传感器网络与地理信息系统.