文章编号: 1001-0920 (2016) 03-0541-06 **DOI:** 10.13195/j.kzyjc.2014.1963

不确定数据的最优 k 近邻和局部密度聚类算法

陆亿红, 夏 聪

(浙江工业大学 计算机科学与技术学院，杭州 310023)

摘 要: 传统聚类算法一般针对的是确定数据, 无法解决不确定数据的聚类问题; 现有基于密度的不确定数据聚类算法存在参数敏感且计算率低的问题. 对此, 在引进新的不确定数据相异度函数、最优 k 近邻、局部密度和互包含概念的基础上, 提出解决不确定数据聚类问题的不确定数据的最优 k 近邻和局部密度聚类 (OLUC) 算法. 该算法不仅能降低参数敏感性, 提高计算效率, 而且具有动态自适应优化 k 近邻, 快速发现聚类中心和除噪优化的能力. 实验结果表明, 所提出的算法对无论是否存在噪声的不确定数据集都效果良好.

关键词: k 近邻；局部密度；不确定数据；聚类算法

中图分类号: TP391 **文献标志码:** A

Optimal k-nearest neighbors and local density-based clustering algorithm for uncertain data

LU Yi-hong, *XIA Cong*

(College of Computer Science and Technology，Zhejiang University of Technology，Hangzhou 310023，China. Correspondent：LU Yi-hong，E-mail：lyh@zjut.edu.cn)

Abstract: Traditional clustering algorithms aim to certain data in general, which cannot solve the clustering problem for uncertain data. The existing density-based clustering algorithms for uncertain data have the problems that parameters are too sensitive and the computational efficiency is low. Therefore, an algorithm, named optimal k-nearest neighbors and local density-based clustering algorithm for uncertain data(OLUC), is proposed to solve the clustering problem for uncertain data by introducing concepts of new dissimilarity function for uncertain data, optimal k-nearest neighbors, local density and mutual inclusion relation. The algorithm not only can reduce the sensitivity of parameters and improve the computational efficiency, but also has the abilities of optimizing k-nearest neighbors in the dynamic adaptive way, deciding cluster center quickly and optimizing denoising. The experimental results show that the algorithm is effective on clustering for uncertain data whatever with noise or without noise, and achieves good results.

Keywords: k-nearest neighbors；local density；uncertain data；clustering algorithm

0 引 言

聚类分析是数据挖掘中的重要技术, 广泛应用于各个领域. 聚类是将数据分成多个簇, 使得相同簇中的数据具有较高相似度、不同簇中的数据具有较高相异度. 其中基于密度的聚类算法由于可以发现任意形状的簇, 过滤噪声信息, 从而获得了良好的聚类结果, 吸引众多学者进行研究. 文献 [1] 运用局部密度解决分布式数据聚类, 对数据分布异常和高维数据有良好适应性; 文献 [2] 引入 k 邻域概念, 解决高维空间聚类; 文献 [3] 采用相对密度, 运用增量式聚类算法实现混合属性数据集中不同密度等级簇的发现; 文献 [4] 结合维度距离, 对含有混合属性数据的离线微簇聚类.

随着信息技术的快速发展, 大量新型数据不断涌现. 不确定数据作为其中之一, 近年来受到广泛关注. 以往针对确定性数据的聚类算法不能直接适用. Kriegel 等[5]结合概率密度函数, 设计不确定数据间的距离函数, 将其定义为模糊距离, 并提出 FDBSCAN 算法; Ngai 等[6]提出 UK-means 算法, 其基本思想与 k-means[7]相似, 利用最小边界矩形描述数据点可能出现的区域, 通过设计剪枝策略来提高计算效率; 张晨等[8]结合概率引力处理不确定数据流, 提出了 EMicro 算法.

收稿日期: 2014-12-26；**修回日期:** 2015-03-13.

基金项目: 水利部公益性行业科研专项基金项目(201401044).

作者简介: 陆亿红(1968−), 女, 副教授, 从事软件理论、数据挖掘等研究；夏聪(1990−), 男, 硕士生, 从事数据挖掘的研究.

本文鉴于基于密度算法在设置参数和计算效率方面薄弱的问题, 结合不确定数据的存在概率, 重新定义元组之间的相异度函数, 提出一种面向不确定数据聚类的不确定数据的最优 k 近邻和局部密度聚类(OLUC)算法, 采用动态自适应的最优 k 近邻结构, 计算局部相对密度, 降低参数选择和密度等级的敏感性. 通过结合聚类中心是由一些局部密度较低的点围绕, 且这些点距离其他较高局部密度点的距离较远的思想, 快速发现聚类中心, 简化聚类以提高计算效率. 此外, 引入互包含关系对噪声点进行处理, 降低噪声信息的干扰, 实现良好的聚类效果.

1 不确定数据

不确定数据的表示方式有多种, 例如: 测量数据的区间数, 决策数据的三角模糊数, 传输数据的点概率模型等. 本文研究点概率模型, 通过一个 $[0,1]$ 之间的概率值和确定的元组属性值表示.

定义 1(不确定元组) 不确定元组 $(\boldsymbol{X},p)$ 是一个由 d 维向量 $\boldsymbol{X}$ 和标量 p 构成的元组. 其中: $\boldsymbol{X}$ 表示该元组的 d 维属性值, $\boldsymbol{X}=(x_1,x_2,\cdots,x_d)$, x_i 是第 i 维的属性值, p 是该元组的存在概率, $0\leqslant p\leqslant 1$.

定义 2(不确定数据集) 不确定数据集 S 是一个由相互独立的 d 维不确定元组 (X_i,p_i) 构成的集合, $S=\{(\boldsymbol{X}_1,p_1),\cdots,(\boldsymbol{X}_n,p_n)\}$, 其中: $\boldsymbol{X}_i$ 是第 i 个元组的值, p_i 是该元组的存在概率, $0\leqslant p_i\leqslant 1$.

传统距离或相似度公式仅能用于确定数据之间的度量, 未考虑元组存在级的不确定性, 因而无法直接用于不确定元组之间的度量. 本文结合传统欧氏距离公式提出一种针对不确定元组的度量方式.

定义 3(不确定相异度) 不确定相异度 ds_{ij} 指两个不确定元组 $(\boldsymbol{X}_i,p_i)$ 和 $(\boldsymbol{X}_j,p_j)$ 之间的相异程度, 其公式如下:

$$ds_{ij}=\frac{\|\boldsymbol{X}_i-\boldsymbol{X}_j\|}{(1-|p_i-p_j|)p_ip_j}=\frac{\sqrt{(\boldsymbol{X}_i-\boldsymbol{X}_j)^2}}{(1-|p_i-p_j|)p_ip_j}=\frac{\sqrt{\sum_{k=1}^{d}(x_{ik}-x_{jk})^2}}{(1-|p_i-p_j|)p_ip_j}. \tag{1}$$

由式(1)可看出, 两个不确定元组之间的相异程度与它们之间的欧氏距离成正比, 同时与两个不确定元组的存在概率之积成反比. 此外, 式(1)综合考虑两个不确定元组之间的存在概率差异, 存在概率差值越小, 它们的不确定相异度越小, 反之越大. 当不存在不确定性时, 它们之间的不确定相异度等于传统的欧氏距离.

2 最优 k 近邻和局部密度

本文算法不同于传统基于密度的聚类算法 DBSCAN[9], 采用 ε 邻域, 限定数据点的邻域半径, 邻域内的点构成一个集合; 又区别于传统的 k 近邻[10]算法, 采用固定的 k 值, 限定近邻点的个数, 最近的 k 个数据点构成一个集合. 本文所采用的聚类集合表示方式是对文献[11]中用于数据流聚类表示结构的一种改进.

定义 4(不确定近邻集合) 给定不确定数据集 $S=\{(\boldsymbol{X}_1,p_1),\cdots,(\boldsymbol{X}_n,p_n)\}$, 且 $|S|\geqslant k$, 则以 (X_i,p_i) 为核心的 k 不确定近邻集合是 S 中与 (X_i,p_i) 之间不确定相异度最小的 k 个元组构成的集合, 记作 $N_k((X_i,p_i))$. 不失一般性, 假设数据集 S 中的不确定元组已按照与 (X_i,p_i) 的相异度升序排序为 $S=\{(\boldsymbol{X}_1,p_1),\cdots,(\boldsymbol{X}_{n-1},p_{n-1})\}$, 即 $ds_{i1}<\cdots<ds_{i(n-1)}$, 则 $N_k((X_i,p_i))=\{(X_1,p_1),\cdots,(X_k,p_k)\}$.

定义 5(最优不确定近邻集合) 最优不确定近邻集合是对不确定近邻集合的近邻数进行优化后产生的集合. 设已有以 (X_i,p_i) 为核心且以不确定相异度升序排序的 k 不确定近邻集合 $N_k((X_i,p_i))=\{(X_1,p_1),\cdots,(X_k,p_k)\}$, 记与 (X_i,p_i) 之间的不确定相异度最小的 l 个不确定元组的平均不确定相异度为 $ds_{\rm pre}$, 称为先前平均相异似度, 即 $ds_{\rm pre}=\sum_{j=1}^{l}ds_{ij}\Big/l$; 对于任意给定的 $m=l+1$, $ds_{\rm now}=\sum_{j=1}^{m}ds_{ij}\Big/m$, 称为当前平均相异度. 最优不确定近邻集合 $\mathrm{Opt}((X_i,p_i))$ 是指以不确定元组 (X_i,p_i) 为核心且满足 $|(ds_{\rm now}-ds_{\rm pre})/ds_{\rm pre}|\leqslant\omega$ 的最大集合, 其中 ω 作为近邻扩展的阈值参数.

阈值 ω 作为其唯一参数, 具有如下重要意义:

1) 使得每个核心元组在满足一定精度的前提下最大化不确定近邻集合, 不同于传统近邻算法因固定近邻数而造成部分有价值的周围数据被忽略, 从而改善了聚类效果、提高了聚类速度;

2) 贯彻数据量较大的情况下, 尽量挖掘有价值数据的思路, 对处理大数据量保证效率和精度具有重要的指导意义.

定理 1 最优不确定近邻集合所满足的条件使得其自身可以自适应扩展, 相同阈值 ω 情况下, 其扩展能力会随周围元组的数量和元组的不确定相异度发生变化, 当元组数量较多且相应不确定相异度较大时, 扩展能力较强, 反之较弱.

证明 因

$$|(ds_{\rm now}-ds_{\rm pre})/ds_{\rm pre}|\leqslant\omega\Leftrightarrow$$

$$\Big(\sum_{j=1}^{l}ds_{ij}+ds_{im}\Big)\Big/\sum_{j=1}^{l}ds_{ij}\leqslant(\omega+1)(l+1)/l\rightarrow$$

$$ds_{im} \leqslant \left(\omega l \sum_{j=1}^{l} ds_{ij} + \omega \sum_{j=1}^{l} ds_{ij} + \sum_{j=1}^{l} ds_{ij}\right) \Big/ l \rightarrow$$

$$ds_{im} \leqslant \omega l ds_{\mathrm{pre}} + \omega ds_{\mathrm{pre}} + ds_{\mathrm{pre}},$$

所以扩展能力与先前平均相异度和周围元组的数量正相关. □

定义 6(最优不确定近邻相异度)　最优不确定近邻相异度是指, 以不确定元组 (X_i, p_i) 为核心与其最优不确定近邻集合 $\mathrm{Opt}((X_i, p_i))$ 中元组的不确定相异度的平均值, 其公式如下:

$$\mathrm{Opt\_}ds((X_i, p_i)) = \frac{\sum\limits_{j \in \mathrm{Opt}((X_i, p_i))} ds_{ij}}{|\mathrm{Opt}((X_i, p_i))|}.$$

最优不确定近邻相异度将作为判断元组是否为互包含的重要指标, 结合定义 7 解决近邻方式可能将噪声点作为近邻的缺点.

定义 7(互包含)　互包含是指任意两个元组的最优不确定近邻相异度邻域内互相包含对方. 若 (X_i, p_i) 和 (X_j, p_j) 满足互包含关系, 则

$$(X_i, p_i) \in \{(X_k, p_k) | ds_{jk} \leqslant \mathrm{Opt\_}ds(X_j, p_j)\} \bigwedge$$

$$(X_j, p_j) \in \{(X_k, p_k) | ds_{ik} \leqslant \mathrm{Opt\_}ds(X_i, p_i)\},$$

即 $ds_{ij} \leqslant \min(\mathrm{Opt\_}ds(X_i, p_i), \mathrm{Opt\_}ds(X_j, p_j))$, 互包含可以判断元组间的真实差异. 考虑 X_1 邻域包含 X_2, X_3, X_7; X_2 邻域包含 X_4, X_5, X_6; X_3邻域包含 X_7, X_8, X_9 的情况, 可以发现 X_1 邻域包含 X_2 和 X_3, 但是 X_2 和 X_3 邻域未包含 X_1, 这说明 X_1 与 X_2 和 X_3 存在虚假的近邻关系, 通过互包含处理后, 可以剔除这类关系, 从而提高聚类质量.

为简单快速地实现不确定数据的聚类, 本文算法结合文献 [12] 中对聚类中心的假设, 认为聚类中心是由一些局部密度较低的点围绕, 且这些点距离其他较高局部密度点的距离较远, 该假设对确定性数据聚类的有效性已在原文献中证明, 故不再详细赘述. 本文的 OLUC 算法, 借鉴这种假设, 将其运用在不确定数据的聚类问题上, 重新定义了用于聚类中心决策的参数、局部密度和最小聚类相异度.

定义 8(局部密度)　给定不确定元组 (X_i, p_i) 的最优不确定近邻集合 $\mathrm{Opt}(X_i, p_i)$ 和最大不确定近邻相异度 $\max(ds)$, 则该不确定元组的局部密度为

$$\rho_i = \sum_{j \in \mathrm{Opt}(X_i, p_i)} \chi(ds_{ij} - \max(ds)).$$

其中

$$\chi(x) = \begin{cases} p_j \mathrm{e}^{-\frac{ds_{ij}^2}{\max(ds)^2}}, & x < 0; \\ 0, & \text{otherwise}; \end{cases}$$

$\max(ds)$ 为任意两个不确定元组之间的最大不确定相异度.

定义 9(最小聚类相异度)　不确定元组与相对局部密度更大的元组之间的不确定相异度的最小值 δ_i 称为最小聚类相异度, $\delta_i = \min\limits_{j:\rho_j > \rho_i} (ds_{ij})$; 对于局部密度最大的元组, 其 $\delta_i = \max(ds_{ij})$, 即该元组到其他元组的最大不确定相异度.

根据重新定义的参数得到推论: 只有那些局部密度是局部或者全局最大的元组才会有远大于正常的最小聚类相异度, 而往往这些局部密度是局部或者全局最大的元组是聚类中心.

3　OLUC 算法

本文提出的 OLUC 算法主要分为三大步骤:

1) 构造最优不确定近邻集合并计算局部密度和最小聚类相异度;

2) 构造聚类中心决策图, 实现聚类中心的发现;

3) 根据聚类中心进行聚类并对噪声点进行处理.

详细过程如下:

Step 1: 计算不确定相异度矩阵 ds, 同时得到最大不确定相异度 $\max(ds)$;

Step 2: 构造每个不确定元组的不确定近邻集合 $N_k(X, p)$;

Step 3: 优化所有不确定近邻集合, 得到最优不确定近邻集合 $\mathrm{Opt}(X, p)$, 并计算最优不确定近邻相异度 $\mathrm{Opt\_}ds((X, p))$;

Step 4: 根据最优不确定近邻集合计算局部密度 ρ;

Step 5: 根据局部密度和不确定相异度矩阵计算最小聚类相异度 δ, 并记录相应的最近邻元组;

Step 6: 根据局部密度和最小聚类相异度构造聚类决策图;

Step 7: 根据聚类决策图获取聚类中心;

Step 8: 根据聚类中心和最近邻元组进行聚类;

Step 9: 处理噪声点.

3.1　不确定近邻集合最优化算法

该算法用于最大化有效近邻数, 在一定精度条件下保留周围有价值的不确定近邻元组, 剔除无效的不确定近邻元组, 减少其对决策参数的影响.

输入: ds_i, k, ω;

输出: $\mathrm{Opt\_}ds((X_i, p_i)), \mathrm{Opt\_}p((X_i, p_i))$.

Step 1: 令 ds_i 按升序排序;

Step 2: 计算 $ds_{\mathrm{now}} = \sum\limits_{j=1}^{k} ds_{ij} \Big/ k$;

Step 3: 若 $j \leqslant 1$, 则输出

$$\mathrm{Opt\_}ds((X_i, p_i)) = ds_{\mathrm{now}} \mathrm{Opt\_}p((X_i, p_i)) = j,$$

并结束, 否则转 Step 4;

Step 4: 计算 $ds_{\rm pre}=(d_{\rm now}\times j-ds_{ij})/(j-1)$;

Step 5: 若 $|(ds_{\rm now}-ds_{\rm pre})/ds_{\rm pre}|\leqslant\omega$, 则输出

$$\mathrm{Opt}\_ds((X_i,p_i))=ds_{\rm now},\mathrm{Opt}\_p((X_i,p_i))=j,$$

并结束, 否则令 $d_{\rm now}=d_{\rm pre}$, $j=j-1$, 转 Step 3.

3.2 聚类中心决策算法

该算法是将计算出的局部密度和最小聚类相异度转化为二维决策图后选取聚类中心. 例如, 图 1 是将仿真数据集转化后的聚类中心决策图, 纵坐标是最小聚类相异度, 横坐标是局部密度. 由图 1 可以发现, 右上部分存在 5 个映射点, 明显区别于其他映射点的决策坐标, 将其标记为不同形式, 表示不同簇的聚类中心.

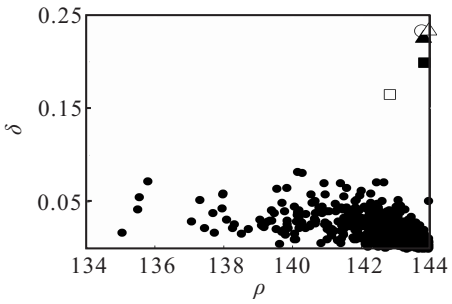

图 1 聚类中心决策图

本文采用双阈值的方法从决策图中划分出聚类中心, 即满足 $\rho>\rho_{\rm cent}\bigwedge\delta>\delta_{\rm cent}$. 根据重新定义参数后的推论, 位于决策图右上部分, 即局部密度和最小聚类相异度同时较大的点往往是聚类中心. 通过用户点击包含所有迥异点的矩形区域的左下角, 将获得的坐标转化成双阈值作为中间输入判别聚类中心. 双阈值非固定数值, 对于同一数据集只要是包含所有聚类中心的坐标即可, 对于不同数据集更是不同. 图 2 是根据图 1 决策出的聚类中心, 未经过噪声处理所得出的聚类结果.

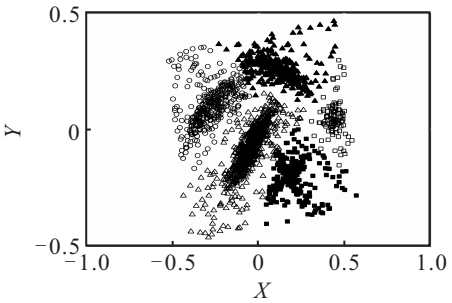

图 2 未除噪聚类结果

3.3 聚类除噪算法

经过聚类中心决策算法处理后, 已经得到所有的聚类中心, 最容易想到的处理方式是 k-means[7]算法, 将剩余元组归类到与其最近的聚类中心所在的簇. 事实上, 本文算法采用的聚类方式有所不同, 因为经过决策算法处理获得的聚类中心是结合局部密度和最小聚类相异度的, 与 k-means[7]意义上的聚类中心不完全等价. 本文算法采用计算最小聚类相异度时记录的最近邻元组标记, 对剩余的元组添加簇标记. 其中 $cl(x)$ 表示元组 x 所在簇的编号.

输入: ρ,nn,n;

输出: cl.

Step 1: 令 (X,p) 按 ρ 的降序映射为 $d\rho$, $i=1$;

Step 2: 若 $i>n$, 则结束, 否则转 Step 3;

Step 3: 若 $cl(d\rho_i)$ 无簇标记, 则令

$$cl(d\rho_i)=cl(nn(d\rho_i)),\ i=i+1,$$

转 Step 2, 否则令 $i=i+1$, 转 Step 2.

经过聚类算法处理后, 所有元组都添加了簇标记, 需要进行噪声处理. 虽然, 对于每个元组都有局部密度和最小聚类相异度两个参数, 可以采用密度除噪或相异度除噪, 但是鉴于 k 近邻对密度等级的不敏感性, 本文的除噪算法采用不确定相异度作为参考指标, 同时利用最优不确定近邻近相异度判断互包含, 自适应计算除噪阈值. 详细算法如下:

输入: $ds,\mathrm{Opt}\_ds((X,p)),ncl,n,ds_{\max}$;

输出: cl.

Step 1: 令所有簇的 $\mathrm{bord}\_ds=ds_{\max}$, $i=1$, $k=1$;

Step 2: 若 $i>n-1$, 则转 Step 7, 否则, 令 $j=i+1$, 转 Step 3;

Step 3: 若 $j>n$, 则令 $i=i+1$, 转 Step 2, 否则, 转 Step 4;

Step 4: 若 $cl(i)\neq cl(j)$ 且 $ds_{ij}\leqslant\min(ods_i,ods_j)$, 则令 $\mathrm{aver}\_ds=(ods_i+ods_j)/2$, 转 Step 5, 否则令 $j=j+1$, 转 Step 3;

Step 5: 若 $\mathrm{aver}\_ds<\mathrm{bord}\_ds_{cl(i)}$, 则令 $\mathrm{bord}\_ds_{cl(i)}=\mathrm{aver}\_ds$, 转 Step 6, 否则转 Step 6;

Step 6: 若 $\mathrm{aver}\_ds<\mathrm{bord}\_ds_{cl(j)}$, 则令 $\mathrm{bord}\_ds_{cl(j)}=\mathrm{aver}\_ds$, $j=j+1$, 转 Step 3, 否则令 $j=j+1$, 转 Step 3;

Step 7: 若 $k>n$, 则结束, 否则转 Step 8;

Step 8: 若 $\mathrm{ods}_k>\mathrm{bord}\_ds_{cl(k)}$, 则令 $cl(k)=-1$, $k=k+1$, 转 Step 7, 否则令 $k=k+1$, 转 Step 7.

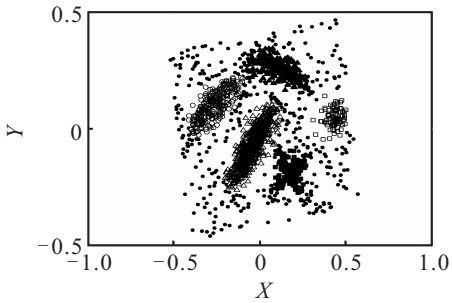

图 3 除噪后聚类结果

至此, 结束OLUC算法的所有处理. 图3是仿真数据集经过噪声处理后的聚类结果, 可以发现除噪处理后, 不同的簇之间有明显的分隔.

4　实验与分析

4.1　实验设置

OLUC算法实验均由Matlab R2012b工具实现, 硬件环境为Intel(R) Core(TM)2 Dou Cpu P7450 2.13 GHz, 操作系统为Windows 7家庭普通版. 实验采用一个仿真数据集(包含2 000个二维数值属性数据), 一个形状数据集jain(包含373个二维数值属性数据), 两个uci真实数据集iris(包含150个4维数值属性数据)和seeds(包含210个7维数值属性数据), 然后再将其转化成相应的不确定数据集.

OLUC算法以文献[12]的cluster_dp算法作为对比参照, cluster_dp算法同样是一种基于密度快速发现聚类中心实现任意形状聚类的算法, 作为参照较为合理. 由于cluster_dp算法未考虑不确定因素, 为实现相同条件下的对比, 先将两个不确定数据集转化成点对相异度的形式输入, 达成合理可比性. 聚类结果采用对所有簇的簇内各点之间相异度的平均值求平均SA作为质量评价指标, 该指标主要体现簇内紧凑程度, 该求和值越小表示聚类质量越好.

4.2　参数选择

OLUC算法主要涉及到的参数有近邻数 k 和近邻优化参数 ω. 为避免参数选择的盲目性, 本文借鉴文献[11-12]中的参数选择思想, k 取总元组数量的7%～8%较佳, ω 取0.4%～0.5%时优化效果较好, 同时容易从聚类中心决策图中判断最终聚类数. 表1是仿真数据集实验中参数 k 和 ω 与指标SA的变化关系. 从表1中可以发现, 在区间范围内SA指标值变化平缓, 对于两个参数具有一定鲁棒性, 相比传统基于密度的算法DBSCAN[9]拥有更低参数敏感性.

表 1　参数 k, ω 与 SA 值关系变化

k/n	0.06	0.07	0.08	0.09	0.10
SA	0.104	0.096	0.094	0.098	0.097
ω	0.001	0.002	0.003	0.004	0.005
SA	0.142	0.125	0.118	0.106	0.096

从表1可以看出, k 值在6%～8%之间变化时SA值呈下降趋势, 而当 k 值达到9%～10%时, SA值回升. 其原因是: 当 k 值在正常范围浮动时, 非互近邻数据可以被正确地剔除而不影响正确的近邻关系; 当 k 值过大时, 较多无关数据被添加进近邻关系, 导致无法优化完全而影响正确的聚类; 当 k 值极小时, 使得近邻关系局部收敛, 产生较多较小的簇, 使得SA下降, 却非真实地提高了聚类质量. 从表1中还可以看出, ω 值在0.1%～5%之间变化时SA值呈下降趋势. 其原因是: 当 ω 值过小时, 导致近邻关系局部收敛, 最终在聚类数不变情况下无法正确聚类而使得SA值较大; 当 ω 值过大时, 等价于未优化不确定近邻集合, 使得SA值将趋于固定值不变.

4.3　算法性能分析

OLUC算法是在最优不确定近邻集合的基础上计算各个决策参数, 然后进行判断聚类的过程. 每次判断近邻对象, 需要消耗冗长的时间, 为此预先计算所有数据的不确定相异度矩阵进行存储, 可以大大减少进一步计算所需的时间. 算法的复杂度也从 $O(n^2)$ 骤降为查询近邻对象 $O(n\log n)$ 和查询近邻相异度 $O(n)$ 的花费和, 因此OLUC算法的时间复杂度为 $O(n\log n)$, 与传统基于密度的聚类算法DBSCAN[9]的时间复杂度同阶.

图4是OLUC和cluster_dp对不同数据集的实际处理时间对比, 横轴表示数据集, 纵轴表示处理速率. 从图4中可以看出, OLUC算法相比cluster_dp算法在小数据集上运行速率更快.

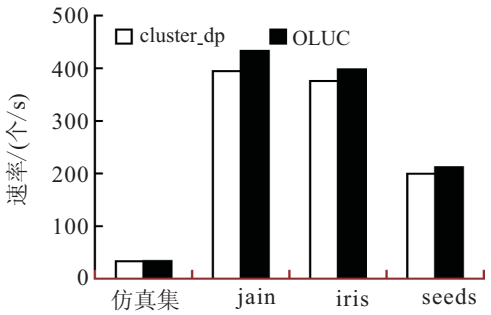

图 4　运行速率对比

4.4　实验结果

表2展示了OLUC和cluster_dp在仿真和真实数据集上的SA值表现. 从表2中可以看出, OLUC算法的SA值相较更小, 簇内的数据更为紧凑.

表 2　聚类 SA 值对比

参数(SA)	cluster_dp	OLUC
仿真集	0.11	0.10
jain	9.81	9.70
iris	0.84	0.83
seeds	2.16	2.14

图5分别比较了OLUC和cluster_dp在仿真数据

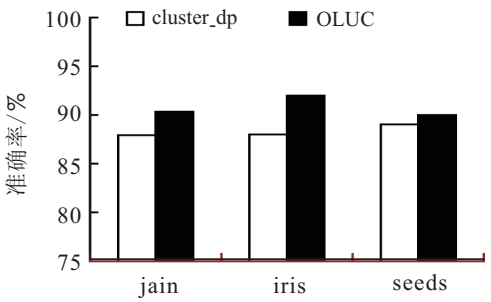

图 5　聚类准确率对比

集和真实数据集 seeds 上的聚类结果正确率, 横轴是数据集, 纵轴是准确率. 从图 5 中可以看出, OLUC 算法相比 cluster_dp 算法在聚类准确率方面更高.

5 结 论

本文提出了不确定数据的最优 k 近邻和局部密度聚类 OLUC 算法, 该算法针对点概率模型的数据进行聚类, 不仅考虑元组间的距离和不确定概率的影响, 还考虑了不确定概率之间的影响. 采用自适应优化的最优不确定近邻结构, 克服了传统基于密度聚类算法中对于邻域半径难以确定的问题. OLUC 算法推广了文献 [12] 的思想, 认为聚类中心是由一些局部密度较低的元组围绕, 且这些元组与其他较高密度元组的最小聚类相异度较大, 从而实现不确定数据的快速聚类, 并采用互包含关系优化除噪, 实验结果表明了 OULC 算法的有效性. 由于无法预知数据集的最终聚类数, 算法中需要用户参与聚类中心数的决策, 以半监督的方式得到最终聚类结果, 因此进一步的工作是研究如何以自适应的方式从聚类中心决策图中判断最终聚类数.

参考文献(References)

[1] 倪巍伟, 陈耿, 吴英杰, 等. 一种基于局部密度的分布式聚类挖掘算法[J]. 软件学报, 2008, 19(9): 2339-2348.
(Ni W W, Chen G, Wu Y J, et al. Local density based distribute clustering algorithm[J]. J of Software, 2008, 19(9): 2339-2348.)

[2] 倪巍伟, 孙志挥, 陆介平. k-LDCHD —— 高维空间 k 邻域局部密度聚类算法[J]. 计算机研究与发展, 2005, 42(5): 784-791.
(Ni W W, Sun Z H, Lu J P. k-LDCHD — A local density based k-neighborhood clustering algorithm for high dimensional space[J]. J of Computer Research and Development, 2005, 42(5): 784-791.)

[3] 黄德才, 李晓畅. 基于相对密度的混合属性数据增量聚类算法[J]. 控制与决策, 2013, 28(6): 815-822.
(Huang D C, Li X C. Incremental relative density-based clustering algorithm for mixture data sets[J]. Control and Decision, 2013, 28(6): 815-822.)

[4] 黄德才, 吴天虹. 基于密度的混合属性数据流聚类算法[J]. 控制与决策, 2010, 2010, 25(3): 416-421.
(Huang D C, Wu T H. Density-based clustering algorithm for mixture data sets[J]. Control and Decision, 2010, 25(3): 416-421.)

[5] Kriegel H P, Pfeifle M. Density-based clustering of uncertain data[C]. Proc of the 11th ACM SIGKDD Int Conf on Knowledge Discovery in Data Mining. Chicago, 2005: 672-677.

[6] Ngai W K, Kao B, Chui C K, et al. Efficient clustering of uncertain data[C]. The 6th Int Conf on Data Mining. Hong Kong, 2006: 436-445.

[7] MacQueen J. Some methods for classification and analysis of multivariate observations[J]. Proc of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, 1(14): 281-297.

[8] 张晨, 金澈清, 周傲英. 一种不确定数据流聚类算法[J]. 软件学报, 2010, 21(9): 2173-2182.
(Zhang C, Jin C Q, Zhou A Y. An uncertain data stream clustering algorithm[J]. J of Software, 2010, 21(9): 2173-2182.)

[9] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]. Kdd. Portland, 1996, 96: 226-231.

[10] Cover T, Hart P. Nearest neighbor pattern classification[J]. IEEE Trans on Information Theory, 1967, 13(1): 21-27.

[11] 沈仙桥, 黄德才, 陆亿红. 三层流数据聚类框架与最优 $2k$ 近邻聚类算法[J]. 小型微型计算机系统, 2013, 34(11): 2451-2455.
(Shen X Q, Huang D C, Lu Y H. Three layers data clustering framework and optimal $2k$-nearest neighbor clustering algorithm[J]. J of Chinese Computer Systems, 2013, 34(11): 2451-2455.)

[12] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

(责任编辑：孙艺红)