第47卷第1期 中国科学技术大学学报 Vol. 47, No. 1
2017年1月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Jan. 2017

文章编号:0253-2778(2017)1-0070-10

面向 LBSN 的 *k*-medoids 聚类算法

罗维佳[1],乔少杰[2],韩 楠[3],元昌安[4],闭应洲[4],舒红平[5]

(1. 西南交通大学信息科学与技术学院,四川成都 610031;2. 成都信息工程大学信息安全工程学院,四川成都 610225;
3. 成都信息工程大学管理学院,四川成都 610103;4. 广西师范学院科学计算与智能信息处理广西高校重点实验室,广西南宁 530023;
5. 成都信息工程大学 软件工程学院,四川成都 610225)

摘要:常用的聚类算法存在诸多不足,为此提出了一种基于初始半径 r 的 k-medoids 改进算法,主要针对 LBSN 中的位置数据进行聚类,改善初始聚类中心敏感对聚类结果的影响,其本质是基于密度聚类,不同之处在于 k 值的选取是依赖于半径 r. 通过大量真实签到数据集进行实验,结果显示本文算法聚类结果更稳定. 本文算法在基于位置的社交网络应用中获得更好的聚类效果和更快的收敛速度. 实验中将距离平方和作为准则函数进行对比,相对于传统 k-medoids 算法优势明显,对退化的 k-medoids 算法也能够缩小 1.2%到 2%.

关键词:社交网络;密度聚类;k-medoids;签到数据;距离相似度

中图分类号:TP 391 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2017.01.010

引用格式:罗维佳,乔少杰,韩楠,等. 面向 LBSN 的 k-medoids 聚类算法[J]. 中国科学技术大学学报,2017,47(1):70-79.

LUO Weijia, QIAO Shaojie, HAN Nan, et al. A k-medoids based clustering algorithm in location based social networks[J]. Journal of University of Science and Technology of China, 2017, 47(1): 70-79.

A *k*-medoids based clustering algorithm in location based social networks

LUO Weijia[1], QIAO Shaojie[2], HAN Nan[3], YUAN Changan[4], BI Yingzhou[4], SHU Hongping[5]

(1. *School of Information Science and Technology, Southwest Jiaotong University, Chengdu* 610031, *China*;
2. *College of Information Security Engineering, Chengdu University of Information Technology, Chengdu* 610225, *China*;
3. *School of Management, Chendu University of Information Technology, Chendu* 610103, *China*;
4. *Science Computing and Intelligent Information Processing of GuangXi higher education Key Laboratory, Guangxi Teachers Education University, Nanning* 530023, *China*;
5. *School of Software Engineering, Chengdu University of Information Technology, Chengdu* 610225, *China*)

Abstract: The commonly-used clustering algorithms have several drawbacks. Aiming to solve the above problems, an improved k-medoids algorithm was proposed based on the initial radius r, which is used for clustering using location data. The algorithm is actually a density-based clustering approach. The difference is that the k value depends on the radius r. Extensive experiments are conducted on real check-in data, and the results show that the improved k-mediods algorithm on the radius r is more stable. In

收稿日期:2016-03-01;**修回日期**:2016-09-17

基金项目:国家自然科学基金(61100045, 61165013, 61363037),教育部人文社会科学研究规划基金(15YJAZH058),教育部人文社会科学研究青年基金(14YJCZH046),四川省教育厅资助科研项目(14ZB0458),成都市软科学项目(2015-RK00-00059-ZF),科学计算与智能信息处理广西高校重点实验室开放课题(GXSCIIP201407)资助.

作者简介:罗维佳,女,1988年生,硕士生. 研究方向:数据挖掘. E-mail: weijialuo1026@gmail.com

通讯作者:乔少杰,博士/教授. E-mail: sjqiao@home.swjtu.edu.cn

addition, by comparing the sum of the square of distance between objects in the same cluster among different algorithms, the proposed algorithm can obtain better clustering results and convergence speed when applied to location based social networks. Compared to the traditional k-medoids algorithm, the cost has obviously reduced, as for and the degraded k-medoids algorithm, the cost can be reduced among 1.2% and 2%.

Key words: social networks; density-based clustering; k-medoids; check-in data; distance similarity

0　引言

随着手机等便携式移动设备的普及，几乎每个用户都能便捷地获取自己的位置信息，人们在日常生活中获得的基于位置信息的服务也越来越多，基于位置的社交网络 LBSN（location-based social network）日渐普及. 当前主流的 LBSN 研究网站包括 Foursquare、Gowalla、Geolife、Brightkite 等，用户在线分享周边有趣的、好玩的、好吃的信息已是基于位置的社交网络最基本的功能[1]，基于 LBSN 的个性化推荐技术带来了可观的商业利益，受到了学术界的广泛重视.

在 LBSN 门户网站中，用户通常将自己的位置信息通过手机 App 实时上传到社交网站，这些信息通常包含较高精度的经纬度. 通过对这些经纬度数据进行聚类，可以预测用户即将行进的轨迹路线. 传统的研究模型通常采用 k-medoids 聚类方法，将数据划分为 k 个簇集，通过计算对象之间的相似度进行聚类；然而 k 值的选择会受用户签到范围的影响，进而影响聚类精度，并且聚类效果还会受到随机选择的初始化对象的影响[2]. 文献[3]通过计算对象间的距离关系来选择新的中心点，虽然提高了算法效率，但未能很好地提高聚类精度. 文献[4]提出了一种基于密度的优化 k-means 算法 PKM（optimal k-means），算法首先通过计算密度信息产生高密度集，然后在高密度集中产生初始化聚类中心. 文献[5]提出了一种高效的 k-medoids 算法，采用了增量候选集策略来替换中心. 文献[6]通过半径 r 来确定高密度集，克服了 k-medoids 算法对初值的敏感，但该算法以时间换取聚类效果. 文献[7]提出了一种优化的 k-medoids 算法，在初始化中心确定后，采用局部迭代方法对算法进行优化，提高了算法的时间效率. 在 LBSN 网络中，上述算法都不能很好地使聚类结果集中于半径 r 之下. 针对传统 k-medoids 算法以及改进 k-medoids 算法在社交网络应用中的不足，本文提出了一种基于半径 r 的改进算法，与以往的聚类算法不同，本文算法主要针对 LBSN 网络中的位置数据进行聚类，算法的创新性主要包括两点. ①算法在构建初始聚类中心的策略上与以往算法有所不同；②算法只需要设定可接受聚类半径 r，使得聚类效果不依赖于用户签到数据的时空范围大小，算法本身不需要对 k 值进行预估，k 个初始聚类中心由算法计算产生，使最终的聚类结果更加准确.

1　聚类技术

1.1　LBSN 中相似性距离计算

传统的聚类算法通过计算对象之间的距离来衡量两个对象之间相似度，距离越大，相似度越低，距离越小，相似度越高. 在基于位置的社交网络中，对象相似度计算是推荐算法中必不可少的部分，常见的计算方法有：

（Ⅰ）Cosine-based similarity，该方法也称为基于余弦的相似度. 首先，将数据空间索引用 k 维度欧式空间中的随机点表达；然后，利用余弦距离计算两个对象的相似度. 这一方法常用于旅游推荐系统中，可以将对象之间的属性转化为向量，具有相似属性的行为归结为一个向量属性值，常应用于对文本、JSON，XML 等一系列格式数据进行相似度计算[8]. 公式如下：

$$\underset{u,v\in R}{\mathrm{sim}}(u,v)=\frac{\sum_{i=1}^{n}u_iv_i}{\sqrt[2]{\sum_{j=1}^{n}u_j^2}\sqrt[2]{\sum_{k=1}^{n}v_k^2}} \tag{1}$$

（Ⅱ）Euclidean distance-based similarity，该方法又叫做欧几里德距离相似度，可以计算多维空间集合中对象的相似度，是通过对属性的差异值来判定的，一般用于平面或者三维空间中. 该方法在计算两点间的相似度时，相似度与两点间距离成反比. 在 LBSN 中，可以利用该方法计算位置相似度，为相似度较高的用户提供相同的推荐地点，进而提供用户可能需要的服务[9]. 具体公式如下：

$$\underset{u,v\in R}{\mathrm{sim}}(u,v)=\frac{1}{1+D(u,v)} \tag{2}$$

$$\underset{u,v\in R}{D(u,v)} = \sqrt[2]{\sum_{i=1}^{n}(u_i - v_i)^2} \quad (3)$$

公式(1)、(2)、(3)中，R 代表所有对象组成的集合，u、v 为 R 中的对象，u_i 代表对象 u 的第 i 个属性，整个对象的属性是 n 维向量，n 是常量.

1.2 相关聚类算法

k-means 算法又称为 k 均值算法，是最常用的聚类算法，其基本思想是计算两个对象的欧式距离来评价其相似度，距离越远，相似度越小，距离越近，相似度越高[10]. 算法将相似度大的集合确认为一个簇，最终目标是要得到相互独立的簇群. 算法的主要步骤如下：

(Ⅰ)随机选取 k 个点作为起始点，随机对象的选取对聚类的效果会有很大的影响；

(Ⅱ)通过计算，把剩余的点按距离远近归结到距离最近的簇集中；

(Ⅲ) 重新计算簇集的中心，依次循环，直到所有簇集的新质心与原始质心的距离小于预设的阈值.

k-means 算法简单、效率高，能够对较大的数据集进行高效聚类，并且随着数据集的伸缩具有很强的可扩展性. 算法的时间复杂度是 $O(n*k*t)$，n 表示数据规模，k 代表聚类的个数，t 代表迭代的次数. 由于 k-means 算法的 k 个初始聚类中心是随机产生的，聚类效果并不好，因而出现了一些改进算法，本质思想还是对初始聚类中心进行高效选择[11]. k-means 算法使用质心作为推荐簇集的中心，在 LBSN 的推荐算法中，该质心值可能是不可达点，因此在实际研究 LBSN 时一般使用 k-medoids 算法对数据进行聚类.

k-medoids 算法又称为 k 中心算法，与 k-means 算法类似，区别是 k-medoids 算法在当前簇集中选取一个到其他所有点距离之和最小的点将其作为新的簇集中心[3]，而 k-means 算法在迭代时是以簇集质心作为中心. k-medoids 算法基本思路如下[12]：

(Ⅰ)选取任意 k 个对象作为初始聚类中心；

(Ⅱ)将剩余的点以相似度的高低分配到各聚类中；

(Ⅲ)对每一个簇进行中心点选择，计算出每一个簇中心点到该簇集中所有其他点距离和最小的点，将其作为新的中心，重新进行迭代，直到所有簇集中心不再变化为止.

该算法的时间复杂度近似为 $O(k*n^2)$，k 为簇集的个数，n 表示集合中对象个数.

由于传统 k-medoids 算法对初值 k 的选取敏感，不同初值会造成聚类结果的不稳定，因此往往是通过多次计算来得到较好的聚类结果. 孙秀娟等提出了一种优化的 k-means 算法(PKM)，该算法基于密度的思想来选取初始的 k 个聚类中心，定义密度函数为

$$\mathrm{density}(x) = \{p \in C \mid \mathrm{dist}(x,\ p) \leqslant r\} \quad (4)$$

式中，dist 表示距离函数，在基于位置的社交网络聚类中，通常定义为欧几里德距离. r 为半径，C 表示样本集. $r=u\times\theta$，θ 为用户给定的常数，u 代表 C 样本集中的均值距离，且

$$u = \frac{\sum \mathrm{dist}(x_i,x_j)}{n\times n},\ i,\ j = 1,\ 2,\ \cdots,\ n \quad (5)$$

样本的平均密度函数为

$$\mathrm{AvgDensity}(x_i) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{density}(x_i) \quad (6)$$

将密度大于平均密度的样本放入集合 S 中，具体计算 k 个初始聚类中心的算法如下：

算法 1.1 PKM 计算初始聚类中心 k 的算法

输入：高密度集合 S，聚类中心个数 k

输出：k 个初始的聚类中心

步骤 1：在高密度集合 S 中首先将密度最高的点作为第 1 个聚类中心 Z_1，然后将该点从 S 中去除；

步骤 2：在 S 中选取距离 Z_1 最远的点作为第二个聚类中心 Z_2，将该点从 S 中去除，取得满足条件 max(min(d(x_i, Z_1), d(x_i, Z_2)))的点作为第 3 个聚类中心 Z_3，i=1, 2, …, n，其中 $x_i\in$S，n 表示集合当前的对象个数，并将 Z_3 从 S 中去除；

步骤 3：依次类推，取得满足条件 max(min(d(x_i, Z_1), d(x_i, Z_2), …, d(x_i, Z_{q-1})))的对象作为 Z_q，其中 $1\leqslant q\leqslant k$，i=1, 2, …, n，$x_i\in$S，n 为集合当前的对象个数，并将 Z_q 从 S 中删除；

步骤 4：重复步骤 3 直到得到 k 个初始聚类中心.

PKM 算法后面的聚类步骤与 k-means 算法相同，直到最终得到 k 个聚类中心使得每一个簇集的中心不再发生变化. 该算法在产生 k 个初始聚类中心时，选取了距离较远且密度较高的点集作为聚类算法的初值中心，很大程度上解决了随机 k 值带来的分布不均问题，使算法具有较好的稳定性[4]. 聚类的最终结果是使得代价 E 最小，E 为准则函数，可表述为距离平方和. 公式表示如下：

$$E = \sum_{i=1}^{k}\sum_{j\in S_i}D(d_i,d_j) \quad (7)$$

式中，D 表示距离函数，d_i 代表簇集 S_i 的中心点，在

LBSN 网络中 D 代表欧几里德距离. E 在一定程度上反映了聚类效果，E 值越小，聚类效果越好.

2　面向 LBSN 的 k-medoids 聚类算法

2.1　问题描述

传统的基于位置的推荐算法是借助聚类思想对距离相近的点进行聚合，只要用户签到地点足够接近，聚类算法就将这些点视作是具有相同意义的地点，并对用户提供统一的推荐，如图 1 所示.

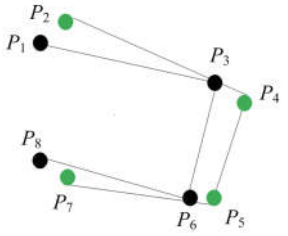

图 1　用户签到路径示例

Fig. 1　Example of the check-in path

用户 u 的签到数据为 $P_u=\{p_1, p_3, p_6, p_8\}$，用户 v 的签到数据为 $P_v=\{p_2, p_4, p_5, p_7\}$. 从图 1 可以看出，用户 u 与用户 v 在签到地点以及行为路径上是具有很高相似性的. 如果不对位置数据进行聚类，就很难分析出用户 u 与用户 v 的这种相似性. 此时我们可以通过聚类算法对签到数据集 $P=\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$ 进行聚类，如图 2 所示.

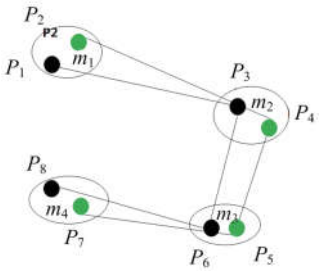

图 2　用户签到路径 k-medoids 聚类结果图

Fig. 2　Results of clustering the check-in path by k-medoids

当 k 等于 4 时，能够聚类得到 $\{p_1, p_2\}=m_1$，$\{p_3, p_4\}=m_2$，$\{p_5, p_6\}=m_3$，$\{p_7, p_8\}=m_4$，$M=\{m_1, m_2, m_3, m_4\}$，对于用户 u 和用户 v 来说 $P_u=P_v=\{m_1, m_2, m_3, m_4\}$；然后可以通过用户相似度计算方法如式(1)，求得用户 u 和用户 v 的相似度，进而实现对相似度较高用户提供更精准的推荐服务.

2.2　k-medoids 算法分析

由于 k-means 和 k-medoids 在处理数据时都是随机选择 k 个初始点作为算法开始的中心，所以最终的聚类效果可能并不理想. 如图 3 中所示.

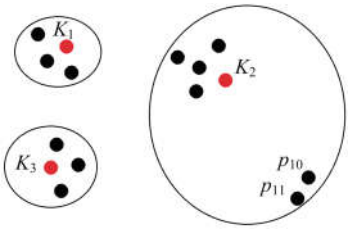

图 3　k-medoids 算法聚类图

Fig. 3　Results of clustering by k-medoids

假设 k-medoids 算法的 k 值等于 3，在随机产生初始点时生成的簇集为 K_1, K_2, K_3，在进行迭代运算后，最终的聚类效果并不理想. 原因是 p_{10}, p_{11} 这两个点位于簇集 K_2 中，它们与 K_2 的中心点仍有较大的距离. 由此可见，较小的 k 值产生的聚类效果会使簇集中对象间的距离大，相似度低，但单纯增大 k 值也并不能解决这一问题. 如图 4 所示.

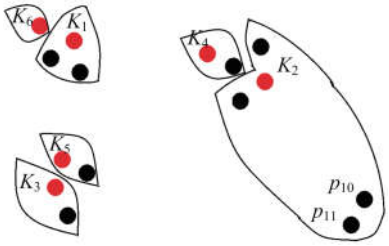

图 4　k-medoids 算法聚类

Fig. 4　Results of clustering by k-medoids

当 k 取 6 时，由于随机产生的 $K_1, K_2, K_3, K_4, K_5, K_6$ 没有覆盖到 p_{10}, p_{11}，所以对于聚类结果来说，K_2 所在的簇集中仍然存在距离较大的点，问题没有得到解决. 算法 1.1 思想产生的 k 个初始聚类中心虽然能够有效的覆盖 p_{10}, p_{11}，但是 k 值的增加导致 K_1 与 K_6，K_4 与 K_2，K_3 与 K_5 分属两个簇集，使得在可接受半径 r 下，相似性很高的两个点被分到了两个簇集，产生了类似图 1 问题. 现实世界中，存在可以接受的半径 r，使得将 r 内的地点看作是一个点对用户进行推荐，首先对一些现有算法进行分析. k-medoids 和 k-means 算法一样，都是用随机产生的 k 值来对数据进行聚类，一般可以采用多次聚类，选择 E 值较小的簇集作为最终的聚类结果，来消除敏感初始点.

2.3　算法描述

本节在研究基于位置的社交网络中，提出一种基于半径 r 的 k-medoids 改进算法，通过可接受范围 r 来计算 k 值大小，并选取出较好的 k 个点作为

k-medoids 算法的初始聚类中心，提升聚类效果.

r 的确定方法为：给定 LBSN 网络 C，应满足对任意 x_i，$x_j \in C$，$r_{\min} > \min(\mathrm{dist}(x_i, x_j))$，$r_{\max} < \max(\mathrm{dist}(x_i, x_j))$. 当 $r \leqslant r_{\min}$ 时，整个系统每一个对象都是一个簇集的中心，簇集个数 k 等于对象个数 n. 当 $r \geqslant r_{\max}$ 时，整个系统只有一个簇集.

算法 2.1 面向 LBSN 的 k-medoids 聚类算法

输入：原始签到数据集 S，半径 r，迭代最大次数 m

输出：簇集个数 k，收敛速度 v，簇集距离平方和 E

步骤 1：设定可接受半径 r，迭代最大次数 m，签到数据集合 S；

步骤 2：依次选取集合中的所有节点，统计以该节点作为中心，r 为半径所包含的节点个数作为该节点的密度值，选取密度最大节点以及该节点包含点作为集合 S_i；

步骤 3：将属于集合 S_i 的点从集合 S 中去除，重复第 2 步，直到最终划分出 k 个集合，满足对任意 i，$j \in K$，$S_i \cap S_j = \Phi$，$S = \bigcup_{i \in K} S_i$，K 是高密度中心所组成的集合，k 依赖 r 以及集合 S 产生，代表簇集个数，；

步骤 4：依次选择 K 中的点作为该集合聚类中心，并初始化空心簇集 O_i，以此类推得到 k 个初始聚类中心 medoids $(O_1, O_2, \cdots, O_k)$；

步骤 5：依次将其余的点以距离大小归结到每个 O_i 中；

步骤 6：对每个簇集，依次选取其中的点来计算每个簇集的代价 E_i，用 E_i 最小的点替换原来 O_i 中的中心；

步骤 7：重复步骤 5 和 6 直到算法稳定，使总代价 E 最小.

2.4 算法分析

在基于 LBSN 的研究中，需要对位置点聚类，把可接受半径 r 内的点作为一个点，来研究用户的位置轨迹. 对于常见的聚类算法，k 值都只能是估算，k 值过大可能会造成很多具有高相似度的用户轨迹位置数据被忽略；k 值过小则会使推荐算法的精度过低，计算出大量无效的相似性用户，降低推荐效果.

本文提出的算法，初始 k 值并不是随机产生，每一个初始点都是一个簇集的近似中心点，并且分布较为松散，能够覆盖整个集合中对象数据，包括高密度点以及孤立点. 该算法在没有任何优化的前提下，时间复杂度为 $O(\alpha n^3/r^2 + (nk + n^2/k)m)$，$r$ 为聚类半径，α，k 依赖集合 S 和 r 产生，m 为迭代次数. 对于数据较大的情况，如果半径 r 较小，计算任务将非常繁重，可以先将 r 取较大值，然后对聚类出的簇集进行区域划分；再将 r 取较小值，对每一个区域进行精度更高的计算. 本文算法的最终聚类结果并不是百分之百按照半径 r 进行聚类，因为在后期处理数据时，本文算法会对中心点进行重新选定，即算法通过牺牲 r 的严格性来换取聚类效果的优越性，如图 5 所示.

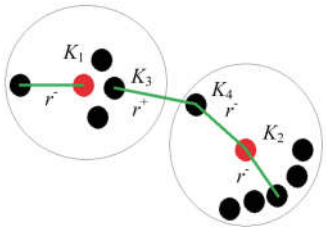

图 5 本文算法初始聚类中心 k 值确定

Fig. 5 Initializing the k value in the proposed algorithm

r^+ 代表半径大于 r 并且足够接近 r 的值，r^- 代表半径小于 r 并且足够接近 r 的值. 利用算法 2.1 可以得到，density(K_1) = 5，density(K_2) = 7，density(K_3) = 4. 本文算法将选取 K_1，K_2 作为初始聚类中心，聚类结果如图 6 所示.

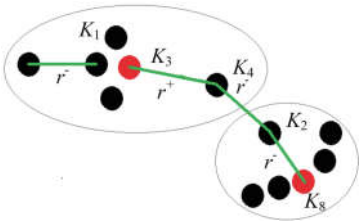

图 6 本文算法聚类结果

Fig. 6 Results of clustering by the proposed algorithm

K_3，K_8 成为最终的聚类中心. K_3 所在的簇集中，K_3 与 K_4 的距离是 r^+，超过了半径 r. 将 K_4 归结到 K_3 所在的簇集中，使得整个系统的距离平方和 E 最小.

2.5 *k*-medoids 改进算法

优化的 k-medoids 算法 PKMD (optimal k-medoids)首先使用 PKM 算法的密度思想，得到 k 个初始的聚类中心；然后利用 k-medoids 算法进行聚类. PKMD 算法解决了 k-medoids 算法的初值敏感问题.

退化的优化 k-medoids 算法 LPKMD(limiting of optimal k-medoids)将 PKMD 算法中的半径 r 设为无穷大，并将选取高密度集 S 的策略从

$$\mathrm{density}(x_i) > \mathrm{AvgDensity}(x) \tag{8}$$

改为

$$\mathrm{density}(x_i) \geqslant \mathrm{AvgDensity}(x) \tag{9}$$

式中，$x_i \in C$，C 为数据全集所组成的集合. 那么高密度集 S 就变成全集 C，依据 PKM 算法的密度思想，此时该算法就变为全局范围内找出最分散的点集作为初始聚类中心.

2.6 本文算法与 PKMD 算法比较

本文提出的算法是一种应用于基于 LBSN 的

签到数据进行聚类的方法，与 PKMD 算法的不同点在于：

（Ⅰ）本文算法与 PKMD 算法在初始聚类中心产生的方式上不同. PKMD 算法利用的是密度思想，在高密度集合中选取初值，并且利用最远距离选取下一个初始中心，而本文算法则是在整个集合中寻找密集点作为初始中心；

（Ⅱ）本文算法与 PKMD 算法的半径 r 虽然都代表聚类半径，但是其含义并不相同. 本文算法选取初值阶段是严格按照半径 r 进行分类，每一个初始聚类中心至少相距 r. 而 PKMD 算法只是确定高密度集 S 时严格按照 r 来选取. 由算法 1.1 可知，在产生初始聚类中心时，对 S 集合的处理并不是按照 r 来确定，因此不能保证 k 个初始聚类中心严格按照半径 r 划分. 该算法按照 S 中最远距离来选取初始聚类中心，只能保证初始聚类中心在高密度集中较为分散，并不能保证在全局范围内分散；

（Ⅲ）在基于 LBSN 的社交网络中，r 为可接受半径，利用 2.3 节的方法确定 r 后，由全集 S 和 r 生成 k，k 是一个被计算出的固定值. PKMD 算法的 k 为给定的值，可以任意设置，但是 k 的大小受到高密度集 S 的数量限制，必须满足 $0 \leqslant k \leqslant \mathrm{Num}(S)$；

（Ⅳ）本文算法与 PKMD 算法如果取相同的半径 r，在对社交网络签到数据进行聚类时，产生的结果完全不同，本文算法的聚类结果更接近于 LPKMD 算法. 原因是在 LPKMD 算法中，当半径 r 取无穷大时，高密度集合 S 即为全集 C，使得选取的初始聚类中心能够在全集范围内均匀的分布.

使用 PKMD 算法得到高密度集合 S 以及 k 个初始聚类中心，如图 7 所示，灰色代表聚类中心.

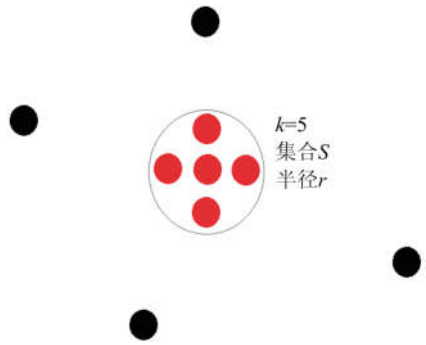

图 7　PKMD 算法初始聚类中心

Fig. 7　Initializing the k value in PKMD algorithm

半径 r 给定，$k=5$ 时，由此进行 k-medoids 算法聚类得到的聚类结果以及聚类中心如图 8 所示.

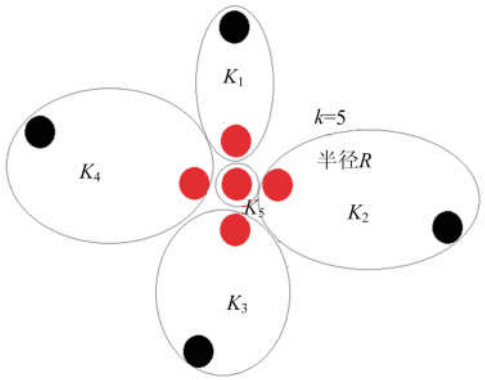

图 8　PKMD 算法聚类结果

Fig. 8　Results of clustering by PKMD algorithm

本文算法中，半径 r 与图 7、图 8 相同，$k=5$，从数据集 C 中选择初始聚类中心，最终的聚类结果如图 9 所示.

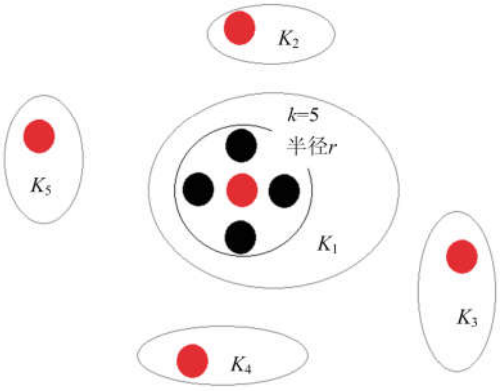

图 9　本文算法聚类结果

Fig. 9　Results of clustering by the proposed algorithm

传统 k-medoids 算法随机产生 k 个聚类中心，初值的选取是任意的. 按照概率统计的思想，在图 9 的 K_1 簇集内，初始聚类中心个数与概率关系如表 1 所示，因此传统的 k-medoids 算法极有可能出现如图 10 所示聚类结果.

表 1　簇集个数和概率关系图

Tab. 1　Relation of the cluster number and probability value

序号	计算公式	值
1	$C_{51} * C_{44} / C_{95}$	5/126
2	$C_{52} * C_{43} / C_{95}$	40/126
3	$C_{53} * C_{42} / C_{95}$	60/126
4	$C_{54} * C_{41} / C_{95}$	20/126
5	$C_{55} * C_{40} / C_{95}$	1/126

对于 LPKMD 算法，由于在全集 C 中选择距离最远的点作为初值中心，因此 LPKMD 算法聚类结果如图 11 所示.

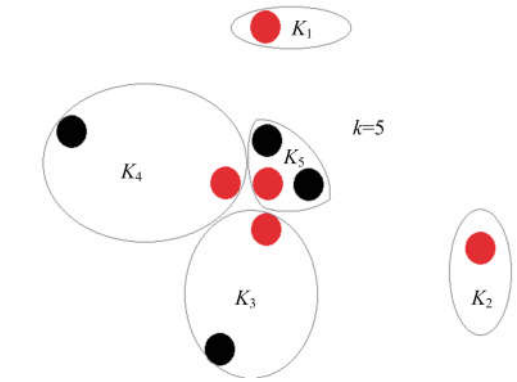

图 10 k-medoids 算法聚类结果

Fig. 10 Results of clustering by k-medoids

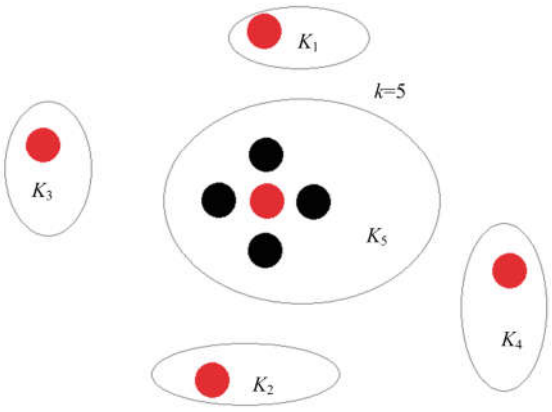

图 11 LPKMD 算法聚类结果

Fig. 11 Results of clustering by LPKMD

LPKMD 算法聚类效果与本文算法类似，都有较好的距离平方和 E. 比较这几种算法在集合 C 上的聚类结果，可以得出结论，对于距离平方和 E，E(本文算法)= E(LPKMD) $\leqslant$ E(k-medoids) $\leqslant$ E(PKMD). 我们可以发现，对于 PKMD 算法，适当地扩大半径 r 有助于提高聚类效果.

本文算法与 PKMD 在时间复杂度上不同，由于 PKMD 算法在高密度集 S 中产生初始聚类中心，而本文算法在全集 C 中产生聚类中心，因此 PKMD 算法在时间复杂度上优于本文算法. 由于 LPKMD 算法也是在全集 C 中构造初始聚类中心，因而在时间复杂度上与本文算法相近.

3 实验结果与分析

3.1 实验数据集简介

本文使用的数据集来源于 Gowalla 网站，该网站是一个用于研究基于位置的社交网络的平台. 本文使用的数据集含 196 591 个节点，950 327 条边，签到数据 Check-in 共 6 442 890 条. 该数据集包含全球范围的签到行为，本实验提取了经度在西经 76°到 78°，纬度在北纬 38°到 40°之间的城市 Washington 的签到数据作为样本数据，并对数据进行预处理，得到19 739条签到数据. 实验结果的评价指标使用了计算聚类后各个簇集的标准距离平方和 E 以及收敛速度 V 来衡量. E 越小，说明聚类效果越好. 实验包含五组测试结果，对比本文算法与 PKMD 算法、LPKMD 算法以及 k-medoids 算法. 由于k-medoids算法的结果会对初始聚类中心敏感，因此每一组实验中 k-medoids 算法采用 10 次统计.

3.2 算法性能对比

当 r 较小时，得到的聚类个数 k 近似等于签到个数 n；当 r 较大时，不符合可接受半径的意义，因此本文首先估算 r 的最小值. 对数据集进行范围选取处理后，结果包含 19 739 条签到数据，为了能够估算半径 r 的最小值，我们将签到范围所在的球面近似看作是一个矩形，假设签到点均匀的分散在其中，那么半径 r 的粗略计算公式如下：

$$r = \sqrt{\frac{l * w}{n * \pi}} \tag{10}$$

式中，r 为半径估计值，l 为矩阵长度，w 为矩阵的宽度，n 为签到个数. L 和 w 可以根据经纬度计算公式得到，最后得出 r 估计值为 r=791m. 考虑到签到数据具有一些高密度区域，因此本文实验选取 r=500m、r=2500m、r=5000m 作为实验半径. 由实验结果可以看出，当 r=500m 时，聚类个数 k 与全集个数 n 的比值已经很大.

(Ⅰ) 实验结果一

实验参数设置：n=1000，r=5000，k=157，d=10，结果如表 2 和图 12 所示. 其中，x 轴表示 10 组不同的聚类数据，y 轴表示距离平方和 E.

表 2 E 和 V 对比

Tab. 2 Comparison of E and V in experiment 1

算法	E_{max}	E_{min}	E_{avg}	V_{max}	V_{min}	V_{avg}
PKMD	9 455 078	9 455 078	9 455 078	9	9	9
k-medoids	2 741 398	1 969 688	2 214 873.7	8	5	6
LPKMD	1 644 900	1 644 900	1 644 900	5	5	5
本文算法	1 612 706	1 612 706	1 612 706	5	5	5

实验一中，本文算法与 LPKMD 算法的聚类结果均好于传统的 k-medoids 算法，优势明显. 本文算法的代价 E 好于 LPKMD 算法，优化约 2%；收敛速

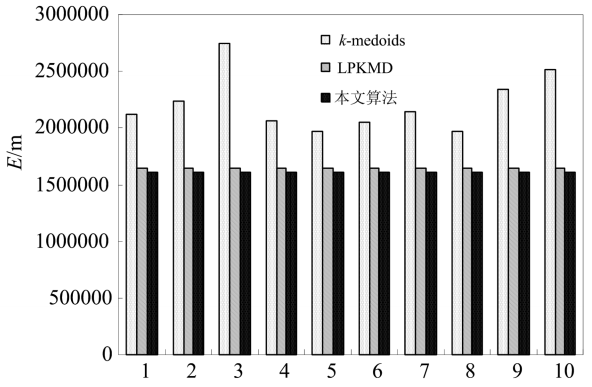

图 12　不同算法实验结果一对比图

Fig. 12　Comparison of different algorithms in the first set of experiments

度 V 与 LPKMD 算法相同. k-medoids 算法的代价 E_{max} 与 E_{min} 比值为 1.39，收敛速度比值为 1.6，可见初值对 k-medoids 算法的影响较大. 本文算法通过算法 2.1 选取高密度点作为初始聚类中心，减少了初始聚类中心不稳定对聚类结果的影响，具有较好的聚类代价和收敛速度，与 2.6 小节分析结果一致.

（Ⅱ）实验结果二

实验参数设置：$n=1000$，$r=500$，$k=632$，$f=10$，结果如表 3 和图 13 所示.

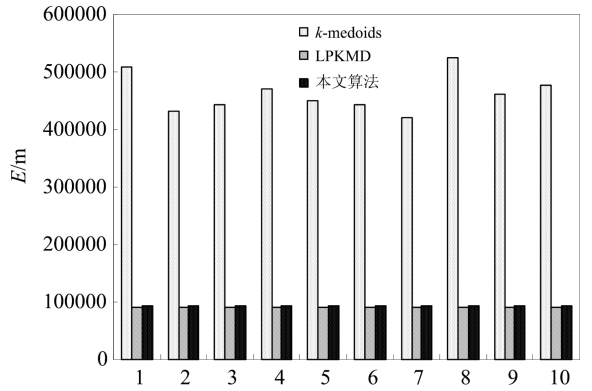

图 13　不同算法实验结果二对比图

Fig. 13　Comparison of different algorithms in the second set of experiments

表 3　E 和 V 对比

Tab. 3　Comparison of E and V in experiment 2

算法	E_{max}	E_{min}	E_{avg}	V_{max}	V_{min}	V_{avg}
PKMD	Null	Null	Null	Null	Null	Null
k-medoids	524 091	420 853	463 199.9	9	4	5.2
LPKMD	91 119	91 119	91 119	3	3	3
本文算法	92 936	92 936	92 936	3	3	3

由于 PKMD 算法选取了密度大于平均密度的点作为高密度集合 S，当 r 取 500 时，AvgDensity(x)=2.828，集合 S 的数量为 344，无法构造出 632 个聚类中心，因此此时给出结果 Null.

实验二中，本文算法在代价 E 上较传统 k-medoids算法缩小到其 20%左右. 当可接受半径 r 接近估计的最小半径时，簇集个数 k 与全集个数 n 的比重较大，聚类结果包含很多孤立簇集，整个系统的 E 值较实验一明显减少. 本文算法和 LPKMD 算法具有相同的收敛速度，代价 E 上也非常接近，整个聚类结果明显好于传统的 k-medoids 算法. 本实验中，k-medoids 算法的收敛速度 V_{max} 与 V_{min} 比值为 2.25，稳定性较实验一中有所下降，因为较小的半径 r 使得簇集个数 k 增加，k 值增加使得传统 k-medoids算法对初值敏感加剧，算法收敛速度更加不稳定，最小收敛速度仍落后于本文算法和 LPKMD 算法.

（Ⅲ）实验结果三

实验参数设置：$n=5000$，$r=5000$m，$k=262$，$d=20$，结果如图 14 和表 4 所示.

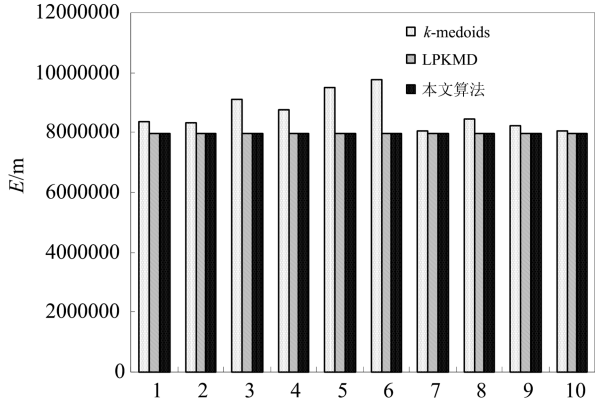

图 14　不同算法实验结果三对比图

Fig. 14　Comparison of different algorithms in the third set of experiments

表 4　E 和 V 对比

Tab. 4　Comparison of E and V in experiment 3

算法	E_{max}	E_{min}	E_{avg}	V_{max}	V_{min}	V_{avg}
PKMD	22 461 927	22 461 927	22 461 927	17	17	17
k-medoids	9 757 022	8 042 535	8 658 782.9	7	10	8.9
LPKMD	7 982 653	7 982 653	7 982 653	11	11	11
本文算法	7 981 678	7 981 678	7 981 678	9	9	9

实验三中，本文算法的代价 E 略微好于 LPKMD 算法；收敛速度 V 具有优势. 对比实验一与实验三可以看出，当 $r=5000$m 时，全集个数 n 增加 5 倍，本文算法与 LPKMD 算法在代价 E 上也增加近 5 倍，E 与 n 的比例系数约为 1，而传统的 k-

medoids算法在代价 E 上增加了近3.5倍，比例系数约为0.7.由于本文算法代价 E 的比率系数高于传统的 k-medoids 算法，因此当 n 增加后，本文算法较传统的 k-medoids 算法优势将会降低，针对本组实验：E(k-medoids)/E(本文算法)=1.22，本文算法聚类结果仍然具有较大优势.

(Ⅳ) 实验结果四

实验参数设置：n=5 000，r=500m，k=1 811，d=10，结果如图15和表5所示.

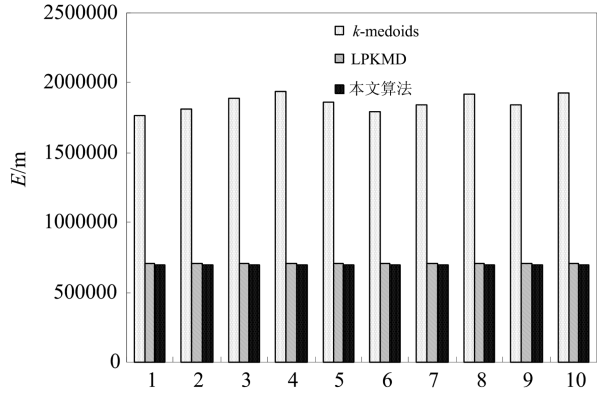

图15 不同算法实验结果四对比图

Fig. 15 Comparison of different algorithms in the fourth set of experiments

表5 E 和 V 对比

Tab. 5 Comparison of E and V in experiment 4

算法	E_{max}	E_{min}	E_{avg}	V_{max}	V_{min}	V_{avg}
PKMD	Null	Null	Null	Null	Null	Null
k-medoids	1 933 165	1 768 171	1 859 022.4	10	5	6.3
LPKMD	702 902	702 902	702 902	7	7	7
本文算法	694 019	694 019	694 019	6	6	6

由于PKMD算法选取了密度大于平均密度的点作为高密度集合 S，当 R 取500时，AvgDensity(x)=11.173 2，集合 S 的数量为1 429，无法构造出1 811个聚类中心，因此此时给出Null.

实验四中，本文算法在代价 E 上优于LPKMD算法及传统的 k-medoids 算法，相对于LPKMD算法，优化约1.2%；收敛速度优于LPKMD算法，均值上好于传统的 k-medoids 算法. 实验二与实验四对比可以看出，当 r=500m时，全集个数 n 增加5倍，本文算法与LPKMD算法在代价 E 上增加近7.7倍，E 与 n 的比例系数约为1.54，而传统的 k-medoids算法在代价 E 上增加了近3.7倍，E 与 n 的比例系数约为0.74，与实验三中分析一致，当 n 增加后，本文算法较传统的 k-medoids 算法优势将会降低. 本实验中 E(k-medoids)/E(本文算法)=2.79，本文算法聚类结果仍然具有较大优势，证明本文算法在初始聚类中心选择上较优.

(Ⅴ) 实验结果五

实验参数设置：n=19 739，r=2 500m，k=847，d=30，结果如图16和表6所示.

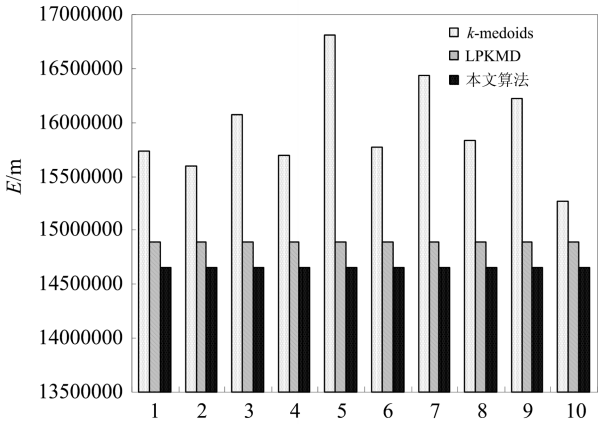

图16 不同算法实验结果五对比图

Fig. 16 Comparison of different algorithms in the fifth set of experiments

表6 E 和 V 对比

Tab. 6 Comparison of E and V in experiment 5

算法	E_{max}	E_{min}	E_{avg}	V_{max}	V_{min}	V_{avg}
PKMD	85 688 877	85 688 877	85 688 877	23	23	23
k-medoids	16 817 081	15 263 320	15 942 745.9	15	9	11.3
LPKMD	14 889 934	14 889 934	14 889 934	19	19	19
本文算法	14 652 102	14 652 102	14 652 102	10	10	10

在实验五中，本文算法在代价 E 上好于LPKMD算法及传统的 k-medoids 算法，相对于LPKMD算法，优化约1.5%. 收敛速度 V 好于LPKMD算法，在均值上好于传统的 k-medoids 算法，与实验一、实验三和实验四结果一致. LPKMD算法在收敛速度上落后传统的 k-medoids 算法，与实验三和实验四结果一致，可见LPKMD在收敛速度上比起 k-medoids 算法并无优势. 除了实验二，LPKMD算法的距离平方和 E 略小于本文算法外，其他几组实验，本文算法都好于LPKMD，拥有较快的收敛速度和较小的距离平方和. 所有的实验中，本文算法的聚类效果都好于传统的 k-medoids 算法.

半径越小，产生的 k 值相对于整个集合 C 的数目来说越大，本文算法的距离平方和越优；随着半径

r 的增加，本文算法的优势会减小. 研究 LBSN 时，由于大部分用户所在城市的签到数据都比较密集，因此缩小半径 r 对地点进行聚合显得更加实际.

(Ⅵ) 实验结果六

分析 2.4 节可知，本文算法在选取初始化中心后，采用 k-medoids 算法进行聚类. 中心点的偏移会导致一些点到该点所属聚类中心的距离大于半径 r，实验(f)对以上五组实验进行了统计，结果如表 7 所示.

表 7　聚类统计结果

Tab. 7　Clustering results and statistics

实验名称	n	r	n_r	β_{max}	β_{min}	β_{avg}	p
实验(Ⅰ)	1 000	5 000	14	1.250	1.009	1.103	1.40%
实验(Ⅱ)	1 000	500	7	1.290	1.007	1.098	0.70%
实验(Ⅲ)	5 000	5 000	55	1.556	1.002	1.116	1.10%
实验(Ⅳ)	5 000	500	62	1.702	1.001	1.139	1.24%
实验(Ⅴ)	19 739	2 500	305	1.783	1.000	1.122	1.54%

n 代表节点个数，r 代表半径，n_r 代表与聚类中心距离大于 r 的节点个数，p 代表 n_r 与 n 的比值，β 代表节点到聚类中心的距离与 r 的比值. 实验结果显示，聚类半径以外的点到聚类中心的平均距离约为 1.1 倍 r，且占整个集合的比例不超过 1.54%.

相比 k-medoids 算法，本文算法时间复杂度更高. k-medoids 算法为 $O((nk+n^2/k)m)$，其中 n 代表节点个数，k 代表聚集数，m 代表迭代次数；本文算法为 $O(\alpha n^3/r^2+(nk+n^2/k)m)$，$r$ 为聚类半径，α，k 依赖集合和 r 产生，m 为迭代次数. 故后续可以在保证聚类精度的基础上，继续对算法进行优化，进一步降低时间复杂度.

4　结论

本文首先介绍了基于 LBSN 的位置相似度计算方法；然后给出签到数据聚类处理的算法，包括 k-means 和 k-medoids 算法以及基于密度思想的 PKM 算法、PKMD 算法和 LPKMD 算法. 在此基础上，分析了 k-medoids 算法在研究基于位置的社交网络上的不足，即其聚类效果受到随机初始值影响. 基于上述分析，本文提出了改进的 k-medoids 算法，详细阐述了算法的思想，并使用 Gowalla 数据集对算法的聚类效果进行了实验评估，验证了本文算法在收敛速度和聚类效果上的优势. 最后，分析了本文所提算法的不足之处，如它在时间复杂度上比 k-medoids 算法高很多，为算法的进一步研究指明了方向.

参考文献(References)

[1] WANG H, TERROVITIS M, MAMOULIS N. Location recommendation in location-based social networks using user check-in data[C]// Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Orlando: ACM Press, 2013: 374-383.

[2] WAKAMIYA S, LEE R, SUMIYA K. Urban area characterization based on semantics of crowd activities in Twitter[C]// International Conference on Geospatial Semantics. Springer-Verlag, 2011: 108-123.

[3] PARK H S, JUN C H. A simple and fast algorithm for K-medoids clustering[J]. Expert Systems with Applications, 2009, 36(2): 3336-3341.

[4] 孙秀娟，刘希玉. 基于初始中心优化的遗传 K-means 聚类新算法[J]. 计算机工程与应用，2008，44(23)：166-168，182.

[5] 夏宁霞，苏一丹，覃希. 一种高效的 K-medoids 聚类算法[J]. 计算机应用研究，2010，27(12)：4517-4519.

[6] PARDESHI B, TOSHNIWAL D. Improved k-medoids clustering based on cluster validity index and object density[C]// Proceedings of the 2nd International Advance Computing Conference. Patiala, India: IEEE Press, 2010: 379-384.

[7] 姚丽娟，罗可，孟颖. 一种新的 k-medoids 聚类算法[J]. 计算机科学与应用，2013，49(19)：153-157.

[8] 李巍，孙涛，陈建孝，等. 基于加权余弦相似度的 XML 文档聚类研究[J]. 吉林大学学报，2010，28(1)：68-76.

[9] JANG J W, KIM B M, JANG S B, et al. Application of Euclidean distance similarity for smartphone-based moving context determination[J]. Journal of the Korea Industrial Information System Society, 2014, 19(4): 53-63.

[10] KANUNGO T, MOUNT D M, NETANYAHU N S, et al. An efficient k-means clustering algorithm: analysis and implementation [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002, 24(7): 881-892.

[11] KHAN F. Short communication: An initial seed selection algorithm for k-means clustering of georeferenced data to improve replicability of cluster assignments for mapping application[J]. Applied Soft Computing, 2012, 12(11): 3698-3700.

[12] 杨志，罗可. 一种改进的基于粒子群的粗糙 K-medoids 算法[J]. 计算机工程与应用，2014，50(20)：110-114.