基于 MapReduce 的强连通网格聚类算法

胡赢双 陆亿红
(浙江工业大学计算机科学与技术学院 杭州 310014)

摘 要 随着位置大数据的爆炸式增长,传统的串行算法已无法对其进行高效地聚类处理,因此,基于 MapReduce 框架的并行聚类算法研究逐渐成为热点。聚类算法并行化后的聚类质量通常难以保证,因此对并行化聚类结果进行归约的方法极为重要。首先提出基于网格的改进 DBSCAN 并行化聚类算法,通过该步骤得到每个数据子集的聚类结果。然后在分析网格与簇的关系,定义网格簇和网格簇的连通、强连通概念的基础上,通过计算网格簇之间的连通权值矩阵,对具有强连通关系的网格簇进行归约,构成基于 MapReduce 的强连通网格聚类算法。该算法可实现位置大数据集的高效聚类。实验分析表明,基于 MapReduce 的强连通网格聚类算法对位置大数据的处理具有较高的效率和聚类质量。

关键词 位置大数据,网格,MapReduce,强连通,DBSCAN
中图法分类号 TP274 **文献标识码** A

Cell Clustering Algorithm Based on MapReduce and Strongly Connected Fusion

HU Ying-shuang LU Yi-hong
(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China)

Abstract With the explosive growth of large location data, most of the traditional serial clustering algorithms can not process big data efficiently. In order to solve this problem, more and more people are studying parallel clustering algorithm. It is difficult to guarantee the clustering quality of parallel clustering algorithm, so it is important to study the algorithm of reducing the result of parallel clustering. Therefore, a grid clustering algorithm based on strongly connected fusion was proposed. Firstly, clustering result of data subsets is obtained according to the improved DBSCAN algorithm based on MapReduce. Next, the relationship between grid and cluster is analyzed and the concepts of Gird-cluster, connectivity and strong connectivity of Gird-clusters are defined. Then the connectivity weights matrix between Gird-cluster and Gird-cluster is calculated. Finally, whether to reduce two Gird-clusters or not is decided according to connectivity weight. The experimental results show that the proposed algorithm has high efficiency and high clustering quality in processing large location data.

Keywords Big data of position, Gird, MapReduce, Strongly connected fusion, DBSCAN

1 引言

随着定位设备的普及、定位卫星精确性的提高,各类基于位置服务的应用大幅增加,如各种社交网络的位置签到打卡、车辆和智能手机等各种终端的定位服务[1]。伴随着 5G 时代的到来,物联网行业必将快速发展,位置数据也将在其中发挥重要作用。基于位置大数据的服务在实际生活中已得到广泛应用,比如在交通调度与控制[2]、推荐系统[3]、城市规划[4]、广告投放、公共设施选址等方面都有巨大应用价值。在不久的将来,位置大数据服务在智能城市建设、自动驾驶、物联网等方面也将有着重要应用。鉴于位置数据采集终端的数量快速增长,其数据量也必将呈指数级增长趋势。因此如何在大量位置数据中快速地挖掘出有效信息变得尤为重要。

位置大数据的研究包括数据采集、数据处理与分析、数据存储和数据可视化等方面[1]。其中数据处理与分析是位置大数据中研究内容最为丰富的领域之一。如郭迟等[5]基于价值提取与协同挖掘方法提出一种位置大数据分析整体框架,用来分析交通问题、人类社会经济活动等。林乐轩[6]通过对行人轨迹和停留点的分析来进行行人路径预测与人群密度预估。Tobler 等[7]将地理信息栅格化以进一步分析人口密度分布情况。对位置大数据的处理与分析也会运用到各种聚类分析算法,如李斯凡[8]利用 K-means 算法分析公交车上乘客的上下车活跃点,得到乘客出行热点区域的分布规律。DBSCAN 算法可以处理任意形状的数据并能处理噪声点,也常用于对位置数据的处理与分析,但 DBSCAN 算法复杂度较高($O(n^2)$),难以在大规模位置数据的分析与处理中得到应用。针对该问题,Guttman[9]利用 R-Tree 技术来减小 DBSCAN 算法步骤中邻域的搜索空间,以此提高 DBSCAN 算法的效率。Zhao 等[10]对数据进行网格划分,选择 m 个高密度网格作为初始簇,然后对其相邻网格进行检索从而进行聚簇扩展,该方

本文受浙江省基础公益研究计划项目(GG19E090005)资助。
胡赢双(1995—),女,硕士生,主要研究方向为数据挖掘,E-mail:huyingshuang2@163.com;陆亿红(1968—),女,硕士,副教授,主要研究方向为软件理论、数据挖掘等,E-mail:lyh@zjut.edu.cn(通信作者)。

法减少了数据对象的查询与距离计算次数，提高了算法效率。Kumar 等[11]提出了基于图结构的 DBSCAN 算法，利用图结构的邻居节点搜索来进行聚簇扩展，进而提高了 DBSCAN 算法的效率。

随着 MapReduce 框架的提出，大数据分析处理的效率得以大幅提高，因此 MapReduce 框架下并行聚类算法的研究也逐渐成为热点。He 等[12]利用 MapReduce 平台实现了一个并行密度聚类算法，根据空间分布进行数据划分，将划分后的数据分配到各个节点进行聚类，最后对各节点的数据子集聚类结果进行聚合。Kim 等[13]同样提出了一种基于 MapReduce 的密度聚类算法 DBCURE-MR，他们首先提出了一种基于 k-椭圆邻居的密度聚类算法 DBCURE，在 Map 阶段并行实现数据点的 k-椭圆邻居搜索和核心簇的发现，最后在 Reduce 阶段对每个 Map 节点得到的聚类结果进行聚合，得到最终聚类结果。

本文首先基于一种改进的 DBSCAN 算法[14]在 MapReduce 框架下对位置大数据进行并行聚类，然后提出基于网格的强连通方法对子集数据聚类结果进行聚合，得到最终聚类结果。DBSCAN 算法可以处理任意形状的数据，适用于位置大数据的海量性、空间分布任意性。对数据进行网格划分，然后基于网格进行聚簇扩展，可提高算法效率。而在 MapReduce 框架下对数据进行并行处理，则大幅提高了大规模位置大数据的处理效率，但通常情况下无法保证聚类质量。因此除了并行处理的算法外，对并行处理后的数据子集聚类结果进行聚合的方法也非常关键。本文基于网格簇的强连通关系，形成网格簇之间的权值连通图，计算网格簇之间的连通权值矩阵。根据设定的阈值判断两个网格簇是否强连通，对数据子集聚类结果进行聚合。通过实验分析，通过该聚类过程对位置大数据进行分析处理，可以保证较高的效率和聚类质量。

2　相关定义

2.1　DBSCAN 算法相关定义

DBSCAN 算法于 1996 年由 Martin 等[15]提出，是一种基于密度的聚类算法，能够发现任意形状的簇。DBSCAN 算法需给定两个参数：Eps 和 $MinPts$。

设有数据集 D 和实数 $Eps>0$，则对任意数据点 $p\in D$，称 $Eps(p)=\{Y|Y\in D,d(Y,X)\leqslant Eps\}$ 为 p 在 D 中的 Eps-邻域，简称 p 的 Eps-邻域。$N^D_{Eps}(p)$ 代表 p 的邻域集合，数目表示为 $|N^D_{Eps}(p)|$。而基于邻域集合和数目的定义，所有数据点可分为 3 种类型：核心点、边界点和噪声点[16]。

核心点：对于给定的实数 $Eps>0$、正整数 $MinPts$ 和任意点 $p\in D$，如果 $|N^D_{Eps}(p)|\geqslant MinPts$，则称点 p 为核心点[16]。

边界点：对于给定的实数 $Eps>0$、正整数 $MinPts$ 和任意点 $p\in D$，如果 $|N^D_{Eps}(p)|<MinPts$，但在集合 $N^D_{Eps}(p)$ 中存在点 $q(q\in N^D_{Eps}(p))$ 是核心点，则称点 p 为边界点[16]。

噪声点：对于给定的实数 $Eps>0$、正整数 $MinPts$ 和任意点 $p\in D$，若 p 既不是核心点也不是边界点，则称点 p 为噪声点[16]。

基于数据点的 3 种类型定义，数据集 D 中的所有数据点之间的密度相连关系可分为 3 种：直接密度可达、密度可达和密度相连[16]。

直接密度可达：对于给定的实数 $Eps>0$、正整数 $MinPts$ 和任意点 $p\in D$，如果点 p 为核心点，则点 p 直接密度可达到其邻域集合 $N^D_{Eps}(p)$ 中的任一数据点[16]。

密度可达：如果 D 存在一个对象链 $X_1,X_2,X_3,\cdots,X_n$，$X_1=X,X_n=Y$，且从 $X_i(1\leqslant i\leqslant n-1)$ 到 X_{i+1} 是直接密度可达的，则称 X 到 Y 是密度可达的[16]。

密度相连：对于给定的实数 $Eps>0$、正整数 $MinPts$ 和任意点 $p,q\in D$，如果存在 $o\in D$，使从 o 到 p，o 到 q 都密度可达，则称 p 和 q 密度相连[16]。

2.2　基于网格的 DBSCAN 算法的相关定义

本文基于网格对数据集中的数据对象进行网格划分[14]，并根据 DBSCAN 算法的原理对网格之间的密度相连关系进行分析。

定义 1（单元网格[14]）　设定数据集 $D=\{X_1,X_2,\cdots,X_n\}$ 和实数 $Eps>0$，对数据集 D 中的每个数据点进行网格划分，网格边长为 $Eps/2\sqrt{2}$。对数据点进化网格划分后，每个网格称为单元网格，表示为 $G=\{v_1,v_2,\cdots,v_d\}$，对于数据集中任意数据点 $X_i(x_{i1},x_{i2},\cdots,x_{id})$，都可对应到相应网格 $G(X_i)=\{v_1,v_2,\cdots,v_d\}$，其中 $v_i=|2\sqrt{2}\cdot x_{ii}/Eps|$，$X_i$ 为该划分网格的中心点。$|G(X_i)|$ 表示该网格内数据点的数目。

DBSCAN 算法是基于数据点之间的距离计算数据点的密度相连关系，而单元网格的划分大小是由参数 Eps 定义，因此数据点的密度相连关系可转换为网格之间的密度相连关系，然后基于网格进行聚簇扩展，而核心点的判断也可由网格中的数目进行判断。

给定网格 $G(X_i)$ 和正整数 $MinPts$，设该网格的相邻网格集为 $NG(X_i)$，数目表示为 $|NG(X_i)|$，若 $|NG(X_i)|\geqslant MinPts$，则网格 $G(X_i)$ 中的所有点一定为核心点。

如图 1 所示，单元网格 1 中的任意一点的 Eps-邻域都包含其相邻网格 2－9，相邻网格 2－9 的数据点之和大于 $Minpts$，则网格 1 中任意一点的 Eps-邻域内的邻居点数目都大于 $MinPts$，因此这些点肯定为核心点。若 $|NG(X_i)|<MinPts$，则判断网格中的某一点是否为核心点，则至多需要检查网格顶点的 Eps-邻域包含的网格中的数据点数目，如图 1 所示。

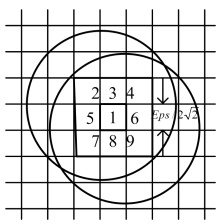

图 1　数据点网格划分图[7]

定义 2（核心网格）　设有网格 $G(X_i)$，如果 $|NG(X_i)|\geqslant MinPts$，则 $\forall p(p\in G(X_i))$ 是核心点，该网格也称为核心网格。而对于核心网格，根据 DBSCAN 算法原理，只要其中任意一个数据点属于某个簇，则网格中的所有点都可划分到该簇中。

定义 3（边界网格）　设有网格 $G(X_i)$，如果 $|NG(X_i)|<MinPts$，但 $\exists p(p\in G(X_i))$ 是核心点，那该网格内的任意两个数据点之间都直接密度可达，该网格也称为边界网格。对于边界网格，根据 DBSCAN 算法原理，若网格中任意一点属于

某个簇，则网格中的所有点都可划分到该簇中。

定义 4（包含网格） 设有网格 $G(X_i)$ 和密度簇 C_i，如果 $\forall p(p\in G(X_i))$ 都有 $p\in C_i$，则称网格 $G(X_i)$ 是密度簇 C_i 的包含网格[14]。

定义 5（网格簇） 设数据集 D 可划分为 u 个网格 $\{G_1,\cdots,G_u\}$，C_j 是某个并行聚类算法得到 D 的一个簇 C_j。如果 G_{if} 是 u 个网格中满足 $G_{if}\subseteq C_j(f=1,2,\cdots,v)$ 的网格，则称 C_j 为 $\{G_{i1},\cdots,G_{iv}\}$ 组成的网格簇。

由定义 1－定义 5 可知，若一个边界网格属于网格簇 C_i，则其相邻的网格一定是网格簇 C_i 的包含网格。因为若 $G(X_i)$ 是边界网格，则 $\exists p(p\in G(d_1,d_2))$ 是核心点。根据定义 1，所有在 $G(X_i)$ 相邻网格内的数据点都是该网格内所有点的 Eps-邻域的邻居点，因此数据点之间都直接密度可达，因此相邻网格内的数据点也都可划分到簇 C_i 中。

2.3 网格簇之间的强连通关系

设数据集 D 用某种并行聚类算法进行聚类后得到 m 个子集聚类结果 $\{C^{(1)},C^{(2)},\cdots,C^{(m)}\}$，其中 $C^{(i)}=\{C_{i1},C_{i2},C_{i3},\cdots,C_{ik}\}$，$C_{ik}$ 为网格簇。若将 m 个聚类结果的每个网格簇作为图中的一个结点，并定义结点 C_{ix} 与 C_{jy} 有边邻接的充分必要条件是 $C_{ix}\cap C_{jy}\neq\varnothing$，即两个网格簇有交叠网格并且相叠网格中的数据点数目不为零，则由各个网格簇形成一个图[16-17] $P(V,E)$，其中：

结点集 $V=\{C_{11},C_{12},C_{13},\cdots,C_{1k},\cdots,C_{m1},C_{m2},C_{m3},\cdots,C_{mk'}\}$。

边集 $E=\{(C_{ix},C_{jy})|C_{ix}\cap C_y\neq\varnothing$，且 $i\neq j,i=1,2,\cdots,k;j=1,2,\cdots,k'\}$。

边权值 $W(C_{ix},C_{jy})=|C_{ix}\cap C_{jy}|/|C_{ix}\cup C_{jy}|$，$|C_{ix}\cap C_{jy}|$ 为两个网格簇的相叠网格中的数据点数目，$|C_{ix}\cup C_{jy}|$ 为两个网格簇包含的数据点总数目。

若 W 的值足够大，可以推断，对于数据集 D 而言这两个簇应该合并为一个簇。

定义 6（强直接连通） 设定一个阈值 $\omega\in(0,1)$，两个网格簇 C_{ix} 与 C_{jy}，若 $W(C_{ix},C_{jy})>\omega$，则称 C_{ix} 与 C_{jy} 之间强直接连通。

定义 7（强间接连通） 设 C_{ix} 与 C_{jy} 之间强直接连通，C_{jy} 和 C_{mz} 之间强直接连通，则称 C_{ix} 和 C_{mz} 之间强间接连通，$C_{ix}\rightarrow C_{jy}\rightarrow C_{mz}$ 称为一条强连通链。

3 基于 MapReduce 的强连通网格聚类算法

MapReduce 是一种用于分布式平台的编程框架，一般分为两个阶段：Map 阶段和 Reduce 阶段。如图 2 所示，对于输入的数据若不用初步预处理，则将划分成若干数据块分配到 Mapper 处理器执行 Map 函数，Map 函数处理后形成中间结果，若处理过程较为复杂，则传到 Reduce 处理器之前用 Combine 函数进行处理，然后 Reduce 函数对传入的中间结果数据进行归约，执行定义的 Reduce 函数，最后产生最终结果作为输出。

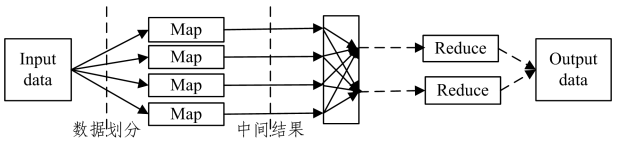

图 2 MapReduce 处理流程图[18]

3.1 Map 过程

首先将数据分块，分块时要保证分布式计算的负载均衡。各个 *Map* 节点分配到数据子集后，对各自数据集 D_m 中的数据进行网格划分，对于数据集中任意数据点 $X_i(x_{i1},x_{i2},\cdots,x_{id})$ 将其对应到相应网格 $G(X_i)$，$G(X_i)=\{v_1,v_2,\cdots,v_d\}$。其中 $v_i=|2\sqrt{2}\cdot x_{ii}/Eps|$。进行网格划分后，在每个节点用基于网格的 DBSCAN 算法[14]进行聚类，得到每个数据子集的聚类结果。

基于网格的 DBSCAN 算法与 DBSCAN 算法相同，先判断网格的类型，再进行网格簇扩展。

算法 1 基于网格的 DBSCAN

输入：数据集 $D\{X_1,X_2,\cdots,X_n\}$，给定参数 Eps 和 Minpts

输出：聚簇结果

1. 对每个数据点 X_i 计算对应的网格 $G(X_i)=\{v_1,v_2,\cdots,v_d\}$，其中 $v_i=|2\sqrt{2}\cdot x_{ii}/Eps|$，形成网格集合 $GG\{G_1,G_2,\cdots,G_m\}$。
2. 遍历数据点，当数据点 X_i 没有被聚类且该数据点所在表格 $G(X_i)$ 没有被聚类时，创建新簇 C_i，对网格 $G(X_i)$ 进行网格扩展聚类。
3. 重复步骤 2 直至所有点遍历结束，输出聚类结果。

网格扩展聚类步骤如下：

1. 对于点 X_i 及其所在的网格 $G(X_i)$，首先判断网格 $G(X_i)$ 是否为核心网格或边界网格，若是则继续进行步骤 2，否则将 X_i 暂时置为噪声点，结束网格扩展聚类步骤。
2. 对 $G(X_i)$ 进行核心、边界网格扩展步骤，得到网格簇 C_i 包含的网格和 C_i 的核心、边界网格列表 CenterCellList。
3. 将 CenterCellList 中网格的外围网格加入 AroundCellList。
4. 遍历 AroundCellList 中网格，对于每一个网格中的数据点 p，先判断 p 是否是 CenterCellList 中某一网格核心点的邻居点，若不是则点 p 为噪声点，若是则点 p 一定属于簇 C_i；然后再判断 p 是不是核心点，若 p 不是核心点，则 p 是簇 C_i 的边界点，若 p 是核心点，则对 p 所在的网格执行核心、边界网格扩展步骤，更新 C_i，CenterCellList 和 AroundCellList。
5. 重复步骤 4，直至 AroundCellList 中的网格遍历结束。

核心、边界网格扩展步骤如下：

1. 对于一个核心网格或边界网格 $G(X_i)$，将其 9 个相邻网格（包括自身）加入待扩展网格列表 CellList，并将 $G(X_i)$ 加入网格簇 C_i 的核心、边界网格列表 CenterCellList。
2. 将 $G(X_i)$ 加入簇 C_i，将 $G(X_i)$ 从 CellList 中移除。
3. 遍历 CellList，得到新的网格 G，若网格 G 为核心网格或边界网格，则继续扩展网格，将 G 的 9 个相邻网格（包括自身）加入待扩展网格列表 CellList，若 CellList 中已存在相应网格，则不重复加入。将 G 加入网格簇 C_i 的核心、边界网格列表 CenterCellList，跳至步骤 2；否则网格 G 不再扩展，G 是包含网格，因此将 G 加入簇 C_i，将 G 从 CellList 中移除。
4. 重复步骤 3，直至 CellList 中的网格遍历结束。

每个 *Map* 节点根据基于网格的 DBSCAN 算法对数据进行并行处理之后得到数据子集的聚类结果，以键值对的形式输出，其存储数据点的数据、所处网格标记、所在网格簇标记。

3.2 Reduce 过程

数据经过 Map 阶段得到各数据子集的聚类结果，在 Reduce 阶段采用基于网格的强连通聚类方法对数据子集的聚类结果进行归约，形成最终的聚类结果。

算法 2 基于网格的强连通聚类方法

输入：数据集 D 经过 m 个 Map 节点处理得到的 m 个聚类结果 $\{C^1,C^2,\cdots,C^m\}$，阈值 w

输出：最终聚类结果 C

1. i=1，2，…，m，计算聚类 $C^{(i)}$ 中每个网格簇与其他所有聚类的网格簇的权值矩阵 $W(C^1,C^2,\cdots,C^m)$；
2. 根据阈值 w 遍历 $W(C^1,C^2,\cdots,C^m)$，找出权值大于 w 的强连通链；
3. 融合每一条强连通链，每一条强连通链代表的网格簇形成新的簇；
4. 对无强连通关系的网格簇再进行判断，计算两个网格簇中心点的距离，若距离小于 $2\max(\mathrm{dis}(G_i,C_i))(\max(\mathrm{dis}(G_i,C_i))$表示网格簇簇中心所在网格与离它最远的网格的距离），则对两个网格簇进行融合。
5. 输出最终聚类结果。

在 Reduce 处理器上经过强连通聚类方法得到最终聚类结果，以键值对的形式输出，其存储数据点的数据、最终所在簇的中心点。

4　实验与分析

4.1　实验平台与数据

硬件环境：本次实验部署 6 个节点，1 个 Master 节点，作为 namenode，其余 5 个为 Slave 节点，作为 DataNode，CPU 为 2.20 Hz，内存 4 GB，硬盘 256 GB。

Hadoop 集群环境：Ubuntu 14.04 版，VirtualBox Portable，Hadoop 2.6.0 版，Jdk7.0 版，Eclipse 3.8 版。

实验数据集：实验采用两个数据量较小的和两个数据量较大的二维数据集，如表 1 所列。Aggregation[19] 数据集分为 7 个簇，这 7 个簇都呈非高斯分布。Compound[20] 包含了 6 个不同形状的聚类簇，结构相对复杂，包括内嵌、相连等情况。Taxi1 和 Taxi2 数据来自 T-Drive 项目[21]，该项目采集了某城市上千辆出租车的 GPS 数据。

表 1　实验数据描述

名称	数据点数目	类别数
Aggregation	788	7
Compound	399	6
Taxi1	1194976	未知
Taxi2	2148225	未知

4.2　实验及结果分析

在 MapReduce 框架下用 K-means 算法、DBSCAN 算法、本文算法对数据集进行处理，并对结果进行分析。

对于两个预先设置聚类结果的数据集，分别在 MapReduce 框架下用 3 个算法分别对其进行聚类，比较聚类结果的准确率，如表 2 所列。

表 2　算法准确率对比

（单位：%）

算法名称	数据集 Aggregation	数据集 Compound
K-means 算法	90.1	67.0
DBSCAN 算法	98.9	92.1
本文算法	98.6	91.3

从实验结果可以看出，本文算法和 DBSCAN 算法的准确率相差较小，因为 DBSCAN 算法根据密度相连的关系，可以发现任意形状的簇。本文算法虽然对数据对象进行网格划分，再利用网格进行聚簇扩展，但也是基于 DBSCAN 算法的密度相连关系，在归约阶段再利用强连通融合方法对子集聚类结果进行融合，因此聚类准确率与 DBSCAN 算法相当。而 K-means 算法无法发现任意形状的簇，对于数据集 Compound 准确率相对较低。

对算法进行并行化处理不仅为了保证聚类结果的质量，最关键的是要提高聚类效率。与串行化的 K-means 和 DBSCAN 算法相比，在数据量较大和并行处理节点较多时，其并行化算法的处理效率明显提高。对 Taxi1 和 Taxi2 数据集分别在 MapReduce 框架下用 3 个算法并行化并分别对其进行聚类，K-means 算法设定聚簇数量 $k\approx n^{\frac{3}{8}}$，除去分配数据时间，比较算法的运行时间，结果如表 3 所列。

表 3　算法处理时间对比

（单位：s）

算法名称	数据集 Taxi1	数据集 Taxi2
K-means 算法	2551.19	5653.25
DBSCAN 算法	3889.89	10941.37
本文算法	386.20	787.12

从实验结果可以看出，本文算法相较于 K-means 和 DBSCAN 算法效率有明显提高，在数据量越大时，提高的效率越明显。

结束语　针对位置大数据的数据规模，本文首先基于 MapReduce 对一种改进的 DBSCAN 算法进行并行化研究，进一步提高对数据网格划分带来的高效率。而同时为了保证聚类质量，对数据子集聚类结果的归约方法进行研究。本文提出一种基于网格的强连通方法对子集聚类结果进行归约，计算网格簇的连通权值矩阵，根据预设的阈值对网格簇进行融合，得到最终聚类结果。通过实验分析，该算法提高了聚类效率，同时也有较高的聚类质量。由于本文是基于 DBSCAN 算法的改进，会预先设置两个参数 *Eps* 和 *MinPts*，因此初始参数的选取对聚类结果也会有一定的影响，如何选取合理的邻域半径值也是下一步应研究的内容。

参 考 文 献

[1]　刘经南，方媛，郭迟，等. 位置大数据的分析处理研究进展[J]. 武汉大学学报(信息科学版)，2014，39(4)：379-385.

[2]　YUAN J，ZHENG Y，XIE X，et al. T-Drive：Enhancing driving directions with taxi drivers' intelligence[J]. IEEE Trans. on Knowledge & Data Engineering，2013，25(1)：220-232.

[3]　ZHENG Y，XIE X，MA W Y. GeoLife：A collaborative social networking service among user，location and trajectory[J]. Bulletin of the TechnicalCommittee on Data Engineering，2010，33(2)：32-39.

[4]　YUAN J，ZHENG Y，XIE X. Discovering regions of different functions in a city using humanmobility and POIs[C]// Knowledge Discovery and Data Mining. ACM Press，2012：186-194.

[5]　郭迟，刘经南，方媛，等. 位置大数据的价值提取与协同挖掘方法[J]. 软件学报，2014，25(4)：713-730.

[6]　林乐轩. 基于位置大数据的行人路径预测及人群密度预估系统研究[D]. 北京：北京邮电大学，2018.

[7]　TOBLER W，DEICHMANN U，GOTTSEGEN J，et al. World population in a grid of spherical quadrilaterals[J]. International Journal of Population Geography，1997，3(3)：203-225.

[8]　李斯凡. 基于无监督学习技术的位置大数据分析[D]. 杭州：浙江理工大学，2017.

[9]　GUTTMAN A. R-trees：A dynamic index structure for spatial searching[C]// International Conference on Management of Data. Boston：1984：47-57.

（下转第 215 页）

均明显高于 ReliefF 算法的分类准确率且本文选取的特征数明显少于 ReliefF 算法选取的特征数。

综上所述，本文方法在分类准确率、分类稳定性上均优于传统的特征选择算法，且特征选择数目较少，达到了高维空间维度约简的目标。

结束语　大数据时代，如何降低高维空间中特征选择的计算成本并提高分类准确率是数据挖掘领域中的重要研究问题之一。本文在分析传统特征选择算法的基础上，提出了一种融合蚁群算法和随机森林的特征选择方法，并在 UCI 数据集上将两种方法进行了对比，实验表明本文的特征选择方法能够有效地减少数据集中的特征数量，同时提高了数据分类的准确率。

参 考 文 献

[1] 姚登举，杨静，詹晓娟. 基于随机森林的特征选择算法[J]. 吉林大学学报(工学版)，2014，44(1)：137-141.

[2] 刘飞飞. 特征选择算法及应用综述[J]. 办公自动化，2018，23(21)：47-49.

[3] 张翠军，陈贝贝，周冲，等. 基于多目标骨架粒子群优化的特征选择算法[J]. 计算机应用，2018，38(11)：3156-3160，3166.

[4] 刘依恋. 模式分类中特征选择算法研究[D]. 哈尔滨：哈尔滨理工大学，2014.

[5] BREIMEN L. Random Forests [J]. Machine Learning，2001，45(1)：5-32.

[6] 徐少成，李东喜. 基于随机森林的加权特征选择算法[J]. 统计与决策，2018，34(18)：25-28.

[7] 杨凯，侯艳，李康. 随机森林变量重要性评分及其研究进展[OL]. http://www.paper.edu.cn/releasepaper/content/201507-212.

[8] ALBERTO C，MANIEZZO D. Distributed optimization by ant colonies[C]// Proc of the First European Conf on Artificial Life. Paris：Elsevier Publishing. 1991：134-142.

[9] 黄丹凤，祁云嵩，许姗娜. 基于粗糙集和蚁群算法的特征基因选择方法[J]. 计算机技术与发展，2012，22(6)：68-70，74.

[10] 马军建，董增川，王春霞，等. 蚁群算法研究进展[J]. 河海大学学报(自然科学版)，2005(2)：139-143.

[11] 杨丽. 基于 ReliefF 和蚁群算法的特征基因选择方法分析[J]. 电脑知识与技术，2017，13(32)：199-200.

[12] MURPHY P M，AHA D W. UCI repository of machine learning database [DB/OL]. (2006-05-12). http://www.ics.uci.edu/mlearn/MLRepository.html.

[13] KIRA K，RENDELL L A. The feature selection problem：Traditional methods and a new algorithm[C]// AAAI. 1992：129-134.

[14] 卜华龙，夏静，韩俊波. 特征选择算法综述及进展研究[J]. 巢湖学院学报，2008(6)：41-44.

[15] 许行，张凯，王文剑. 一种小样本数据的特征选择方法[J]. 计算机研究与发展，2018，55(10)：2321-2330.

[16] 朱振国，赵凯旋，刘民康. 基于强化学习的特征选择算法[J]. 计算机系统应用，2018，27(10)：214-218.

[17] 闫春，李亚琪，孙海棠. 基于蚁群算法优化随机森林模型的汽车保险欺诈识别研究[J]. 保险研究，2017(6)：114-127.

[19] 雷海锐，高秀峰，刘辉. 基于机器学习的混合式特征选择算法[J]. 电子测量技术，2018，41(16)：42-46.

[20] 邱宁佳，周稳，王鹏，等. 一种结合改进 CHI 和 RFFS 的特征选择算法研究[J]. 计算机工程与应用，2018，54(21)：133-140.

[21] 叶志伟，郑肇葆，万幼川，等. 基于蚁群优化的特征选择新方法[J]. 武汉大学学报(信息科学版)，2007(12)：1127-1130.

[22] 张文杰，蒋烈辉. 一种基于遗传算法优化的大数据特征选择方法[J]. 计算机应用研究，2019：1-5.

[23] 李晓岚. 基于 Relief 特征选择算法的研究与应用[D]. 大连：大连理工大学，2013.

[24] 蔡萌萌，张巍巍，王泓霖. 大数据时代的数据挖掘综述[J]. 价值工程，2019，38(5)：155-157.

[25] 魏茂胜. 数据挖掘中的分类算法综述[J]. 网络安全技术与应用，2017(6)：65-66.

(上接第 207 页)

[10] ZHAO Q，SHI Y，LIU Q，et al. A grid-growing clustering algorithm for geospatial Data[J]. Pattern Recognition Letters，2014，53(53)：77-84.

[11] KUMAR K M，REDDY A R M. A fast DBSC-AN clustering algorithm by accelerating neighbor searching using Groups method[J]. Pattern Recognition，2016，58：39-48.

[12] HE Y，TAN H，LUO W，et al. MR-DBSCAN：An efficient parallel density-based clustering algorithm using MapReduce[C]// 2011 IEEE 17th International Conference on Parallel and Distributed Systems. IEEE Computer Society，2011：473-480.

[13] KIM Y，SHIM K，KIM M S，et al. DBCURE-MR：An efficient density-based clustering algorithm for large data using MapReduce[J]. Information Systems，2014，42(2)：15-35.

[14] 于彦伟，贾召飞，曹磊，等. 面向位置大数据的快速密度聚类算法[J]. 软件学报，2018，29(8)：2470-2484.

[15] ESTER M，KRIEGEL H P，SANDER J，et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// International Conference on Knowledge Discovery & Data Mining. Portland：AAAI Press，1996：226-231.

[16] 黄德才. 数据仓库与数据挖掘教程[M]. 北京：清华大学出版社，2016.

[17] 钱潮恺，黄德才. 基于维度频率相异度和强连通融合的混合数据聚类算法[J]. 模式识别与人工智能，2016，29(1)：82-89.

[18] 余长俊，张燃. 云环境下基于 Canopy 聚类的 FCM 算法研究[J]. 计算机科学，2014，41(S1)：316-319.

[19] GIONIS A，MANNILA H，TSAPARAS P. Clustering aggregation[J]. ACM Transactions on Knowledge Discovery from Data (TKDD)，2007，1(1)：1-30.

[20] ZAHN C T. Graph-theoretical methods for detecting and describing gestalt clusters[J]. IEEE Transactions on Computers，1971，100(1)：68-86.

[21] YUAN J，ZHENG Y，XIE X，et al. T-Drive：Enhancing driving directions with taxi drivers' intelligence[J]. IEEE Transactions on Knowledge & Data Engineering，2013，25(1)：220-232.