中国科学: 信息科学 2023 年 第 53 卷 第 7 期: 1281–1298
SCIENTIA SINICA Informationis
论文

基于模糊密度峰值聚类的区域同位模式并行挖掘算法

蒋希文[1], 王丽珍[1,2*], Vanha TRAN[3]

1. 云南大学信息学院, 昆明 650504, 中国
2. 云南大学滇池学院, 昆明 650228, 中国
3. Department of Information Technology Specialization, FPT University, Hanoi 155514, Vietnam
* 通信作者. E-mail: lzhwang@ynu.edu.cn

收稿日期: 2022–03–14; 修回日期: 2022–09–01; 接受日期: 2022–11–18; 网络出版日期: 2023–06–30

国家自然科学基金 (批准号: 62276227, 61966036)、云南省基础研究计划重点项目 (批准号: 202201AS070015)、云南省创新团队项目 (批准号: 2018HC019) 和云南大学研究生科研创新基金 (批准号: 2021Y279) 资助项目

摘要 区域同位模式挖掘 (RCPM, regional co-location pattern mining) 是为了发掘某个局部区域内存在的同位 (co-location) 模式, 以发现在全局中无法发现的信息. 传统的区域挖掘大多会采用明确界限的几何体框定同位模式产生的区域. 但是现实中的各类区域可能是无明确边界的. 另外, 数据的分布情况作为区域的重要特征之一, 也应该成为区域选择的因素. 基于上述思考, 本文引入密度峰值聚类 (DPC, density peak-based clustering), 提出新的密度度量函数, 并结合模糊集理论与 k 近邻距离, 设计了一个行之有效的并行区域同位模式挖掘算法. 实验结果表明, 利用本文方法挖掘到的结果更具有现实意义, 并且并行化极大地提升了挖掘算法的效率. 在真实数据上, 2 线程下的加速比达到了 1.89.

关键词 空间数据挖掘, 区域同位模式, 模糊密度峰值聚类, 并行算法, k 近邻

1 引言

随着全球卫星定位系统与遥感技术的高速发展, 人类每天都产生大量的空间数据. 由于空间数据的多变性以及其数据规模往往大于传统事务型数据, 因此对空间数据的利用处理也变得更为复杂. 空间数据除了包含位置数据和属性数据外, 还可能包含不同范围内空间实体的空间关系 [1]. 如何利用好空间数据就成为了近年来专家学者热衷于研究的一个问题. 空间数据挖掘的目的在于在空间数据中挖掘出潜在知识, 并将其实际应用. 目前, 空间数据挖掘技术已经在大气治理 [2]、地理信息系统 (geographic information system, GIS) [3]、城市建设 [4] 等领域得到了广泛的应用.

空间同位 (co-location) 模式的挖掘是空间数据挖掘的一个重要方向. 所谓空间同位模式, 就是空间特征的子集, 并且它们的实例在空间中频繁相关. 例如, 鸡枞菌生长的地方一定有白蚁, 就可以得到

引用格式: 蒋希文, 王丽珍, Tran V. 基于模糊密度峰值聚类的区域同位模式并行挖掘算法. 中国科学: 信息科学, 2023, 53: 1281–1298, doi: 10.1360/SSI-2022-0004
Jiang X W, Wang L Z, Tran V. A parallel algorithm for regional co-location mining based on fuzzy density peak clustering (in Chinese). Sci Sin Inform, 2023, 53: 1281–1298, doi: 10.1360/SSI-2022-0004

空间同位模式 {鸡枞菌, 白蚁}. 传统的空间同位模式挖掘都是在全局的数据上进行的. 使用全局挖掘的方法挖掘出来的潜在知识往往只在全局范围内有意义. 基于现实问题, 有学者提出了区域空间同位模式挖掘. 通过区域空间同位模式挖掘, 可以发现存在于某些区域内的频繁空间同位模式. 当然, 这些模式的实际意义也只存在于该区域内, 而在全局范围中不一定成立. 例如, 某个城市的犯罪率总体不高, 但是可能在酒吧附近会较高 [5]; 公交车在始发站附近氮氧化物排放偏高, 这与汽车发动机未充分加热达到最佳工作状态相关 [6]. 类似的实例在社会治理、环境保护等领域的价值促使更多的学者在区域空间同位模式挖掘领域开展研究工作.

传统的区域空间同位模式挖掘致力于寻找一种几何图形作为区域. 但实际上区域不一定是类似于国界这样的硬划分的区域. 许多软划分的区域在生活和研究中扮演了重要的角色. 例如, 台风影响的区域除了我国东部沿海的几个省份外, 对于受到影响的内陆地区的范围却不应该使用明确的界限划定. 往往要采用一种软划分的方式对台风的影响进行评估, 从而在应急预案的有效性和对当地的经济影响中找到更加合理的平衡点. 在空间同位模式挖掘中, 传统邻近距离阈值将实例之间的关系硬划分成了两类: 有关系和无关系. 这样做会导致邻近距离阈值的设定成为一种挑战 [7]: (1) 过高的阈值会使实际上不存在的关系被认为存在; (2) 过低的阈值会导致原本存在的关系被丢失; (3) 不同的区域空间内的数据分布情况也往往不同, 统一的阈值显然不适合. 受到文献 [8] 的启发, 本文将模糊集理论引入区域划分中以对全局空间进行软划分. 聚类技术将一系列的实例划分成多个簇, 并且实例与处于相同簇内的实例更加相似, 与处于不同簇的实例则较为不相似. 因为空间数据往往分布不均匀, 若将数据分布情况定义为相似性, 引入聚类后即可完成数据分布不平衡情况下的区域划分.

基于上述背景和理论, 本文提出了一种新的区域同位模式挖掘算法. 首先利用改进后的密度峰值聚类算法确定簇中心实例, 再使用模糊集理论生成多个极大模糊簇, 这也就相当于进行了区域的划分. 由于模糊集理论的引入, 这种划分不是传统意义上的硬划分, 模糊隶属度使得划分的区域更加合理. 此外, 本文引入了 k 近邻距离并提出了新的密度度量函数, 这使得聚类算法效率得到提升的同时自适应性也得到加强. 最后, 对算法进行并行化处理, 并均衡各线程的任务. 在真实数据中进行的实验表明, 本文算法具有较高的效率和实际应用价值.

2 相关工作

2.1 空间同位模式挖掘

空间同位模式挖掘首先由 Morimoto [9] 提出. 文献 [9] 中定义了基于距离的模式, 并且模式实例的个数作为模式的频繁性度量, 此度量不满足反单调性. Shekhar 等 [10] 提出了以参与度作为频繁性度量, 这种度量满足反单调性, 算法也因此在效率上有显著的提升. Huang 等 [11] 又提出了一种基于模式实例的连接算法 (Join-based), 但是这种算法在大数据集上效率欠佳. 之后 Yoo 等 [12,13] 又提出了两种算法: 部分连接算法 (Partial-join) 和无连接算法 (Joinless). Partial-join 算法使用团划分模型物化邻近关系以减少子连接操作, 但是需要额外的空间用以存储切断的邻近关系. Joinless 算法使用星型实例进行剪枝操作以降低时间消耗. Wang 等 [14,15] 提出了基于前缀树的 CPI-Tree 算法及其改进算法 iCPI-Tree. 两者将空间邻近关系以树的形式进行存储, 但前缀树存储开销是该类算法的瓶颈. Huang 等 [16] 提出基于聚类的空间同位模式挖掘算法, 并以密度率作为聚类近似函数的衡量标准. 文献 [16] 还给出了近似函数的三项评价准则, 这对于之后的研究有着重要的意义. 文献 [8] 将模糊集理论引入到聚类中, 并且提出了新的相似函数用于聚类. 文献 [17] 引入 Delaunay 三角化生成实例的邻近关系,

再使用统计方法去除不恰当的邻近关系, 最后在生成的图上进行空间同位模式挖掘.

2.2 区域空间同位模式挖掘

区域空间同位模式挖掘主要分为两类方法: 基于区域划分的方法和基于适应度函数的聚类方法.

基于区域划分的方法根据算法指定的划分策略对全局空间进行划分, 如网格、四分树等. 在空间划分后得到的区域内进行挖掘, 通常这种挖掘使用的是全局空间同位模式使用的算法. Celik 等 [18] 用四分树划分全局域, 但是这样的做法需要大量的先验知识. 文献 [6] 提出了 QGFR 算法, 该算法使用最小正交包围矩形作为模式的区域, 但得到的区域形状固定为正交矩形, 且即使在带有剪枝策略的情况下, 算法时间复杂度依然极高. 在数据分布情况的研究中, Qian 等 [19] 提出了基于 k 近邻的划分. 这样的划分将全局空间划分成几个数据分布相似的区域, 并且降低了对先验知识的要求. 文献 [20] 使用 Delaunay 三角化方法将全局域以一种自适应的方式产生实例之间的邻近关系, 并且使用类似于 Apriori [21] 的方式迭代生成候选超模式及其候选区域. 该算法自适应地解决了数据分布不均匀情况下的区域划分问题. 文献 [22] 利用用户设定的网格对全局域进行划分, 再利用核函数衡量网格内各特征实例的分布情况进行挖掘. 但是以上方法挖掘出的区域都给定了明确的界线, 得到的都是硬划分的区域.

基于适应度函数的聚类方法将聚类问题转化为目标函数的最优化问题. 通常认为最优化的情况下, 所得的聚类也是最优的. 文献 [23] 通过使兴趣度函数最大以得到热区域和冷区域, 再通过在热点和冷点地区的关联规则挖掘及区域合并以发现最终的模式区域.

2.3 基于密度峰值的聚类算法

基于密度峰值的聚类算法根据以下事实进行聚类: (1) 簇中心点所处位置的密度高于其他点; (2) 簇中心点离其他簇中心点较远. 文献 [24] 据此对空间中的每个点计算两个数量值以进行聚类: 局部密度 ρ_i 以及相对距离 δ_i. 其表达式如下所示:

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \tag{1}$$

$$\delta_i = \min_{j:\rho_i<\rho_j} d_{ij}, \tag{2}$$

其中 d_{ij} 表示点 i 到点 j 的距离, d_c 为人为设定的阈值. $\chi(\cdot)$ 定义如下所示:

$$\chi(t) = \begin{cases} 1, & t < 0, \\ 0, & t \geqslant 0. \end{cases} \tag{3}$$

对于局部密度最大的点, 其 $\delta_i = \max_j(d_{ij})$.

得到每个点的 ρ_i 与 δ_i 值后, 聚类算法利用决策图选择簇中心点, 该过程就是选择 ρ_i 与 δ_i 同时较高的点作为簇中心点, 也就是尽可能选择靠近决策图右上方的点作为簇中心点. 而有低 ρ_i 值、高 δ_i 值的点则被认为是噪声点. 在确定完簇中心点后, 其余的非簇中心点选择最近的簇中心点进行聚类. DPC 算法拥有较好的时间效率, 其时间消耗大多发生在局部密度和相对距离的计算上. 但是, 给出的局部密度对于小数据集情况并不能客观衡量局部密度情况. 因此, 文献 [24] 还提出了一种基于高斯核函数的局部密度, 以改善算法在小数据集中的局部密度. 其表达式如下所示:

$$\rho_i = \sum_j \exp\left(-\frac{d_{ij}^2}{d_c^2}\right). \tag{4}$$

但评定数据集的大与小是一项因人而异的工作. 局部密度函数的选择也因此成为了一项没有客观标准的工作. 基于此, 文献 [25] 在 DPC-KNN 算法中提出了新的局部密度. 该密度度量使用 k 近邻距离作为密度度量的依据. 这不仅减少了局部密度度量的计算量, 而且在大小数据集上都可以较为准确地衡量局部密度. 其表达式如下所示:

$$\rho_i = \exp\left(-\frac{1}{k}\sum_{j\in \mathrm{KNN}_i} d_{ij}^2\right), \tag{5}$$

其中 k 为用户设定的值, KNN_i 为 i 点的 k 近邻点集.

文献 [26] 从物理学场理论获得启发, 提出利用势衡量数据点密度, 再通过不确定性度量自适应寻找最优势参数, 并以此生成簇中心点进行聚类. 文献 [27] 的 ADPC-KNN 算法中又提出了一种新的局部密度, 并且作者证明了该密度相较于文献 [24, 25] 更能突出局部密度的高低. 因为处于高密度位置的点的局部密度值会因此变得更高, 所以这有利于在决策图中寻找簇中心点. 其表达式如下所示:

$$\rho_i = \sum_{j\in \mathrm{KNN}_i} \exp\left(-\frac{d_{ij}^2}{d_c^2}\right). \tag{6}$$

式 (6) 虽然可以一定程度上加大最大最小局部密度的差值, 但是差值还有进一步的上升空间.

3 相关概念

3.1 空间同位模式挖掘与区域空间同位模式挖掘

本小节介绍空间同位模式挖掘的相关定义及算法将要使用到的一些概念.

空间特征指在空间中的不同事物类型, 如学校、医院等. 空间实例指具有空间属性的具体事物, 如具体到名字的某家医院. 空间实例因为拥有位置属性, 因此可以将其视作空间中带有特征属性的点. 空间特征集为空间内所有特征的集合, 记为 $F = \{f_1, f_2, \ldots, f_m\}$. 空间实例集为所有空间实例的集合, 记为 $I = \{s_1, s_2, \ldots, s_n\}$. 空间同位模式是空间特征集的子集, 并且其实例在指定空间 (全局或者某个区域内) 内频繁具有邻近关系. 实例之间的邻近关系通常根据数据的坐标系使用欧几里得距离或球面距离等方式进行衡量. 空间同位模式的频繁度 (PI, participation index) 用于衡量该模式在指定空间中的频繁强度. 当某个模式在指定空间内的 PI 值大于等于频繁度阈值时, 则认为该模式在对应空间内频繁. 通常使用的 PI 值由文献 [10] 提出的度量方法求得, 本文也使用该度量作为频繁度度量.

区域空间同位模式由两部分组成: 空间同位模式和模式的频繁区域, 记为 $P = (\mathrm{pattern}, c)$, 其中 $\mathrm{pattern} \subseteq F$, c 表示频繁区域.

3.2 基于 k 近邻的模糊密度峰值聚类

在 2.3 小节中总结了几种局部密度度量函数. 受文献 [27] 的启发, 本文提出了一种拥有新的局部密度度量的聚类方法, 并且将同样拥有点的空间性质的实例作为这种聚类的对象.

定义1 (实例的局部密度值)　对于给定空间上的实例集 I. 对于其中的每个实例 s_i, 其局部密度如下所示:

$$\rho_i = \sum_{j\in \mathrm{KNN}_i} R(d_{ij}), \tag{7}$$

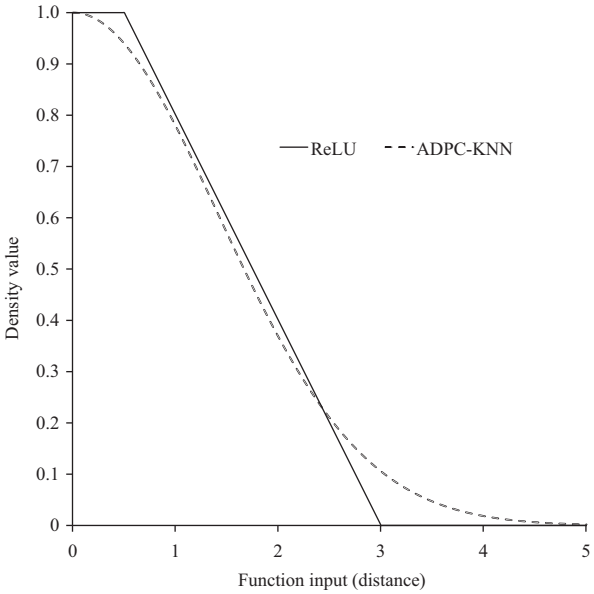

图 1 两种密度函数对比

Figure 1 Comparison of two density functions

其中 KNN_i 为 s_i 的 k 近邻实例集. $R(\cdot)$ 为一种整流线性单位函数 (ReLu, rectified linear unit), 其表达式如下所示:

$$R(d)=\begin{cases}1, & d\leqslant d_1,\\ 1-\frac{d-d_1}{d_2-d_1}, & d_1<d<d_2,\\ 0, & d\geqslant d_2,\end{cases} \tag{8}$$

其中, d_1, d_2 为两个距离阈值, 可以由用户设定.

图 1 对比了整流线性单位函数和 ADPC-KNN 算法使用的密度函数, 其中 $d_1=0.5$, $d_2=3$, $d_c=2$. 整流线性单位函数对于处于高密度位置的点能产生更高的密度值, 而对于处于低密度位置的点则降低了其密度值.

以实验部分将要使用的 "深圳 POI 数据集" 为例. 当 $k=150$ 时, 所有实例的第 k 近邻距离均值为 $\mu^k=657.406$, 标准差为 $S^k=448.563$. 取 $d_1=208.843$, $d_2=1105.969$, $d_c=1105.969$. 如图 2(a) 决策图所示, 在同时拥有较高局部密度和较高相对距离的实例中, 本文提出的 ReLU 方法生成的实例更位于决策图的右上方. 这也说明, 本文的方法认为这些实例作为簇中心实例的可能性更高. 此外, 图 2(b) 和 (c) 分别展示了两种算法得到的前 15 个最低和前 15 个最高局部密度的实例. ReLU 算法得到的低值低于 ADPC-KNN, 而高值则高于 ADPC-KNN. 两种算法在局部密度高低差值上存在差异: 使用本文定义的 ReLU 函数计算得出的最高和最低局部密度之差更大, 这更有利于识别簇中心实例 [27]. 因此, 本文定义的 ReLU 函数能更好地反映各实例的局部密度.

定义2 (模糊簇中心实例的选择) 在实例集 I 上, 簇中心实例集 $\mathrm{CC}=\{s_i|s_i\in I\wedge\delta_i\geqslant d_c\}$, 即簇中心实例集由实例集 I 中相对距离大于等于人为设定阈值 d_c 的实例构成. 剩余实例集 I 中的实例被称为非簇中心实例, 其集合表示为 $\mathrm{NCC}=\{s_i|s_i\in I\wedge s_i\notin \mathrm{CC}\}$. 为了方便区分两个集合中的实例, 本文 CC 中的实例以 c 表示, 即 $\mathrm{CC}=\{c_1,c_2,\ldots\}$; NCC 中的实例以 o 表示, 即 $\mathrm{NCC}=\{o_1,o_2,\ldots\}$.

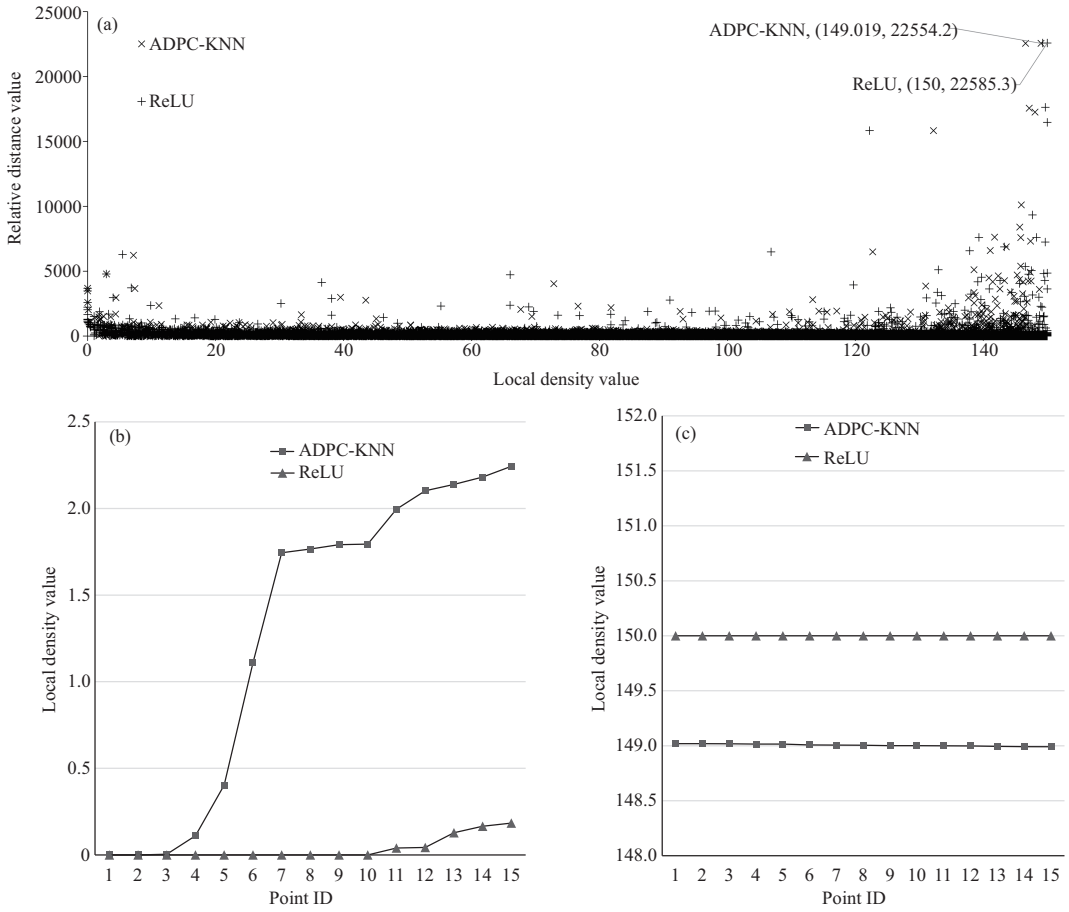

图 2 (a) 决策图; (b) 两种方法生成的前 15 个最低局部密度; (c) 两种方法生成的前 15 个最高局部密度

Figure 2 (a) The decision graph; (b) top-15 lowest local densities generated by two methods; (c) top-15 highest local densities generated by two methods

定义3 (对簇的模糊隶属度) 非簇中心实例 o_i 对于以 c_j 为中心实例的簇的模糊隶属度 u_{ij} 如下所示:

$$u_{ij} = \sum_{q=1}^{l_1} \left(\frac{d_{ij}}{d_{iq}} \right)^{-\frac{2}{r-1}}, \tag{9}$$

其中 r 为平滑参数, 本文始终设置其为 2; d_{ij} 为非簇中心实例 o_i 到簇中心实例 c_j 的距离; l_1 为簇中心实例个数.

其合理性说明如下: 定义目标函数 J, 并最小化目标函数. 目标函数 J 的表达式如式 (10) 所示:

$$J = \sum_{i=1}^{l_2} \sum_{j=1}^{l_1} u_{ij}^r d_{ij}^2, \quad \text{s.t.} \begin{cases} \sum\limits_{j=1}^{l_1} u_{ij} = 1, \\ 0 \leqslant u_{ij} \leqslant 1, \\ 1 \leqslant j \leqslant l_1, \\ 1 \leqslant i \leqslant l_2. \end{cases} \tag{10}$$

对上述最优化问题使用拉格朗日乘数法 (Lagrange multiplier) 即可求得式 (9), 其中 l_1 为簇中心实例个数, l_2 为非簇中心实例个数.

定义4 (λ 隶属关系) 若非簇中心实例 o_i 对于以 c_j 为中心实例的簇 C_j 的隶属度 $u_{ij} \geqslant \lambda$, 则称 o_i 与簇 C_j 为 λ 隶属关系, 记作 $\lambda(o_i, C_j)$.

定义5 (极大模糊簇) 若实例集 I' 满足 $I' \subseteq \mathrm{NCC}$ 且对 $\forall p_i \in I'$, $\lambda(p_i, C_j)$ 成立, 则 $I' \cup \{c_j\}$ 称为以 c_j 为中心实例的 λ 模糊簇. 若不存在 I'', 满足 $I' \subset I''$ 且 I'' 为以 c_j 为中心实例的 λ 模糊簇, 则称 $I' \cup \{c_j\}$ 为 λ 极大模糊簇, 简称极大模糊簇.

4 FDPC-RCPM 算法

本文提出的基于模糊密度峰值聚类的区域空间同位模式挖掘算法 (FDPC-RCPM) 首先根据第 3.2 小节定义的改进的密度峰值聚类算法生成簇中心实例, 再根据每个实例的模糊隶属度生成 λ 极大模糊簇, 最后在 λ 极大模糊簇上进行空间同位模式挖掘. 挖掘的算法采用传统的空间同位模式挖掘算法在每个极大模糊簇上进行. 其伪代码如算法 1 所示.

Algorithm 1 FDPC-RCPM

Input: d_1, d_2, d_c, d_j, λ, k, minPI, I;
Output: P;

```
 1: CC ⇐ ∅;
 2: KNN = CalculateKNN(I, k);
 3: ρ ⇐ CalculateLocalDensity(I, d1, d2);
 4: δ ⇐ CalculateRelativeDistance(I, dc);
 5: for si ∈ I do
 6:     if δi ⩾ dc then
 7:         CC ⇐ CC ∪ si;
 8:     end if
 9: end for
10: u ⇐ {uij | (1 ⩽ i ⩽ |I \ CC|) ∧ (1 ⩽ j ⩽ |CC|)};
11: for oi ∈ I \ CC, cj ∈ CC do
12:     if uij ⩾ λ then
13:         Fj ⇐ Fj ∪ {oi};
14:     end if
15: end for
16: F ⇐ F ∪ {Fj ∪ {cj}};
17: P ⇐ ∅;
18: for Fj ∈ F do
19:     P ⇐ P ∪ (Joinless(Fj, minPI, dj), Fj);
20: end for
```

算法 1 首先初始化存储簇中心实例的集合 CC (行 1) 并计算每个实例的 k 近邻 (行 2). 再根据 k 近邻计算每个实例的局部密度, 并存储在集合 ρ 中 (行 3). 得到每个实例的局部密度后, 每个实例再寻找最近的密度大于自身的实例并计算与其最小距离以作为相对距离 (行 4). 如果某个实例的相对距离大于阈值 d_c, 那么认为该实例为簇中心实例, 将其放入集合 CC (行 5~9). 之后分别计算每个非簇中心实例对每个簇的隶属度 (行 10). 若某个非中心实例 o_i 对于以簇中心实例 c_j 为中心的簇的隶属度大于等于 λ, 则将非中心实例 o_j 放入到该簇对应的模糊簇 F_j 中 (行 11~15). 最后, 对每个极大模

糊簇进行空间并置模式挖掘, 这里使用 Joinless 算法进行挖掘. Joinless 使用的邻近距离阈值与频繁度阈值使用人为设定的 d_j 和 minPI. 每个极大模糊簇上的频繁模式与该极大模糊簇形成的元组构成了区域空间并置模式 (行 18∼20).

为了使算法具有较高的自适应性, 本文对改进密度峰值聚类算法的 3 个参数 d_1, d_2, d_c 使用式 (11) 和 (12) 自适应生成:

$$d_1 = \max(0, \mu^k - S^k), \tag{11}$$

$$d_2 = d_c = \mu^k + S^k. \tag{12}$$

其中 μ^k 为每个实例的第 k 近邻实例与该实例距离的均值, S^k 为上述距离的标准差. 对空间同位模式挖掘的距离阈值 d_j 和模糊聚类参数 λ 使用式 (13) 和 (14) 自适应生成:

$$d_j = \frac{1}{|F_j|} \sum_{s_i \in F_j} u_{ij} d_{ij}, \tag{13}$$

$$\lambda = \frac{1}{|\mathrm{CC}|}, \tag{14}$$

其中, u_{ij} 与 d_{ij} 分别为实例 s_i 对簇 C_j 的隶属度和到簇中心实例的距离.

式 (11) 和 (12) 中标准差的引入使得其在损坏的数据集上更具有鲁棒性 [27], 并且也可减少制图时离散连续空间带来的误差. 另外, 如果数据集满足正态分布, 在均值前后 1 个标准差的范围内有超过 68% 的数据落入, 也就是说大多数第 k 近邻距离将落入 d_1 与 d_2 之间, 因此式 (11) 和 (12) 是合适的. 式 (13) 自适应生成邻近距离阈值, 从而避免统一阈值带来的各类问题, 并且平均加权距离也反映了极大模糊簇内实例的分布情况. 式 (14) 保证每个实例都至少属于一个极大模糊簇. 因为考虑到异常点也可能是有价值的信息, 所以不将其舍去, 其包含信息的价值高低应在空间同位模式挖掘过程中进行评估.

5 并行化改进 (FDPC-PRCPM)

在 FDPC-RCPM 聚类过程中, 时间开销主要集中在每个实例的 k 近邻的搜索过程中, 但每个实例的搜索过程相互独立. 在挖掘区域同位模式时, FDPC-RCPM 在每个簇上执行一次空间同位模式挖掘算法, 该过程也是互相独立进行的. 因此本文提出将 FDPC-RCPM 并行化以提升效率.

5.1 聚类中的并行化

在 FDPC-RCPM 中, 聚类的时间开销主要集中在 k 近邻搜索过程中. 对于线性查找法, 该过程即使使用网格进行优化, 也可能因为实例分布的原因导致优化效果不佳. 例如实例集中分布在某一较小区域中, 使用网格进行优化也会因为落在单元外的实例过少减弱优化效果. 若此时减小网格的大小, 又有可能出现实例分散分布情况, 这就会导致需要搜索的单元数快速增加. 因此, 本文采用线性查找法搜索 k 近邻并将该过程并行化.

以同样的思路, 局部密度 ρ 的计算、相对距离 δ 的计算、簇中心实例的选择、实例对簇的隶属度 u 的计算以及极大模糊簇的生成都可以并行化处理, 因为这些过程均具有独立性, 计算过程互不干扰.

表 1 Joinless 算法运行时间及主存峰值

Table 1 Execution time and memory usage peak of the Joinless

Number of instances	Average runtime (s)	Average memory peak (KB)
1000	12.2	1183.2
2000	40.2	1880
5000	103.2	5749.6
10000	361	10477.6
20000	1884	33484.8
Correlation coefficient	0.96	0.98

5.2 空间同位模式挖掘过程的并行化

虽然各模糊簇的空间同位模式挖掘过程均相互独立, 但空间同位模式挖掘的过程极其消耗计算、存储资源, 因此不可与聚类过程中的并行化作同样处理. 该过程的并行化不仅仅需要考虑线程的负载均衡, 还需要考虑主存的消耗问题.

FDPC-PRCPM 使用 Joinless [13] 算法作为传统空间同位模式挖掘算法. Joinless 算法主要有 5 个挖掘过程: (1) 星型邻居关系生成; (2) 从星型邻居关系中过滤出候选模式的星型实例; (3) 根据星型实例计算候选模式的粗频繁度并以此剪枝; (4) 从剩余候选模式的星型实例中过滤出其团实例; (5) 计算剩余候选模式的频繁度并生成频繁模式.

文献 [13] 指出, Joinless 算法的时间消耗主要在于过滤候选模式星型实例和从星型实例中过滤出团实例两个过程中. 过滤候选模式星型实例的时间消耗与星型邻居关系的数量相关, 而随着实例个数增加, 必定使得星型邻居关系数量的增加, 这导致 Joinless 算法的运行时间的增加. 另外, 星型邻居关系的增加也往往导致候选模式的星型实例数增加, 过滤团实例的工作量上升, 因此算法的运行时间也相应上升.

Joinless 算法的空间消耗主要在于对星型邻居关系和候选模式星型实例的存储. 实例数的增加必然导致星型邻居关系和候选模式星型实例的增加, 因此实例数的增加也使得主存开销的增加.

此外, 本文还使用文献 [13] 提出的人工数据集生成方法随机生成了 100 组数据, 每组数据控制实例个数, 分别为 1000, 2000, 5000, 10000 以及 20000 个, 拥有特征个数固定为 20. 使用 Joinless 算法在上述人工数据集上进行挖掘, 并且对挖掘时主存峰值及算法运行时间进行统计. 最终使用两者的平均值分别与实例个数计算相关系数. 其结果如表 1 所示. 通过观察发现, Joinless 算法的平均运行时间及平均内存峰值与实例个数均存在强正相关性.

综上, 空间同位模式挖掘算法的主存消耗量和算法运行时长与数据实例个数呈正相关. 从而, 有如下线程任务分配问题:

有簇集 $C=\{c_1,c_2,\ldots,c_m\}$, $|c_i|$ 表示簇内实例个数. 将簇集分为 t 个子集, 即 $\{C_1,C_2,\ldots,C_t\}$, 分别对应每个线程的任务, $|C_i|$ 表示子集内所有簇的实例个数之和. 且对于 $1\leqslant i,j\leqslant t$, $C_i\cap C_j=\emptyset$, $\cup_{i=1}^{t}C_i=C$, C_i 计算时长为 T_i. 现使得目标函数 J 最小, J 的定义为

$$J=\sum_{i=1}^{k}\sum_{j=i+1}^{k}|T_i-T_j|=\sum_{i=1}^{t}\sum_{j=i+1}^{t}|\Gamma(|C_i|)-\Gamma(|C_j|)|, \tag{15}$$

且满足同一时间点 t 个正在执行的任务主存消耗量接近主存总容量. Γ 为任务量到同位模式挖掘算法运行时间的映射函数.

Algorithm 2 FDPC-PRCPM

Input: t, KNN_i, k, minPI, I;
Output: P;
1: $\mathrm{CC} \Leftarrow \emptyset$;
2: **parallel** $\mathrm{KNN} \Leftarrow \mathrm{CalculateKNN}(I, k)$;
3: Calculate d_1, d_2, d_c, λ;
4: **parallel** $\rho \Leftarrow \mathrm{CalculateLocalDensity}(I, d_1, d_2)$;
5: **parallel** $\delta \Leftarrow \mathrm{CalculateRelativeDistance}(I, d_c)$;
6: **for parallel** $s_i \in I$ **do**
7: **if** $\delta_i \geqslant d_c$ **then**
8: $\mathrm{CC} \Leftarrow \mathrm{CC} \cup s_i$;
9: **end if**
10: **end for**
11: **parallel** $u \Leftarrow \{u_{ij} \mid (1 \leqslant i \leqslant |I \setminus \mathrm{CC}|) \wedge (1 \leqslant j \leqslant |\mathrm{CC}|)\}$;
12: **for parallel** $o_i \in I \setminus \mathrm{CC}, c_j \in \mathrm{CC}$ **do**
13: **if** $u_{ij} \geqslant \lambda$ **then**
14: $F_j \Leftarrow F_j \cup \{o_i\}$;
15: **end if**
16: **end for**
17: $F \Leftarrow F \cup \{F_j \cup \{c_j\}\}$;
18: $P \Leftarrow \emptyset$;
19: $\mathrm{TASK} = \mathrm{TaskDivide}(F, t)$;
20: $M =$ current available memory;
21: **for parallel** $\mathrm{TASK}_t \in \mathrm{TASK}$ **do**
22: **for** $F_j = \mathrm{TASK}_t.\mathrm{pop}(M)$ **do**
23: $M -=$ Amount of memory F_j requires;
24: Calculate neighboring distance threshold d_j;
25: $P \Leftarrow P \cup (\mathrm{Joinless}(F_j, \mathrm{minPI}, d_j), F_j)$;
26: $M +=$ Amount of memory F_j requires;
27: **end for**
28: **end for**

J 的最优化问题被称为 0/1 多背包问题 (zero/one multiple knapsack problem), 该问题是一类 NP 完全问题 [28]. 但是本文的目的是均衡线程的任务负担以达到减少算法总时间消耗, 若在该问题求解上消耗大量时间则会背离本意. 所以本文使用贪心策略以期快速将任务划分, 而非花费大量时间求得准确的划分. 并行化后的 FDPC-RCPM 算法 (FDPC-PRCPM) 如算法 2 所示. 其中 t 为并行线程数.

算法 2 首先初始化存储簇中心实例的集合 CC (行 1), 并计算每个实例的 k 近邻距离 (行 2). 之后再根据式 (11), (12) 和 (14) 计算密度峰值聚类所需的参数 d_1, d_2, d_c, λ, 局部密度以及相对距离 (行 3∼5). 根据定义 2 寻找簇中心实例并将其放入集合 CC (行 6∼10) 中, 该过程也为并行过程. 计算每个非簇中心实例对每个簇的隶属度 (行 11), 得到模糊簇 F_j (行 12∼16). 行 19 根据线程数 t 对任务进行划分, 划分算法如算法 3 所示. 临界资源 M 用于存储主存的可用容量 (行 20). 划分完任务之后, 每个线程都各有属于自己的任务清单 TASK_t. 每个线程根据记录当前可用内存大小的临界资源 M 决定接下来执行任务清单中的哪个任务 (行 22). 行 25 对每个极大模糊簇在对应的线程上进行空间同位模式挖掘, 这里使用 Joinless [13] 算法进行挖掘. Joinless 的邻近距离阈值使用根据式 (13) 自适应生成的 d_j, 人为设定频繁度阈值 minPI. 每个极大模糊簇上的频繁模式与该极大模糊簇形成的元组构成了

区域空间同位模式, 所有区域空间同位模式存储在结果集 P 中.

算法 3 使用贪心策略, 每次将模糊簇分配给当前任务最轻的线程, 以此达到尽可能的均衡.

Algorithm 3 TaskDivide

Input: F, t;
Output: TASK;
1: $B_{1,2,\ldots,t} \Leftarrow 0$;
2: **for** $f \in F$ **do**
3: $\quad i \Leftarrow \text{argmin}(B)$;
4: $\quad \text{TASK}_i \Leftarrow \text{TASK}_i \cup \{f\}$;
5: $\quad B_i \mathrel{+}= |f|$;
6: **end for**
7: $\text{TASK} = \{\text{TASK}_i \mid 1 \leqslant i \leqslant t\}$.

6 复杂度分析

本节将分析 FDPC-RCPM 及 FDPC-PRCPM 在时间与空间上的复杂度. 以下均假设数据集拥有 n 个实例、m 个极大模糊簇, 考虑 k 个近邻. FDPC-PRCPM 拥有 t 个线程.

6.1 FDPC-RCPM

FDPC-RCPM 算法计算 k 近邻实例需要 $O(n^2)$ 的时间复杂度, 计算局部密度需要 $O(kn)$ 时间复杂度, 计算相对距离需要 $O(n^2)$ 时间复杂度. 确定簇中心实例需要 $O(n)$ 时间复杂度. 计算非簇中心实例对于每个模糊簇的隶属度需要 $O((n-m)m^2)$. 生成极大模糊簇需要 $O((n-m)m)$. 每个簇上进行挖掘的时间复杂度取决于使用的空间同位模式挖掘算法. 这里假设 Joinless 算法在每个簇上的平均时间复杂度 T_{jl}. 因此 FDPC-RCPM 总的时间复杂度为 $O(n^2 + kn + (n-m)m^2 + mT_{jl})$.

FDPC-RCPM 算法的空间消耗主要发生在存储 k 近邻、局部密度、相对距离、模糊隶属度以及空间同位模式挖掘几个过程中. 存储 k 近邻的空间复杂度为 $O(kn)$. 存储局部密度、相对距离的空间复杂度均为 $O(n)$. 存储模糊隶属度需要的空间复杂度为 $O((n-m)m)$. 空间同位模式挖掘算法 Joinless 在每个簇上的平均空间复杂度以 M_{jl} 表示. 因此 FDPC-RCPM 算法的空间复杂度为 $O(kn + (n-m)m + mM_{jl})$.

6.2 FDPC-PRCPM

FDPC-PRCPM 算法的加速比在后续 $|I| = 100000$, $|F| = 5$, $k = 100$ 的实验中 2 线程验证约为 1.89, 16 线程约为 6.51.

平均空间复杂度除了空间同位模式挖掘过程尽最大可能占用了主存大小的策略外, 大致与 FDPC-RCPM 相同. 但该过程的差异并不产生额外的空间消耗, 只是在时间上尽最大可能提前使用.

7 实验结果

本文将围绕各参数对算法结果的影响和在真实数据集中的挖掘进行实验.

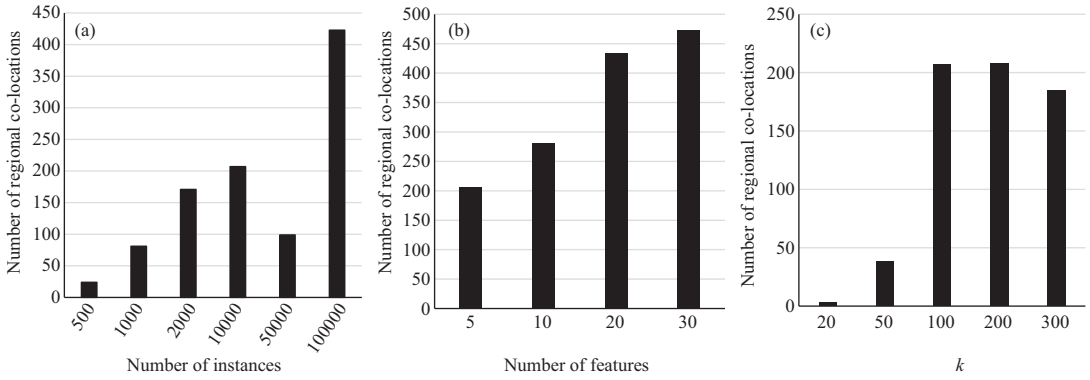

图 3　各参数对算法运行结果的影响

Figure 3　Effect of parameters on the results. (a) The number of instances; (b) the number of features; (c) k.

7.1　参数对算法的影响

本小节主要论述输入实例个数 $|I|$、特征个数 $|F|$、近邻数 k 以及线程数 t 对算法结果和运行时间的影响. 由于 FDPC-PRCPM 的任务分配算法使用尽量占满主存空间的策略, 因此对于主存空间的消耗不予考虑. 使用的数据集是以 2020 年芝加哥犯罪数据集[1)]为基础, 随机抽样得到的.

7.1.1　对结果的影响

显而易见, 算法的并行化不会对挖掘区域空间同位模式的结果造成影响. 因此本文只考虑输入实例个数 $|I|$、特征个数 $|F|$、近邻数 k 对算法结果造成的影响. 通过控制参数变量进行对比实验以完成各参数对算法结果影响的观察.

若输入实例个数作为变量, 控制特征个数 $|F|=5$、近邻数 $k=100$、频繁度阈值 minPI 为 0.5, 结果如图 3(a) 所示. 图 3(a) 中, 有 50000 个输入实例时, 区域空间同位模式个数变小. 这是因为实例个数的增加使得数据分布发生变化, 极大模糊簇数量及簇内实例分布发生改变, 挖掘结果的数量也由此有了非单调增长的可能.

图 3(b) 展示了输入实例的特征个数对区域空间同位模式的结果数量的影响. 该实验控制实例个数 $|I|$ 为 10000, $k=100$, 频繁度阈值 minPI 为 0.5. 随着输入特征个数的增加, 挖掘出的区域空间同位模式个数呈现增长的态势. 这是因为在不改变实例个数和分布的情况下, 特征数的增加只会导致邻近关系个数的增加 (原本拥有同特征的实例无法邻近, 现在却因为特征不同而可以有邻近关系), 而极大模糊簇的大小及含有的实例个数并不会发生改变. 特征数的增加因邻近关系的增加带来了更多的参与实例并且原先特征拥有的实例数变少, 这些变化都可能会使模式的参与度变大, 更多的区域空间同位模式也因此随着特征数的增加而产生.

图 3(c) 展示了邻近数 k 对区域空间同位模式的结果数量的影响. 该实验控制实例个数 $|I|$ 为 10000, 特征数量 $|F|$ 为 5, 频繁度阈值 minPI 为 0.5. 邻近数 k 的增加, 一方面导致自适应参数 d_c 的增加, 这往往导致极大模糊簇数量的减少. 另一方面, 由于极大模糊簇数量的减少, 单个极大模糊簇的平均大小增加, 每个极大模糊簇内的区域空间同位模式数量也因此在增长. 两方面的共同制约使区域空间同位模式数量可能如图 3(c) 所示的: 随着 k 增大而增大后减少.

1) https://data.cityofchicago.org/.

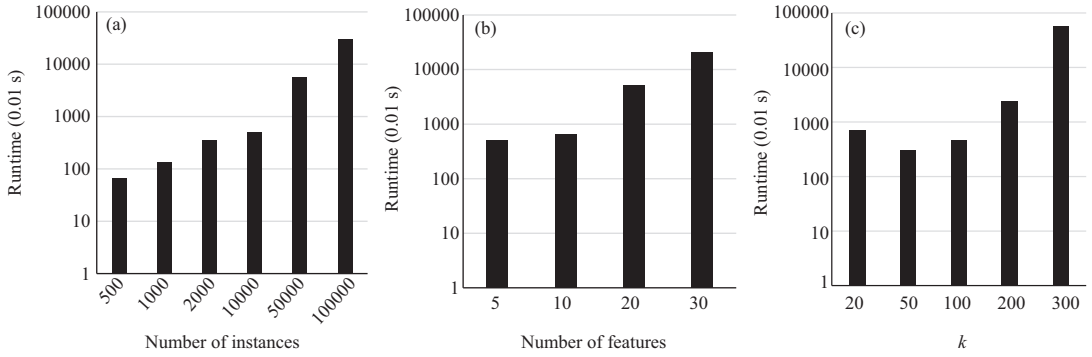

图 4 各参数对算法运行时间的影响

Figure 4 Effect of parameters on execution time. (a) The number of instances; (b) the number of features; (c) k.

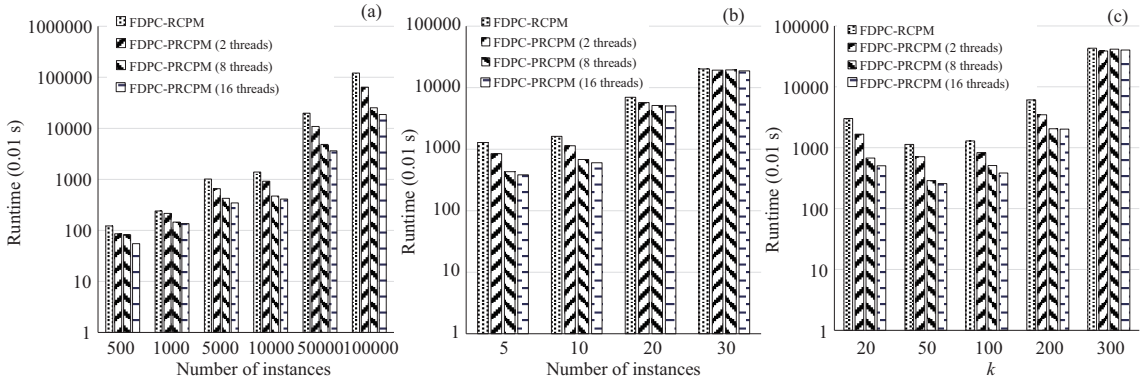

图 5 控制不同参数时非并行算法与并行算法的运行时间. (a) 控制参数为 $|I|$; (b) 控制参数为 $|F|$; (c) 控制参数为 k

Figure 5 Runtime of the non-parallel algorithm and parallel algorithm under different control parameters. (a) The control variable is $|I|$; (b) the control variable is $|F|$; (c) the control variable is k

7.1.2 对运行时间的影响

接下来探讨各参数对于算法运行时间的影响. 分别考虑输入实例个数 $|I|$、特征个数 $|F|$、邻近数 k 以及线程数 t 对运行时间的影响. 依然通过控制参数变量进行对比以达到实验目的. 得到以下结果使用的计算机配置为: Intel(R) Core(TM) i7-7820X 8 核 16 线程、64 GB 内存.

本文先对输入实例个数 $|I|$ 进行实验. $|F|$ 设定为 5, k 设定为 100, 频繁度阈值为 0.5, 线程数 t 设置为 8. 其结果如图 4(a) 所示. 对特征数 $|F|$, 设定 $|I|$ 为 10000, $k = 100$, 频繁度阈值为 0.5, 线程数为 8. 其结果如图 4(b) 所示.

对近邻数 k, 设定 $|I|$ 为 10000, $|F|$ 为 5, 频繁度阈值为 0.5, 线程数为 8. 结果如图 4(c) 所示. 在 k 较小时出现了算法运行时间随着 k 的增加而减小的情况, 其原因是在 k 较小时, 多线程额外的时间消耗相对于节省的时间消耗过于庞大, 而随着 k 的进一步增大, 额外的时间消耗就会显得微不足道了.

图 5 展示了 FDPC-RCPM 和其并行化版本 FDPC-PRCPM 在控制不同参数时的运行时间消耗. FDPC-PRCPM 的线程数分别设置为 2, 8 和 16, 当参数为非当前控制参数时, 设置 $|I| = 10000$, $|F| = 5$, $k = 100$.

表 2　案例分析使用的数据集

Table 2　The datasets used in case study

Dataset	Number of instances	Number of features	Distribution
Three Parallel Rivers	3879	16	Even
Shenzhen POI	71606	13	Clustered

从图 5 中可以得出, 并行化算法 FDPC-PRCPM 相较于非并行化算法 FDPC-RCRM 在运行时间上总有优势. 而深入分析优势程度可以发现: 随着线程数的成倍数增加, 算法的运行时间并未出现成倍数的减少. 究其原因, 一是不同的邻近距离阈值会物化不同数量的邻近关系, 邻近关系数量上的变化也会导致模式团实例数量的变化, 从而影响了同位模式挖掘过程; 二是 FDPC-PRCPM 使用的任务分配算法解决多背包问题采用的是贪心策略, 非最优解策略, 这导致每个线程的任务非最均衡; 三是 FDPC-PRCPM 算法的任务分配是预分配的, 每条线程在其任务清单内选择最优执行, 这就导致了耗时最长的线程决定了算法的运行时间. 如某线程只需要在一个簇上进行挖掘, 但需要大量时间, 而其他线程任务均衡, 但运行时间较短, 这就导致前者的运行时间决定了程序的运行时间. 第三点尤其在图 5(b) 的 $|F| = 30$ 和图 5(c) 的 $k = 300$ 时明显得到体现, 两组实验的加速效果相对不明显 (但仍然具有时间上的优势).

7.2　案例分析

本小节将在真实数据上运行 FDPC-PRCPM 算法, 并且将运行结果与全局空间同位模式挖掘算法及其他区域空间同位模式挖掘算法的挖掘结果进行对比. 本文采用两类数据集进行实验 —— 分布较为均匀的三江并流数据集以及分布呈簇状的深圳 POI 数据集. 数据集信息如表 2 所示.

7.2.1　与全局空间同位模式挖掘算法对比

为了更好地比较全局空间同位模式挖掘算法与本文提出的算法结果, 全局空间同位模式挖掘算法的邻近距离阈值将被设定为不小于所有极大模糊簇的最大值.

在三江并流数据集上, 设定近邻数 k 为 100, 频繁度阈值为 0.2, 全局空间同位模式算法的邻近距离阈值为 1670, 大于所有极大模糊簇的最大邻近距离阈值 1666.52. 本文提出的 FDPC-PRCPM 挖掘出了无法存在于全局空间同位模式的区域模式 65 个.

在深圳 POI 数据集上, 设定近邻数 k 为 150, 频繁度阈值为 0.6, 全局空间同位模式算法的邻近距离阈值为 250, 大于所有极大模糊簇的最大邻近距离阈值 248.31. 本文提出的 FDPC-PRCPM 挖掘出了无法存在于全局空间同位模式的区域模式 6 个.

7.2.2　与其他区域空间同位模式挖掘算法对比

本文提出一种衡量算法挖掘结果新颖程度的度量以便客观衡量挖掘结果质量, 其表达式如下所示:

$$D_i = \frac{|\mathrm{Result}_i \setminus \bigcup_{j=1,j\neq i}^{A} \mathrm{Result}_j|}{|\bigcup_{j=1}^{A} \mathrm{Result}_j|}, \tag{16}$$

其中 D_i 为第 i 种算法挖掘结果的新颖性度量, Result_i 为第 i 种算法挖掘得到的空间同位模式集合, A 为算法个数, 在本小节恒为 4. 本文提出的新颖性度量可以用于衡量某种算法的挖掘结果与其他算法挖掘结果的差异程度, 值越大说明越能挖掘出其他算法不能得到的结果, 其值域介于 0 与 1 之间.

表 3 三江并流数据集挖掘结果

Table 3 Mining results on the "Three Parallel Rivers" dataset[a)]

Algorithm	Number of regional co-locations	Novelty value	Runtime (s)
FDPC-PRCPM	**116**	**0.22**	**1.07**
QGFR-quick	93	0.14	105.59
Ref. [19]	10	0	5.63
Multi-level	70	0.18	6.11

a) 粗体表示最佳结果.

表 4 深圳 POI 数据集挖掘结果

Table 4 Mining results on the "Shenzhen POI" dataset[a)]

Algorithm	Number of regional co-locations	Novelty value	Runtime (s)
FDPC-PRCPM	38	0.08	**69.61**
QGFR-quick	**77**	**0.38**	259843.51
Ref. [19]	4	0	1250.15
Multi-level	15	0	2651.42

a) 粗体表示最佳结果.

在 2.2 小节中简单介绍了文献 [6] 提出的 QGFR 算法、文献 [19] 提出的区域同位模式挖掘算法及文献 [20] 提出的多层 (multi-level) 挖掘方法. 文献 [6] 提出的 QGFR 方法复杂度与实例个数相关且极高 [6], 在实验中我们无法在 24 小时内获得 QGFR 算法的全部挖掘结果. 因此本文对 QGFR 算法作出改进: 只要可以挖掘任一空间同位模式的一块频繁区域, 即判定其能够挖掘出该空间同位模式. 这样可以在可观时间内不遗漏空间同位模式, 从而可以衡量算法在数据集上挖掘结果的新颖度. 本文使用上述 3 种算法的挖掘结果与本文算法的结果进行对比.

在三江并流数据集上, 设定 FDPC-PRCPM 的参数 k 为 100, 频繁度阈值为 0.2, 线程数为 8, 算法表现如表 3 所示. FDPC-PRCPM 在结果新颖度与运行时间上均有良好的表现.

在深圳 POI 数据集上, 设定 FDPC-PRCPM 的参数 k 为 150, 频繁度阈值为 0.6, 线程数为 8, 算法表现如表 4 所示. 由于 QGFR 的改进算法使用枚举正交矩形区域的方式, 该方式可以发现所有在某个正交矩形区域内频繁的区域同位模式, 因此其拥有极高的新颖度. 但是这种枚举的复杂度极高, 如果某个模式拥有庞大的实例个数而不存在可用正交矩形表示的频繁区域, 那么将会消耗大量时间用于枚举.

结果表明, 本文提出的算法拥有较好的时间效率并且结果相对较新颖.

7.2.3 挖掘结果合理性

本小节将对 FDPC-PRCPM 算法在两个数据集上的挖掘结果进行合理性说明.

FDPC-PRCPM 算法在三江并流数据集上得到了如图 6(a) 的某个极大模糊簇. 在该极大模糊簇上发现了模式 {干热灌丛, 河流}, 其实例分布如图 7(a) 所示. 干热灌丛是典型的干热河谷地区植被 [29]. 同时具备干、热的河谷地区的区域称为干热河谷, 即河流是干热灌丛的必要条件, 而河流存在的区域并不一定满足干热条件. 另外, 我们还发现了模式 {温性针叶林, 针阔混交林}, 如图 7(b) 所示. 针阔混交林是温带地区的针叶林与落叶阔叶林的过渡类型, 两者很可能同位共存 [30], 因此为空间同位模式是合理的.

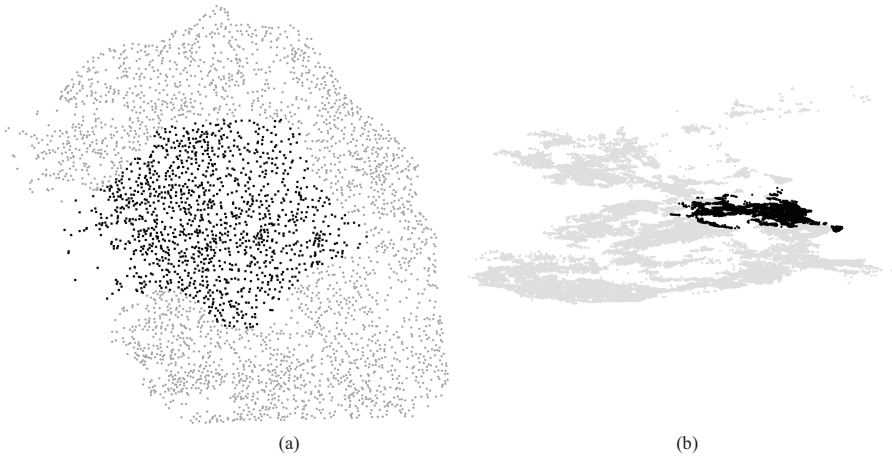

图 6　在真实数据集 (a) 三江并流和 (b) 深圳 POI 上存在的某个极大模糊簇

Figure 6　One of the maximal fuzzy clusters on the real datasets. (a) The Three Parallel Rivers dataset; (b) the Shenzhen POI dataset.

图 7　(网络版彩图) 图 6 所示的极大模糊簇中存在的空间同位模式的分布情况. (a) 和 (b) 为图 6(a) 中存在的两个空间同位模式; (c)∼(e) 为图 6(b) 中存在的三个空间同位模式

Figure 7　(Color online) Two co-location patterns' distribution in the maximal fuzzy clusters shown in the Figure 6. (a) and (b) Displays of two co-location patterns existing in Figure 6(a); (c)∼(e) displays of three co-location patterns existing in Figure 6(b)

在深圳 POI 数据集上, 算法挖掘得到了如图 6(b) 的某个极大簇. 在该极大模糊簇上有模式 {公司, 停车场}、{超市, 停车场} 以及 {家居市场, 停车场} 3 个区域空间同位模式, 如图 7(c)∼(e) 所示.

这符合生活常识, 因此可以认为上述区域空间同位模式合理.

8 结论

本文提出了一种基于模糊密度峰值聚类的区域空间同位模式挖掘算法. 该算法使用改进密度度量后的密度峰值聚类, 寻找簇中心实例, 再使用模糊集理论对非簇中心实例进行划分. 与传统的硬划分不同, 使用基于模糊集理论的划分使得每个实例可能属于多个簇, 这样划分的簇也更加符合实际. 基于得到的簇, 本文使用并行算法, 在多个簇上并行挖掘空间同位模式使得算法的效率得到提升.

本文提到的实例均为点实例, 即对象的地理空间信息仅有一个坐标点. 但实际问题中的实例不一定是以点的形式存在的. 例如城市的道路与建筑都以线段的形式或者多边形的形式存在. 这些不以点的形式存在的空间实例对象被称为扩展空间对象. 如何在扩展空间对象上挖掘区域空间同位模式将成为我们接下来要思考的问题. 此外, 我们还将在区域同位模式的评价上进行研究, 以期提出更好的评价方法用于评价挖掘结果.

参考文献

1 Li D R, Wang S L, Li D Y. Spatial Data Mining: Theory and Application. Berlin: Springer, 2015. 4–6
2 Akbari M, Samadzadegan F, Weibel R. A generic regional spatio-temporal co-occurrence pattern mining model: a case study for air pollution. J Geogr Syst, 2015, 17: 249–274
3 Kim S K, Lee J H, Ryu K H, et al. A framework of spatial co-location pattern mining for ubiquitous GIS. Multimed Tools Appl, 2014, 71: 199–218
4 Yao X J, Chen L J, Peng L, et al. A co-location pattern-mining algorithm with a density-weighted distance thresholding consideration. Inf Sci, 2017, 396: 144–161
5 Mohan P, Shekhar S, Shine J A, et al. A neighborhood graph based approach to regional co-location pattern discovery: a summary of results. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, 2011. 122–132
6 Li Y, Shekhar S. Local co-location pattern detection: a summary of results. In: Proceedings of the 10th International Conference on Geographic Information Science, Melbourne, 2018. 1–15
7 Fang Y, Wang L Z, Hu T. Spatial co-location pattern mining based on density peaks clustering and fuzzy theory. In: Web and Big Data. New York: Springer, 2018. 298–305
8 Wang X X, Lei L, Wang L Z, et al. Spatial colocation pattern discovery incorporating fuzzy theory. IEEE Trans Fuzzy Syst, 2021, 30: 2055–2072
9 Morimoto Y. Mining frequent neighboring class sets in spatial databases. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, 2001. 353–358
10 Shekhar S, Huang Y. Discovering spatial co-location patterns: a summary of results. In: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, Redondo Beach, 2001. 236–256
11 Huang Y, Shekhar S, Xiong H. Discovering colocation patterns from spatial data sets: a general approach. IEEE Trans Knowl Data Eng, 2004, 16: 1472–1485
12 Yoo J S, Shekhar S, Smith J, et al. A partial join approach for mining co-location patterns. In: Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems, Washington DC, 2004. 241–249
13 Yoo J S, Shekhar S. A joinless approach for mining spatial colocation patterns. IEEE Trans Knowl Data Eng, 2006, 18: 1323–1337
14 Wang L Z, Bao Y Z, Lu J, et al. A new joinless approach for co-location pattern mining. In: Proceedings of International Conference on Computer and Information Technology, Sydney, 2008. 197–202
15 Wang L Z, Bao Y Z, Lu Z Y. Efficient discovery of spatial co-location patterns using the iCPI-tree. Open Inf Syst J, 2009, 3: 69–80
16 Huang Y, Zhang P S. On the relationships between clustering and spatial co-location pattern mining. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, Washington DC, 2006. 513–522

17 Tran V, Wang L Z. Delaunay triangulation-based spatial colocation pattern mining without distance thresholds. Stat Anal Data Min-ASA Data Sci J, 2020, 13: 282–304
18 Celik M, Kang J M, Shekhar S. Zonal co-location pattern discovery with dynamic parameters. In: Proceedings of the 7th IEEE International Conference on Data Mining, Omaha, 2007. 433–438
19 Qian F, Chiew K, He Q M, et al. Mining regional co-location patterns with kNNG. J Intell Inf Syst, 2014, 42: 485–505
20 Deng M, Cai J N, Liu Q L, et al. Multi-level method for discovery of regional co-location patterns. Int J Geogr Inf Sci, 2017, 31: 1846–1870
21 Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, San Francisco, 1994. 487–499
22 Yu W H. Identifying and analyzing the prevalent regions of a co-location pattern using polygons clustering approach. ISPRS Int J Geo-Inf, 2017, 6: 259
23 Ding W, Eick C F, Yuan X J, et al. A framework for regional association rule mining and scoping in spatial datasets. Geoinformatica, 2011, 15: 1–28
24 Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014, 344: 1492–1496
25 Du M J, Ding S F, Jia H J. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. Knowledge-Based Syst, 2016, 99: 135–145
26 Wang S L, Wang D K, Li C Y, et al. Clustering by fast search and find of density peaks with data field. Chin J Electron, 2016, 25: 397–402
27 Liu Y H, Ma Z M, Yu F. Adaptive density peak clustering based on k-nearest neighbors with aggregating strategy. Knowledge-Based Syst, 2017, 133: 208–220
28 Khuri S, Bäck T, Heitkötter J. The zero/one multiple knapsack problem and genetic algorithms. In: Proceedings of the 1994 ACM Symposium on Applied Computing, Phoenix, 1994. 188–193
29 Liu Y, Li P, Xu Y, et al. Quantitative classification and ordination for plant communities in dry valleys of Southwest China. Biodiversity Science, 2016, 24: 378–388 [刘晔, 李鹏, 许玥, 等. 中国西南干旱河谷植物群落的数量分类和排序分析. 生物多样性. 2016, 24: 378–388]
30 Shang F, Saito T, Ohi S, et al. Coniferous and broad-leaved forest distinguishing using L-band polarimetric SAR data. IEEE Trans Geosci Remote Sens, 2021, 59: 7487–7499

A parallel algorithm for regional co-location mining based on fuzzy density peak clustering

Xiwen JIANG[1], Lizhen WANG[1,2*] & Vanha TRAN[3]

1. *School of Information Science and Engineering, Yunnan University, Kunming* 650504*, China*;
2. *Dianchi College of Yunnan University, Kunming* 650228*, China*;
3. *Department of Information Technology Specialization, FPT University, Hanoi* 155514*, Vietnam*
* Corresponding author. E-mail: lzhwang@ynu.edu.cn

Abstract Regional co-location pattern mining (RCPM) is designed to discover co-location patterns that exist within some local regions to address the patterns that cannot be found globally. Traditional RCPM techniques use geometry with well-defined boundaries as the regions of prevalent co-locations. However, the regions should not have such determined bounds. Moreover, data distribution is another important feature of a region, and this feature should also affect region selection. Based on the above considerations, we introduced density peak-based clustering (DPC) and proposed a novel density metric, combining with the fuzzy set theory and k-nearest neighbor distance to design an applicable paralleled RCPM algorithm. Experiment results show that our method can mine more meaningful results, and parallelization improves our algorithm efficiency. On real data, the speedup ratio under two threads reached 1.89.

Keywords spatial data mining, regional co-location pattern, fuzzy density peak clustering, parallel algorithm, k-nearest neighbor