计算机科学与探索 1673-9418/2000/00(00)-0000-00
Journal of Frontiers of Computer Science and Technology doi: 10.3778/j.issn.1673-9418.0000000

基于密度峰值聚类的 Top-K 低冗余 Co-location 模式挖掘算法

作者名 [1+], 作者名 [2]

1. 单位全名 学院(系)全名, 省 市(或直辖市) 邮政编码
2. 单位全名 学院(系)全名, 省 市(或直辖市) 邮政编码
+ 通信作者 E-mail: ****, Phn: +86-**-****-****, http://****

摘 要: 随着空间数据集规模的不断增大，空间 co-location 模式挖掘所产生的频繁模式快速增加。为了解决大规模频繁 co-location 模式集对于数据分析造成的困难，提出了一种基于密度峰值聚类的 Top-K 低冗余 co-location 模式提取方法。首先，为了更加准确的衡量 co-location 模式之间的相似度，设计了一种基于表实例的模式距离度量，实现了计算所有 co-location 模式之间相似度的功能。其次，利用 K 邻近密度峰值聚类对频繁 co-location 模式进行聚类。基于聚类的结果，进一步的筛选出代表性 co-location 模式，达到低冗余的压缩效果。通过在合成数据集以及实际数据集上的实验，证实了该方法具有良好的压缩性能。

关键词: 空间数据挖掘; co-location 模式挖掘; 二次挖掘; 聚类; 模式压缩

文献标志码：A **中图分类号：**TP***

Mining Redundancy-aware Top-k Co-location Patterns Based on Density Peak Clustering

NAME Name[1+], NAME Name[2]

1. College/School (Department) of ****, University, City ZipCode, China
2. College/School (Department) of ****, University, City ZipCode, China

Abstract: With the continuous increasing in the scale of spatial data sets, the number of prevalent co-location patterns generated by traditional mining framework is rapidly increased. To solve the difficulty of data analysis caused by large-scale prevalent co-location patterns, this paper proposed an approach to discover compressed co-location patterns based on the density peak clustering. First, we proposed a strategy to measure the distance between two co-location patterns by using comparing their table instances. Second, we cluster the prevalent co-location patterns by a KNN approach of the Density Peak Clustering (DPC). Finally, by evaluating our proposed approach compared with the state-of-the-art approaches, the experimental results on both synthetic and real data sets demonstrate the effectiveness of our approach.

Key words: spatial data mining; co-location pattern mining; post-mining; clustering; pattern compression

基金项目：基金中文完整名称（******编号）。

This work was supported by the **** Foundation/Program/Project of China (****** No.).

收稿日期：2000-00-00 **修回日期：** 2000-00-00.

随着卫星与遥感技术的不断发展，空间数据库所收集到的信息变得越发庞杂。传统数据挖掘方法难以处理空间数据中的空间相关性，为了解决此问题，空间数据挖掘成为了一个新的研究方向。空间数据挖掘技术使得使用者能够更加快速地获得空间数据中有用的信息，然后通过对数据的分析来获取空间数据中隐藏的规律。除了与地理空间有关的领域，例如地理信息系统、图像数据勘测，空间数据挖掘的前景也十分广阔，在国防、导航、生态保护、交通系统与公共卫生等领域，该技术都有广泛的应用。

由于空间数据没有自然的事务概念，Huang Yan 和 Jin Soung Yoo 提出了空间 co-location 模式的概念[1]，空间 co-location 模式是空间特征的一个子集，隶属于这些特征的实例在空间中频繁关联。例如，对交通设施的空间分布进行分析时，{地铁站，共享单车停放点}是一个频繁的二阶 co-location 模式，其中地铁站、共享单车停放点是空间中所有相应设施的统称，即空间特征。该模式代表在一定范围内，地铁站与共享单车停放点会相邻出现，代表它们在人们日常交通出行中具有一定的相关性。

空间 co-location 模式挖掘是空间数据挖掘中的一个重要分支，已经有大量高效的算法[2]可以快速挖掘出频繁的 co-location 模式。在传统的 co-location 模式挖掘过程中，常用参与度来衡量 co-location 模式的频繁度，但参与度阈值是由用户主观设置的，如果参与度阈值设置过大，则产生的 co-location 模式过少，无法发现潜在的联系，如果参与度阈值设置过小，则会产生大量的 co-location 模式，使得结果难以分析。如果空间数据中包含 n 个空间特征，则最多会产生 2^n-n-1 个频繁 co-location 模式，随着 n 的增大，频繁 co-location 模式的数量呈指数增长，而其中的频繁 co-location 模式可能存在极大的冗余，这使得用户对数据的分析极为困难。

为了解决上述问题，学者提出两种 co-location 模式的表示方法，极大 co-location 模式[3]与闭频繁 co-location 模式[4]。极大 co-location 模式是一种特殊的频繁 co-location 模式，它们的超集均不频繁。使用极大 co-location 模式代表频繁 co-location 模式，可以大幅减少模式的数量，但与此同时，极大 co-location 模式普遍为高阶模式，并且会覆盖它们的子集，损失了频繁模式集中的有用信息。闭 co-location 模式是指一个 co-location 模式的参与度不等于其所有直接超集的参与度，该方法可以尽可能地保留频繁 co-location 模式中的有用信息，但闭 co-location 模式的压缩能力低下，极端情况下，甚至无法起到压缩的效果。

由于上述两种方法在压缩上的局限性，需要有一种可以压缩频繁 co-location 模式数量，并保留有用信息的 co-location 模式压缩方法。本文提出了一种基于聚类的方法以发现 co-location 模式集中的 top-k 低冗余 co-location 模式。主要贡献包括：

1.提出了一种可以衡量所有模式之间相似度的模式距离度量，该度量考虑了 co-location 模式表实例的具体分布，这使得压缩 co-location 模式的方法可以准确判断模式之间的冗余，并且更加灵活。

2. 为了达到低冗余 top-k 模式的压缩效果，我们采用密度峰值聚类方法对 co-location 模式进行聚类，并且可以由用户设定获得的簇的个数 k。

3. 在获得的 k 个簇中，通过计算簇中的每个模式与其他模式之间的平均模式距离，获得具有代表性的 k 个 co-location 模式。

4.利用实际数据集与合成数据集进行了对比实验，证明了提出的压缩方法具有较高的压缩率和且压缩结果冗余度较小。

1 基本概念

空间特征是对空间中同种事物的概括，每个空间特征包含一个或多个空间实例，空间特征集是空间中所有空间特征的集合。为了表示实例与特征之间的隶属关系，每一个空间实例都由<实例 id，空间特征，空间坐标>组成，每个空间特征的实例集由隶属于该空间特征下的所有空间实例组成。在衡量空间实例之间的关系时，常用欧几里得距离作为距离度量，设置一个距离阈值 d，当两个实例的距离小于 d 时，即认定这对实例满足空间邻近关系 R。当隶属于不同空间特征的多个空间实例两两相邻时，这些实例的组合被称为团 (Clique)，常用 cl 表示。

一个 co-location 模式 c 是空间特征的子集，c

中空间特征的个数称为 c 的阶数。隶属于 c 中各个不同特征的实例组成的团，称为 c 的行实例(Row Instance)，所有行实例的集合称为表实例(Table Instance)。

在挖掘空间 co-location 模式时，采用参与率 PR(Participation Ratio)和参与度 PI (Participation Index)来衡量 co-location 模式的频繁程度，参与率的计算公式为：

$$PR(c, f_i) = \frac{\pi_i(|table\_instance(c)|)}{|table\_instance(f_i)|} \quad (1)$$

其中，f_i为 co-location 模式 c 的一个特征。参与率代表 co-location 模式中一个特征的实例出现的频率。参与度为$PI(c)=\min_{f_i \in c}\{PR(c,f_i)\}$,参与度代表 co-location 模式的频繁度。

当一个 co-location 模式 c 的参与度不小于用户设定的最小参与度阈值 min_pr 时，称 c 为频繁 co-location 模式。

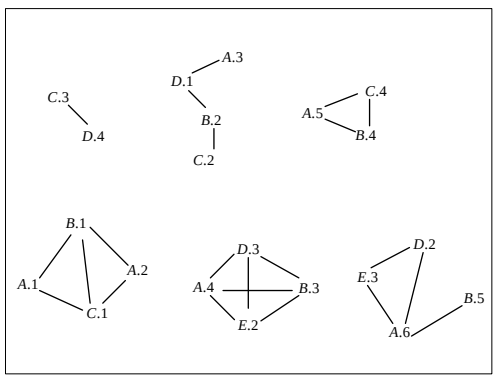


Fig.1 The distribution of spatial instances

图一 空间实例分布

图一给出了一个空间实例分布的例子，包含 5 个特征 A、B、C、D 和 E，A.1 表示 A 特征的第一个实例；该空间中共有 6 个 A 的实例，5 个 B 的实例，4 个 C 的实例，4 个 D 的实例以及 3 个 E 的实例，实例间的连线代表两个实例之间存在邻近关系，例如 A.1 和 B.1 是相互邻近的。{A.1, B.1, C.1}中的三个实例两两邻近，构成一个团，为 3 阶 co-location 模式{A, B, C}的一个行实例。除此之外，{A, B, C}的表实例还包含{A.2, B.1, C.1}和{A.5, B.4, C.4}，由此可计算得到 co-location 模式{A, B, C}的参与率 PR{{A, B, C},A}=3/6, PR{{A, B, C},B}=2/5, PR{{A, B, C}, C}=2/4,因此，PR{A, B, C}=min{PR{A, B, C},A}, PR{{A, B, C}, B},PR{{A, B, C}, C}=2/5,若参与度阈值为 0.4，则{A, B, C}是一个频繁的 co-location 模式。

为了减少 co-location 模式的数量，使结果尽量具有代表性，除了极大 co-location 模式和闭频繁 co-location 模式外，有学者提出了利用模式距离对 co-location 模式之间的相似性进行判断。文献[5]、[6]基于 co-location 模式的实例定义了父子模式之间的距离，并采取自顶向下的方式对 co-location 模式集进行压缩。具体方式为给定模式距离阈值，当模式距离小于阈值时，两个模式存在冗余，如图二所示，实线表示模式距离小于距离阈值，虚线表示模式距离大于距离阈值。利用父模式覆盖与其相似度高的冗余子模式，并输出代表模式集{$P2, P3, P5, P7, P9$}，实现 co-location 模式的压缩。相比于极大频繁 co-location 模式和闭 co-location 模式，这种压缩方法的结果具有更高的代表性。然而，自顶向下的结构限制了该方法的压缩效果，这是因为模式距离只能衡量父子 co-location 模式之间的相似度，而忽略了不相交或部分相交的 co-location 模式之间存在的冗余。

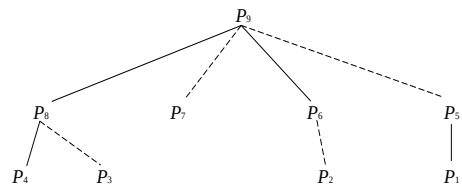


Fig.2 Traditional method of co-location pattern compression

图二 传统的模式压缩方法

本文提出了一种新的模式距离度量方法来衡量模式之间的相似性，这种方法并不局限于父子模式之间，而是可以衡量所有模式之间的距离，并利用聚类方法实现了 Top-K 低冗余模式的提取，使压缩能力提高。

2 co-location 模式间距离度量

Co-location 模式之间的相似性代表两个模式在空间分布上的相关性。表实例是 co-location 模式在空间分布上的具体表示，为了更加精确的衡量两个模式之间的相似性，需要分析它们的表实例在空间中的关系。给出如下定义：

定义 1（co-location 模式距离） 给定两个 co-location 模式 A 和 B，他们之间的模式距离为：

$$D(A,B)=1-\frac{|T(A\cup B)|}{|T(B)-R_B(A\cup B)|+|T(A)-R_A(A\cup B)|+|T(A\cup B)|} \quad (2)$$

其中，T(A)代表 co-location 模式 A 的表实例，$R_A(A\cup B)$表示$A\cup B$的表实例中包含的模式 A 的行实例。模式距离 D 的取值范围为[0,1]，当 A、B 之间的距离等于 1 时，$A\cup B$不存在表实例，这表示 A 和 B 的行实例不会出现在同一个邻域内，即 A 与 B 不具有空间相关性。当 A 与 B 的距离为 0 时，$A\cup B$的表实例包含了 A 和 B 的所有行实例，这代表在 A 的行实例的邻域内，至少存在一个 B 的行实例使它们构成$A\cup B$的行实例，即 A 与 B 在空间中完全相关。通过计算模式距离，可以获得 A、B 表实例在空间中的相关性，接下来通过不同模式关系说明该度量方法的合理性。

A	B	C
A.1	B.1	C.1
A.2	B.2	C.3
A.2	B.3	C.3
A.3	B.4	C.2
A.4	B.4	C.4

C	D	E
C.1	D.2	E.2
C.2	D.1	E.3
C.2	D.3	E.1
C.3	D.4	E.4
C.3	D.5	E.2

A	B	C	D	E
A.1	B.1	C.1	D.2	E.2
A.3	B.4	C.2	D.3	E.1

Fig.3 Table instance of ABC, CDE, ABCDE

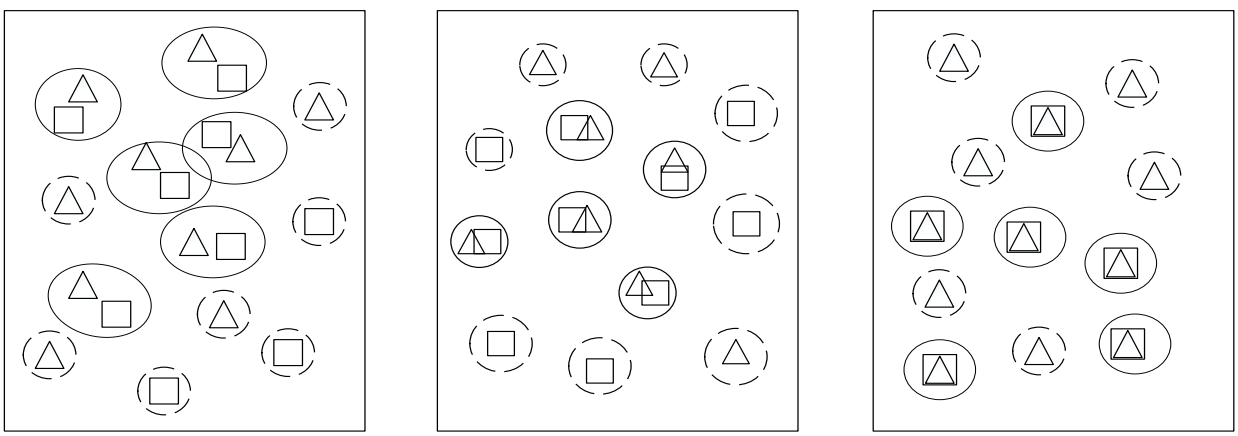
图三 ABC、CDE 与 ABCDE 的表实例

合理性分析：基于表实例的模式距离可以衡量所有 co-location 模式之间的相似度。为了解释该模式距离的合理性，图四列举了 co-location 模式之间的三种关系下的表实例分布，分别为不相交模式、部分相交模式和父子模式。(1)中不相交的两个 co-location 模式不存在共同的特征，所以各自的行实例在空间中不会相交。当两个模式具有较高的空间相关性时，双方的行实例会频繁分布在同一空间邻域内，使得模式距离变小，反之则变大。(2)中部分相交的 co-location 模式之间会共有部分特征，所以两个 co-location 模式的行实例可能会相交，当两个模式具有较高的空间相关性，并集模式的行实例会频繁分布在一个空间邻域内，此时模式距离较小。在构成父子模式的一对 co-location 中(图四(3))，父模式包含了子模式的所有特征，根据 co-location 模式的特性，父模式的行实例都包含了子模式的行实例，而子模式的行实例不一定能构成父模式的行实例，因此，公式 2 会变成：

$$D(A,B)=1-\frac{|T(A)|}{|T(B)-R_B(A)|+|T(A)|} \quad (3)$$

其中 $B\subseteq A$。若两个之间模式距离越小，则子模式的行实例构成了更多父模式的行实例，说明两个模式间具有更高的空间相关性。

通过上述分析，可以计算图四中三种情况下的模式间距离，分别为 7/13，8/13 和 1/2,。



(1) Not intersecting pattern
(1) 不相交模式

(2) Partially intersecting pattern
(2) 部分相交模式

(3) Completely intersecting pattern
(3) 父子模式

Fig.4 The distribution of row instance

图四 模式行实例分布

3 Co-location 模式压缩方法

本节介绍一种基于密度峰值聚类的 co-location 模式压缩算法，该方法为二次挖掘方法(Post-Mining)，分为 5 个步骤，包括 co-location 模式挖掘、获取 co-location 模式实例集、计算 co-location 模式距离、密度峰值聚类以及获取代表性 co-location 模式，接下来将对每个步骤进行介绍。

首先，co-location 模式挖掘获取所有的频繁 co-location 模式以及它们的参与度信息。为了便于在二次挖掘的过程中获取每个频繁模式的表实例，还需要输出包含每个频繁模式全部实例的 co-location 模式实例集。

算法 1　获取 co-location 模式实例

输入:

PC:频繁 co-location 模式集合

INS:频繁 co-location 模式的实例集

DT:距离阈值

P:co-location 模式

输出：

P_TINS:*P* 的表实例集

步骤：

```
1:  IF (! is_prevlant_pattern(P))  THEN
2:     P1_INS = get_ins(P1);
3:     P2_INS = get_ins(P2);
4:     P_INS = union (P1_INS, P2_INS);
5:  ELSE
6:     P_INS = get_ins(P);
7:  END IF
8:  STN = gen_star_neighbor (P_INS);
9:  C_INS = gen_coarse_ins (STN);
10:  P_TINS = filter(C_INS);
```

3.1 获取 co-location 模式的行实例

算法 1 给出了获取 co-location 模式表实例的算法步骤。通过 co-location 模式实例集，get_ins(P)(第 2、3、6 行)获取了频繁 co-location 模式的全部实例，并将实例按照不同特征分类，便于获取表实例。两个频繁 co-location 模式的并集模式可能不频繁，所以对于非频繁并集 co-location 模式，通过 union(第 4 行)将两个子集模式的实例合并，作为并集模式的实例集。Gen_star_neighbor(第 8 行)通过 co-location 模式的实例集，按照字典顺序将特征排序，构建出星型邻居集。利用 Join-less 中的方法，gen_coarse_ins(第 9 行)从星型邻居集中获取 co-location 模式的粗糙表实例。最后，Filter(第 10 行)遍历粗糙表实例中的每个行实例，然后判断行实例中的每个实例是否与其他实例相邻，若存在不相邻的实例，则删除这个行实例。过滤得到的结果即为 co-location 模式的表实例。

算法 2　计算 co-location 模式距离

输入：

P_1：co-location 模式 1

P_2：co-location 模式 2

输出：

d：模式距离

步骤：

```
1:  P1_TINS = get_TINS(P1);
2:  P2_TINS = get_TINS(P2);
3:  P = Union_Pattern (P1, P2);
4:  P_TINS = get_TINS(P);
5:  FOR EACH (RI1 in P1_TINS) DO
6:    IF (P_TINS. contain (RI1))
7:       P1_TINS. remove(RI1);
8:    END IF
9:  END FOR
10: FOR EACH (RI2 in P2_TINS) DO
11:   IF (P_TINS. contain (RI2))
12:      P2_TINS.remove(RI2);
13:   END IF
14: END FOR
15: d = |P_TINS|/(|P_TINS|+|P1_TINS|+|P2_TINS|);
16: return d;
```

3.2 计算模式距离

算法 2 给出了模式距离的计算方法。首先，通过利用算法 1 分别获取待比较的两个模式 P1、P2 的表实例(第 1、2 行)，并利用 union_pattern 获取并集模式 P 及其表实例(第 3、4 行)。分别遍历 P1 和 P2 的表实例，若某个行实例构成了 P 的行实例，则将该行实例从对应的表实例中删去(第 5 到 14 行)。最后，分别统计 P1、P2 和 P 三个模式的表实例包含的行实例个数，计算得到 P1 与 P2 的模式距离(第 15 行)。

3.3 密度峰值聚类

本文采用密度峰值聚类(Density Peak Clustering，DPC)[7]对频繁 co-location 模式进行聚类，原始的密度峰值聚类算法是基于截断距离进行密度的计算的，但在 co-location 模式距离中，截断距离难以评估，而截断距离的选取对聚类的效果会产生巨大的影响。为了获得更好的聚类效果，本文采用了密度峰值聚类的 k 邻近优化算法[8]，在此算法中，密度的表达式为：

$$\rho=\sum_{j\in KNN_i}\exp(-d_{ij}) \tag{4}$$

其中，KNN_i是 co-location 模式 i 的 k 个最邻近的模式集合[9]。经过测试[10]，通过公式(2)，可以较好地计算出每个 co-location 模式的密度，并且具有较高的区分度[11]。

3.4 代表性 co-location 模式的选取

密度峰值聚类将获得 K 个 co-location 模式簇，这里的 K 值是由用户定义的代表性 co-location 模式个数，之后将在每个簇中选出最具有代表性的 co-location 模式作为结果输出。

获取代表性 co-location 模式的方法有很多种，这里采取最简单有效的方法，即寻找每个簇中距离该簇其他 co-location 模式距离的和最小的 co-location 模式作为代表性模式，这代表了该 co-location 模式与簇内其他 co-location 模式具有最高的相似度，适合作为代表性 co-location 模式。

3.5 算法性能分析

在算法 1 的 gen_starneighbor 方法中，设 n 为 co-location 模式的特征数量，m 为每个特征平均包含的实例数量，需要获取的星型邻居集的大小为 m，且需要比较第一个特征的实例与其他实例之间的邻居关系，所以计算次数为$(n\text{-}1)\times m^2$，时间复杂度为$O(n\times m^2)$。在算法 2 中，获取并集模式 P 的实例集需要合并两个子模式的实例集，时间复杂度为$O(m\times n)$。设子模式以及并集模式平均包含 s 个表实例，则 5 到 9 行以及 10 到 14 行都要运行 s 次，并且判断并集模式是否包含实例的时间复杂度为 $O(s)$，所以计算 co-location 模式距离的时间复杂度为$O(s^2)$。

4 实验结果与分析

本节将分别在实际数据和合成数据上进行对比实验来测试算法的压缩能力、压缩结果的代表性以及算法的运行时间。实验中的算法均采用 Java 编写，实验环境为 Win10 系统，Core i7 CPU，8GB 内存计算机。

4.1 实际数据集及实验设计

实际数据选取云南省三江并流保护区珍惜植被的分布数据，该数据集包含 31 个空间特征，空间实例个数为 337 个，空间范围为 130km×130km，距离阈值设置为 10km，为了衡量压缩方法在不同规模 co-location 模式集上的压缩效果，选取 0.3, 0.25, 0.22 作为 co-location 模式挖掘的参与度阈值分别进行频繁模式挖掘，产生的频繁模式数量分别为 1577, 2562, 4050。

本节将通过压缩率与代表性模式的平均距离来展示基于密度峰值聚类的 co-location 模式压缩方法(DPCTK)的压缩效果，压缩率公式如下：

$$CR=\left(1-\frac{|\text{代表性co-location模式}|}{|\text{频繁co-location模式}|}\right)\times 100\% \tag{5}$$

图五展示了 DPCTK 与 Top-K CCP[4]、Top-K-size MCP[3]以及基于父子模式距离的 co-location 模式压缩方法(RCPFast)[6]在三种频繁模式集上的压缩效果比较(对所有 Top-K 算法，均有 K=150)，avr_D 代表压缩产生的 co-location 模式之间的平均模式距离，avr_D 越小，压缩产生的 co-location 模式之间的冗余度越高，反之则越低。Patterns Set 代表原始模式集的。图六展示了各个压缩方法在不同模式集下的压缩率。

如图五和图六所示，Top-K CCP 虽然可以压缩产生 K 个模式，但 K 个模式的选取标准为参与度最高的 K 个闭模式并不考虑模式之间的相关性，这使得压缩得到的结果平均模式距离较低，冗余度较高。Top-K-size MCP 以模式的阶数为标准，从高到低输出 K 个极大模式，所以高阶模式在压缩结果中占较大比例，这使得压缩结果的平均距离较大，但由于每阶都可能有多个极大模式，使得压缩结果并不只有 K 个，所以压缩率较低与 Top-K CCP。RCPFast

利用父子模式距离来衡量父子模式之间的相似度，但由于模式距离的限制，只能以自顶向下的方法进行压缩，所以压缩率较低，压缩的结果冗余度也较高。与上述三个压缩方法相比，本文提出的 DPCTK 在不同规模的模式集上都具有最高的平均模式距离和压缩率，这代表 DPCTK 具有良好的压缩性能。

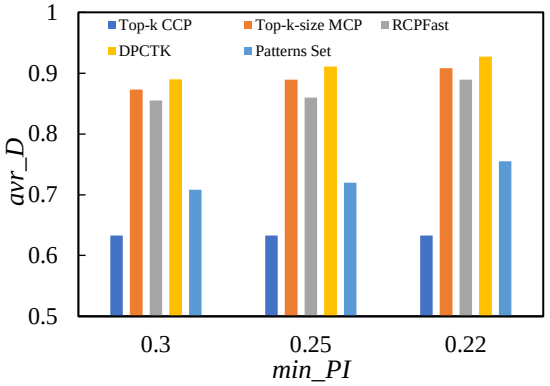


Fig.5 Average distance between representative patterns
图五　代表性模式之间的平均模式距离

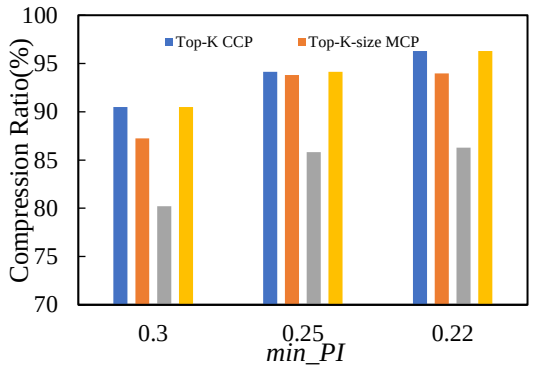


Fig.6 Compression ratio
图六　压缩率

为证明 DPCTK 的压缩结果具有代表性，表 1 列举了参与度为 0.22 时，原始模式集及每个压缩方法产生的各阶 co-location 模式数量及占比。由表 1 可知，原始模式集中 3 阶，4 阶和 5 阶 co-location 模式占有很大比例，Top-K CCP 压缩结果中的 co-location 模式大多为 2 阶和 3 阶 co-location 模式，这是由于参与度的反单调性，导致低阶 co-location 模式的参与度高于高阶 co-location 模式。Top-K-size MCP 选择高阶极大 co-location 模式作为输出，产生了更多 7 阶，8 阶和 9 阶模式，但低阶模式损失较多。RCPFast 的压缩结果较为平均，压缩结果包含每阶 co-location 模式，具有更好的代表性，但产生的 co-location 模式数量较多，压缩率低。DPCTK 相较于其他压缩方法，不仅压缩结果中的 co-location 模式分布均匀，且压缩率更高，具有更好的压缩效果。

4.2　合成数据集及实验设计

在 3.1 节所介绍的获取 co-location 模式表实例的步骤里，需要在挖掘完毕后输出每个 co-location 模式的实例集，为了验证该方法对于算法速度的提升效果，本节将在合成数据集上对比有实例输入的 DPCTK 与无实例输入的 DPCTK 在不同模式数量下的运行时间。合成数据集包含 30 个空间特征，4076 个空间实例，范围为 500×500，距离阈值为 15，通过选取不同的参与度，产生不同模式数量的模式集，具体数量如图七所示。

表 1 各阶 co-location 模式数量及占比

Table.1 The number and proportion of each order co-location pattern

项目	2 阶	3 阶	4 阶	5 阶	6 阶	7 阶	8 阶	9 阶	总计
原始频繁模式集	327 (100%)	970 (100%)	1243 (100%)	906 (100%)	437 (100%)	139 (100%)	26 (100%)	2 (100%)	4050 (100%)
Top-K CCP	81 (24.77%)	57 (5.88%)	11 (0.88%)	1 (0.11%)	0 (0%)	0 (0%)	0 (0%)	0 (100%)	150 (3.7%)
Top-K-Size CCP	0 (0%)	0 (0%)	118 (9.49%)	66 (7.28%)	36 (8.24)	13 (9.35)	9 (34.62%)	2 (100%)	244 (6.02%)
RCPFast	52 (15.9)	151 (15.57)	173 (13.92%)	95 (10.49%)	56 (12.81%)	18 (12.95)	9 (34.62)	2 (100%)	556 (13.98%)
DPCTK	27 (8.26%)	39 (4.02%)	49 (3.94%)	20 (2.21%)	5 (1.14%)	5 (3.6%)	3 (11.54%)	2 (100%)	150 (3.7%)

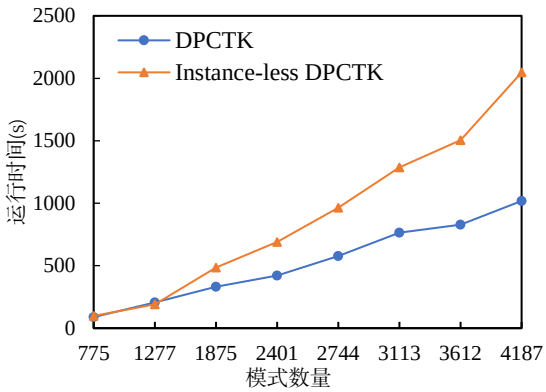


Fig.7 Running time

图七 运行时间

无实例的 DPCTK 在计算模式距离时直接从原始实例集中通过星型邻居集获取每个 co-location 模式的实例。当模式数量上升时，需要获取更多的 co-location 模式表实例，而直接从所有实例集构成的星型邻居集获取实例信息要更为复杂。在有实例输入的 DPCTK 中，可以从 co-location 模式的实例集中产生一个更加精确的星型实例集，并且在该星型实例集中，所有星型实例都必然包含相应模式的表实例，所以节省了更多的计算时间。

5 总结

本文提出了一个基于密度峰值聚类的 co-location 模式压缩算法。与之前利用父子模式距离进行压缩的算法相比，本文提出的模式距离度量可以衡量任意 co-location 模式之间的相似度，使得模式的压缩更加方便。其次，本文将密度峰值聚类加入到压缩过程中，使得压缩效率更高，获得的结果更具有代表性。由于提出的方法为二次挖掘方法，本文设计了优化方法用来提高算法的运行速度，并提供了获取代表性模式的方法。通过实验与分析，验证了本文提出算法的高压缩率与高代表性，并证明了优化方法对运行时间与代表性的提升。

未来，我们将继续研究 co-location 模式的压缩方法，并对压缩算法做优化，减少计算 co-location 模式距离的计算复杂度，提高压缩结果的代表性，更加准确、快速地获得代表性 co-location 模式集。

参考文献：

[1] Huang Y, Shekhar S, Xiong H. Discovering colocation patterns from spatial data sets: a general approach [J]. IEEE Transactions on Knowledge and data engineering, 2004, 16(12): 1472-1485.

[2] Yoo J S, Shekhar S, Celik M. A join-less approach for co-location pattern mining: A summary of results [C]//Fifth IEEE International Conference on Data Mining (ICDM'05). IEEE, 2005: 4 pp.

[3] Bao X, Wang L, Zhao J. Mining top-k-size maximal co-location patterns [C]//2016 International Conference on Computer, Information and Telecommunication Systems (CITS). IEEE, 2016: 1-6.

[4] Yoo J S, Bow M. Mining top-k closed co-location patterns [C]//Proceedings 2011 IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services. IEEE, 2011: 100-105.

[5] Wang L, Bao X, Zhou L. Redundancy reduction for prevalent co-location patterns [J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 30(1): 142-155.

[6] Liu B, Chen L, Liu C, et al. RCP mining: Towards the summarization of spatial co-location patterns [C]//International Symposium on Spatial and Temporal Databases. Springer, Cham, 2015: 451-469.

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

[8] Du M, Ding S, Jia H. Study on density peaks clustering based on k-nearest neighbors and principal component analysis [J]. Knowledge-Based Systems, 2016, 99: 135-145.

[9] Xie J, Gao H, Xie W, et al. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors[J]. Information Sciences, 2016, 354: 19-40.

[10] Hou J, Zhang A, Qi N. Density peak clustering based on relative density relationship[J]. Pattern Recognition, 2020, 108: 107554.

[11] Chen Y, Hu X, Fan W, et al. Fast density peak clustering for large scale data based on kNN[J]. Knowledge-Based Systems, 2020, 187: 104824

[12] Xu X, Ding S, Wang L, et al. A robust density peaks clustering algorithm with density-sensitive similarity[J]. Knowledge-Based Systems, 2020, 200: 106028.

[13] Wang L, Bao X, Cao L. Interactive probabilistic post-mining of user-preferred spatial co-location patterns[C]//2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, 2018: 1256-1259.

[14] Wang L, Bao X, Cao L. Interactive probabilistic post-mining of user-preferred spatial co-location patterns[C]//2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, 2018: 1256-1259.

[15] Bao X, Wang L. A clique-based approach for co-location pattern mining[J]. Information Sciences, 2019, 490: 244-264.

[16] Bao X, Gu T, Chang L, et al. Knowledge-Based Interactive Postmining of User-Preferred Co-Location Patterns Using Ontologies[J]. IEEE Transactions on Cybernetics, 2021, doi: 10.1109/TCYB.2021.3054923.