Evolutionary Intelligence (2020) 13:197–210
https://doi.org/10.1007/s12065-019-00321-7

SPECIAL ISSUE

Mining high influence co-location patterns from instances with attributes

Dianwu Fang[1] · Lizhen Wang[1] · Peizhong Yang[1] · Lan Chen[2]

Received: 17 June 2019 / Revised: 12 October 2019 / Accepted: 29 October 2019 / Published online: 18 November 2019

Abstract
A spatial co-location pattern describes coexistence of spatial features whose instances frequently appear together in geographic space. Numerous studies have been proposed to discover interesting co-location patterns from spatial data sets, but most of them only use the location information of instances. As a result, they cannot adequately reflect the influence between instances. In this paper, we take additional attributes of instances into account in the process of co-location pattern mining, and propose a new approach for discovering the high influence co-location patterns. In our approach, we consider the spatial neighboring relationships and the similarity of instances simultaneously, and utilize the information entropy approach to measure the influence of any instance exerting on its neighbors and the influence of any feature in a co-location pattern. Then, an influence index for measuring the interestingness of a co-location pattern is proposed and we prove the influence index measure satisfies the downward closure property that can be used for pruning the search space, and thus an efficient high influence co-location pattern mining algorithm is designed. At last, extensive experiments are conducted on synthetic and real spatial data sets. Experimental results reveal the effectiveness and efficiency of our method.

Keywords High influence co-location pattern · Influence index · Spatial instances with attributes · Information entropy

1 Introduction

In recent decades, rapid development of spatial related technologies made large amounts of spatial data available. It becomes popular to analyze spatial phenomena and discover knowledge from spatial data sets. Spatial co-location pattern mining plays an important role in this domain. A spatial co-location pattern is a subset of spatial features, whose instances frequently appear in the spatial proximity. For instance, the location where collybia albuminosa grows usually has nest of termites, so {collybia albuminosa, nest of termites} is a co-location pattern. {rhinoceros, oxpeckers} is another real example of co-location pattern because the rhinoceros often live with oxpeckers. Co-location patterns may yield important insight in many applications, such as Earth science, public health, biology, transportation, etc. Due to the wide application of co-location patterns, researchers have developed multiple approaches of mining co-location patterns.

So far, most of the spatial co-location pattern mining did not consider the influence of different features [1–9], a few paper related to this topic studied either the spatial co-location pattern mining on extended spatial objects (points, linestrings, polygons) [10–13], or the high impact co-location pattern mining on point-type objects [14], based on buffer overlap of instances. Almost all the spatial co-location pattern mining approaches made their progress on the proximity of instances, without considering the non-spatial attributes of instances. The influence between different features is hard to be reflected only by the spatial proximity, since we need to consider more non-spatial attributes of instances to help us analyze the implicit interaction between features. For example, supermarkets and convenience stores have different influence on the same residents, besides the distance factors, many non-spatial attributes of them such as scale, commodity diversity, service level, etc. also work. Attributes can reflect the influence between instances, while previous spatial co-location pattern mining approaches did not focus on the influence reflected by instances' attributes.

✉ Lizhen Wang
Lzhwang2005@126.com

[1] School of Information Science and Engineering, Yunnan University, Kunming 650504, Yunnan Province, China

[2] China Mobile Group Anhui Co., Ltd., Lu'an Branch, Lu'an 237005, Anhui Province, China

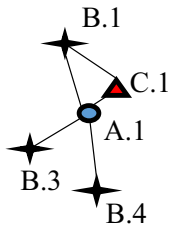

Fig. 1 An example of spatial instances

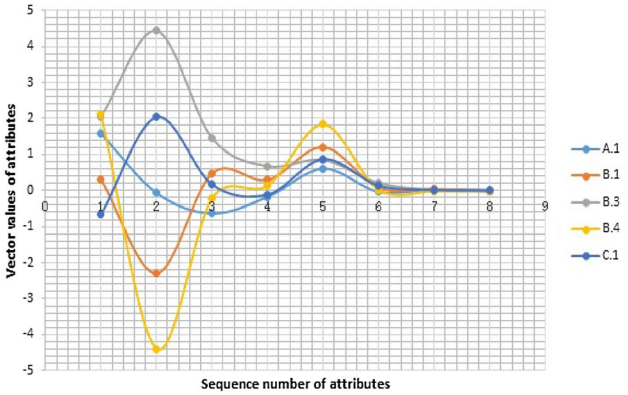

Fig. 2 Similarity of instances in terms of attributes

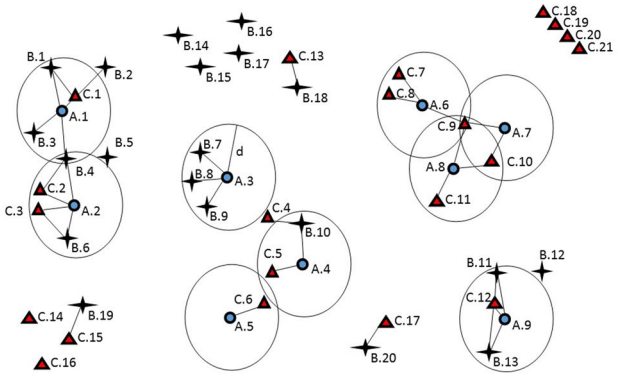

Fig. 3 Illustration of spatial co-location patterns

An example of five spatial instances is shown in Fig. 1, where the instances belong to feature A (shopping center), B (store) and C (grocery), and adjacent instances within a preset distance threshold are linked with solid lines. Although instances A.1, B.1 and C.1 form a clique for candidate co-location patterns {A, B, C}, we cannot accurately describe the influence between the three instances. Therefore, we add 8-dimensional attributes as described in Sect. 3.1 to the sample instances in Fig. 1. The principle of our insight is depicted in Fig. 2, in which one curve denotes an attribute vector for one of the five instances in Fig. 1, we can conclude the influence between instances by calculating the similarity of instances via vector values of attributes, as well as other computation.

It is challenging to introduce the influence of non-spatial attributes to co-location pattern mining. In this paper, we propose a novel approach that mines high influence co-location patterns by using spatial neighborhoods of instances and similarity of instances with attributes. In summary, the contributions of this paper are listed as follows:

(1) We introduce the additional attributes of spatial instances, construct a new equation for calculating the similarity of instances with cosine similarity function and Mahalanobis distance function, and apply information entropy approaches for computing the influence of instances.
(2) We define the concepts of influence ratio and influence index for co-location patterns, and prove them satisfy the downward closure property which can be used for pruning search space. Then, a novel High Influence Co-location Pattern Mining (HICPM) algorithm is designed to mine high influence co-location patterns.
(3) We conduct extensive experiments on synthetic and real data sets, the results show that our approach can find high influence co-location patterns effectively and efficiently.

The remainder of this paper is organized as follows: related works are introduced in Sect. 2. Section 3 gives definitions and equations, proof of downward closure property, HICPM algorithm, and time complexity analysis. Section 4 conducts experiments on real and synthetic data sets and analyzes results. Finally, Sect. 5 summarizes the paper and proposes the work in the future.

2 Related works

Related work can be elaborated in two aspects as follows:

Spatial co-location pattern mining Agrawal et al. [15, 16] firstly published a founding paper of mining association rules between sets of items in large databases in 1993, it animated the pattern mining researches on methodologies and applications actively so far [17–19]. Koperski et al. [1] introduced spatial association rules to analyze geographic information databases in 1995. Shekhar et al. [2] pioneered to define the concept of co-location pattern mining in 2001, and propose join-based method for discovering prevalent co-location patterns from spatial data sets.

A spatial co-location pattern c is defined as a set of spatial features. The spatial features represent the kinds of instances in space, denoted as f_i, while instances of f_i represent individual objects at a specified location, denoted as $f_i \cdot j$. Instances are adjacent when they are within the distance threshold range. Adjacent instances of diverse features form

cliques, those instances in cliques will be defined as row_instance(c) of a co-location pattern c only if they present all the features of the pattern and no subsets of them can do so. Table_instance(c) denotes T(c), is a set which contains all row_instance(c). Participation ratio PR(c, f_i) is defined as the fraction of number of non-repetitive instances of feature f_i involved in table_instance(c) divided by total instances of the feature f_i. Participation index PI(c) takes the minimum of PR (c,f_i). A co-location pattern c will be considered prevalent if PI(c) is no less than the preset threshold $\mathrm{PI}_{threshold}$.

Example 1 Figure 3 shows features A, B, C with 9, 20 and 21 instances respectively. A co-location pattern $c = \{A, B, C\}$ has table-instances of {{A.1, B.1, C.1}, {A.2, B.4, C.2}, {A.2, B.6, C.3}, {A.9, B.11, C.12}, {A.9, B.13, C.12}}. Thus, $\mathrm{PR}(c, A) = 3/9$, $\mathrm{PR}(c, B) = 5/20$, $\mathrm{PR}(c, C) = 4/21$, and $\mathrm{PI}(c) = \min\{\mathrm{PR}(c, f_i)\} = 4/21 \approx 0.19$. Assumed $\mathrm{PI}_{threshold} = 0.15$, the co-location pattern {A, B, C} is prevalent.

A landmark event in this field is the full-join algorithm proposed by Huang et al. [3] in 2004, it can mine complete and correct prevalent co-location patterns. As the running time of the full-join algorithm rises significantly with instances increase, Yoo et al. [4] proposed a partial-join algorithm, which divides instances into disjoint clusters to reduce computation for join operations. Yoo et al. [5, 6] proposed join-less algorithm based on star neighbor materialized model and integrated the ideas of join-less and partial-join, to make it run faster than full-join method in dense data sets, and compared their time complexity. Huang et al. [20] use a compact prefix tree structure called *FP*-tree for mining prevalent co-location patterns, Wang et al. [7] propose iCPI-tree for updating previous CPI-tree based algorithm [8] for improving efficiency of co-location pattern mining. In order to compress the large number of co-location patterns mined, researchers proposed new concepts and algorithms for mining closed patterns [21], maximal frequent patterns and compressed prevalent patterns [9]. The forms of instances studied tend to be diversified into uncertain data [22], interval data [23], fuzzy data [24], extended spatial objects [10–13], incremental data [25, 26], suitable for practical scenarios. Different from participation index mostly applied to co-location pattern mining, Xiong et al. [11] introduced extended spatial objects with buffer, and proposed coverage computation with MBBR model and apply it to test route selection. Kim et al. [12] constructed a transaction-based framework for co-location pattern mining, making association analysis applicable and extended spatial objects usable, for getting geographic context awareness of ubiquitous GIS. Li et al. [13] proposed a grid based transaction for extended spatial objects, and introduced a statistical test to validate the significance of candidate co-location patterns rather than a global threshold, for identifying correlation between child cancer cases and pollutant emissions. Chai et al. [27] suggested a node-priority based large-scale overlapping community detection method. Chen [14] noticed participation index and coverage computation did not reflect the effect of instances thus defined buffer as the effect of instances and proposed an algorithm for co-location pattern mining with high impact. In view of the shortcomings of locational information and geometric computation cost, this paper introduces attributes to instances for further exploration.

Influence analysis With the rapid development of social networks, influence analysis has been widely studied. Xiang et al. [28] evaluated relationship power from interactive action and user similarity. Sathanur et al. [29] introduced transfer entropy as a measure of directed causal influence for online social interactions. Peng et al. [30] evaluated influence of a node via two factors. One was intimacy degree reflecting the proximity between users, the other was activity degree determining active nodes. Bakshy et al. [31] counted pairwise influence by score and predict global influence of a user by sum of scores, and suggested disjoint influence tree with features to estimate user's global influence. Huang et al. [32] tended to measure individual social influence by the number of followers and the sensitivity of finding good items. Peng et al. [33] presented an evaluation model to measure direct and indirect influence based on social relationship graph, by introducing friend entropy and interaction frequency entropy to describe social influence in mobile social networks. This paper concerns the influence between adjacent instances, without consideration to the influence beyond distance threshold and indirect influence of instances.

3 Formal definitions and the algorithm

In this section, we define the high influence co-location pattern formally and prove the downward closure property of the pattern. Then, we design an algorithm with a pruning strategy for mining high influence co-location patterns and analyze time complexity of the algorithm. Table 1 lists the key notations used in this paper.

3.1 Definitions and properties

In the data sets of this paper, spatial instances are expressed as vectors, which contain feature symbol, instance number, latitude, longitude, attributes. Vector values of some spatial instances in Figs. 1 and 3 are listed in Table 2.

We compute similarity of instances on attribute vectors of instances, with integration of cosine similarity and Mahalanobis distance equations as follows:

Table 1 Key notations used in this paper

Notations	Definitions	Brief description
F	A set of features	Different types of features in space
O	A set of instances	Entity objects at specific spatial locations
k	Size of a pattern c	The number of features in a spatial co-location pattern c
R	Distance threshold	A threshold for judging whether an instance is in the proximity of other instances
Sim(X, Y)	Cosine similarity	A measure that computes the cosine of the angle between vectors X and Y
$D^2(X, Y)$	Mahalanobis distance	A measure that computes the difference between vectors X and Y
Icm	Inverse covariance matrix	A variable is used for computing $D^2(X, Y)$
S	Composite similarity	An index that integrates Sim(X, Y) and $D(X, Y)$
Ne	Neighbor entropy	A measure that reflects the aggregation of neighbors around an instance
Ase	Attribute similarity entropy	A measure that reflects the attribute similarities of instance with its neighbors
ω_1	Weight of Ne	–
ω_2	Weight of Ase	–
Inf	Influence of an object	The degree that an instance (or a feature) affects its neighbors (or other features)
InR	Influence ratio	The ratio of the influence of a feature in a co-location pattern c to the total influence of the feature
InI	Influence index	The minimum among influence ratios of all the features in the pattern c
$\text{InI}_{threshold}$	High influence threshold	A preset threshold for checking whether a pattern c is a high influence co-location pattern

Table 2 Vector values of some spatial instances in Figs. 1 and 3

Name	Latitude	Longitude	Attri_1	Attri_2	Attri_3	Attri_4	Attri_5	Attri_6	Attri_7	Attri_8
A.1	40.0133	116.4103	1.5808	−0.0691	−0.6302	−0.1745	0.6037	−0.0387	0.0332	0.0099
B.1	40.0028	116.4305	0.3046	−2.2936	0.4754	0.2930	1.2066	0.0778	0.0424	−0.0235
C.1	40.0238	116.4180	−0.6603	2.0363	0.1616	−0.1130	0.8702	0.1334	0.0156	0.0006
…	…	…	…	…	…	…	…	…	…	…

- Cosine similarity on attribute vectors of instances

Cosine similarity represents the angle between two vectors with cosine function. Assumed that $X_{f_i \cdot j}$ and $Y_{f_{i'} \cdot j'}$ are two attribute vectors of instances $f_i \cdot j$, $f_{i'} \cdot j'$, then their cosine similarity shall be computed as follows:

$$\text{Sim}\left(X_{f_i \cdot j}, Y_{f_{i'} \cdot j'}\right) = \frac{X_{f_i \cdot j} \cdot Y_{f_{i'} \cdot j'}}{||X_{f_i \cdot j}|| ||Y_{f_{i'} \cdot j'}||} \tag{1}$$

where dot $\cdot$ denotes vector dot product, $X_{f_i \cdot j} \cdot Y_{f_{i'} \cdot j'} = \sum x_{f_i \cdot j} \cdot y_{f_{i'} \cdot j'}$, $||X_{f_i \cdot j}||$ is the length of vector $X_{f_i \cdot j}$, $||X_{f_i \cdot j}|| = \sqrt{\sum (X_{f_i \cdot j})^2} = \sqrt{X_{f_i \cdot j} \cdot X_{f_i \cdot j}}$.

- Mahalanobis distance on attribute vectors of instances

Mahalanobis distance (also called *Dz statistics*) is an effective approach to calculate similarity between two unknown sample sets, it can be defined as the difference between two random variables which obey the same distribution and share same covariance matrix. Mahalanobis distance for the attribute vectors $X_{f_i \cdot j}$ and $Y_{f_{i'} \cdot j'}$ shall be computed as follows:

$$D^2(X_{f_i \cdot j}, Y_{f_{i'} \cdot j'}) = (X_{f_i \cdot j} - Y_{f_{i'} \cdot j'})^T \Sigma^{-1} (X_{f_i \cdot j} - Y_{f_{i'} \cdot j'}) \tag{2}$$

where D^2 denotes Mahalanobis distance of attribute vectors of instances $f_i \cdot j$ and $f_{i'} \cdot j'$, Σ^{-1} (also notated Icm) denotes inverse covariance matrix of variables, T denotes vector shall be transposed.

Although cosine similarity and Mahalanobis distance can express similarity between vectors, their emphases are different. Cosine similarity emphasizes the consistency of directions of vectors and is no sensitive to the numerical differences of vectors, while Mahalanobis distance emphasizes the numerical difference and is no sensitive to the consistency of directions of vectors. As the attributes of our data sets contain sequence data and numerical data, we expect to integrate cosine similarity and Mahalanobis distance and construct a new measure for reflecting the similarity of instances with attributes more accurately.

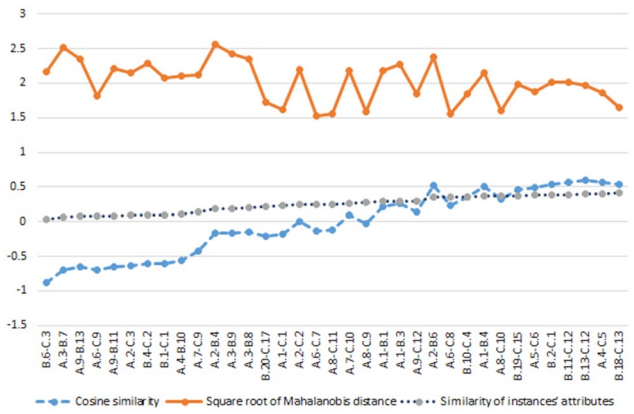

Fig. 4 Comparison on cosine similarity, square root of Mahalanobis distance, and similarity of instances

- Similarity between instances with attributes

Noticing the cosine similarity ranges in [− 1, 1] and the Mahalanobis distance ranges in [0, M] (M denotes a finite constant), we construct a new measure called similarity between instances with attributes which ranges in [0, 1]. When cosine similarity takes 1 and Mahalanobis distance takes 0, that means the attribute vectors of the neighbor pairs are coincident, i.e. the attribute vectors are in the same orientation and length, at this moment, the similarity takes 1 as the maximum. And when cosine similarity takes − 1 and Mahalanobis distance is M, that is to say, the attribute vectors of the neighbor pairs are in the opposite orientation and the attribute vectors have different length, at this moment, the similarity takes 0 as the minimum. The similarity of instances $f_i \cdot j$ and $f_{i'} \cdot j'$ is defined as follows.

$$S(f_i \cdot j, f_{i'} \cdot j') = \frac{1 + \cos(X_{f_i \cdot j}, Y_{f_{i'} \cdot j'})}{2 + \mathrm{D}(X_{f_i \cdot j}, Y_{f_{i'} \cdot j'})}, \quad S(f_i \cdot j, f_{i'} \cdot j') \in [0, 1] \tag{3}$$

where $S(f_i \cdot j, f_{i'} \cdot j')$ denotes similarity of instances $f_i \cdot j$ and $f_{i'} \cdot j'$, $\mathrm{D}(X_{f_i \cdot j}, Y_{f_{i'} \cdot j'})$ denotes square root of the Mahalanobis distance.

Using the example data set in Fig. 3, we give an illustrated comparison of cosine similarity, square root of Mahalanobis distance and similarity between instances as in Fig. 4, where the similarity of instances rises sequentially in [0, 0.5], the cosine similarity increases in minor fluctuations, while the square root of Mahalanobis distance decreases in relatively large fluctuations. We compare the values of ordinates that the neighbor pairs on the abscissa reflect on the three curves and find that when the values of ordinates for cosine similarity are the same, the smaller the square root of the Mahalanobis distance is, the greater the similarity of instances is, as depicted by the neighbor pairs of A.2–B.4, A.3–B.9 and when the values of ordinates for square root of the Mahalanobis distance are the same, the greater the cosine similarity is, the greater the similarity of instances is, as depicted by the neighbor pairs of A.7–C.10, A.1–B.1. Therefore, we reckon that the constructed similarity of instances can better smooth the floating variation of cosine similarity and Mahalanobis distance and reasonably reflect our expectation on the concept of influence which is defined on the basis of similarity between instances with attributes.

- Introduction of information entropy

Information entropy was introduced by Shannon, in his paper "A Mathematical Theory of Communication" published in 1948. It tells how much information that an event contains, the more uncertain an event is, the more information it contains. The computation of information entropy is widely applied in many areas for decades. In this paper, we use the concept and computation method of "average self-information" in Shannon's information entropy theory, that is, the average self-information of random variable X, symbolized as $\mathrm{H}(X)$, is defined as:

$$\mathrm{H}(X) = \sum_{i=1}^{n} P(x_i) I(x_i) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{4}$$

where X denotes a discrete random variable whose output is x_i, $i = 1, 2, \ldots,$ n, and $P(x_i)$ is the probability of occurrence of output x_i, $I(x_i)$ denotes the self information of event $X = x_i$.

As we aim at discovering high influence co-location patterns in this paper, the available information of data sets is the number of neighbors of instances and the similarity of instances with attributes, so we use the Eq. 4 to calculate the entropies for measuring the information contained in neighbor of instance and attribute similarity of instance as follows.

- Neighbor entropy of instance

Neighbor entropy denoted as $\mathrm{Ne}(f_i \cdot j)$, is computed on neighbor number of instance $f_i \cdot j$ with Eq. 5:

$$\mathrm{Ne}(f_i \cdot j) = -\sum_{1}^{N_{f_i \cdot j}} \frac{1}{N_{f_i \cdot j}} \log_{10} \frac{1}{N_{f_i \cdot j}} - \left(\left(-\frac{1}{N_{f_i \cdot j}} \log_{10} \frac{1}{N_{f_i \cdot j}} \right) \right) \tag{5}$$

where $N_{f_i \cdot j}$ denotes the number of neighbors and itself of instance $f_i \cdot j$. The entropy is arranged for calculating the influence of instances on their surrounding neighbors.

- Attribute similarity entropy of instance

Attribute similarity entropy denoted as $\mathrm{Ase}(f_i \cdot j)$, is computed on the similarities of instance $(f_i \cdot j)$'s attributes with its neighbors' attributes, the equation is as follows:

$$\mathrm{Ase}(f_i \cdot j) = -\sum_{1}^{N_{f_i \cdot j}} \frac{S(f_i \cdot j, f_{i'} \cdot j')}{\sum_1^{N_{f_i \cdot j}} S(f_i \cdot j, f_k \cdot m)} \log_{10} \frac{S(f_i \cdot j, f_{i'} \cdot j')}{\sum_1^{N_{f_i \cdot j}} S(f_i \cdot j, f_k \cdot m)} - \left(-\frac{S(f_i \cdot j, f_i \cdot j)}{\sum_1^{N_{f_i \cdot j}} S(f_i \cdot j, f_k \cdot m)} \log_{10} \frac{S(f_i \cdot j, f_i \cdot j)}{\sum_1^{N_{f_i \cdot j}} S(f_i \cdot j, f_k \cdot m)} \right) \tag{6}$$

where $S(f_i \cdot j, f_i \cdot j)$ denotes the similarity of instances with attributes for the instance $f_i \cdot j$ with itself, $S(f_i \cdot j, f_k \cdot m)$ denotes the similarity of instances with attributes between the instance $f_i \cdot j$ and any of its neighbors (i.e. instances $f_k \cdot m$).

Definition 1 (*Influence of instance*) Influence of instance is defined as the degree that an instance affects its adjacent instances to tend to be the same on attributes, denoted as $\mathrm{Inf}(f_i \cdot j)$, which represents all the influence that instance $f_i \cdot j$ exerts on its adjacent instances. By means of information entropy approaches, we construct the neighbor entropy and attribute similarity entropy for instances. The influence of instance $f_i \cdot j$ can be computed with the entropies as follows:

$$\mathrm{Inf}(f_i \cdot j) = \omega_1 \cdot \mathrm{Ne}(f_i \cdot j) + \omega_2 \cdot \mathrm{Ase}(f_i \cdot j), \quad \omega_1 + \omega_2 = 1 \tag{7}$$

where $\mathrm{Ne}(f_i \cdot j)$ denotes the neighbor entropy of instance $f_i \cdot j$, $\mathrm{Ase}(f_i \cdot j)$ denotes the attribute similarity entropy of instance $f_i \cdot j$. ω_1 denotes the weight of $\mathrm{Ne}(f_i \cdot j)$, ω_2 denotes the weight of Ase $(f_i \cdot j)$. In this paper, we let $\omega_1 = 0.2$, $\omega_2 = 0.8$ and assume that the influence of attributes is more bigger than that of neighbors as per our practical experience.

Definition 2 (*Influence of feature*) Given a spatial feature f_i and its instance set $\mathrm{S}(f_i)$, the influence of feature f_i is defined as the total influence of all spatial instances belonging to this feature. It is described as follows:

$$\mathrm{Inf}(f_i) = \sum_{f_i \cdot j \in S(f_i)} \mathrm{Inf}(f_i \cdot j) \tag{8}$$

Definition 3 (*Influence of feature in a co-location pattern*) Given a k-size co-location pattern $c = \{f_1, f_2, \ldots, f_k\}$, $f_i \in c$, $\mathrm{k} \geqslant 2$. The influence of feature f_i in pattern c is defined as the sum of the minimum influence of instances belonging to the feature f_i exert on other features' instances in table_instance(c). It is described as follows:

$$\mathrm{Inf}(c, f_i) = \sum_{f_i \cdot j \in T(c)} \mathrm{Inf}_{min}(f_i \cdot j) \tag{9}$$

Definition 4 (*Influence ratio of feature in a spatial co-location pattern*) Given a k-size co-location pattern $c = \{f_1, f_2, \ldots, f_k\}$, $f_i \in c$, $\mathrm{k} \geqslant 2$. The influence ratio of feature f_i in pattern c ($\mathrm{InR}(c, f_i)$) is defined as the ratio of the influence of feature f_i in the co-location pattern c to the total influence of feature f_i. It is described as follows:

$$\mathrm{InR}(c, f_i) = \frac{\mathrm{Inf}(c, f_i)}{\mathrm{Inf}(f_i)} \tag{10}$$

Definition 5 (*Influence index of the co-location pattern*) Given a k-size co-location pattern $c = \{f_1, f_2, \ldots, f_k\}$, $f_i \in c$, $\mathrm{k} \geqslant 2$. The influence index of the pattern c ($\mathrm{InI}(c)$) is defined as the minimum among influence ratios of all the features in the pattern c. It is described as follows:

$$\mathrm{InI}(c) = \min_{i=1}^{k} \{\mathrm{InR}(c, f_i)\} \tag{11}$$

Definition 6 (*High- influence co-location pattern*) A spatial co-location pattern c will be defined as a high influence co-location pattern only if its $\mathrm{InI}(c)$ is not smaller than the preset threshold $\mathrm{InI}_{threshold}$.

Example 2 Based on the table-instances and instances' attributes from the sample data set of Fig. 3, we illustrate the process of mining high influence co-location patterns in Table 3. Assumed $\mathrm{InI}_{threshold} = 0.001$, as $\mathrm{InI}(c) \geqslant \mathrm{InI}_{\mathrm{threshold}}$, we can conclude that all the co-location patterns in Table 3 are high influence patterns.

3.2 The downward closure property

Downward closure property (also called *antimonotone property*) and Apriori principle are two cornerstones for mining spatial co-location patterns. It's already proven that participation ratio (PR) and participation index (PI), which are the measures of traditional co-location pattern mining satisfy the downward closure property and Apriori principle [3, 34], i.e. PR and PI decrease monotonously with the increase of pattern sizes, and any instance $f_i \cdot j$ which belongs to the table_instances(c) surely belongs to the table_instance(c') when $c' \subseteq c$. Through calculation and proof, we find that influence ratio and influence index also satisfy the downward closure property.

Lemma 1 *(Downward closure property of the high influence co-location patterns) Influence ratio (InR) and influence index (InI) decrease monotonously as the increasing of co-locations' sizes.*

Proof Assumed $c' \subseteq c$, $|c| = \mathrm{k}$, $|c'| = \mathrm{m}$, so $\mathrm{k} > \mathrm{m}$. ($|c|$ denotes the size of c). $\mathrm{T}(c)$ denotes table-instance of the co-location pattern c. According to the above definitions and equations, the following proofs can be made:

Table 3 The process of HICPM based on sample data in Fig. 3

Co-location patterns	C_2			C_3	
	{A, B}	{A, C}	{B, C}	{A, B, C}	
T(c)	{A.1, B.1}, {A.1, B.3}, {A.1, B.4}, {A.2, B.4}, {A.2, B.6}, {A.3, B.7}, {A.3, B.8}, {A.3, B.9}, {A.4, B.10}, {A.9, B.11}, {A.9, B.13}	{A.1, C.1}, {A.2, C.2}, {A.2, C.3}, {A.4, C.5}, {A.5, C.6}, {A.6, C.7}, {A.6, C.8}, {A.6, C.9}, {A.7, C.9}, {A.7, C.10}, {A.8, C.9}, {A.8, C.10}, {A.8, C.11}, {A.9, C.12}	{B.1, C.1}, {B.2, C.1}, {B.4, C.2}, {B.6, C.3}, {B.10, C.4}, {B.11, C.12}, {B.13, C.12}, {B.18, C.13}, {B.19, C.15}, {B.20, C.17}	{A.1, B.1, C.1}, {A.2, B.4, C.2}, {A.2, B.6, C.3}, {A.9, B.11, C.12}, {A.9, B.13, C.12}	
Inf(f_i)	–	–	–	–	
Inf(c,f_i)	$Inf(c,A) = 0.419, Inf(c,B) = 1.012$	$Inf(c,A) = 0.889, Inf(c,C) = 1.341$	$Inf(c,B) = 1.201, Inf(c,C) = 0.899$	$Inf(c,A) = 0.25, Inf(c,B) = 0.36, Inf(c,C) = 0.331$	
InR(c,f_i)	$InR(c,A) = 0.156, InR(c,B) = 0.432$	$InR(c,A) = 0.331, InR(c,C) = 0.467$	$InR(c,B) = 0.512, InR(c,C) = 0.313$	$InR(c,A) = 0.093, InR(c,B) = 0.154, InR(c,C) = 0.115$	
InI(c)	0.156	0.331	0.313	0.093	

$$\because \mathrm{InR}(c,f_i) = \frac{\mathrm{Inf}(c,f_i)}{\mathrm{Inf}(f_i)} = \frac{\sum_{f_i \cdot j \in T(c)} \mathrm{Inf}_{min}(f_i \cdot j)}{\mathrm{Inf}(f_i)},$$

$$\mathrm{Inf}_{f_i \cdot j \in T(c)}(f_i \cdot j) < \mathrm{Inf}_{f_i \cdot j \in T(c')}(f_i \cdot j),$$

$$f_i \cdot j \in \mathrm{T}(c) \text{ and } f_i \cdot j \in \mathrm{T}(c'),$$

$$\Rightarrow \sum_{f_i \cdot j \in T(c)} \mathrm{Inf}_{min}(f_i \cdot j) < \sum_{f_i \cdot j \in T(c')} \mathrm{Inf}_{min}(f_i \cdot j),$$

$\Rightarrow \mathrm{Inf}(c,f_i) < \mathrm{Inf}(c',f_i)$. As $\mathrm{Inf}(f_i)$ is a positive constant,

$$\therefore \mathrm{InR}(c,f_i) < \mathrm{InR}(c',f_i).$$

$$\mathrm{InI}(c) = \min_{i=1}^{k}\{\mathrm{InR}(c,f_i)\},$$

$$\min_{i=1}^{m}\{\mathrm{InR}(c,f_i)\} < \min_{i=1}^{m}\{\mathrm{InR}(c',f_i)\},$$

Assumed $f_l \in c$, $f_l \notin c'$.

If $\mathrm{InR}(c,f_l) = \min_{i=1}^{k}\{\mathrm{InR}(c,f_i)\}$, then $\min_{i=1}^{k}\{\mathrm{InR}(c,f_i)\} < \min_{i=1}^{m}\{\mathrm{InR}(c',f_i)\}$ holds. Otherwise, the inequality is also true.

$\therefore \mathrm{InI}(c) < \mathrm{InI}(c')$. So proof is completed. The lemma holds. □

Theorem 1 (Apriori principle of high influence co-location patterns) *If a co-location pattern c is with high influence, then all its sub-patterns $c' \subseteq c$ are also with high influence. Conversely, if a co-location pattern c is not with high influence, then all of its super-patterns $c''(c \subseteq c'')$ must not be with high influence.*

Proof Based on Lemma 1, Theorem 1 is clearly true. □

3.3 High influence co-location pattern mining algorithm

We propose the HICPM algorithm for mining high influence co-location patterns with a pruning strategy. The pseudo code is listed in Table 4.

The description of pseudo code is as follows:

(1) Set variables ω_1, ω_2, R, $\mathrm{InI}_{threshold}$, compute inverse covariance matrices, and use R for generating neighbor pairs, as for data preprocessing;
(2) Generate 1-size candidate co-location patterns (i.e. features) and table-instances (i.e. instances) (steps 1–2); compute cosine similarity and Mahalanobis distance then generate similarity of attributes for any neighbor pair (steps 3–7); generate star-type neighborhoods and compute influence of instances and features (steps 8–13);
(3) Initialize data structure; start the iteration from 2-size candidate co-location patterns: *k*-size candidate co-location patterns come from *k-1*-size high influence co-

Table 4 The pseudo code of HICPM algorithm (key notations please refer to Table 1)

Algorithm 1: HICPM algorithm

Input: $F=\{f_1, f_2, ..., f_m\}$, $O=\{O_1, O_2, ..., O_n\}$, R, $\text{InI}_{threshold}$;

Output: All the high influence co-location patterns satisfying $\text{InI}_{threshold}$.

Variables: ω_1, ω_2, k, Nb, , Sim, D^2, S, SN, Ne, Ase, Inf, C_k, SI_k, CI_k, E_k

Nb: neighbor pairs of spatial instances

SN=$\{SN_{f_1}, SN_{f_2}, ..., SN_{f_m}\}$: star-type neighbor sets of features

C_k : k-size candidate co-location patterns

SI_k: star-type instances set of co-location patterns C_k

CI_k: table-instances set of co-location patterns C_k

E_k : k-size high influence co-location patterns

Data Preprocessing:

1) Nb = gen_neighbor_pairs(F, O, R);
2) Icm = gen_inverse_covariance_matrix(O);
3) Set ω_1, ω_2, R, $\text{InI}_{threshold}$;

Steps:

1) k=1; C_1=F;
2) CI_1 = gen_table_instances(C_1, O);
3) While (for any neighbor pair in Nb) {
4) Sim = compute_ cosine_similarity(Nb);
5) D^2 = compute_Mahalanobis_distance(Nb, Icm);
6) S = compute_similarity(Sim, D^2);
7) }
8) SN=gen_star_neighborhoods(F, Nb);
9) While (for any instance in SN) {
10) Ne = compute_neighbor_entropy(SN);
11) Ase = compute_attribute_similarity_entropy(S, SN);
12) Inf = compute_influence_of_instances(Ne, Ase, ω_1, ω_2, SN);
13) }
14) k=2; E_1=F; Initialize data structure C_k, CI_k, E_k to be empty;
15) While (not empty E_{k-1}) {
16) C_k = gen_candiate_colocations(E_{k-1});
17) For i in 1 to n do
18) for t $\in SN_{f_i}$, where $f_i = cf_1$, cf_1 is the first feature of $C_k(cf_1, ..., cf_k)$;
19) SI_k = filter_star_instances(C_k, t);
20) end do
21) if k=2 then CI_k = SI_k;
22) else do C_k = select_coarse_highInfluence_colocations(C_k, SI_k, $InI_{threshold}$);
23) CI_k = filter_clique_instances(C_k, SI_k);
24) end do
25) E_k = select_highInfluence_colocations(C_k, CI_k, Inf, $\text{InI}_{threshold}$);
26) k=k+1;
27) }
28) Return U(E_2, ..., E_k);

location patterns, where any sub-sets of a candidate co-location pattern were not high influential, the candidate co-location pattern would be pruned (steps 14–16); generate star-type instances for k-size candidate co-location patterns (steps 17–20); As 2-size star-type instances are clique ones, they can be directly processed (step 21); for 3-size or larger size, it's necessary to check if the star-type instances were clique ones, before that, the candidate co-location patterns would be coarsely filtered, i.e. if influence ratio of star-type instances in a candidate co-location pattern was less than preset $\text{InI}_{threshold}$, the candidate co-location pattern would be pruned (step 22); generate the clique instances for k-size candidate co-location patterns (steps 23–24); generate k-size high influence co-location patterns (step 25); continue the iteration and return all sizes high influence co-location patterns (steps 26–28).

3.4 Time complexity

The HICPM algorithm shares the same process of forming star-neighborhoods and cliques with join-less. With reference to the time complexity analysis of join-less [6], we compare the time complexity of the two algorithms at first:

Let T_{hi}, T_{jl} be the time cost for HICPM and join-less respectively, S denotes the input spatial data sets, $T_{star\_neighborhoods}(S)$ denotes the time cost for materialization of star-type neighborhoods from neighbor pairs set Nb, $T_{hi}(2), T_{jl}(2)$ denotes the time cost that HICPM and join-less respectively spend for finding 2-size co-location patterns, $\sum_{k>2} T_{hi}(k)$, $\sum_{k>2} T_{jl}(k)$ denotes the time cost that HICPM and join-less respectively spend for finding k-size ($k>2$) co-location patterns, $T_{compu\_influ}(S)$ denotes the time cost that HICPM spends for computing the influence of instances and features, $T_{compu\_InI(c)}(2)$, $T_{compu\_PI(c)}(2)$ denotes the time cost that HICPM and join-less respectively spend for computing the influence index or participation index for 2-size co-location patterns, $T_{compu\_InI(c)}(k)$, $T_{compu\_PI(c)}(k)$ denotes the time cost that HICPM and join-less respectively spend for computing the influence index or participation index for k-size co-location patterns. Then,

$$\because T_{hi} = T_{gen\_neighborhoods}(S) + T_{star\_neighborhoods}(S) + T_{hi}(2) + \sum_{k>2} T_{hi}(k)$$

$$T_{jl} = T_{gen\_neighborhoods}(S) + T_{star\_neighborhoods}(S) + T_{jl}(2) + \sum_{k>2} T_{jl}(k)$$

$$T_{hi} - T_{jl} = T_{hi}(2) - T_{jl}(2) + \sum_{k>2}(T_{hi}(k) - T_{jl}(k))$$

$$= T_{compu\_influ}(S) + T_{compu\_InI(c)}(2) - T_{compu\_PI(c)}(2) + \sum_{k>2}(T_{compu\_InI(c)}(k) - T_{compu\_PI(c)}(k))$$

As $T_{compu\_influ}(S) > 0$, $T_{compu\_InI(c)}(2) > T_{compu\_PI(c)}(2)$, $T_{compu\_InI(c)}(k) > T_{compu\_PI(c)}(k)$

$$\therefore T_{hi} > T_{jl}$$

So we know that the HICPM algorithm spends more time than join-less does.

Secondly, we analyze the time complexity of HICPM algorithms:

Let n denotes the number of instances, $|C_k|$ denotes the number of k-size candidate patterns, $t_{compu\_InI(c)}(k)$ denotes the average time cost that HICPM spends for computing the influence index for k-size co-location patterns, c denotes a constant.

Table 5 Comparison on Top-5 co-location patterns

	2-size patterns	3-size patterns	4-size patterns
Top-5 high influence co-location patterns (mined by HICPM algorithm)	{B,F} 0.234660 {A,B} 0.227399 {B,H} 0.179887 {B,I} 0.177815 {A,H} 0.136592	{A,B,H} 0.022428 {A,H,I} 0.016476 {A,B,I} 0.015820 {B,F,H} 0.013153 {B,H,I} 0.012246	{A,B,H,I} 0.002328 {A,D,H,I} 0.001127 {A,F,H,I} 0.001034 {A,B,D,H} 0.001019
Top-5 prevalent co-location patterns (mined by join-less algorithm)	{A,H} 0.057796 {B,F} 0.055469 {A,B} 0.053645 {A,I} 0.050394 {B,H} 0.044271	{A,H,I} 0.007392 {A,B,H} 0.006510 {A,B,I} 0.004167 {B,F,H} 0.003516 {B,H,I} 0.003385	Null

$$\begin{aligned}\because T_{hi} &= T_{gen\_neighborhoods}(S) + T_{star\_neighborhoods}(S) + T_{hi}(2) + \sum_{k>2} T_{hi}(k)\\ &= T_{gen\_neighborhoods}(S) + T_{star\_neighborhoods}(S) + T_{compu\_influ}(S)\\ &\quad + T_{compu\_InI(c)}(2) + \sum_{k>2}(T_{compu\_InI(c)}(k))\end{aligned}$$

As the data preprocessing takes time of $O(n^2)$ to generate the spatial relationships between instances, and the number of instances is usually much larger than the number of features and the average number of neighbors per instance.

$$\begin{aligned}\therefore T_{hi} &= T_{gen\_neighborhoods}(S) + T_{star\_neighborhoods}(S) + T_{compu\_influ}(S)\\ &\quad + T_{compu\_InI(c)}(2) + \sum_{k>2}(T_{compu\_InI(c)}(k)\\ &\le O(n^2) + c \times O(n) + |C_k| \times t_{compu\_InI(c)}(k)\\ &\approx O(n^2)\end{aligned}$$

Therefore, the time complexity of HICPM algorithm is $O(n^2)$.

4 Experimental evaluation

The experiments use a synthetic data set and a real data set, performed by R 3.5.2 and Java 1.7.0_51, Java SE 1.7.0_51-b13, Java Server VM (24.51-b03, mixed mode), run on a normal PC with Intel core i7-6700 CPU @ 3.40 GHz, 3.41 GHz, 16.0 GB memory, Windows 10 (64-bit). We use R for synthesizing data and calculating the Icm and use Java for executing other tasks. The real data set and synthetic data set come from Beijing's points of interest (POIs) with attributes, downloaded from Baidu Map API on April 22, 2018 [35].

4.1 Data description

The real data set contains 16,307 instances with 8-dimensional attributes, i.e. unit_price, overall_rating, service_rating, facility_rating, hygiene_rating, image_num, groupoon_num and comment_num.

The synthetic data set contains 23,083 instances with 8-dimensional attributes, i.e. ave_price, person_visit, investment, turnover, loyalty, comment_num, rank_rating, complaint_num, synthesized by R with the means and variances referring to the annual growth rate of industries in Beijing in 2018.[1]

The two data sets involve 9 kinds of features i.e. beauty store, restaurant, school, enterprise, hospital, hotel, residence, health-care center, shop.

4.2 Effectiveness of HICPM algorithm

In this section, the effectiveness of HICPM algorithm is shown on the real data set, by comparison with that of join-less algorithm.

(1) Comparison on Top-5 Co-location Patterns Mined by HICPM Algorithm and Join-less Algorithm

When the experimental variables are set with $R = 10$ m, $\omega_1 = 0.2, \omega_2 = 0.8$, $\text{InI}_{threshold} = \text{PI}_{threshold} = 0.001$, we can see in Table 5 that the HICPM algorithm can find patterns that the join-less algorithm miss, i.e. the 2-size pattern {B, I} and 4-size patterns {A, B, H, I}, {A, D, H, I}, {A, F, H, I}, {A, B, D, H}. Besides, the mining results show that another 2-size patterns {C, F}, {B, C} and twelve 3-size patterns {A, F, I}, {B, D, F}, etc. can solely be mined by HICPM algorithm. That is because when R is small, there are few instances found in neighborhoods, the join-less algorithm seldom find prevalent co-location patterns as the participation ratios are too small to satisfy the threshold, however, the HICPM algorithm uses the more effective InI which is only related to attributes of instances and regardless of the prevalence.

Both of them can find the same size co-location patterns at same rank, e.g. 3-size patterns {A, B, I}, {B, F, H}, {B,

[1] National bureau of statistics of China: http://data.stats.gov.cn/search.htm. Accessed 20 Jan 2019.

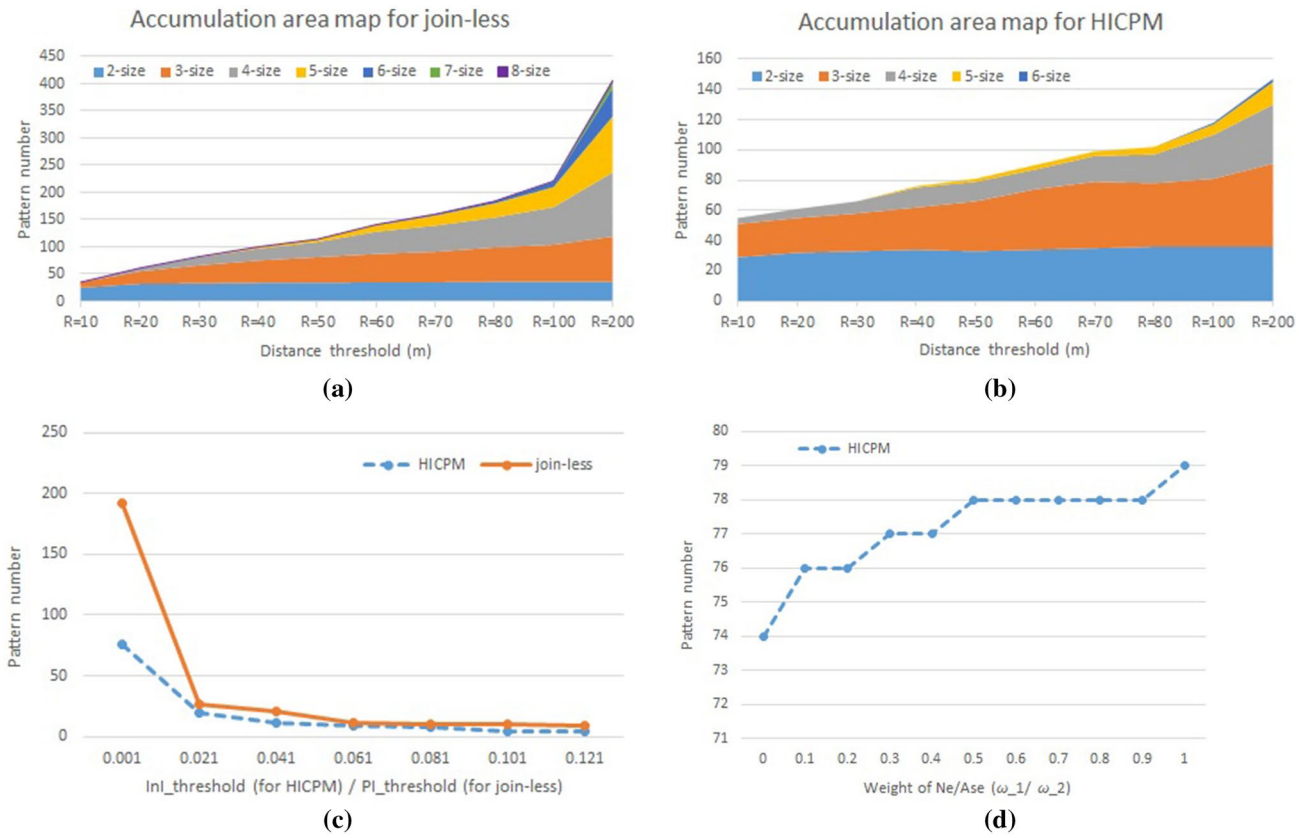

Fig. 5 Effect of distance threshold [for join-less in (**a**) and HICPM in (**b**)], $\mathrm{InI}_{threshold}/\mathrm{PI}_{threshold}$ [for HICPM/join-less in (**c**)] and weight ω_1/ω_2 [for HICPM in (**d**)], on pattern number

H, I}, or the same size patterns at different rank, e.g. 2-size patterns {B, F}, {A, B}, {B, H}, {A, H} and 3-size patterns {A, B, H}, {A, H, I}. It happens because the two algorithms share the same R, i.e. they use same table-instances, so some table-instances with high influence and participation ratio can be found. However, the assumption in this paper that influence of instances only exists in neighborhood is set manually to simplify the analysis, once influence range is set beyond R, more co-location patterns with high influence will be solely discovered by HICPM algorithm.

Besides, when $R = 20$ m while other variables remain, the results show that the two algorithms can mine same number of co-location patterns at diverse sizes, i.e. the 2-size patterns are identical, one pattern is different in 3-size patterns and in 4-size patterns respectively. When R varies in [30, 200] while other variables remain, the join-less algorithm mines more co-location patterns at all size and finds co-location patterns at higher size. Although most of high influence patterns can also be found by join-less algorithm, a few patterns can only be mined by the HICPM algorithm. It appears that the two algorithms show similar characteristics at higher size when $\mathrm{InI}_{threshold}$ is set at a relatively low level. Nevertheless, the HICPM algorithm can sequence the patterns as per their influential level. Therefore, the HICPM algorithm can be reckoned as a new co-location pattern mining approach different from previous ones.

(2) Comparison on Mining Results of HICPM Algorithm and Join-less Algorithm

The mining results of the two algorithms are compared with the varying variables.

- Effect of distance threshold on pattern number

In Fig. 5a, b, $\omega_1 = 0.2$, $\omega_2 = 0.8$, $\mathrm{InI}_{threshold} = \mathrm{PI}_{threshold} = 0.001$, R varies in [10, 200]. When $R = 10$ m, the HICPM algorithm can discover 2-size patterns four more, 3-size patterns twelve more, 4-size patterns four more, than the patterns mined by join-less algorithm. When $R = 20$ m, the two algorithms mine same number of patterns at all sizes. When $R \geqslant 30$ m, the join-less algorithm can mine more co-location patterns, as more cliques appear, the participation ratio (PR) grow faster than InR. The opposite is true when $R < 20$ m.

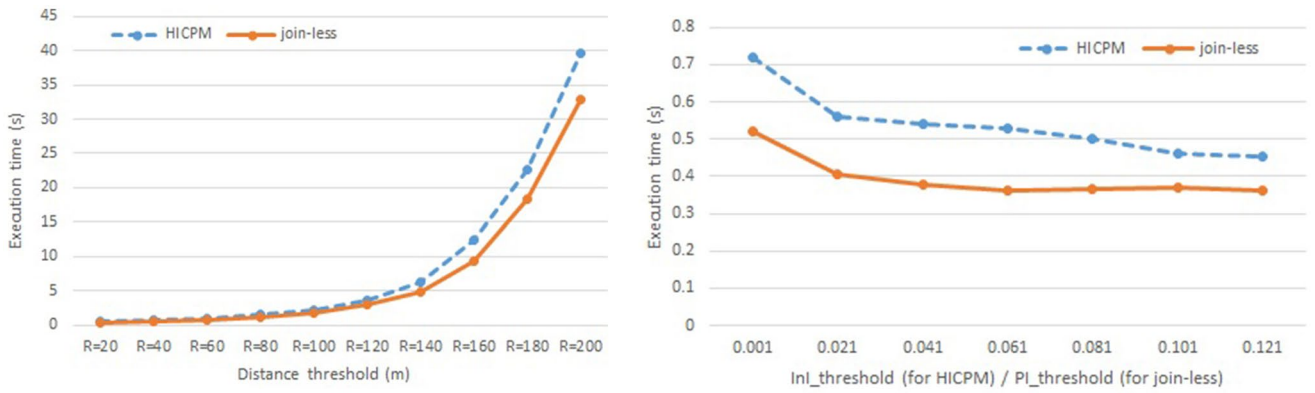

Fig. 6 Comparison on time performance of HICPM algorithm and join-less algorithm running on the real data set, with the variation of R in (**a**) and $\mathrm{InI}_{threshold}/\mathrm{PI}_{threshold}$ in (**b**)

- Effect of $\mathrm{InI}_{threshold}/\mathrm{PI}_{threshold}$ on pattern number

In Fig. 5c, $R=40$ m, $\omega_1 = 0.2$, $\omega_2 = 0.8$, $\mathrm{InI}_{threshold}/\mathrm{PI}_{threshold}$ varies in [0.001, 0.121]. It can be seen that pattern number falls sharply when the $\mathrm{InI}_{threshold}/\mathrm{PI}_{threshold}$ rises from 0.001 to 0.021. The reason is that most of patterns have their InI/participation index (PI) below 0.001.

- Effect of weight ω_1/ω_2 on pattern number

In Fig. 5d, $R=40$ m, $\mathrm{InI}_{threshold}=\mathrm{PI}_{threshold}=0.001$, the weight ω_1/ω_2 varies in [0, 1]. The figure depicts the growth of pattern number when ω_1 rises from 0 to 1 (i.e. ω_2 falls from 1 to 0, as $\omega_1 + \omega_2 = 1$). It shows that when ω_1 rises, that means the importance of neighborhood number increases or the importance of attribute similarity decreases, we can find more high influence patterns. A compromise is made for balance as $\omega_1 = 0.2$, $\omega_2 = 0.8$ in this paper.

4.3 Performance of HICPM Algorithm

In this section (1), time performance of HICPM algorithm with variation of thresholds is shown on the real data set and the synthetic data set, by comparison with that of join-less algorithm. In this section (2), scalability of HICPM algorithm with variation of the number of instances, attribute dimensions and features is verified on the synthetic data set, by comparison with that of join-less algorithm.

(1) Comparison on time performance of HICPM algorithm and join-less algorithm running on the real data set and synthetic data set

Time performance of the algorithms is shown with the variation of two kinds of threshold in Fig. 6 (using real data) and Fig. 7 (using synthetic data).

- Effect of R on time performance

In Fig. 6a, $\omega_1 = 0.2$, $\omega_2 = 0.8$, $\mathrm{InI}_{threshold}=\mathrm{PI}_{threshold}=0.001$, R varies in [20, 200]. It shows that the curves run closely, the gap between them expands in general with the rise of R. The gap is 0.189 s(s) at $R=20$ m as the minimum, and is 6.769 s at $R=200$ m as the maximum. The HICPM algorithm runs slightly slower than the join-less algorithm as the former uses the same clique-forming of the latter and runs more steps to calculate the influence of features.

- Effect of $\mathrm{InI}_{threshold}/\mathrm{PI}_{threshold}$ on time performance

In Fig. 6b, $R=40$ m, $\omega_1 = 0.2$, $\omega_2 = 0.8$, $\mathrm{InI}_{threshold}/\mathrm{PI}_{threshold}$ (InI/PI) varies in [0.001, 0.121]. It can be seen that the curves decline with the rise of InI/PI and the curve of HICPM algorithm runs over that of join-less algorithm. The gap between curves is 0.091 s at InI/PI = 0.121 as the minimum, and is 0.197 s at InI/PI = 0.001 as the maximum. Additional experiments conducted with InI/PI < 0.001 indicate that most of the features have InI and PI values concentrated in (0, 0.001].

- Effect of thresholds on time performance

The following curves in Fig. 7 reflect the time performance of the two algorithms running on the synthetic data set with the same experimental conditions as aforementioned for analyzing Fig. 6.

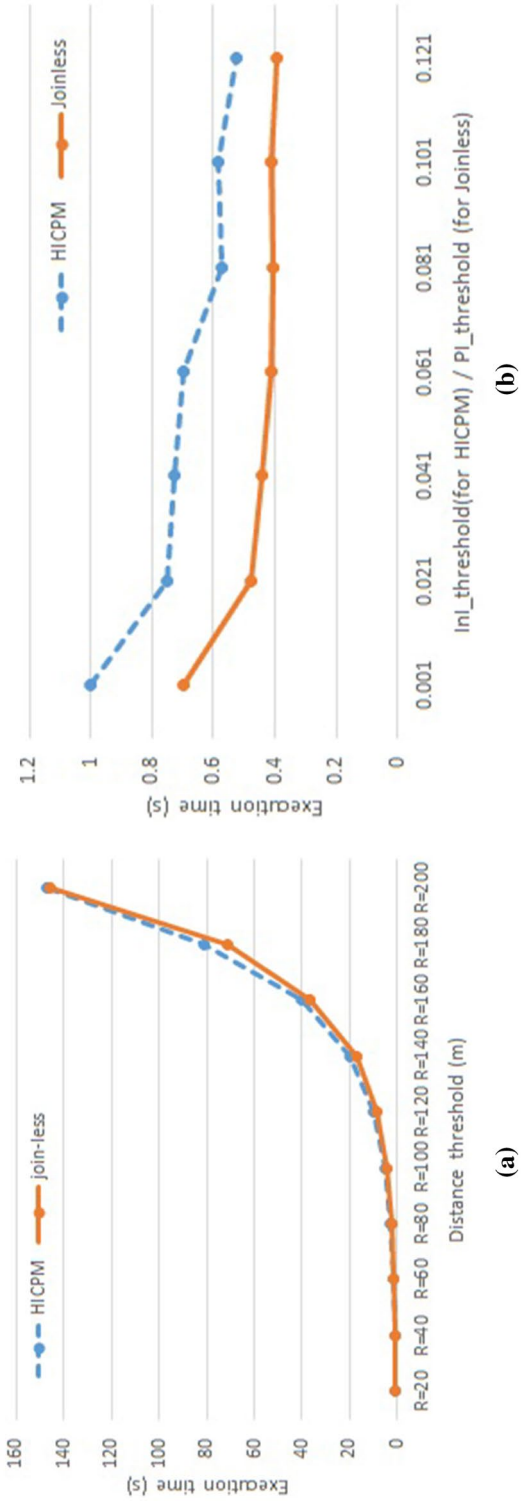

Fig. 7 Comparison on time performance of HICPM algorithm and join-less algorithm running on the synthetic data set, with the variation of R in (**a**) and InI/PI in (**b**)

In Fig. 7a, the minimal gap between curves is 0.33 s at $R=40$ m, and the maximal gap is 9.685 s at $R=180$ m. In Fig. 7b, the minimal gap is 0.128 s at InI/PI = 0.121 and the maximal gap is 0.305 s at InI/PI = 0.001.

(2) Comparison on scalability of HICPM algorithm and join-less algorithm running on the synthetic data set

To verify the scalability, we test the scalability of the algorithms on the synthetic data set. Variation of variables share the same experimental conditions: $R=40$ m, $\omega_1 = 0.2$, $\omega_1 = 0.8$, $\text{InI}_{threshold}=\text{PI}_{threshold}=0.001$.

- Effect of number of instances on scalability

Figure 8a depicts that the two curves run closely, the gap between them expands in general with the increase of instances. The minimal gap is 0.033 s at n = 600, while the maximal gap is 0.262 s at n = 20,600. It demonstrates from actual data that the efficiency of the two algorithms is different as they run at same level of time complexity.

- Effect of number of attribute dimensions on scalability

It can be seen in Fig. 8b that the two algorithms are insensitive to the variation of number of attribute dimensions, as the points on the curves fluctuate in a tiny range which is approximately 4%. So it seems hopeful to apply more dimensional attributes to explore more details of spatial relationships and changes. Time performance of the HICPM algorithm is in average 0.257 s more than that of the join-less algorithm, it is the cost for the former to run more steps.

- Effect of number of features on scalability

In Fig. 8c, it is shown that the curve of the HICPM algorithm runs above that of the join-less algorithm, while the curves rise steeply at f = 6, 8 and 9, as the number of instances belonging to the features at f = 6, 8, 9 is much more than the number of instances of features at f = 3, 4, 5 and 7. It is also noticed that the gap between the curves changes a lot at f = 8, the repeated experiments reveal that it may be a bit deviation caused by computer system.

5 Conclusion

The main work of this paper is to mine high influence co-location patterns from spatial instances with attributes. Based on number of neighbors and similarity between instances, the InI of features in co-location patterns is calculated for mining high influence co-location patterns.

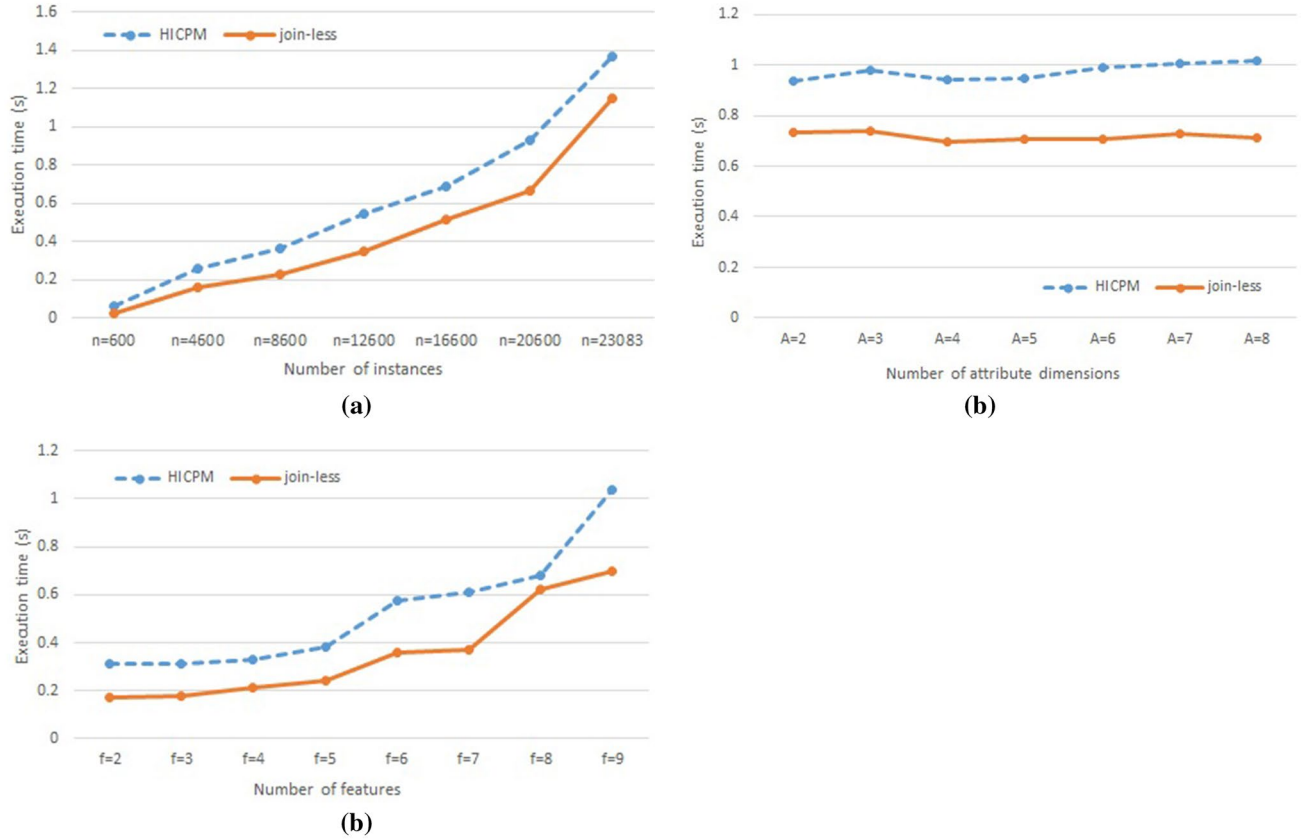

Fig. 8 Comparison on scalability of HICPM algorithm and join-less algorithm with the variation of number of instances in (**a**), number of attribute dimensions in (**b**) and number of features in (**c**)

This paper proves the InI satisfies the downward closure property, and proposes a HICPM algorithm with pruning strategy. With extensive experiments and comparisons on real and synthetic data sets, we analyzed the effectiveness, performance and scalability of the HICPM and join-less algorithms and found HICPM could discover high influence co-location patterns, at a time cost level slightly higher than join-less. A high influence co-location pattern can be reckoned as a concise compression of traditional co-location patterns in the aspect of influence. This paper establishes a basic framework for mining high influence co-location patterns, many details shall be probed further, e.g. flexible influential distance, principal component analysis for high-dimensional data [36], weight of attributes and influence of extended spatial objects.

Acknowledgement This work was supported in part by Grants (Nos. 61966036, 61662086) from the National Natural Science Foundation of China, and in part by the Project of Innovation Research Team of Yunnan Province (No. 2018HC019), and in part by the Research and Innovation Project of Yunnan University (No. 2018216).

References

1. Koperski K, Han J (1995) Discovery of spatial association rules in geographic information databases. In: Proceedings of international symposium on large spatial data bases, Portland, ME, pp 47–66
2. Shekhar S, Huang Y (2001) Discovering spatial co-location patterns: a summary of results. In: Proceedings of the 7th international symposium on advances in spatial and temporal database (SSTD). Springer, Berlin, pp 236–240
3. Huang Y, Shekhar S, Xiong H (2004) Discovering colocation patterns from spatial data sets: a general approach. IEEE Trans Knowl Data Eng (TKDE) 16(12):1472–1485
4. Yoo JS, Shekhar S (2004) A partial join approach for mining co-location patterns. In: Proceedings of the 12th annual ACM international workshop on geographic information systems. ACM Press, pp 241–249
5. Yoo JS, Shekhar S, Celik M (2005) A join-less approach for co-location pattern mining: a summary of result. In: Proceedings of the 5th IEEE international conference on data mining (ICDM). IEEE Press, pp 813–816
6. Yoo JS, Shekhar S (2006) A join-less approach for mining spatial co-location pattern. IEEE Trans Knowl Data Eng (TKDE) 18(10):1323–1337

7. Wang L, Bao Y, Lu Z (2009) Efficient discovery of spatial co-location patterns using the iCPI-tree. Open Inf Syst J 3(1):69–80
8. Wang L, Bao Y, Lu J et al. (2008) A new join-less approach for co-location pattern mining. In: Proceedings of the IEEE 8th international conference on computer and information technology (CIT2008). The IEEE Computer Society Press, Piscataway, NJ, pp 197–202
9. Wang L, Zhou L, Lu J et al (2009) An order-clique-based approach for mining maximal co-locations. Inf Sci 179(19):3370–3382
10. Bembenik R, Jozwicki W, Protaziuk G (2017) Methods for mining co-location patterns with extended spatial objects. Int J Appl Math Comput Sci 27:681–695
11. Xiong H, Shekhar S, Huang Y, Kumar V, Ma XB, Yoo JS (2008) A framework for discovering co-location patterns in data sets with extended spatial objects. In: Proceedings of the 4th SIAM international conference on data mining, Lake Buena Vista, Florida, USA. ACM Press, California, pp 1–10
12. Kim SK, Lee JH, Ryu KH (2014) A framework of spatial co-location pattern mining for ubiquitous GIS. Multimed Tools Appl 71:199–218
13. Li JD, Adilmagambetov A, Jabbar MSM, Zaiane OR, Osornio-Vargas A, Wine O (2016) On discovering co-location patterns in datasets: a case study of pollutants and child cancers. Geoinformatica 20:651–692
14. Chen L (2017) Spatial high impact co-location pattern mining. Dissertation, Yunnan University (in Chinese)
15. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on management of data. ACM, Washington, DC, pp 207–216
16. Giacometti A, Li DH, Marcel P et al (2014) 20 years of pattern mining: a bibliometric survey. ACM SIGKDD Explor Newsl 15(1):41–50
17. Bayardo R J (1998) Efficiently mining long patterns from databases. In: Proceedings of 1998 ACM-SIGMOD international conference on management of data. ACM, Seattle, WA, pp 85–93
18. Pan F, Tung A K H, Cong G et al. (2004) COBBLER: combining column and row enumeration for closed pattern discovery. In: Proceedings of 2004 international conference on scientific and statistical database management, Santorini Island, Greece, pp 21–30
19. Wang C, Zheng X (2019) Application of improved time series Apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint. Evolutionary intelligence. Springer, Berlin. https://doi.org/10.1007/s12065-019-00234-5
20. Huang Y, Zhang L, Yu P (2005) Can we apply projection-based frequent pattern mining paradigm to spatial co-location mining? In: Proceedings of the Pacific-Asia conference on the methodologies for knowledge discovery and data mining (PAKDD 2005). Springer, Berlin, pp 719–725
21. Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Proceedings of the 7th international conference on database theory (ICDT'99), Jerusalem, Israel, pp 398–416
22. Wang L, Han J, Chen H et al (2016) Top-k probabilistic prevalent co-location mining in spatially uncertain data sets. Front Comput Sci 10(3):1–16
23. Wang L, Chen H, Zhao L et al. (2010) Efficiently mining co-location rules on interval data. In: Proceedings international conference on advanced data mining and applications, ADMA 2010, Springer, Berlin, pp 477–488
24. Ouyang Z, Wang L, Wu P (2017) Spatial co-location pattern discovery from fuzzy objects. Int J Artif Intell Tools 26:2
25. Lin C, Lan G, Hong T (2012) An incremental mining algorithm for high utility itemsets. Expert Syst Appl 39(8):7173–7180
26. Chefrour A (2019) Incremental supervised learning: algorithms and applications in pattern recognition. Evol Intell 12:97. https://doi.org/10.1007/s12065-019-00203-y
27. Chai Z, Liang S (2019) A node-priority based large-scale overlapping community detection using evolutionary multi-objective optimization. Evol Intell. https://doi.org/10.1007/s12065-019-00250-5
28. Xiang R, Neville J, Rogati M (2010) Modeling relationship strength in online social networks. In: Proceedings of the 19th international world wide web conference (WWW2010), Raleigh, NC, USA, pp 981–990
29. Sathanur AV, Jandhyala V (2014) An activity-based information-theoretic annotation of social graphs. In: Proceedings of the 2014 ACM conference on web science (WEBSCI 2014), Bloomington, USA, pp 187–191
30. Peng S, Wang G, Yu S (2013) Mining mechanism of top-k influential nodes based on voting algorithm in mobile social networks. In: Proceedings of the 11th IEEE/IFIP international conference on embedded and ubiquitous computing (EUC 2013), Zhangjiajie, China, pp 2194–2199
31. Bakshy E, Hofman JM, Mason WA, Watts DJ (2011) Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the 4th ACM international conference on web search and data mining (WSDM 2011), Hong Kong, China, pp 65–74
32. Huang J, Cheng X, Shen H, Zhou T, Jin X (2012) Exploring social influence via posterior effect of word-of-mouth recommendations. In: Proceedings of the 5th ACM international conference on web search and data mining (WSDM 2012), Seattle, Washington, USA, pp 573–582
33. Peng SC, Yang AM, Cao LH, Yu S, Xie DQ (2016) Social influence modeling using information theory in mobile social networks. Inf Sci 379:146–159
34. Wang L, Chen H (2014) Spatial pattern mining theory and method. Science Press, Beijing **(in Chinese)**
35. Baidu Map API (developer interface) (2018) Baidu Map Open Platform. http://lbsyun.baidu.com/. Accessed 22 Apr 2018
36. Behdad M, French T, Barone L et al (2012) On principal component analysis for high-dimensional XCSR. Evol Intell 5:129. https://doi.org/10.1007/s12065-012-0075-6