Fuzzy Systems and Data Mining VIII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220365

Mining Local Tight Spatial Sub-Prevalent Co-Location Patterns

Qiuqing He, Hongmei Chen[1], and Qing Xiao
School of Information Science and Engineering, Yunnan University, Yunnan, China

Abstract. Spatial sub-frequent co-location patterns reveal the rich spatial relationship of spatial features and instances, which are widely used in real applications such as environmental protection, urban computing, public transportation, and so on. Existing sub-frequent pattern mining methods cannot distinguish patterns whose row instance spatial distributions are significantly different. Additionally, patterns whose row instances are tightly located in a local area can further reveal the particularity of the local area such as special environments and functions. Therefore, this paper proposes mining Local Tight Spatial Sub-frequent Co-location Patterns (LTSCPs). First, a relevancy index is presented to measure the local tightness between sub-frequent pattern row instances by analyzing mutual participation instances between row instances. The concept of LTSCPs is then proposed followed by an algorithm for mining these LTSCPs. Finally, a large number of experiments are carried out on synthetic and real datasets. The results show that the algorithm for mining LTSCPs is efficient and LTSCPs are practical.

Keywords. Spatial data mining, sub-prevalent co-location pattern, local tight spatial sub-frequent co-location patterns, relevancy index.

1. Introduction

Spatial co-location patterns are an important type of spatial pattern which are widely used in many fields. For example, botanists mined co-location patterns in plant data and found that 80% of the semi-humid evergreen broad-leaved forest grow orchids [1]. A traditional co-location pattern uses the clique instance model, requiring two instances in the pattern to form a clique, which ignores other important non-clique spatial relationships. Therefore, sub-frequent co-location patterns are proposed to find richer spatial relationships [2-3].

However, locally tight row instances in space not only help to understand deeply the spatial relationship of features and instances, but also help to identify special environments or functions of the local spatial area. Figure 1 shows an example of a spatial data set with six spatial features. Table 1 shows that when the participation index threshold is 0.3, the pattern {A, B, C} and pattern {D, E, F} are both sub-frequent patterns. But as can be seen from Figure 1, the row instances of the pattern {A, B, C} are closely distributed in the local area shown by the dotted circle, while the row instances of the pattern {D, E, F} are loosely distributed throughout the spatial region, which are two distinct sub-frequent patterns. Moreover, the pattern of closely distributed row instances {A, B, C} can further reveal the characteristics of this local area; for example, the local area

[1] Corresponding Author, Hongmei Chen; E-mail: hmchen@ynu.edu.cn.

may be a tourist service center. However, existing sub-frequent pattern mining algorithms cannot distinguish these two patterns with significantly different distributions of row instances.

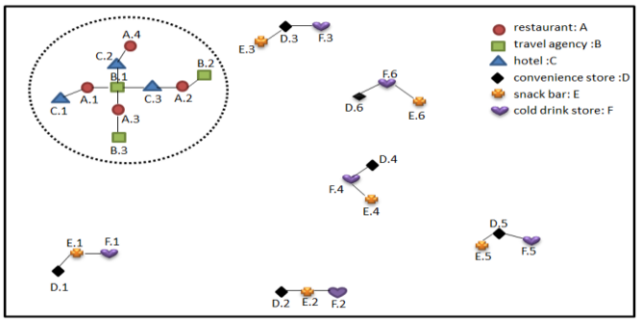

Figure 1. An example of a spatial dataset.

Thus, the following contributions are made by this paper. First, by analyzing the mutually participating instances in the row instances of the pattern, the correlation degree can be defined to measure the degree of local compactness of the pattern row instances. The concept of a local tight sub-prevalent co-location pattern is then proposed. Second, an efficient algorithm is developed, named the local tight sub-prevalent co-location pattern mining algorithm. Third, the efficiency of the proposed algorithm is evaluated and the practicability of local tight sub-frequent co-location patterns is verified.

2. Related Work

The traditional co-location pattern based on the clique instances model was first proposed by Huang et al. Then, a join-based pattern mining algorithm was proposed [4]. In order to address the problem of low efficiency caused by too many connection operations, the partial join algorithm [5] and the join-less algorithm [6] were proposed successively. In addition, an algorithm for mining maximum frequent co-location patterns and closed frequent co-location patterns was proposed [7]. Exploring the upward inclusion property of negative co-location patterns, a minimal negative co-location pattern was proposed [8]. A method for mining sub-prevalent co-location patterns based on graph databases was explored [9]. The authors then developed the star instance model and sub-frequent co-location patterns, in addition, validated two efficient mining algorithms, namely PTBA and PBA [2]. Based on sub-frequent patterns, an effective mining algorithm was proposed [10]. Considering that the location of spatial instances changes with time, an effective spatio-temporal sub-frequent pattern mining algorithm was proposed [11].

Researchers have carried out many related studies on spatial instance distributions and table instance distributions. An efficient local region pattern mining method based on user-specified local regions was proposed [12]. Considering the wideness of a spatial instance distribution, pattern mining with a wide spatial instance distribution using information entropy as the measure of interest was proposed [13]. Also, a uniform distribution pattern mining method was proposed [14] based on the division of spatial regions.

Different from the above studies, by considering the spatial relationship between the row instances of a pattern, this paper studies sub-frequent co-location patterns and mining algorithms with row instances that are tightly located in a local area.

3. Basic Concepts and Definition

Table 1. Star row instance, *SPR* and *SPI* of pattern {A, B, C} and {D, E, F}

Pattern	Feature	Star participation instance	Star row instance	*SPR*	*SPI*
pattern {A,B,C}	A	A.1	{A.1#,B.1,C.1}	2/4	1/3
		A.2	{A.2#,B.2.C.3}		
	B	B.1	{B.1#,A.1,C.2}	1/3	
			{B.1#,A.1,C.3}		
			{B.1#,A.3,C.2}		
			{B.1#,A.3,C.3}		
	C	C.2	{C.2#,A.4,B.1}	2/3	
		C.3	{C.3#,A.2,B.1}		
pattern {D,E,F}	D	D.3	{D.3#,E.3,F.3}	1/3	1/3
		D.5	{D.5#,E.5,F.5}		
	E	E.1	{E.1#,D.1,F.1}	1/3	
		E.2	{E.2#,D.2,F.2}		
	F	F.4	{F.4#,D.4,E.4}	1/3	
		F.6	{F.6#,D.6,E.6}		

Spatial features are various spatial entities, such as restaurant A and travel agency B in Figure 1. Spatial instances are specific instances of spatial features at a certain spatial location, such as restaurant A.1 and travel agency B.1 in Figure 1. In a spatial data set, let F be a set of n features $F = \{f_1, f_2, \ldots, f_n\}$. Let S be a set of instances of F, $S_i (1 \le i \le n)$ is the instance set of spatial features f_i. Given a distance threshold d, if $dis(i_j, i_k) \le d$, instance i_j satisfies a spatial proximity relationship R, such as the instances connected by solid lines in Figure 1. The star neighborhoods instance of instance i_j is the set consisting of i_j and the other spatial instances located within distance d from i_j, that is, $SNsI(i_j) = \{i_k \mid \text{dis}(i_j, i_k) \le d\}$, the instances i_j and i_k satisfy R. The star participation instance $SPIns(f_i, c)$ is the set consisting of instances of f_i whose star neighborhood instances contain all features in pattern c. Each minimal subset of the star neighbor instances whose features cover all features of pattern c is called a star row instance of pattern c. The star participation ratio is the fraction of instances of f_i that occur in the star participation instance of f_i in pattern c, that is $SPR(f_i, \text{c}) = |SPIns(f_i, \text{c})|/|S_i|$.The star participation index of c is the minimum star participation ratio among all features f_i in c, that is $SPI(c) = \min\{SPR(f_i, \text{c}) \mid f_i \in c\}$.

Given a sub-prevalence threshold *min_sprev,* if the star participation index of pattern c is no less than *min_sprev*, that is, $SPI(c) \ge min\_sprev$, then the pattern c is a sub-prevalent co-location pattern.

Definition 1 ***Neighborhoods Instance (NsI)*****:** $NsI(i_j, c)$ is a set of instances of other features except feature f_i in pattern c in the star neighborhood instance set $SNsI(i_j)$ of instance i_j. *NsI* is defined as:

$$NsI(i_j, c) = \{i_k \mid i_k \in SNsI(i_j), f_k \in c, f_k \in f_i\} \quad (1)$$

As shown in Figure 1, *NsI*(B.1, {A, B, C}) = {A.1, A.3, C.2, C.3}.

Definition 2 ***Center Number (CN)*****:** $CN(i_j, c)$ is the product of the number of instances of each feature that belong to pattern c in the set of neighborhood instances $NsI(i_j, \text{c})$ of i_j. *CN* is defined as:

$$CN(i_j, c) = \prod_{f_k \in c, f_k \neq f_j} | \pi_{f_k} NsI(i_j, c) | \quad (2)$$

Π is the projection operation of the instance set on the features.

As shown in Figure 1, *CN* (B.1, {A, B, C}) =| {A.1, A.3} | * | {C.2, C.3} |=2*2=4.

Definition 3 ***Relevancy Number (RN)*****:** $RN(i_j, c)$ is the sum of the center number of i_j and the center number of each star participation instance in the neighborhood instance of i_j. *RN* is defined as:

$$RN(i_j, c) = CN(i_j, c) + \sum_{i_k \in NsI(i_j, c), i_k \in SPIns(f_k, c)} CN(i_k, c) \quad (3)$$

As shown in Figure 1, *RN*(B.1, {A, B, C}) = *CN*(B.1, {A, B, C}) + *CN*(A.1, {A, B, C}) +*CN*(C.2, {A, B, C}) + *CN*(C.3, {A, B, C}) = 4 + 1 + 1 + 1 = 7.

Definition 4 ***Relevancy Ratio (ReR)*****:** $ReR(i_j, c)$ is the ratio of the relevancy number of i_j to the sum of the ***CN*** of all star participation instances of pattern *c*. *ReR* is defined as:

$$ReR(i_j, c) = \frac{RN(i_j, c)}{\sum_{i_k \in SPIns(f_k, c), f_k \in c} CN(i_k, c)} \quad (4)$$

As shown in Figure 1, *ReR*(B.1, {A, B, C}) = 7 / (1 + 1 + 4 + 1 + 1) = 7/8.

Definition 5 ***Relevancy Index (ReI)*****:** *ReI*(*c*) is the maximum relevancy rate of all star participation instances in pattern *c*. *ReI* is defined as:

$$ReI(c) = \max\{ReR(i_j, c) \mid i_j \in SPIns(f_j, c), f_j \in c\} \quad (5)$$

In Figure 1, *ReR*(A.1,{A,B,C}) = 5/8, *ReR*(A.2,{A,B,C}) = 2/8, *ReR*(B.1,{A,B,C}) = 7/8, *ReR*(C.2,{A,B,C}) = 5/8, *ReR*(C.3,{A,B,C}) = 6/8, thus, *ReI*({A,B,C}) = 7/8.

Definition 6 ***Local Tight Sub-prevalent Co-location Pattern (LTSCP)*****:** An LTSCP is a sub-prevalent co-location pattern whose $ReI(c) \geq min\_rei$, *min_rei* is a relevance threshold.

As shown in Figure 1, if *min_rei* is 0.7, the sub-frequent co-location pattern {A, B, C} is a local tight sub-frequent co-location pattern.

4. Mining Algorithm

Since a sub-frequent co-location pattern satisfies anti-monotonicity [5-6], the pattern search space can be reduced. On the basis of sub-frequent pattern mining, this paper further calculates the local compactness of the pattern, and proposes a local tight sub-frequent co-location pattern mining algorithm (LTSCP algorithm). The specific process is shown in Algorithm 1.

Algorithm 1. LTSCP algorithm

Input. (a) spatial feature set *F*; (b) spatial instance set *S*; (c) neighbor relationship *R*; (d) minimum sub-prevalence threshold *min_sprev*; (e) minimum relevancy threshold *min-rei*

Output. A set of local tight sub-prevalent co-locations

Variables. (a) *SNsI*: star neighborhood instances; (b) *k*: co-location size; (c) P_k: set of *k* size sub-prevalent co-locations; (d) LT_k: set of *k* size local tight sub-prevalent co-locations; (e) *LT*: set of local tight sub-prevalent co-locations

Method.

1) *SNsI* = Gen_Star_Neighs(*F, S, R*) // generate *SNsI*

2) k=2, P1=F
3) while (P_{k-1} is not empty)
4) P_K = Gen_k_SCP(P_{k-1},SN,min_sprev) //generate P_k
5) LT_K = Gen_k_LTSCP(Pk, SN, min_rei) //generate size-k LTSCP
6) for each $p \in P_k$
7) calculate the relevance ratio of pattern p $ReR(i_j, p)$
8) if $ReR(i_j, p) \geq min\_rei$
9) $LT_k \leftarrow p$ // join LT_k
10) $LT \leftarrow LT_k$ // join LT
11) k=k+1
12) end while

Step 1 generates the star neighborhood instance set according to the spatial feature set, the spatial instance set, and the neighbor relationship. In step 2, each feature $f_i \in$ F is considered as the size-1 prevalent co-location for the start of the iteration. Step 4 adopts the join-based method to generate size-k candidate sub-frequent patterns by connecting the size-(k-1) sub-frequent patterns. It then calculates the star participation instance set and participation index of the size-k candidate sub-frequent patterns and obtains the size-k sub-frequent patterns. Steps 5-12 calculate the relevancy index of the size-k local tight candidate patterns and obtains the size-k local tight sub-frequent patterns.

Time complexity. The time complexity of generating a star neighborhood instance set is $O(m^2)$. In the generation of the size-k sub-frequent patterns, first, $|P_{k-1}|$ size-(k-1) sub-frequent patterns are connected to generate $|C_k|$ size-k candidate sub-frequent patterns, and then, according to the star neighborhood instance set of each instance, calculate the $|C_k|$ star participation instance set and participation index of size-k candidate sub-frequent patterns; the time complexity for this is $O(|P_{K-1}|^2+|C_k|)$. In the generation of size-k local tight sub-frequent patterns, according to the star neighborhood instances of $k\bar{m}$ instances, the relevancy index of $|P_k|$ size-k local tight candidate patterns is calculated, and its time complexity is $O(m^2 + \sum_{k=2}^{n}(|P_k|^2 + \bar{m}|C_k|)$. Usually, $|P_k|<|C_k|$, so the time complexity of the LTSCP algorithm is $O(m^2 + \sum_{k=2}^{n}(|P_{k-1}|^2 + \bar{m}|C_k|)$.

5. Experimental results

5.1. Experimental settings

Compared algorithm. In order to evaluate the LTSCP algorithm, an algorithm based on the join-based approach [4] is used to mine sub-frequent co-location patterns, denoted JBSCP. All algorithms are implemented in Python and run on a PC with an Intel Core i7 CPU, 8 GB RAM, Windows 10, and PyCharm 2017.

Data sets. This paper randomly generates three synthetic datasets named Synthetic data 1~3 (S_1, S_2, S_3). In order to analyze the mined patterns, this paper selects two real datasets: Plant-data from the "Three Parallel Rivers Region", as shown in Figure 2, with a banded distribution, and Beijing-POI, shown in Figure 3, with a clustered distribution. Information about these datasets is shown in Table 2. The default settings of parameters are shown in Table 3.

Table 2. Data sets

Dataset		Number of features	Number of instances	Spatial scope
Synthetic data	Synthetic data 1	10	10000	500×500

	Synthetic data 2	10	10000	1000×1000
	Synthetic data 3	25	20000	1000×1000
Real data	Plant-data	31	335	8000×13000
	Beijing-POI	16	23025	22000×14000

Table 3. Default values of the experimental parameters of the LTSCP algorithm

Dataset		*d*	*min_sprev*	*min_rei*
Synthetic data	Synthetic data 1	18	0.2	0.1
	Synthetic data 2	18	0.2	0.1
	Synthetic data 3	18	0.2	0.1
Real data	Plant-data	6000	0.3	0.7
	Beijing-POI	50	0.2	0.4

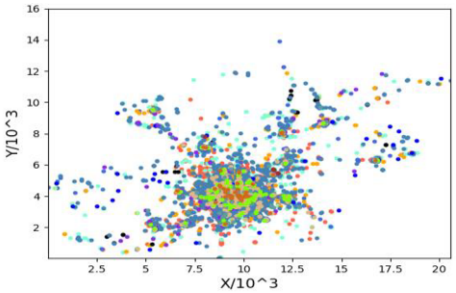

Figure 2. Distribution of the Plant-data dataset.

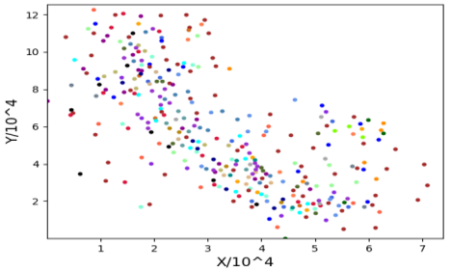

Figure 3. Distribution of the Beijing-POI dataset.

5.2. Influence of different parameters on the efficiency of the LTSCP algorithm

Effect of Distance Threshold *d*. Figure 4 shows that, for all datasets, the running time gradually increases as *d* increases, and as the dataset size increases, so does the running time. Synthetic data 1 has a denser distribution than Synthetic data 2; the effect of *d* is larger, so the running time is also relatively longer. Since Synthetic data 3 has the largest amount of data, with any parameter setting, the runtime is the longest.

Effect of *min_sprev*. Figure 5 shows that the distribution of Synthetic data 1 is denser than that of Synthetic data 2, so more local sub-frequent patterns are generated. Therefore, the runtime for Synthetic data 1 is much higher than for Synthetic data 2. Synthetic data 3 has the largest amount of data, so the impact of the threshold *min_sprev* is also the largest.

Effect of *min_rei*. Figure 6 shows that with the change of *min_rei*, the runtime of the algorithm is basically unchanged. The runtime of the algorithm does not fluctuate greatly with the change of *min_rei* because the relevancy index does not satisfy anti-monotonicity, that is, no matter how *min_rei* is set, the relevancy index needs to be calculated for the sub-frequent patterns mined to determine whether it belongs to local tight sub-frequent patterns.

Comparison of Algorithm Efficiency. Figure 7 shows that, for the three synthetic datasets, the runtimes of the two algorithms are not much different, because the time complexities of the two algorithms are of the same order of magnitude. Therefore, while ensuring relatively high efficiency, the LTSCP algorithm further discovers local tight patterns that cannot be found by the JBSCP algorithm.

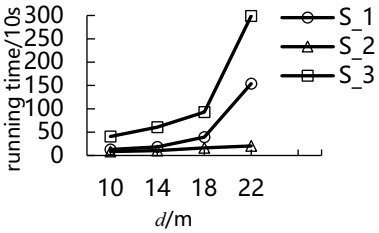

Figure 4. Runtime for different *d*.

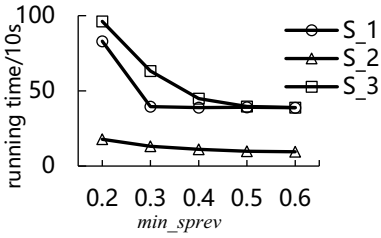

Figure 5. Runtime for different *min_sprev*.

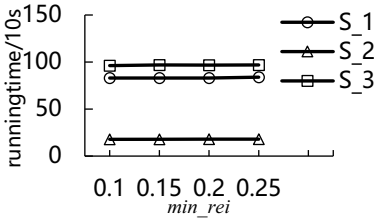

Figure 6. Runtime for different *min_rei*.

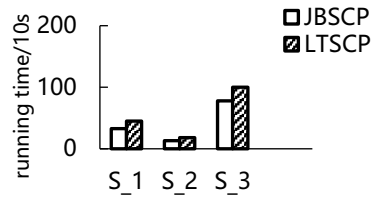

Figure 7. Runtime for three synthetic datasets: LTSCP, JBSCP.

5.3. Analyzing patterns mined by LTSCP

Pattern comparison for the Plant-data dataset. Figure 8(a) shows that the number of LTSCPs is much lower than the number of sub-frequent patterns, because LTSCPs are a subset of sub-frequent patterns, which are sub-frequent patterns of local compactness among row instances. Figure 8(b) shows that as *min_sprev* increases, the number of sub-frequent patterns satisfying the threshold decreases, and the number of LTSCPs also decreases. Likewise, the number of LTSCPs is much lower than the number of sub-frequent patterns.

Pattern comparison for the Beijing-POI dataset. Figure 9(a) shows that when *d* is between 30 and 40, the number of LTSCPs does not change. This is because although the sub-frequent patterns increase, they do not reach the local compactness required by the relevancy index, so the LTSCPs remain invariant. Figure 9(b) shows that with the increase of *min_sprev*, the number of sub-frequent patterns decreases, and the number of LTSCPs also decreases, but the proportion of LTSCPs in sub-frequent patterns increase. This is because row instances with high participation sub-frequent patterns are more likely to be locally dense.

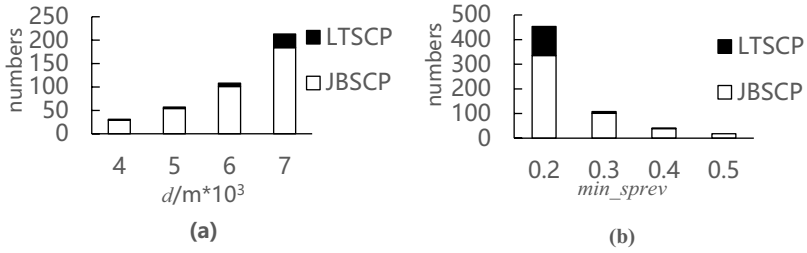

Figure 8. Number of patterns with varying *d* or *min_sprev* of LTSCPs for the Plant-data dataset.

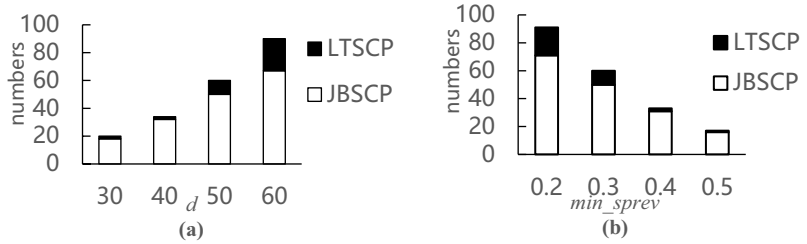

Figure 9. Number of patterns with different *d* or *min_sprev* of LTSCP for the Beijing-POI dataset.

5.4. Case studies

A case study on the Plant-data dataset. The top five LTSCPs are shown in Table 4. In the pattern {Glycyrrhiza yunnanensis, Anisodus acutangulus, Berneuxia thibetica}, all three characteristics are medicinal plant characteristics. It shows that the local area where the pattern is located is suitable for the growth of medicinal plants, and this local area can provide a research environment for pharmacists.

A case study on the Beijing-POI dataset. The top three LTSCPs are shown in Table 5. The pattern in the result {Chinese food, coffee house, hotel, guest house, parking lot, clothing store} indicates that features of the pattern appear in a local area. The area where the local tight pattern is discovered is a leisure and entertainment center. LTSCPs may have high practicability for applications such as the planning and relocation of a city center and commercial location selection.

Table 4. Mining results of LTSCPs for the Plant-data dataset.

LTSCP	Location of LTSCP	*RelI*
Picea brachytyla，Glycyrrhiza yunnanensis	Glycyrrhiza yunnanensis 1	1.0
Glycyrrhiza yunnanensis，Anisodus acutangulus，Berneuxia thibetica	Berneuxia thibetica 5	0.78
Abies georgei Orr，Hemsleya lijiangensis，Trillium tschonoskii Maxim	Hemsleya lijiangensis 2	0.77
Megacarpaea delavayi Franchet，Cephalotaxus lanceolata	Megacarpaea delavayi Franchet 3	0.75
Picea brachytyla，Hemsleya lijiangensis	Hemsleya lijiangensis 4	0.75

Table 5. Mining results of LTSCPs for the Beijing-POI dataset.

LTSCP	Location of LTSCP	*RelI*
Chinese food, coffee house, hotel, guest house, parking lot, clothing store	hotel 1869	0.56
Cafes, hotels, guest houses, parking lots, clothing stores	hotel 1869	0.45
Chinese food, coffee house, hotel, guest house, clothing store	hotel 1869	0.42

6. Conclusion and future work

In this paper, on the basis of sub-frequent pattern mining, the distribution characteristics of the row instances are analyzed. LTSCPs and an associated mining algorithm are proposed to reveal the special functions of local areas. The efficiency of the LTSCP mining algorithm is evaluated through experiments, and the practicability of LTSCPs is verified.

In future research work, we can consider the time-varying location of spatial instances and integrate the temporal dimension into LTSCP mining, which will help us to further understand the spatiotemporal relationship between spatial instances.

References

[1] WANG L, CHEN H. Spatial Pattern Mining Theory and Methods [M]. Beijing: Science Press, 2014: 2-4.

[2] WANG L, BAO X, ZHOU L, et al. Maximal sub-prevalent co-location patterns and efficient mining algorithms [C] // Proceedings of the 2017 International Conference on Web Information Systems Engineering, LNCS 10569. Cham: Springer, 2017: 199-214.

[3] WANG L, BAO X, ZHOU L, et al. Mining maximal sub-prevalent co-location patterns [J]. World Wide Web, 2019, 22(5): 1971-1997.

[4] 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.

[5] YOO J S. SHEKHAR S. A partial join approach for mining colocation patterns [C] // The ACM international Symposium on Advances in Geographic Information System (ACM GIS). Washington USA: ACM, 2004: 241-249.

[6] YOO J S. SHEKHAR S. CELIK M. A join-less approach for colocation pattern mining. A summary of results [C] // The IEEE International Conference on Data Mining (ICDM). Houston, USA: IEEE, 2005: 813-816.

[7] Yoo J. S, Bow M. A framework for generating condensed co-location sets from spatial databases [J]. Intelligent Data Analysis, 2019, 23(2): 333-355.

[8] Wang G, Wang L, Yang P, Chen H. Minimal negative co-location patterns and effective mining algorithm [J/OL]. Journal of Frontiers of Computer Science and Technology (2020-06-10) [2020-11-20]. https://kns.cnki.net/kcms/detai l/11.5602.TP.20200610.1016.002.html.

[9] Hu Z, Wang L, Vanha Tran, Zhou L. Mining Spatial Prevalent Co-location Patterns Based on Graph Databases [J/OL]. Journal of Frontiers of Computer Science and Technology:1-22[2022-01-22]. http://kns.cnki.net/kcms/detail/11.5602.TP.20201207.1544.021.html.

[10] Ma D, Chen H, Wang L, et al. Dominant feature mining of spatial sub frequent co location patterns [J]. Journal of Computer Applications, 2020, 40 (2): 465-472.

[11] Li X, Chen H, Xiao Q, et al. Spatiotemporal sub frequent co location pattern mining [J]. Journal of Southwest University (Natural Science Edition), 2020, 42 (11): 68-76.

[12] Celik M, Kang J M, Shekhar S. Zonal co-location pattern discovery with dynamic parameters [C]. Proc. 7th IEEE International Conference on Data Mining (ICDM 2007), Omaha, Institute of Electrical and Electronics Engineers Inc. 2007: 433-438.

[13] Sengstock C, Gertz M, Van Canh T. Spatial interestingness measures for co-location pattern mining [C]. Proc. 12th IEEE International Conference on Data Mining Workshop (ICDM Workshop 2012), Brussels, Belgium, IEEE Computer Society, 2012: 821-826.

[14] Zhao J. Research on spatial co location pattern mining based on Regional Division [D]. Yunnan University, 2018.