Effectively Updating Co-location Patterns in Evolving Spatial Databases

Jin Soung Yoo and Hima Vasudevan

Computer Science Department
Indiana University-Purdue University Fort Wayne
Indiana, USA 46805
Email: yooj,vasuh02@ipfw.edu

***Abstract*—Spatial co-location mining has been used for discovering spatial event sets which show frequent association relationships based on the spatial neighborhood. This paper presents a problem of finding co-location patterns on evolving spatial databases which are constantly updated with fresh data. Maintaining discovered spatial patterns is a complicated process when a large spatial database is changed because new data points make spatial relationships with existing data points on the continuous space as well as among themselves. The change of neighbor relations can affect co-location mining results with invalidating existing patterns and introducing new patterns. This paper presents an algorithm for effectively updating co-location analysis results and its experimental evaluation.**

Keywords–Spatial association mining; Co-location pattern; Incremental update

I. INTRODUCTION

As one of the spatial data mining tasks, spatial association mining is often used for discovering spatial dependencies among objects [1]–[4]. A spatial co-location represents a set of spatial features which are frequently observed together in a nearby area [3]. Examples of frequently co-located features/events include symbiotic species such as West Nile incidents and stagnant water sources in epidemiology, and interdependent events such as a car accident, traffic jam, policemen and ambulances in transportation. In business, co-location patterns can be used for finding relationships among services requested by mobile users in geographic proximity.

Most of the spatial association mining works [3]–[10] assume that all data is available at the start of data analysis. However, many application domains including location-based services, public safety, transportation and environmental monitoring collect their data periodically or continuously. For example, a police department accumulates, on average, 10,000 crime incidents per month [11]. For Earth observation, daily climate measurement values are collected at every 0.5 degree grid of the globe [12]. For keeping the analysis result coherent with respect to the most recent database status, discovered patterns should be updated.

The problem of updating spatial co-location patterns presents more challenges than updating frequent itemsets in a traditional transaction database. In the classical association analysis, the database update means the simple addition of new transaction records, or the deletion of existing records. Newly added transaction records are separately handled from existing records because the database is a collection of disjoint transaction records. In contrast, when a spatial database is updated, a new data point can make neighbor relationships with existing data points as well as other new data points on the continuous space. Thus, all neighbor relationships in the updated database should be examined for the maintenance of co-location patterns. The spatial pattern mining process is a computational and data intensive task, therefore simply re-executing a state-of-the-art co-location mining algorithm, whenever the database is updated, can result in an explosion of required computational and I/O resources. This paper proposes an algorithm for effectively updating discovered co-location patterns with the addition of spatial data points.

The remainder of this paper is organized as follows. Section II presents the basic concept of co-location pattern mining and the related work. Section III describes our algorithmic design concept for incremental co-location mining and the proposed algorithm. Its experimental evaluation is presented in Section IV. This paper will conclude in Section V.

II. BASIC CONCEPT AND RELATED WORK

The preliminary knowledge of spatial co-location pattern mining and the related work are presented in this section.

A. Basic concept of spatial co-location mining

Let $E = \{e_1, \ldots, e_m\}$ be a set of event types, and $S = \{o_1, \ldots, o_n\}$ be a set of their objects with geographic location. When the Euclidean metric is used for the neighbor relationship R, two objects o_i and o_j are neighbors of each other if the ordinary distance between them is not greater than a neighbor distance threshold d. A **co-location** X is a set of event types, $\{e_1, \ldots, e_k\} \subseteq E$, whose objects are frequently neighbors to each other on space. The **co-location instance** I of X is defined as a set of event objects, $I \subseteq S$, which includes all types in X and makes a clique under R.

The prevalence strength of a co-location is often measured by participation index value [3]. The **participation index** $PI(X)$ of $X = \{e_1, \ldots, e_k\}$ is defined as

$$PI(X) = \min_{e_i \in X}\{PR(X, e_i)\}, \quad (1)$$

where $1 \leq i \leq k$, and $PR(X, e_i)$ is the **participation ratio** of event type e_i in X, which is the fraction of objects of event e_i in the neighborhood of instances of $X - \{e_i\}$, i.e. $PR(X, e_i) = \frac{|distinct\ objects\ of\ e_i\ in\ instances\ of\ X|}{|objects\ of\ e_i|}$. If $PI(X)$ is greater than a given minimum prevalence threshold, we say X is a *prevalent co-located event set* or a *co-location*.

 ISBN: 978-1-61208-343-8

B. Related work

The problem of mining association rules based on spatial relationships (e.g., proximity and adjacency) was first discussed by Koperski et al. [1]. Shekhar, et al. [3] defines the co-location pattern and proposes a join-based co-location mining algorithm. Morimoto [2] studies the same problem to discover frequent neighboring service class sets. A space partitioning and non-overlap grouping scheme is used for finding neighboring objects. Yoo et al. [4], [10] propose join less algorithms to reduce the number of expensive spatial join operations in finding co-location instances. Celik et al. [8] extends the notion of co-location to a local zone-scale pattern. Eick et al. [13] proposes a framework for mining regional co-location patterns and Mohan et al. [14] presents a graph based approach for regional co-location discovery. Recognizing the dynamic nature of database, much effort has been devoted to the problem of incrementally mining frequent itemsets in classical association rule mining literature [15]–[19]. However, to find the problem to update co-location patterns in spatial data mining literature is rare. The most similar work with ours is He et al. [20] which is compared in our experimental evaluation.

III. Incremental Co-location Mining

Let $S_{old} = \{o_1, \ldots, o_n\}$ be a set of old data points in a spatial database and $S_{in} = \{o_{n+1}, \ldots, o_{n+h}\}$ be a set of new data points added in the database. Let S be all data points in the updated database, i.e., S=$S_{old} \cup S_{in}$. There are two types of co-location in the update. The *retained co-location* is an event set prevalent in both S_{old} and S. The *emerged co-location* is an event set not prevalent in S_{old} but prevalent in S. We propose an algorithm of Effective Update of COLOCation patterns (EUCOLOC). The proposed algorithm has two update stages. The first update stage examines only neighbor relationships of new data points, and finds all retained co-locations and some emerged co-locations. If an emerged set is found from the first update, the second update stage is triggered for finding other emerged co-location patterns in the updated database. Figure 3 shows the pseudo code of EUCOLOC algorithm.

A. Neighborhood Process

Directly finding all co-location instances forming clique neighbor relationships from spatial data is computationally expensive. Instead, we process the neighbor relationships related to the new data points S_{in}.

Definition 1: The neighborhood of a new object $o \in S_{in}$, **new neighborhood** $n_{new}(o)$, is defined to $\{o, o_2, \ldots, o_p | o_i \in S \wedge R(o, o_i)$=true $\wedge$ o's event type $<$ o_i's event type$\}$, where $2 \leq i \leq p$.

We assume there is a total ordering among the event types (i.e., a lexicographic order $\preceq_e$). R is a neighbor relationship function. Next, if an existing data point has a neighbor relationship with at least one new data point, its neighborhood is changed.

Definition 2: The **changed neighborhood** of an old object $o \in S_{old}$, $n_{chg}(o)$, is defined to $\{o, o_2, \ldots, o_p | o_i \in S \wedge \exists o_i \in S_{in} \wedge R(o, o_j)$=true $\wedge$ o's event type $<$ o_i's event type $\}$, where $2 \leq i \leq p$.

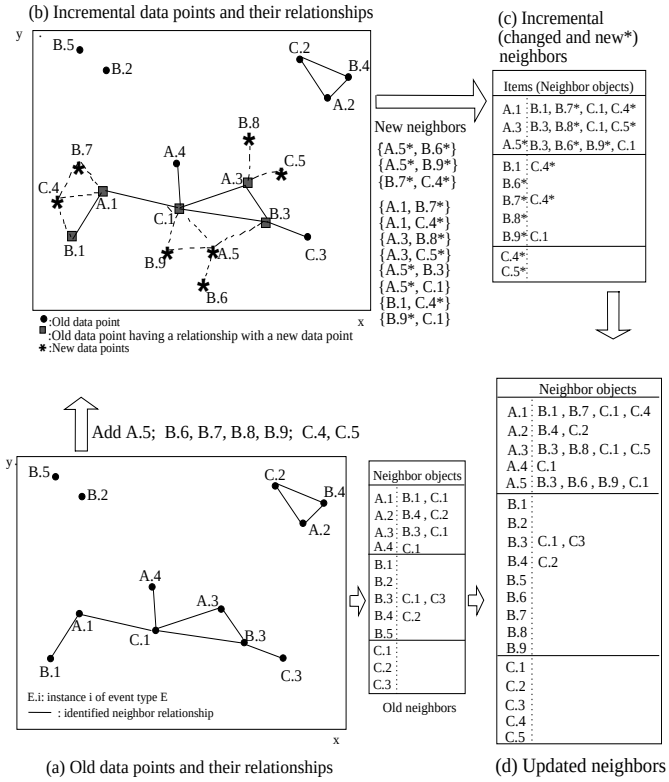


Figure 1. New, Changed and Updated Neighborhoods

Let $N_{new} = \{n_{new}(o_1), \ldots, n_{new}(o_h)\}$ be a set of all new neighborhoods for S_n and $N_{chg} = \{n_{chg}(o_1), \ldots, n_{chg}(o_q)\}$ be a set of all changed neighborhoods where $\{o_1, \ldots, o_q\} \subseteq S_{old}$. We call the union of N_{new} and N_{chg} to **incremental neighborhood set** (N_{inc}).

When an increment S_{in} is added as shown in Figure 1 (b), the EUCOLOC algorithm first finds all neighbors (NP) of new data points in S using a geometric method or a spatial query method (*Algorithm Line 2*). The incremental neighborhood set (N_{inc}) is prepared by finding new neighborhoods from NP and detecting changed neighborhoods from the old neighborhoods N_{old} (*Line 3* & Figure 1 (c)). Figure 1 (d) shows the entire neighborhood information (N) of the updated database (*Line 4*).

B. First update and detection

Let an event set be a *border event set* if the event set's all proper subsets are prevalent, but not prevalent itself. The border sets are used for detecting an emerging co-location without the generation and testing of many unnecessary candidates. The candidate event sets for the first update are previous co-located event sets (P_{old}) and border event sets (B_{old}) (*Line 7*). The incremental co-location instances of the candidate event sets are searched from the incremental neighborhoods (N_{inc}) without examining the entire neighbor relationships. (*Line 8* & Figure 1 (c)). A filter-and-refine search strategy is used for finding co-location instances. Let $SI = \{o_1, o_2, \ldots, o_k\}$ be a set of objects of a candidate set $c = \{e_1, e_2, \ldots, e_k\}$ where $e_1 < e_2 \ldots < e_k$. If the first object o_1 has neighbor relationships with all other objects in the set, SI is called a *star instance* of c. The start instances of $\{e_1, e_2, \ldots, e_k\}$ are

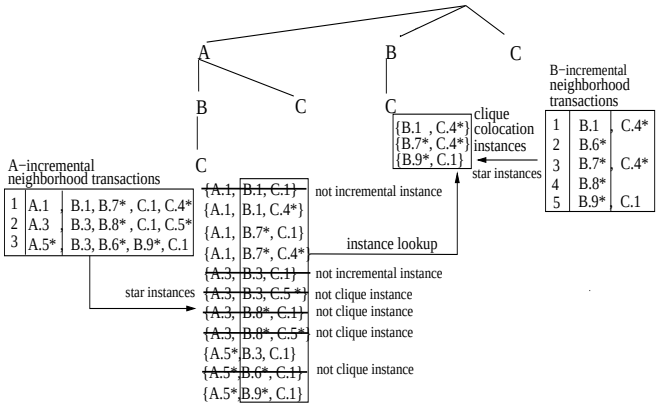


Figure 2. Event subsets and their instance search space

collected from the neighborhoods of e_1 according to Definition 1 and 2. The candidate instance $SI = \{o_1, o_2, \ldots, o_k\}$ of $c = \{e_1, \ldots, e_k\}$ becomes a true co-location instance of c if its subinstance $\{o_2, \ldots, o_k\}$ forms a clique. The cliqueness of the subinstance can be checked by simply querying the co-location instances of c's sub event set $\{e_2, \ldots, e_k\}$ if the subinstance has at least one new point, as shown in Figure 2.

The participation index of a candidate is computed with its incremental co-location instances (CI_c) and previous *instance metadata* (old_PB_info) which has the object information of its old co-location instances (*Line 13*). The prevalence of a candidate $c = \{e_1, \ldots, e_k\}$ is updated with

$$incPI(c) = \min_{e_i \in c}\{incPR(c, e_i)\}, \quad (2)$$

where $1 \leq i \leq k$, and $incPR(c, e_i)$ is the updated participation ratio of event type e_i with the incremental co-location instances of c, $incPR(c, e_i) = \frac{|O_i \bigcup I_i|}{|S_{old_i}|+|S_{in_i}|}$, where $|S_{old_i}|$ is the total number of old objects of e_i, $|S_{in_i}|$ is the total number of new objects of e_i, O_i is a set of distinct objects of e_i in the old co-location instances of c, and I_i is a set of distinct objects of e_i in the incremental co-location instances of c. If the participation index is greater than min_prev, the event set is a co-location ($\in P$) (*Line 14-15*). If this co-location is from the border set B_{old}, it also becomes an emerged co-location ($\in ES$) (*Line 16-17*).

C. Second update stage

If any emerged set is found from the first update stage, there is a possibility of finding other emerged event sets according to the following lemma.

Lemma 1: Let X be a co-located event set that is prevalent in the updated set $S = S_{old} \cup S_{in}$ but not prevalent in the old set S_{old}. Then there exists a subset $Y \subseteq X$ such that Y is an emerged event set.

Proof: Let Y be a minimal cardinality subset of X that is prevalent in S, not in S_{old}. Since Y is a prevalent event set in S, so are all of is proper subsets. However, by the minimality of Y, none of these subsets are new prevalent sets in S. Thus, Y is a border set in S_{old}, and $Y \subseteq X$ as claimed. ■

In the second update, a candidate is an event set which has at least one emerged event set as its subset (*Line 29*).

```
1:  procedure PREPROCESS
2:      NP ← search_neigh_pairs(S_in, S_old, R)
3:      N_inc ← gen_incr_neigh_trans(NP, N_old)
4:      N ← gen_upd_neigh_trans(N_old, N_inc)
5:  end procedure

6:  procedure FIRSTUPDATEDETECTION
7:      C ← P_old ∪ B_old
8:      SI ← scan_incr_star_inst(C, N_inc)
9:      k ← 2
10:     while C_k ≠ ∅ do
11:         for all c ∈ C_k do
12:             CI_c ← find_incr_clique_inst(SI_c, NP)
13:             PI ← compute_incPI(old_PB_info, CI_c)
14:             if PI ≥ min_prev then
15:                 P ← P ∪ c
16:                 if c ∈ B_old then
17:                     ES ← ES ∪ c
18:                 end if
19:             else
20:                 B ← B ∪ c
21:             end if
22:         end for
23:         k ← k+1
24:     end while
25: end procedure

26: procedure SECONDUPDATE
27:     if ES ≠ ∅ then
28:         k ← 3
29:         C_k ← gen_sizeK_candidates(P_{k−1}, ES_{k−1})
30:         while C_k ≠ ∅ do
31:             SI ← scan_star_instances(C_k, N)
32:             for all c ∈ C_k do
33:                 CI_c ← find_clique_instances(SI_c)
34:                 if compute_PI(CI_c) ≥ min_prev then
35:                     P ← P ∪ c; ES = ES ∪ c
36:                 elseB ← B ∪ c
37:                 end if
38:             end for
39:             k ← k+1
40:             C_k ← gen_sizeK_candidates(P_{k−1}, ES_{k−1})
41:         end while
42:     end if
43:     P_old ← P; B_old ← B; N_old ← N; S_old ← S_old ∪ S_in
44:     return P;
45: end procedure
```

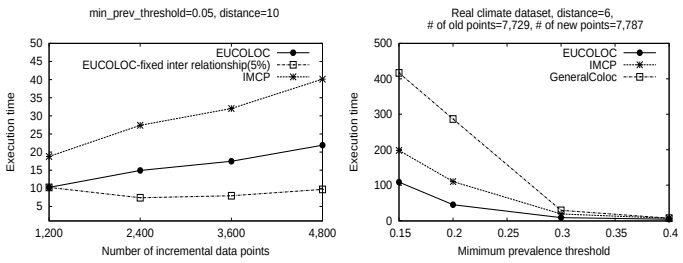
Figure 3. EUCOLOC algorithm

The star instances of candidates are collected from the entirely updated neighborhood transactions (N) (*Line 31*). The true co-location instances are filtered from the candidate instances. The prevalence value of a candidate is calculated using the original participation index (Equation (1)) because this set is a new candidate with no previous instance metadata. If the candidate is prevalent, it becomes an emerged co-location. Otherwise, the set is included in the border set for future update. The second update is repeated with the increase of the pattern size until no more candidate (*Line 30-41*).

IV. EXPERIMENTAL EVALUATION

We compared the performance of EUCOLOC with two other co-location mining algorithms. One (denoted as IMCP in this paper) has an update function [20]. The implementation of this algorithm is based on our understanding of the work. The other (denoted as GeneralColoc) does not have an update function [4]. All the experiments were performed on a Linux system with 8.0 GB memory, and 2.67 GHz CPU.

 ISBN: 978-1-61208-343-8



(a) (b)

Figure 4. Experiment Result (a) By incremental data size (b) By prevalence threshold

In the first experiment, we compared the performance of EUCOLOC and IMCP by varying the incremental size of synthetic data. The number of distinct event types was 50. The number of old data points was 10,020. The first incremental set has 1,200 data points. The second incremental set is two times bigger than the first set. The third one was three times bigger than the first set, and so on. The ratio of old data points which have relationships with new points was increased with the increase of new data (i.e., 5%, 10%, 15% and 20%). As shown in Figure 4 (a), the execution times of both EUCOLOC and IMCP increased with the incremental data size. The EUCOLOC showed better performance than IMCP. When the ratio of relationships with old data points was fixed to 5%, the execution times of EUCOLOC were stable, or very slowly increased. The performance of EUCOLOC depends on the inter-neighbor relationship ratio.

We also conducted the evaluation of EUCOLOC with real climate measurement data [12]. The total number of processed event types was 18. 7,728 event records were used for the old data. 7,787 new event records were added for the incremental data. We used 6 as a neighborhood distance, which means 6 cells on latitude-longitude spherical grids, where each grid cell is 1 degree $\times$ 1 degree. About half of the old event objects had neighbor relationships with the new ones. Figure 4 (b) shows the result. EUCOLOC showed slowly increasing execution time than other algorithms when the prevalence threshold was decreased.

V. Conclusion

In this paper, we presented an algorithm for efficiently mining co-location patterns in evolving spatial databases. The proposed algorithm has two update stages. The first update stage is 1) to avoid the generation and testing of many unnecessary candidates using the border concept, 2) to search only incremental neighborhoods for the update, and 3) to update the prevalence value of current co-locations with their incremental instances and minimal previous co-located object information. The second update stage is used for only finding new co-located event sets (emerged ones). The initial experimental evaluation shows our algorithmic design decision is effective in updating discovered co-location patterns. The proposed algorithm can be easily extended to handle the case of deleted data points. Our approach can be adopted for the cases of change of important parameters, such as neighbor distance and prevalence threshold. In the future, we plan to explore these problems.

References

[1] K. Koperski and J. Han, “Discovery of Spatial Association Rules in Geographic Information Databases,” in Proceedings of the International Symposium on Large Spatial Data bases, 1995, pp. 47–66.

[2] Y. Morimoto, “Mining Frequent Neighboring Class Sets in Spatial Databases,” in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 353–358.

[3] Y. Huang, S. Shekhar, and H. Xiong, “Discovering Co-location Patterns from Spatial Datasets: A General Approach,” IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 12, 2004, pp. 1472–1485.

[4] J. S. Yoo and S. Shekhar., “A Join-less Approach for Mining Spatial Co-location Patterns,” IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 10, 2006, pp. 1323–1337.

[5] J. S. Yoo and S. Shekhar, “A Join-less Approach for Spatial Co-location Mining: A Summary of Results,” in Proceedings of the IEEE International Conference on Data Mining, 2005, pp. 813–816.

[6] H. Xiong, S. Shekhar, Y. Huang, V. Kumar, X. Ma, and J. S. Yoo, “A Framework for Discovering Co-location Patterns in Data Sets with Extended Spatial Objects,” in Proceedings of the SIAM International Conference on Data Mining, 2004, pp. 78–89.

[7] J. Yoo and M. Bow, “Finding N-Most Prevalent Colocated Event Sets,” in Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, 2009, pp. 415–427.

[8] M. Celik, J. M. Kang, and S. Shekhar, “Zonal Co-location Pattern Discovery with Dynamic Parameters,” in Proceedings of the IEEE International Conference on Data Mining, 2007, pp. 433 – 438.

[9] J. S. Yoo and M. Bow, “Mining Spatial Colocation Patterns: A Different Framework,” Data Mining and Knowledge Discovery, vol. 24, no. 1, 2012, pp. 159–194.

[10] J. S. Yoo and S. Shekhar, “A Partial Join Approach for Mining Co-location Patterns,” in Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, 2004, pp. 241–249.

[11] “San Francisco Crime Incidents,” https://data.sfgov.org/.

[12] “Earth Observation Data,” http://data.giss.nasa.gov/.

[13] C. F. Eick, R. Parmar, W. Ding, T. F. Stepinski, and J. Nicot, “Finding Regional Co-location Patterns for Sets of Continuous Variables in Spatial Datasets,” in Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2008, pp. 1–10.

[14] P. M. et al., “A Neighborhood Graph based Approach to Regional Co-location Pattern Discovery: A Summary of Results ,” in Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2011, pp. 122–132.

[15] N. Ayan, A. Tansel, and E. Arkyn, “An Efficient Algorithm to Update Large Itemsets with Early Pruning,” in Proceedings of the International Conference on Knowledge Discovery and Data Mining, 1999, pp. 287–291.

[16] D. Cheung, J. Han, V. Ng, and C. Y. Wong, “Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique,” in Proceedings of the IEEE International Conference on Data Engineering, 1996, pp. 106 – 114.

[17] S. Thomas and S. Chakravarthy, “Incremental Mining of Constrained Associations,” High Performance Computing (HiPC), vol. 1970, 2000, pp. 547–558.

[18] D. Cheung, S. D. Lee, and D. Kao, “A General Incremental Technique for Maintaining Discovered Association Rules ,” in Proceedings of the International Conference on Databases Systems for Advanced Applications, 1997, pp. 185 – 194.

[19] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka, “An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases,” in Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining, 1997, pp. 263–266.

[20] J. He, Q. He, F. Qian, and Q. Chen, “Incremental Maintenance of Discovered Spatial Colocation Patterns,” in Proceedings of Data Mining Workshop, 2008, pp. 399 – 407.

 ISBN: 978-1-61208-343-8