Mining Mixed-drove Co-occurrence Patterns For Large Spatio-temporal Data Sets: A Summary of Results

Xiangxiang Cong, Zhanquan Wang (Corresponding Author), Man Kong, Chunhua Gu
Computer Science&Engineer Department, East China University of Science and Technology, Shanghai, China

Abstract - Discovering mixed-drove spatiotemporal co-occurrence patterns (MDCOPs) is an important field with many applications such as identifying tactics in battlefields, crime detection, etc. In practical applications, it is difficult to mine MDCOPs from large spatio-temporal data sets. Firstly, mining MDCOPs is computationally very expensive because the number of candidate co-occurrence instances is exponential in the number of object-types. Secondly, the spatio-temporal data sets are large and can’t be managed in memory. In order to reduce the number of candidate co-occurrence instances, we present a novel and computationally efficient MDCOP Graph Miner algorithm by using Time Aggregated Graph. The LDMDCOP Graph Miner algorithm is presented, which can deal with large data sets by means of file index. The correctness, completeness and efficiency of the proposed methods are analyzed. Experimental results show that the proposed MDCOP Graph Miner is computationally more efficient than the fast MDCOP-Miner and the LDMDCOP Graph Miner can effectively deal with the large spatiotemporal data sets.

Keywords: Mixed-drove Spatiotemporal Co-occurrence pattern; Large Spatiotemporal Data Set; Time Aggregated Graph (TAG); File Index

1 Introduction

As the volume of spatiotemporal data continues to increase significantly due to both the growth of database archives and the increasing number of spatiotemporal sensors, automatic and semi-automatic pattern analysis becomes more essential. It is meaningful and challenging for us to extract interesting patterns from these large spatiotemporal data sets.

Given a large spatiotemporal database, a neighbor relationship and mixed-drove interest measure thresholds, our aim is to discover mixed-drove spatiotemporal co-occurrence patterns (MDCOPs). To mine co-occurrence patterns, Celik et al. proposed MDCOP-Miner and fast MDCOP-Miner [4]. The two methods are based on the join-based collocation algorithm proposed by Huang et al. [9]. The basic co-occurrence pattern mining procedure involves four steps. First, candidate co-occurrence instances are gathered from the spatiotemporal data set. Prevalent co-occurrence pattern sets satisfying the given prevalence thresholds are filtered. Finally, co-occurrence patterns satisfy the given prevalence thresholds are generated. Most of the computational time of co-occurrence pattern mining is devoted to finding co-occurrence instances. The approach is Apriori like, which is costly as it enumerates all possible co-occurrence instances over all time instances. Thus, we propose adding a step for materializing the neighbor relationships to increase the efficiency of co-occurrence mining.

While the volume of the spatiotemporal data set is large, we can’t manage to discover the MDCOPs by existed methods. Since the existed methods are based on memory, we can’t get enough memory space. For example, we have 300 MB vehicle data. As movements of vehicle change over time, their co-occurrences also move in the same way. We can know the military strategy from the co-occurrences. So mining those patterns is really meaningful. As another example, China’s Zhejiang province public security bureau has more than 2GB of crime data. We can discover a lot of useful information from the large data set, such as the co-occurrence of crime type. For example, gamble and larceny usually occur together and they are co-occurrence. Mining these patterns is very useful for the police to analyze the movement of criminals. But the volume of crime data is too large for us to discover the patterns by using existed methods. As a result, the issue related to mining correct and complete patterns from large spatiotemporal data sets is a difficult problem. In order to solve the problem, new efficient storage method for MDCOPs must be proposed.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 presents basic concepts to provide a formal model of MDCOP Graph and the problem statement of mining MDCOPs. In Section 4, we present our proposed MDCOP mining algorithms. Analysis of the algorithms is given in Section 5. Section 6 presents the experimental evaluation, and Section 7 presents conclusions and future work.

2 Related Work

The MDCOPs problem differs from the co-location pattern. Previous approaches of MDCOP mining can use a spatial co-location mining algorithm for each time slot to find spatial prevalent co-locations, and then apply a post-processing step to discover MDCOPs by check their time prevalence. To mine co-locations, Huang et al. proposed a join-based approach [1] [2]. Celik et al. [4] formalizes the problem, propose a new monotonic mixed-drove interest
measure to discover and mine MDCOPs, and also propose an efficient algorithm (MDCOP-Miner).

MDCOPs represent object types co-located over space and time forming a spatial network (edges between objects in the network indicate existence of a neighborhood relationship) that dynamically changes over time. A common naïve approach to model such a network is to use time expanded graph, as described by Köhler et al. [5] where in the network is replicated across discrete time instants. A more efficient method of modeling temporal spatial networks was proposed by George et al. [6] by incorporating the properties of nodes and edges in the graph as a time series. This paper also proposed efficient algorithms for computing the shortest path and connectivity in time dependent networks modeled using time aggregated graphs. The problem of mining MDCOPs with high spatial and time prevalence is described by Celik et al. [4], However the approach is Apriori like and involves candidate generation, which is costly as it enumerates all possible cliques over all time instances.

Although experts obtained many achievements in MDCOP mining, we still have no correct and efficient approach dealing with large spatiotemporal datasets. In this paper, we materialize the neighbor relationships for efficient co-occurrence pattern mining. We solve the problem of efficient storage of MDCOPs by using the Time Aggregated Graph model [6] and create our own storage model MDCOP Graph for mining MDCOPs. Finally, we provide a correct and efficient co-occurrence pattern mining algorithm to deal with large spatiotemporal datasets.

3 Co-occurrence Pattern Mining

In this section, we present basic concepts to provide a formal model of MDCOP Graph and the problem statement of mining MDCOPs.

3.1 Basic Concepts

3.1.1 Mixed-drove Prevalence Measure

The focus of this study is to mine MDCOPs with multiple prevalence measures from large spatiotemporal data sets. The basic MDCOP algorithm [4] defines two interest measures namely spatial prevalence $\theta_p$ and a time prevalence measure $\theta_{time}$. Hence, a pattern is defined as an MDCOP if it has the property [4]

$$\text{Prob}_{i,\text{prev}}[s_{\text{prev}}\{P_i\}, \text{time_slot} \rightarrow t_n] \geq \theta_p \land \text{Prob}_{i,\text{time}}[t_n] \geq \theta_{\text{time}}$$

(1)

Where, Prob $(.)$ is the probability of overall prevalence time slots, $s_{\text{prev}}$ stands for spatial prevalence. There are more details in [6]

3.1.2 Time Aggregated Graph (TAG)

We propose a graph based data structure to capture the information required to mine MDCOPs from the data set. This data structure is motivated by Time Aggregated Graphs (TAG) which models time varying road conditions as time series on the edges of a road network. Defines the time aggregated graph as follows.

$$\text{TAG} = \{ N, E, TF, f_1, \ldots, f_k, g_1, \ldots, g_n, w_1, \ldots, w_p \} \mid f : N \rightarrow R^n, g : E \rightarrow R^n, w : E \rightarrow R^n$$

(2)

Where $N$ is the set of nodes, $E$ is the set of edges, $TF$ is the length of the entire time interval, $f_1, \ldots, f_k$ are the mappings from nodes to nodes, $g_1, \ldots, g_n$ are mapping from edges to edges, and $w_1, \ldots, w_p$ indicate the dependent weights (eg.travel times) on the edges.

Each edge has an attribute, called an edge time series that represents the time instants for which the edge is present. This enables TAG to model the topological changes of the network with time. . There are more details in [6].

3.2 Modeling MDCOP Graph

Given a set of spatiotemporal mixed object-types $E$, a neighborhood relation $R$, a set of time slots $TF$, a threshold pair $(\theta_p, \theta_{time})$, MDCOP Graph can be represented as a neighbor graph in which a node is an object type and edge between two nodes represents the neighbor relationship over all time slots. We use MDCOP Graph to materialize neighbor relationships. As we know that most of the computational time of co-occurrence pattern mining is devoted to finding co-occurrence instances. By means of MDCOP Graph, we don’t need to generate all possible candidate co-occurrence instances. We just generate real co-occurrence instances through visiting the MDCOP Graph. Thus, we can increase the efficiency of co-occurrence mining.

Definition 3.1 Given a set of co-occurrence instances $CI$, instance type level graph (IG) is used to captures the existence of co-location instances between two instance types over time. We define instance type level graph as follows.

$$IG = \{I, CI, TF, f_i, g_i, e_i \mid I \rightarrow R^n, e_i : CI \rightarrow R^n$$

(3)

Where $I$ is the set of instances of all object-types, $CI$ is the set of co-location instances, $TF$ is the length of the entire time interval, such that $TF = \{T_0, \ldots, T_n\}$, $f_i, g_i, e_i$ are the mappings from object-types to object-types , $e_i$ indicate the existence of co-occurrence instances between two instance types over time on the edges.

For example, we generate instance type level graph (Figure.1) using the data set given [6]. In Figure.1, we use time-series [1 1 1 0] to show that A1 and C1 are co-located at time slot 0, time slot 1, time slot 2 and disappear at time slot 3. Therefore we can easily capture the existence of co-occurrence instances over time by traversing instance type level graph.
Definition 3.2 Given a set of candidate co-occurrence patterns (CP), object type level graph (OG) is used to indicate the participation count of particular object-types contributing to particular co-occurrence patterns. We define object type level graph as follows.

\[
OG = \{E, CP, TF, f_0, \ldots, f_n, p_0, \ldots, p_n | f_i : E \in CP^P; p_i : E \in CP^P \}
\] (4)

Where E is the set of spatiotemporal mixed object-types, CP is the set of co-occurrence patterns, TF is the length of the entire time interval, \(f_0, \ldots, f_n\) are the mappings from particular object-types to the particular co-occurrence patterns, and \(f_0, \ldots, f_n\) indicate the participation count of particular object-types contributing to particular co-occurrence patterns over time on the edges.

For example, we generate object type level graph (Figure 2) using the data set given in[7]. The co-occurrence pattern AC has co-occurrence instances sets \{(A1, C1), (A3, C2)\} at time slot 0, time slot 1 and time slot 2. At time slot 3, AC has no co-location instances. In Figure 2, since A1 and A3 are different instances of A, we use time-series [2 2 0 0] to show the participation count of object-type A contributing to co-location pattern AC. By using object type level graph, we can get the spatial prevalence index values and time prevalence index values.

Definition 3.3 Given instance type level graph and object type level graph, MDCOP Graph is composed of two parts: instance type level graph and object type graph. The instance type level and object type graphs are connected through links. We define MDCOP Graph as follows.

\[
MDCOP\ Graph = \{IG, OG, TF, E, l_0, \ldots, l_n, l_i | I \rightarrow E \}
\] (5)

Where IG is instance type level graph, OG is object type level graph, TF is the length of the entire time interval, E is the set of spatiotemporal mixed object-types, \(l_0, \ldots, l_n, l_i\) are the mappings from object-types to their own instances.

For example, we generate MDCOP graph (Figure 3) using the data set given in[4]. In Figure 3, both the instance type level and object type graphs are connected through links for easy traversal.

4 Mining MDCOPS

In this section, we discuss FastMDCOP-Miner and then propose two novel MDCOP mining algorithms: MDCOP Graph Miner and LDMDCOP Graph Miner to mine MDCOPs. We also give the execution trace of these algorithms.

4.1 FastMDCOP-Miner

FastMDCOP-Miner [8] uses a spatial co-location mining algorithm for each time slots to find spatial prevalent co-locations and prune time non-prevalent patterns as early as possible between the time slots to discover MDCOPs. To mine co-locations, Huang et al. proposed a join-based approach, Yoo et al. proposed a partial join-based approach and a join-less approach [3], [9], [10], this approach is based on the join-based collocation algorithm proposed by Huang et al., but it is also possible to use other approaches. FastMDCOP-Miner [8] will first discover all size k spatial prevalent MDCOPs and prune time non-prevalent patterns as early as possible between the time slots to discover MDCOPs. Then the algorithm will generate size \(k + 1\) candidate MDCOPs using size \(k\) MDCOPs until there are no more candidates. However, this approach is Apriori like and involves candidate generation which is costly as it enumerates all possible cliques over all time instants.

4.2 MDCOP Graph Miner

To eliminate the drawbacks of FastMDCOP-Miner, we propose a MDCOP mining algorithm (MDCOP Graph Miner) to discover MDCOPs by storing all the MDCOPs in the MDCOP Graph. This data structure is motivated by Time Aggregated Graphs (TAG) [7], which models time varying road conditions as time series on the edges of a road network.
In our case, we use two different types of series over the edges. One of the series capture the existence of co-occurrence patterns between two instances over time. Based on the existence time series of co-occurrence patterns between pairs of instances, we aggregate the information to object types. At the object type graph, each time series contains the participation count of a particular object contributing to a particular co-location pattern. Both the instance type level and object type graphs are connected through links for easy traversal.

We give the pseudo code of the algorithm and provide an execution trace of it using the data set in [7]. Algorithm 1 give the pseudo code of the MDCOP Graph Miner algorithm. This pseudo code is used to explain two algorithms: MDCOP Graph Miner and LDMDCCP Graph Miner which will be discussed in the next section. The choice of the algorithm is provided by the user. In the algorithm, steps 1-14 create the MDCOP Graph. Steps 15-21 give an iterative process to mine MDCOPs, steps 15-21 continue until there is no candidate MDCOP to be generated. Step 22 gives a union of the results. The execution trace of the MDCOP Graph Miner are explained below.

Algorithm 1 pseudo code for the MDCOP Graph Miner

Inputs:

| E: a set of spatial object types |
| ST: a spatiotemporal data set < object_type, object_id, x, y, timeslot > |
| R: spatial neighborhood relationship |
| TF: a time slot frame {t0,...,tn-1} |
| θ_p: spatial prevalence threshold |
| θ: time prevalence threshold |

Output: MDCOPs whose spatial prevalence indices, i.e., participation indices, are no less than θ_p, for time prevalence indices are no less than θ time.

Variables:

| t: time slots(0,...,m-1) |
| k: co-occurrences size |
| T_k: set of instances of size k co-occurrences |
| S_P_k: set of spatial prevalent size k co-occurrences |
| S_P_k: set of time prevalent size k co-occurrences |
| C_k: set of candidate size k co-occurrences |
| MDP_k: set of mixed-drove size k co-occurrences |
| MDG: graph store all the MDCOPs |
| Address: address for storing time series to the file |

Method:

1. k = 2
2. C_k(θ) = gen_candidate_co_occ(E)
3. for each time slot t in TF
   4. T_k(t) = gen_co_occInst(C_k(θ), ST, R)
   5. set time_series [t] =1 for T_k(t)
   6. S_P_k(t) = find_time_prev_co_occ(T_k(t), θ_p)
   7. T_P_k(t) = find_time_index(S_P_k(t))
   8. MDP_k(t) = find_time_prev_co_occ(T_P_k(t), θ)
   9. C_k(t) = MDP_k(t)
10. if alg_choice == "LDMDCCP Graph Miner"
11.   Address = gen_co_occ_address(’T_k(t)’)
12.   access the MDG file by address
13. set timeseries [t] = 1 if T_k(t) if required
14. if alg_choice == "MDCOP Graph Miner"
15.   MDG = gen_MDCOP_Graph(MDP_k)
16. while (not empty MDP_k)
17.   C_k(t) = gen_candidate_co_occ(MDP_k)
18.   T_k(t) = gen_instancesTree(C_k(t), MDG)
19.   S_P_k(t) = find_time_index(S_P_k(t), θ_p)
20.   T_P_k(t) = find_time_index(S_P_k(t))
21.   MDP_k(t) = find_time_prev_co_occ(T_P_k(t), θ)
22.   k = k + 1
23. return union( MDP2,..., MDPk)

The execution trace of the MDCOP Graph Miner is given in Figure 4. This data set in [7], contains four object-types A, B, C, and D and their instances in four time slots (i.e., A has four instances). The instances of each object-type have a unique identifier, such as A1. To discover MDCOPs, we use a monotonic composite interest measure which is a composition of the spatial prevalence and time prevalence measure. The spatial prevalence measure shows the strength of the spatial co-location when the index is greater than or equal to a given threshold [8], [9]. The time prevalence measure shows the frequency of the pattern over time.

In step 1, by dividing each entry in Figure 4a with the corresponding number of instances for an object, we get the participation ratio of an object type in co-location. For example, the participation index of collocation AB is [3/5 3/5 3/5 3/5], which is the minimum participation ratio of type A and B in all time slots. We prune time non-prevalent patterns whose participation indices are less than a given threshold as early as possible. For example, there are four time slots and the time prevalence threshold is 0.5. In this case, a size k pattern should be present for at least two time slots to satisfy the threshold. If the time prevalence index of a pattern is 0 for the first (or any) three time slots, there is no need to generate it and check its prevalence for the rest of the time slots even if it is time persistent for the remaining time slots. Spatial prevalent patterns AB, AC, and BC are selected as MDCOPs since they are time prevalent (their time prevalence indices satisfy the given time prevalence threshold 0.5). In contrast, spatial prevalent patterns AD, BD, and CD are pruned since they are time non-prevalent.

In step 2, three sub-graphs on the bottom of Figure 4b are created. It also creates links from the instance types to the object type, for example, Link between A3 and A if A3 is part of at least one co-occurrence. The links between the object and the instance type help in traversing the data set efficiently to calculate the spatial prevalence index values. Connections between the object type graph and the instance type graph are missing in order to reduce clutter and the series in the object graph has not been represented for the same reason. After the algorithm has been executed, the series on edge A and AB would be [3/4 3/4 3/4 3/4] because of co-locations A1B1, A2B1, A3B2 and A3B3. Note that A3 is counted only once at each time interval though it appears in two co-locations at every time instant.

In step 3, the candidate MDCOP ABC is generated through AB, AC, and BC. Generally, the number of candidate patterns is large. Then if we generate temporal time prevalence index for every candidate pattern, candidate pattern whose temporal time prevalence index is less than a given threshold can be pruned. For example, ABC is a candidate pattern, the temporal time series of ABC is [0 1 1 0] equals to time series
of AB [1 1 1 1] & AC [1 1 1 0] & BC [0 1 1 1], the time prevalence index is 0.5 which is no less than the given threshold. So we generate instances of candidate pattern ABC. By building instances-trees for candidate patterns, instances of candidate pattern ABC could be generated.

In step4, the participation indices of pattern ABC are 2/5 in time slots 1 and 2 and its time prevalence index 0.5 equals to the threshold. Since there are not enough subsets to generate the next superset patterns, the algorithm stops at this stage and outputs the union of all size MDCOPs, i.e., A B, AC, BC, and ABC.

4.3 LDMDCOP Graph Miner

In this section, we propose a new algorithm, called LDMDCOP Graph Miner, which can deal with large data sets by using file index. MDCOP Graph is an efficient storage method to capture the information required to mine MDCOPs from the MDCOP Graph in the file. In order to solve these problems, we use adjacency matrix to store the MDCOP Graph. The largest convenience of adjacency matrix is the ability to determine the existence of a particular edge in constant time, and access the storage media only once. According to this method, we can calculate the address for a particular edge to store its time series in the file and access the storage media only once. As a result, we use file to store big MDCOP Graph. This approach brings us two problems. One is how to store the big MDCOP Graph in the file, the other is how to capture the information required to mine MDCOPs from the MDCOP Graph in the file. In order to solve these problems, we use adjacency matrix to store the MDCOP Graph. The largest convenience of adjacency matrix is the ability to determine the existence of a particular edge in constant time, and access the storage media only once. According to this method, we can calculate the address for a particular edge to store its time series in the file and access the time series by the same address, there is no need to store the address of the time-series, we calculate the address
according to the same expression which used to calculate the address for storing. The expression is as follows.

\[
\text{address}(R,C) = \left( R \times N + C - E(R,C) \right) \times S
\]

Where \( R \) is the row number, \( C \) is the column number, \( N \) is the total number of instance, \( E(R,C) \) is the number of patterns whose instances are of the same type. \( S \) is the size of time series.

The pseudo code of the LDMDCOP Graph Miner is given in Algorithm 1. When the LDMDCOP Graph Miner is chosen, the algorithm will activate steps 10, 11, 12 and deactivate steps 13 and 14. We use adjacency matrix to store the MDCOP Graph. At the same time, the addresses for storing time series of co-occurrence instances could be calculated and the addresses also are used to access the file for getting the information required to mine MDCOPs.

5 Experiment Results

We use Real Data Sets and Synthetic to evaluate the proposed algorithm. The real data includes 15 time snapshots and 21 distinct vehicle types and their instances. The minimum instance number is 2, the maximum instance number is 78, and the average number of instances is 19. To evaluate the performance of the algorithms, spatiotemporal data sets were generated based on the spatial data generator proposed by Huang et al. [8]. Synthetic data sets were generated for spatial frame size \( D \times D \). For simplicity, the data sets were divided into regular grids whose side lengths had neighborhood relationship \( R \).

5.1 Experiment Results for Real Data Sets

5.1.1 Effect of Number of Time Slots

We evaluated the effect of the number of time slots on the execution time of the MDCOP algorithms using the real data set. The participation index, time prevalence index, and distance were set at 0.2, 0.8, and 100 m, respectively. Experiments were run for a minimum of 1 time slot and a maximum of 14 time slots. Results show that the MDCOP Graph Miner requires less execution time than the fastMDCOP-Miner (Figure 5a). As the number of time slots increases, the ratio of the increase in execution time is bigger than that of the MDCOP Graph Miner as the number of object-types increases for the real data set.

5.2 Experiment Results for Synthetic Datasets

We evaluated the effect of the spatial prevalence threshold on the execution times of MDCOP mining algorithms. The fixed parameters were participation index, distance, and number of time slots, and their values were 0.4, 20 m, and 100, respectively. Experimental results show that the MDCOP Graph Miner is more computationally efficient than the fastMDCOP-Miner (Figure 6a). The execution time of the MDCOP Graph Miner decreases as the time prevalence threshold increases.

We also evaluated the effect of the time prevalence threshold on the execution time of the LDMDCOP Graph Miner using synthetic data sets. The participation index, distance and number of time slots, were set at 0.5, 20 m, 200, respectively. The results showed that the execution time of the LDMDCOP Graph Miner decreases as the time prevalence threshold increases (Figure 6b).
6 Conclusions and Future Work

We presented a novel and computationally efficient algorithm (the MDCOP Graph Miner) for mining MDCOPs. We also presented an improved MDCOP Graph Miner algorithm (the LDMDCOP Graph Miner) which can deal with large spatiotemporal data sets. We compared the MDCOP Graph Miner with fastMDCOP-Miner, which is Apriori-like and involves candidate generation, which is costly as it enumerates all possible co-occurrence instances over all time instants. We proved that the proposed algorithms are correct, complete and effective in finding mixed-drove prevalent (i.e., spatial prevalent and time prevalent) MDCOPs. Our experimental results using real and synthetic data sets provide further evidence of the viability of our approaches.

Further, we would like to extend the MDCOP graph and the subsequent mining algorithm for insertions of object types at arbitrary time interval. Also we would like to extend the current methodology to address zonal co-location problems where the spatial prevalence changes according to the local patterns observed. Finally, we hope to investigate the idea of multi-scale relationship for different pattern families.

7 References

[1] Shashi Shekhar, Yan Huang, Discovering Spatial Co-location Patterns. A Summary of Results. SSTD 2001: 236-256


