

Mining At Most Top-K% Mixed-drove Spatio-temporal Co-occurrence Patterns: A Summary of Results

Mete Celik¹ Shashi Shekhar¹ James P. Rogers² James A. Shine² James M. Kang¹

¹Department of Computer Science, University of Minnesota, MN, USA
{mcelik, shekhar, jkang}@cs.umn.edu

²U.S. Army ERDC, Topographic Engineering Center, VA, USA
{james.p.rogers.II, james.a.shine}@erdc.usace.army.mil

Abstract

Mixed-drove spatio-temporal co-occurrence patterns (MDCOPs) represent subsets of object-types that are located together in space and time. Discovering MDCOPs is an important problem with many applications such as planning battlefield tactics, and tracking predator-prey interactions. However, determining suitable interest measure thresholds is a difficult task. In this paper, we define the problem of mining at most top-K% MDCOPs without using user-defined thresholds and propose a novel at most top-K% MDCOP mining algorithm. Analytical and experimental results show that the proposed algorithm is correct and complete. Results show the proposed method is computationally more efficient than naïve alternatives.

1. Introduction

At most top-K% mixed-drove spatio-temporal co-occurrence patterns (TopMDCOPs) represent subsets of object-types that are located together in space and time. Formally, given a collection of Boolean spatio-temporal (ST) features (object-types), their instances (objects) over a common ST framework, and a neighborhood relation over neighbors, a TopMDCOP mining algorithm aims to discover correct and complete sets of interesting and non-trivial TopMDCOPs. A TopMDCOP represents a pattern set whose interest measures are in the top-K% of the complete set of MDCOPs and have higher values than patterns which are not found in TopMDCOPs.

This work was partially supported by the US Army Corps of Engineers under contract number W9132V-06-C-0011, the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under contract number DAAD19-01-2-0014, and the NSF grant ISS-0431141.

Discovering TopMDCOPs is important for many spatio-temporal application domains, including military (battlefield planning and strategy), ecology (tracking species and predator-prey interactions), and homeland defense (looking for significant “events”) [7, 13].

However, an MDCOP mining algorithm proposed in previous work, requires user-defined thresholds: a spatial prevalence threshold and a time prevalence threshold [3]. The spatial prevalence measure is used to determine if the pattern is spatially prevalent in a specific timeslot. The time prevalence measure is used to determine if the pattern is frequent. These threshold values are mostly domain-specific and without domain knowledge, it is difficult to set up suitable interest measure thresholds to mine the MDCOPs. If the user-defined thresholds are too small, it is highly likely that too many patterns will be generated. If the threshold values are too large, it is also possible to discover too few patterns and miss possible significant ones. This study aims to discover TopMDCOPs with no need for user-defined spatial or time prevalence thresholds.

An Application Domain Example: TopMDCOPs are of great concern in ecology and animal behavioral science, where there is a need to identify keystone species or the co-existence/co-occurrence of species with keystone species that most affect the food chain/web [16, 19]. A food web shows the feeding relationships between different species. Keystone species are species that have effects on other species through their strong interaction with them and whose absence results in major changes in the food web structure [16, 19].

Figure 1 shows two food webs, one with sea otters and one without sea otters. The species in the food web are connected by arrows. The web shows prey-predator relationships. The thickness of the lines represents the importance of the food link, and dashed

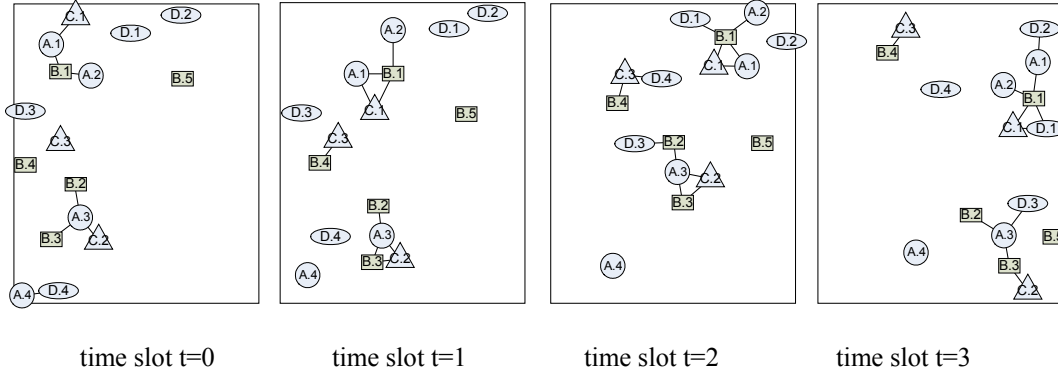


Figure 2. A spatio-temporal input dataset

Table 1. Comparison of outputs of MDCOP mining [3] and at most Top-K% MDCOP mining (\checkmark - pattern is in the output set, X - pattern is pruned, thld = threshold, * pruned patterns by MDCOP-Miner)

Co-occurrence Patterns	Spatial prevalence index values				Time prevalence index values (sorted)	MDCOPs [3] (MDCOP-Miner) for spatial-thld=0.4		At most top-K% MDCOPs (TopMDCOP-Miner)	
	time slot 0	time slot 1	time slot 2	time slot 3		time-thld=0.1	time-thld=0.5	top-10%	top-30%
A B	3/5	3/5	3/5	3/5	4/4	\checkmark	\checkmark	\checkmark	\checkmark
A C	2/4	2/4	2/4	0	3/4	\checkmark	\checkmark	X	\checkmark
B C	0	3/5	3/5	3/5	3/4	\checkmark	\checkmark	X	\checkmark
A B C	0	2/5	2/5	0	2/4	\checkmark	\checkmark	X	X
A D	1/4*	0	0	2/4	2/4	\checkmark	X	X	X
B D	0	0	2/5	1/5*	2/4	\checkmark	X	X	X
C D	0	0	1/4*	1/4*	2/4	X	X	X	X
B C D	0	0	0	1/5*	1/4	X	X	X	X

To illustrate the difference between the MDCOP mining problem and at most top-K% MDCOP mining problem, we use the example spatio-temporal dataset given in Figure 2. It shows the positions of instances of 4 object-types - A, B, C, and D - and their instances in four time slots. Table 1 shows the outputs of both the MDCOP mining algorithm (MDCOP-Miner) and at most top-K% MDCOP mining algorithm (TopMDCOP-Miner). The aim of both approaches is to find MDCOPs based on the spatial and time prevalence index values. The spatial prevalence index and how it is calculated will be discussed in the next section. Spatial prevalence index values of patterns in each time slot are given in Table 1. The time prevalence index of an MDCOP is the number of time slots where the pattern occurs divided by the total number of time slots. For example, the time prevalence index of pattern {A, B} is 4/4 since it occurs in 4 out of 4 time slots. Similarly, the time prevalence index of pattern {A, C} is 3/4 since it occurs in 3 out of 4 time slots (time slots 0, 1, and 2). MDCOP-Miner discovers the patterns that satisfy the user-defined spatial and time prevalence threshold. A pattern is an MDCOP if it is spatial prevalent and the time prevalence index of it

is equal to or above the user-defined threshold. For example, if the spatial prevalence threshold is 0.4 and the time prevalence threshold is 0.1, then the output of the MDCOP-Miner will be patterns {A, B}, {A, C}, {B, C}, {A, B, C}, {A, D}, and {B, D}. If the time prevalence threshold is 0.5 then the MDCOPs will be patterns {A, B}, {A, C}, {B, C}, {A, B, C} since they are spatial prevalent and their time prevalence index is equal to or above the threshold. The limitation of this approach is its dependence of the spatial and time prevalence index thresholds. If thresholds are too small, the entire candidate set can be output. If they are too big, significant patterns may not be discovered.

In contrast, the proposed TopMDCOP-Miner will discover the patterns which are the first K% of the power-set (except singletons and empty set) of the set of all distinct feature types based on the highest value of the interest measure. For example, if we want to find the top-10% MDCOPs of the spatio-temporal dataset given in Figure 2, only pattern {A, B} will be in the output list. Similarly, if we want to find the top 30% of the MDCOPs, the output will contain patterns {A, B}, {A, C} and {B, C}.

Contributions: This paper makes the following contributions:

- It defines the at most top-K% mixed-drove spatio-temporal co-occurrence pattern (MDCOP) mining problem.
- It proposes a novel and computationally efficient top-K% MDCOP mining algorithm (TopMDCOP-Miner).
- It shows that the proposed algorithm is correct and complete with regard to top-K% MDCOPs.
- It experimentally evaluates the proposed top-K% MDCOP mining algorithms using real datasets.

Scope: This paper focuses on the top-K% MDCOPs applied to a typed collection of moving objects. The following issues are beyond the scope of this paper: (i) similarity measures for tracking moving objects due to the focus on object-types rather than objects; (ii) indexing and query processing issues related to mining objects; and (iii) discovering multisets (e.g., {A, A, B}).

Outline: The rest of the paper is organized as follows. Section 2 presents basic concepts to provide a formal model of at most top-K% MDCOPs and the problem statement of mining at most top-K% MDCOPs. Section 3 presents our proposed TopMDCOP mining algorithm. Analysis of the algorithm is given in Section 4. Section 5 presents the experimental evaluation and Section 6 discusses conclusions and future work.

2. Basic Concepts and Problem Statement

The focus of this study is to discover at most top-K% mixed-drove spatio-temporal co-occurrence patterns (MDCOPs) over a spatio-temporal framework and a neighborhood relation R . First we review the modeling of MDCOPs, and then explain the modeling of at most top-K% MDCOPs. Then we propose an algorithm to mine these patterns.

2.1 Basic Concepts of MDCOP Mining

Given a set of spatio-temporal object-types and a set of their instances with a neighborhood relationship R , an MDCOP is a subset of spatio-temporal object-types whose some of instances are neighbors in space and time. The MDCOP mining problem significantly extends the spatial co-location mining problem to include time information.

Spatial co-location mining algorithms are used to discover sets of mixed object-types that are frequently

located together in a spatial framework for a given set of spatial object-types, their instances, and a spatial neighbor relationship R [11]. For example, in Figure 2, in time slot $t=0$, $\{A.1, C.1\}$ is an instance of a co-location if the distance between the objects is no more than a given neighborhood distance threshold. The solid lines show the distance between the objects that satisfies the neighborhood distance threshold. The participation index is used as a spatial prevalence measure to determine the strength of the co-location pattern, in particular, whether the index is greater than or equal to a threshold [11]. Such a co-location is called **spatial prevalent**. The **participation index** is defined as the minimum of the participation ratios (the number of instances on object-types forming co-location instances divided by total number of instances). For example, in Figure 2, $\{A, B\}$ is a co-location in time slot $t=0$, and its instances are $\{A.1, B.1\}$, $\{A.2, B.1\}$, $\{A.3, B.2\}$, and $\{A.3, B.3\}$. In the dataset, object-type A has 4 instances and three of them ($A.1, A.2$, and $A.3$) are contributing to the co-location $\{A, B\}$, so the participation ratio of A is $3/4$. The participation ratio of B is $3/5$ since 3 out of 5 instances are contributing to the co-location $\{A, B\}$. The participation index of the co-location $\{A, B\}$ is $3/5$, which is the minimum of the participation ratios of object-types A and B .

It has been shown that the participation index is anti-monotone in size of co-locations [11]. In other words, $participation_index(P_i) \leq participation_index(P_j)$ if P_i is a subset of P_j . In addition, the participation index has a spatial statistical interpretation as an upper bound on the cross- K function [4].

The **time prevalence measure** of an MDCOP can be defined as the number of time slots where the pattern occurs divided by the total number of time slots. If the ratio is greater than a user-defined threshold, the pattern is time prevalent. For example, in Figure 2, the total number of time slots is 4 and pattern $\{A, B\}$ occurs in all 4 time slots, so it is time prevalent since its time prevalence index - 1 - is above the time prevalence index threshold 0.5.

The **mixed-drove prevalence measure** of an MDCOP is a composition of the spatial prevalence measure and time prevalence measure [3]. The pattern is called a mixed-drove prevalent pattern if its mixed-drove prevalence measure satisfies the following.

$$Prob_{t_m \in all_time_slot} [s_prev(pattern P_i, time_slot t_m) \geq \theta_p] \geq \theta_{time}$$

where $Prob$ stands for the probability of overall prevalence time slots, s_prev stands for spatial prevalence, θ_p is the spatial prevalence threshold, and θ_{time} is the time prevalence threshold.

For example, in Figure 2, {A, B} is an MDCOP because it is spatial prevalent in time slots $t=0$, $t=1$, $t=2$, and $t=3$ since its participation indices are not less than the given threshold 0.4 in these time slots, and it is time prevalent since its time prevalence index - 1 -, is above the time prevalence index threshold 0.5. In contrast, pattern {B, D} is not an MDCOP. Although it is spatial prevalent in time slot $t=2$, it is not time prevalent since its time prevalence index is not more than the given time prevalence index threshold 0.5.

2.2. Modeling At most Top-K% MDCOPs

Given a set of spatio-temporal object-types and a set of their instances with a neighborhood relationship R , an at most top-K% MDCOP is a subset of spatio-temporal object-types whose instances are neighbors in space and time.

Definition 2.1: Given n distinct object-types, all MDCOP subsets include power-sets of these object-types except singletons and the empty set. The number of all MDCOP subsets can be found using the following equation.

$$\text{Number of all MDCOP subsets} = 2^n - (n+1) \quad (1)$$

where 2^n represents the number of all number of subsets of n object-types and $(n+1)$ represents the number of subsets of singletons and the empty set. Singletons and the empty set are not included in the number of all MDCOP subsets since they do not represent co-occurrence.

Definition 2.2: Given a spatio-temporal dataset, and a set T of time slots, such that $T=[T_0, \dots, T_{n-1}]$, a pattern P is in the top-K% MDCOP list if it is in the first K% of the number of all MDCOP subsets of the set of all distinct object-types based on the highest value of the interest measure.

For example, in Figure 2, a spatio-temporal dataset includes four object-types and their power-set includes $2^4 - (4+1) = 11$ subsets (patterns). A top-10% MDCOP list will include only 1 pattern which is the top-10% of 11 subsets. A top-30% MDCOP will include 3 patterns which are {A,B}, {A,C}, and {B,C}. A top-100% MDCOP list will include all 11 MDCOP subsets of the set of 4 object-types regardless of whether their interest measures “0” (zero) or not. Table 1 lists the patterns whose time prevalence indices are not zero.

Definition 2.3: Given a top-K% MDCOP list and its time prevalence indices, a percentile time prevalence

threshold is the minimum time prevalence index of the top-K% MDCOP list.

For example, in Table 1, for top-30% MDCOPs, the percentile time prevalence index threshold is the minimum time prevalence index of patterns {A,B}, {A,C}, and {B,C}, that is, $\min(4/4, 3/4, 3/4) = 3/4$.

Definition 2.4: Given a spatio-temporal dataset, and a set T of time slots, such that $T=[T_0, \dots, T_{n-1}]$, a patterns P is in the **at most top-K% MDCOP list** if it is in the top-K% MDCOP list and interest measure is not equal to zero.

For example, in Figure 2, the at most top-10% MDCOP list and the at most top-30% MDCOP list will include the same patterns as in the top-10% MDCOP list and top-30% MDCOP list. In contrast, the at most top-100% MDCOP list will include 8 of the 11 subsets since the time prevalence interest measure of 8 subsets are not equal to zero, as shown in Table 1.

2.3. Problem statement

Given:

- A set P of Boolean spatio-temporal object-types over a common spatio-temporal framework STF.
- A neighbor relation R over locations.
- A parameter value of K

Find: At most Top-K% mixed-drove spatio-temporal co-occurrence patterns which are the first K% of all MDCOP subsets of the set of all distinct object-types based on the highest value of the interest measure.

Objective: Minimize computation cost.

Constraints: To find a correct and complete set of TopMDCOPs.

Example: The spatio-temporal dataset given in Figure 2(a) contains 4 Boolean object-types, A, B, C, and D for 4 time slots. A distance between the objects may define the neighborhood relation R . For example, A.4 is a neighbor of D.4 in time slot 0, but not in time slots 1, 2, and 3. In this example dataset {A, B}, {A, C}, {B, C}, {A, B, C}, {A, D}, and {B, D} form candidate MDCOPs. Table 1 gives the spatial prevalence indices (participation indices), the time prevalence indices of the MDCOPs, and the possible outputs for two different scenarios. In the first scenario, the K value is 10 and the aim is to find at most top-10% MDCOPs. The output will be pattern {A, B}. In the second scenario, the K value is 30 and the aim is to find at most top-30% MDCOPs. The output will contain patterns {A, B}, {A, C}, {B, C}.

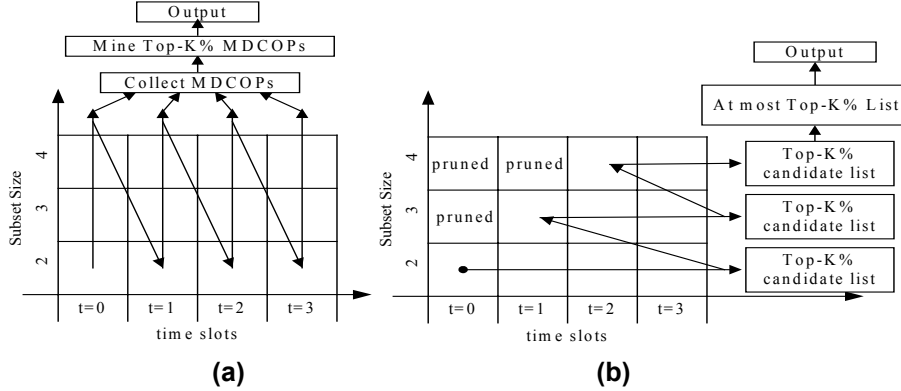


Figure 3: Comparison of the Naïve approach and TopMDCOP-Miner algorithms

3. Mining Top-K% MDCOPs

In this section, we first discuss a naïve approach to finding MDCOPs and then propose a novel at most top-K% MDCOP mining algorithm (TopMDCOP-Miner).

Naïve approach: A naïve approach uses a co-location mining algorithm for each time slot to find co-locations and then applies a post-processing step to discover at most top-K% MDCOPs by checking their time prevalence. This approach will generate all possible MDCOPs and then apply a pruning step to mine at most top-K% MDCOPs (Figure 3(a)). To mine co-locations, Huang, Shekhar and Xiong proposed a join-based approach [11]. Yoo and Shekhar proposed a partial join-based approach [22]. Yoo, Shekhar, and Celik proposed a join-less approach [23]. Zhang et al. proposed a multi-way spatial join-based approach [25]. The approach studied here will be based on the Huang’s join-based approach, but it is also possible to use other approaches [21, 23, 25]. The naïve approach generates size $m+1$ candidate co-locations for each time slot using size m subclasses until there are no more candidate co-locations. After finding all size co-locations in each time slot, a post-processing step is used to discover at most top-K% MDCOPs. This approach will not prune out non-top-K% patterns before the post-processing step thereby resulting in unnecessary computational cost.

TopMDCOP-Miner: In contrast, we propose a TopMDCOP mining algorithm (TopMDCOP-Miner) to discover at most top-K% MDCOPs by incorporating a filtering step in each iteration of the algorithm (Figure 3(b)). It maintains a top-K% candidate list until the correct top-K% MDCOPs are selected (Definition 2.2). The percentile time prevalence threshold is the minimum time prevalence index of

patterns of the top-K% candidate list. If a generated pattern has a higher time prevalence index value than the percentile time prevalence threshold, the pattern will be replaced with the new generated pattern and the percentile time prevalence threshold will be updated. If the time prevalence index of a generated pattern is equal to the percentile time prevalence threshold, it will be added to the top-K% candidate list. In the final step, the algorithm will eliminate the patterns whose time prevalence index value is equal to zero (Definition 2.4). The proposed approach is computationally more efficient than the naïve approach. First we give the pseudo code of the algorithm, and then we provide an execution trace of this algorithm using the spatio-temporal dataset from Figure 2.

Algorithm 1 gives the pseudo code of the TopMDCOP-Miner algorithm. The inputs are a set of spatial object-types E , a spatio-temporal dataset ST , a spatial neighborhood relationship R , and the percentage of patterns to be mined K . The output is a set of at most top-K% MDCOPs.

In the Algorithm 1, steps 1 and 2 include initialization of the parameters, steps 3 through 12 give an iterative process to mine top-K% MDCOPs, and step 13 finds at most top-K% MDCOPs by pruning patterns whose time prevalence value is equal to zero. The functions of the algorithm are explained below.

Generation of candidate co-occurrence patterns (step 5):

This function uses an apriori-based approach to generate size- $m+1$ candidate co-locations C_{m+1} for each time slot, using size- m top-K% mixed-drove co-occurrence patterns MDP_m [2].

Generating spatial co-occurrence instances (step 6):

The instances of candidate C_{m+1} are generated by joining neighbor instances of size- m patterns found in top-K% candidate list for each time slot. This is similar

to the instance generation step of the co-location miner algorithm [11].

Finding spatial prevalence indices of co-occurrence patterns (step 7): Participation indices of the patterns for each time slot are found in this step. Computation of the participation indices follows the same algorithmic ideas as those in the co-location mining algorithm [11].

Calculating the time prevalence index (step 9): The time prevalence indices of the patterns are calculated. It is the number of time slots where the pattern occurs divided by the total number of time slots.

<p style="text-align: center;">Pseudo code for TopMDCOP-Miner Algorithm</p> <p>Inputs: E: a set of spatial object-types ST: a spatio-temporal dataset <object_type, object_id, x, y, time> R : spatial neighborhood relationship TF : a time slot frame {t₀, ..., t_{n-1}} K : percentage of number of patterns to be mined Output: At most top-K% MDCOPs. Variables: m: co-occurrence size T₁ : set of instances of size m co-occurrences C_m: set of candidate size m co-occurrences SP_m, TP_m: set of size m co-occurrences Most_TopK, Cand_list, MDP : sets of MDCOPs</p> <p>Algorithm:</p> <ol style="list-style-type: none"> 1. initialization 2. co-occurrence size m=1, C_m(0)=E, MDP₁(0)=ST // init 3. while (not empty MDP_m) { 4. For each time slot t in (0, ..., n-1){ 5. C_{m+1}(t)=gen_candidate_co-occ(C_m(t), MDP_m(t)) 6. T_{m+1}(t)=gen_co-occurrence_instance(C_{m+1}(t), T_m(t), R) 7. SP_{m+1}(t)=find_spatial_prevalence_index(T_{m+1}(t), C_{m+1}(t), k) 8. } 9. TP_{m+1}=find_time_prevalence_index (SP_{m+1}) 10. [MDP_{m+1}, Cand_list]=Find_candidate_top-K%_list (TP_{m+1}) 11. m=m+1 12. } 13. Most_TopK=find_at_most_top-K% (Cand_list)

Algorithm 1. TopMDCOP-Miner algorithm

Find candidate top-K% list (step 10): The aim of this step is to maintain the top-K% candidate list. First, it will pick size 2 top-K% MDCOPs out of all MDCOP subsets (Definition 2.1) and store them in the top-K% candidate list. The percentile time prevalence threshold will be the minimum time prevalence index of the top-K% candidate list. Size 3 patterns will be generated using size 2 top-K% MDCOPs. If a generated pattern has higher time prevalence index value than the percentile time prevalence threshold, the pattern will be replaced with the new generated pattern and the percentile time prevalence threshold will be updated. If the time prevalence index of generated pattern is equal to the percentile time prevalence threshold, it will be added to the top-K% candidate list. Size 4 patterns will

be generated using size 3 top-K% patterns and if necessary the top-K% candidate list and percentile time prevalence index will be updated.

The algorithm will run iteratively until there is no more candidate MDCOPs to be generated. The algorithm outputs the union of all sizes of top-K% MDCOPs.

Finding the final top-K% mixed-drove co-occurrence patterns (step 13): This function will eliminate the patterns whose time prevalence index value is equal to zero (Definition 2.4) and will output at most top-K% MDCOPs.

An Execution Trace: The execution trace of the algorithm is given in Figure 4 using the spatio-temporal dataset given in Figure 2. This dataset contains four object-types A, B, C, and D and their instances in four time slots. The total number of MDCOP subsets of these object-types will be 11 by Definition 2.1. A has 4 instances, B has 5 instances, C has 3 instances, and D has 4 instances. Each instance of an object-type has a unique identifier, such as A.1.

If we want to discover at most top-30% MDCOPs in the spatio-temporal dataset (Figure 2), we need to mine 3 patterns which is 30% of the 11 MDCOP subsets. The candidate pairs of the object-types and their instances are generated for each time slot in step 1 (Figure 4(a)). The time prevalence indices of patterns are calculated (Step 2). Pattern {A, B} is persistent for all time slots and its time prevalence index is 4/4, and pattern {A, C} is persistent in time slots t=0, t=1, and t=2 and its time prevalence index is 3/4.

The next step is to determine the size 2 top-30% MDCOP candidate list and the percentile time prevalence threshold. The top-30% candidate list will include patterns {A, B}, {A,C}, and {B,C}. The percentile time prevalence threshold will be 3/5 which is the minimum time prevalence index of patterns {A, B}, {A,C}, and {B,C}. These patterns will be used to generate triples (size 3 patterns). In step 4, the triple pattern {A,B,C} is generated. In step 4 (Figure 4(b)), the instances of candidate MDCOP {A, B, C} and participation indices are found, which are 2/5 for time slots t=1 and t=2.

The next step is to determine if we need to update the top-30% candidate list using the triple pattern {A,B,C}. Pattern {A,B,C} will not be included in the top-30% candidate list since its the time prevalence index is less than the percentile time prevalence threshold. The algorithm will not generate any more patterns since there are not enough triple subsets to generate size 4 patterns. The output of the algorithm will include patterns {A,B}, {A,C}, and {B,C} which are on the top-30% MDCOPs list.

Step 1: Find patterns and their instances and their s_{prev} values, e.g., participation index (PI)

Co-occurrence patterns	timeslot t=0					timeslot t=1					timeslot t=2					timeslot t=3						
	AB	AC	AD	BC	BD	CD	AB	AC	AD	BC	BD	CD	AB	AC	AD	BC	BD	CD				
Co-occurrence pattern instances	A1 B.1 A2 B.1 A3 B.2 A3 B.3	A1 C.1 A3 C.2	A4 D.4				A1 B.1 A2 B.1 A3 B.2 A3 B.3	A1 C.1 A3 C.2	B1 C.1 B3 C.2 B4 C.3				A1 B.1 A2 B.1 A3 B.2 A3 B.3	A1 C.1 A3 C.2	B1 C.1 B3 C.2 B4 C.3	B1 D.1 B2 D.3	C3 D.4	A1 B.1 A2 B.1 A3 B.2 A3 B.3	A1 D.2 A3 D.3	B1 C.1 B3 C.2	B1 D.1	C1 D.1
PR	3/4 3/5	2/4 2/3	1/4 1/4				3/4 3/5	2/4 2/3	3/5 3/3				3/4 3/5	2/4 2/3	3/5 3/3	2/5 2/4	1/3 1/4	3/4 3/5	2/4 2/3	2/5 2/3	1/5 1/4	1/3 1/4
PI	3/5	2/4	1/4				3/5	2/4	3/5				3/5	2/4	3/5	2/5	1/4	3/5	2/4	2/5	1/5	1/4

(a)

Step 2: Calculate time prevalence indices

	timeslot t=0	timeslot t=1	timeslot t=2	timeslot t=3	time prevalence index
AB	1	1	1	1	4/4
AC	1	1	1	0	3/4
AD	1	0	0	1	2/4
BC	0	1	1	1	3/4
BD	0	0	1	1	2/4
CD	0	0	1	1	2/4

Step 3: Top-30% candidate list

candidate list	time prevalence index
AB	4/4
AC	3/4
BC	3/4
AD	2/4
BD	2/4
CD	2/4
Percentile time prevalence threshold = 3/4	

Step 4: Generate superset patterns (triplets)

	timeslot t=0	timeslot t=1	timeslot t=2	timeslot t=3
Patterns	ABC	ABC	ABC	ABC
Instances		A1 B.1 C.1 A3 B.3 C.2	A1 B.1 C.1 A3 B.3 C.2	
PR		2/4 2/5 2/3	2/4 2/5 2/3	
PI		2/5	2/5	

Step 5: Calculate time prevalence index

	timeslot t=0	timeslot t=1	timeslot t=2	timeslot t=3	time prevalence index
ABC	-	1	1	-	2/4

Step 6: At most top-30% MDCOPs

	time prevalence index
AB	4/4
AC	3/4
BC	3/4

(b)

Figure 4. Execution trace of the TopMDCOP-Miner algorithm

4. Correctness and Completeness Analysis of the TopMDCOP-Miner

Theorem 4.1: *The TopMDCOP-Miner is complete.*

Proof: The TopMDCOP-Miner is complete if it finds at most top-K% MDCOPs out of all MDCOP subsets defined in Definition 2.1. Algorithm first will find all top-K% MDCOPs (Definition 2.2) and will prune out the patterns whose time prevalence interest measures are equal to zero (Definition 2.4). We can show this by proving that none of the functions of the algorithm miss any patterns, i.e., filter out a TopMDCOP.

The *gen_candidate_co-occur* function does not miss any patterns given the anti-monotone nature of the spatial prevalence interest measure. The input to this function is candidate size-m top-K% MDCOPs and the output is candidate size-m+1 top-K% MDCOPs. If $c_1 = \{f_1, \dots, f_m\}$ and $c_2 = \{f_1, \dots, f_{m-1}, f_{m+1}\}$ are size-m candidate top-K% MDCOPs, candidate size-m+1 patterns $C_{m+1} = \{f_1, \dots, f_{m-1}, f_m, f_{m+1}\}$ will be produced by joining size-m top-K% MDCOPs.

The *gen_co-occur_instance* function does not miss any patterns. This function generates instances of candidate size-m+1 top-K% MDCOPs by joining instances of size m top-K% MDCOPs if they are in the neighborhood distance and forming a clique.

The *find_spatial_prevalence_index* function does not miss any patterns. It finds spatial prevalence indices of the patterns.

The *find_time_prevalence_index* function does not miss any MDCOPs. It finds the time prevalence indices of the patterns.

The *find_candidate_top-K%_list* function does not miss any patterns. This function is used to determine the top-K% candidate list. It will determine top-K% patterns and percentile time prevalence threshold. The patterns whose time prevalence index is not more than threshold will be pruned.

The *find_at_most_top-K%* function does not miss any top-K% MDCOPs. The function finds at most top-K% MDCOPs by pruning the patterns whose time prevalence indices are equal to zero, if there is any. □

Theorem 4.2: *The TopMDCOP-Miner is correct. In other words, if an MDCOP P is returned by the TopMDCOP-Miner algorithm then P is in the top-K% MDCOP list.*

Proof: The proof is easy to establish due to the pruning steps of “find_candidate_top-K%_list” and “find_at_most_top-K%” which weed out candidates not found in the top-K% MDCOP list and whose interest measures are equal to zero.

At most top-K% MDCOPs are the patterns which are in the top-K% of all size MDCOP subsets (Definition 2.1) and have time prevalence indices are not equal to zero. Function “find_at_most_top-K%” will prune out the patterns, if their time prevalence indices are equal to zero. The following properties are applied in the pruning step of “find_candidate_top-K%_list”:

1) No-tie condition: If there is no tie, TopMDCOP-Miner will not miss any at most top-K% pattern in function “find_candidate_top-K%_list”. For example, assume the number of all MDCOP subsets are 11 (for 4 object-types) and each pattern has a

unique time prevalence index value. The TopMDCOP-Miner will return top 3 patterns from the list if we want to find at most top-30% MDCOPs.

2) Tie condition: If there is a tie, TopMDCOP-Miner will not miss any at most top-K% patterns in function “find_candidate_top-K%_list”. In this case, function will include more than the top-K% patterns until there is no tie. For example, in Figure 4, we want to find top-20% of the patterns, there is a tie top-20% corresponds to the 2 patterns out of an 11-patterned list and patterns {A, C} and {B, C} have the same prevalence index values. In this case, the function will include both patterns in the top-20% MDCOPs. The output of function will be patterns {A, B}, {A, C}, and {B, C}. In the worst case, if there is a tie, function will output all possible subsets. □

5. Experimental Evaluation

In this section, we present our experimental evaluations of several design decisions and workload parameters of our TopMDCOP-Miner algorithm. We used a real-world vehicle movement dataset. We evaluated the performance of the TopMDCOP-Miner and a naïve approach by changing the number of time slots and number of object-types. Figure 5 shows the experimental setup to evaluate the impact of design. Experiments were conducted on an Intel Centrino PIV 1.6 GHz computer with 512 MB of RAM.

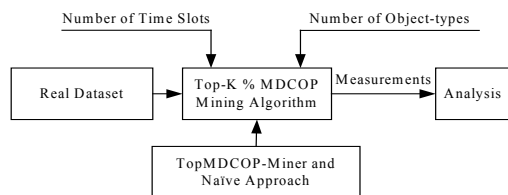


Figure 5. Experimental setup and design

The dataset contains the location and time information of moving objects. It includes 15 time snapshots and 22 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.

5.1 Effect of Number of Time slots

In the first experiment, we evaluated the effect of number of timeslots on the execution time of both algorithms by mining at most top-10% MDCOPs. The neighborhood distance and the parameter K parameter were set at 100m and 10 respectively. The execution time of both algorithms increase, as the number of

timeslots is increased (Figure 6). The TopMDCOP-Miner is computationally more efficient than the naïve approach because of its early pruning strategy (Figure 6). As the number of time slots increases, the ratio of the increase in execution time is smaller for TopMDCOP-Miner than with the naïve approach.

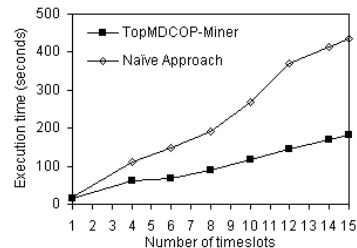


Figure 6. Effect of number of time slots

5.2 Effect of Number of Object-types

In the second experiment we evaluated the effect of number of object-types on the execution times of algorithms. The neighborhood distance and K parameter were set at 100m and 10 respectively. Figure 7 shows that the execution time of both algorithms increases and the TopMDCOP-Miner outperforms the naïve approach as the number of features increases. The increase in the feature number causes an increase in the number of join operations, which is computationally expensive.

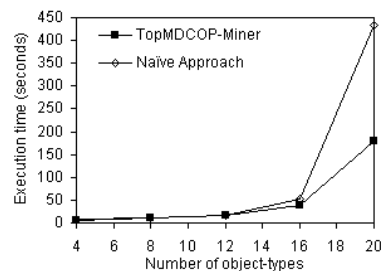


Figure 7. Effect of number of object-types

Experimental results for TopMDCOP-Miner can be optimized further and will be ready in the camera-ready version of the paper.

6. Conclusions and Future Work

We defined at most top-K% mixed-drove spatio-temporal co-occurrence patterns (MDCOPs) and at most top-K% MDCOP mining problem. We also presented a novel algorithm (the TopMDCOP-Miner) for mining these patterns. We proved that the proposed algorithm is correct and complete in discovering at most top-K% MDCOPs. Our experimental results

using a real dataset provide further evidence of the viability of our approach.

The scope of this paper is to determine at most top-K% MDCOPs based on a time prevalence index. In future work, we plan to investigate methods to mine at most top-K% MDCOPs based on both time and spatial prevalence indices. We would also like to develop new computationally efficient algorithms for mining at most top-K% MDCOPs.

The association rule mining literature has explored, problems of mining top-K patterns and N most interesting patterns. Han et. al. defined the problem of mining top-K frequent closed patterns and proposed methods to mine these patterns based on FP-tree [9, 10]. Fu et. al. defined the problem of mining N most interesting itemsets [5]. These proposed methods are based on Apriori-gen functions [2]. Ngan et. al. proposed an approach based on COFI-trees and FP-trees to mine N most interesting itemsets [18]. The proposed approaches to mine top-k patterns and N most interesting patterns in association rule mining literature are not applicable to mining at most top-K% MDCOPs since they do not deal with spatio-temporal datasets and require a transaction database.

Other studies have focused on defining spatio-temporal patterns and algorithms [6, 8, 12, 14, 17, 20]. Laube and Imfeld defined several spatio-temporal patterns, such as leadership, convergence [15] Query processing algorithms have been proposed to extract such patterns [15]. We plan to extend our algorithm to mine these patterns.

7. Acknowledgments

The authors would like to thank Kim Koffolt for her comments.

8. References

- [1] Sea Otter Food Web, <http://cbc.amnh.org/crisis/foodweb.html>.
- [2] R. Agarwal and R. Srikant, Fast algorithms for Mining Association Rules, *VLDB'94*, 1994.
- [3] M. Celik, S. Shekhar, J. P. Rogers, J. A. Shine, and J. S. Yoo, Mixed-Drove Spatio-Temporal Co-occurrence Pattern Mining: A Summary of Results, *In Proc. of ICDM*, Hong Kong, December 2006.
- [4] N. A. C. Cressie, *Statistics for Spatial Data*, Wiley and Sons, ISBN 0471843369, 1991.
- [5] A. W.-c. Fu, R. Wang-wai, and J. Tang, Mining N-most Interesting Itemsets, *In Proc. Int. Symp. on Methodologies for Intelligent Systems*, 2000.
- [6] J. Gudmundsson, M. v. Kreveld, and B. Speckmann, Efficient Detection of Motion Patterns in Spatio-Temporal Data Sets, *ACM-GIS*, 250-257, 2004.
- [7] R. Guting and M. Schneider, *Moving Object Databases*, Morgan Kaufmanns, 2005.
- [8] M. Hadjieleftheriou, G. Kollios, P. Bakalov, and V. J. Tsotras, Complex Spatio-Temporal Pattern Queries, *VLDB'05*, 877-888, 2005.
- [9] J. Han, J. Pei, and Y. Yin, Mining Frequent Patterns without Candidate Generation, *In Proc. ACM-SIGMOD'00*, Dallas, TX, 2000.
- [10] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, Mining Top-K Frequent Closed Patterns without Minimum Support, *In Proc. Int. Conf. on Data Mining (ICDM)*, Japan, 2002.
- [11] Y. Huang, S. Shekhar, and H. Xiong, Discovering Co-location Patterns from Spatial Datasets: A General Approach, *IEEE Trans. on Knowledge and Data Eng. (TKDE)*, vol. 16(12), 1472-1485, 2004.
- [12] P. Kalnis, N. Mamoulis, and S. Bakiras, On Discovering Moving Clusters in Spatio-temporal Data, *9th Int'l Symp. on Spatial and Temporal Databases (SSTD)*, Angra dos Reis, Brazil, 2005.
- [13] M. Koubarakis, T. Sellis, A. Frank, S. Grumbach, R. Guting, C. Jensen, N. Lorentzos, H. J. Schek, and M. Scholl, *Spatio-Temporal Databases: The Chorochronos Approach*, LNCS 2520, vol. 9, Springer Verlag, 2003.
- [14] P. Laube and S. Imfeld, Analyzing relative motion within groups of trackable moving point objects, in *In GIScience, number 2478 in Lecture notes in Computer Science*. Berlin: Springer, 132-144, 2002.
- [15] P. Laube, M. v. Kreveld, and S. Imfeld, Finding REMO - detecting relative motion patterns in geospatial lifelines, *11th Int'l Symp. on Spatial Data Handling*, 201-214, 2004.
- [16] M. A. Leibold, A Graphical Model of Keystone Predators in Food Webs: Tropic Regulation of Abundance, Incidence, and Diversity Patterns in Communities, *The American Naturalist*, vol. 147(5), 784-812, 1996.
- [17] C. Mouza and P. Rigaux, Mobility Patterns, *GeoInformatica*, vol. 9(4), 297-319, 2005.
- [18] S.-C. Ngan, T. Lam, R. C.-W. Wong, and A. W.-c. Fu, Mining N-most Interesting Itemsets Without Support Threshold by the COFI-tree, *Int. J. Business Intelligence and Data Mining*, vol. 1(1), 2005.
- [19] R. T. Paine, Food Web Complexity and Species Diversity, *American Naturalist*, vol. 100(910), 65-75, 1966.
- [20] H. Yang, S. Parthasarathy, and S. Mehta, A Generalized Framework For Mining Spatio-temporal Patterns in Scientific Data, *ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, 716-721, 2005.
- [21] J. S. Yoo and S. Shekhar, A Joinless Approach for Mining Spatial Colocation Patterns, *IEEE Trans. on Knowledge and Data Eng. (TKDE)*, vol. 18(10), 2006.
- [22] J. S. Yoo and S. Shekhar, A Partial Join Approach for Mining Co-location Patterns, *ACM-GIS'05*, Washington D.C., USA, 2005.
- [23] J. S. Yoo, S. Shekhar, and M. Celik, A Join-less Approach for Co-location Pattern Mining: A Summary of Results, *IEEE Int'l Conf. on Data Mining*, Houston, USA, 2005.
- [24] J. S. Yoo, S. Shekhar, S. Kim, and M. Celik, Discovery of Co-evolving Spatial Event Sets, *SIAM Int'l Conf. on Data Mining (SDM)*, Maryland, 2006.
- [25] X. Zhang, N. Mamoulis, D. W. L. Cheung, and Y. Shou, Fast Mining of Spatial Colocations, *10th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 384-393, Seattle, WA, 2004.