EvolvingClusters: Online Discovery of Group Patterns in Enriched Maritime Data

George S. Theodoropoulos, Andreas Tritsarolis, and Yannis Theodoridis

Data Science Lab., University of Piraeus
{gstheo, andrewt, ytheod}@unipi.gr
http://www.datastories.org/

Abstract. In this paper, we propose a novel unified online group pattern mining algorithm, *EvolvingClusters*, that aims to enrich geospatial data through the mapping of their group behaviour. Specifically, *Evolving-Clusters* is used to discover collective movement behaviour (like flocks and convoys) by monitoring the activity of multiple clusters through time and space. We evaluate the aforementioned algorithm using a real-world marine traffic dataset consisting of vessels' movement in Brest Bay, France. Our study demonstrates the efficiency and effectiveness of the proposed algorithm as well as its value towards a semantic enrichment tool that can be used to observe and categorize the behaviour of multiple moving objects in real time.

Keywords: Big data · Data analytics · Maritime intelligence · Collective movement behaviour · Group patterns · Flocks · Convoys · Semantic enrichment

1 Introduction

Mobility Data Analytics [9,2,21] is an ever growing branch of the general spectrum of Data Analytics. GPS-enabled mobile phones, cars, airplanes, and vessels are the most common data sources broadcasting volumes of location information. Using them as-is (i.e., in their "raw" form), offer us limited usefulness; however, with proper processing (cleansing, transformation, enrichment etc.) and analysis (pattern discovery), the vast amount of available data can produce some very interesting and insightful stories. The outcome of data analytics over mobility data is of great interest to researchers and practitioners of the field.

More specifically, in the field of semantic enrichment, behavioural clustering can provide a concise and meaningful base that can be of value to multiple mining methodologies. Classification with the use of artificial neural networks for example, is a process that requires vast amounts of data, computational resources and time. Using a behavioural clustering technique like EvolvingClusters can be very beneficial, especially with respect to time and resources, since the classifier will be able to train on a smaller set of objects that belong to multiple different clusters instead of the full dataset that might contain objects with a lot of similarities.

This paper focuses on mobility data analytics over maritime traffic data. In particular, our purpose is to evaluate group movement behaviour at sea (e.g. flocks, convoys) over enriched trajectories of vessels.

The contributions of our work are summarized in the following lines:

– We enrich vessel movement data with annotations regarding their closeness to ports, etc.
– We design and evaluate a unified group behaviour discovery algorithm able to simulate existing pattern discovery methods, such as flocks and convoys.
– We evaluate the above over a large-volume real-world maritime trajectory dataset [22].

Our paper is structured as follows: In Section 2, we present background knowledge and related work. In Section 3, we provide our problem formulation and discuss what is special about maritime data. In Section 4, we present our *Evolving Clusters* algorithm for unified group pattern mining. In Section 5, we discuss preliminary experimental results. Section 6 concludes the paper, also giving hints for future work.

2 Background Knowledge and Related Work

The field of trajectory data mining [27] is rich in methods capturing collective movement of objects, i.e. sets of objects moving close to each other for a certain time period.

Flocks [4,10,25] take into account the spatial proximity and the direction of moving objects. For a flock pattern to be discovered, a minimal number of trajectories that satisfy such constraints are required. Formally, a flock valid during a time interval I, where I spans for at least k successive timepoints, consists of at least m objects, such that for every timepoint in I, there is a disk of radius r that contains all those entities. Technically, a flock discovery algorithm is tuned by three parameters: k (the minimum number of successive timepoints), m (the minimum number of neighboring objects), and r (the radius that defines the neighborhood). Companion [24] and Gathering [26] are two flock variations, focusing on online / streaming applications.

A *convoy* [12,13,20] is a group of objects consisting of at least m objects that are density-connected with respect to a density-reachability distance threshold e, during at least k consecutive timepoints. Specifically, assuming the partitioning of the database of the objects' locations with respect to a discretization of the time dimension, a snapshot S_i (i.e., the set of objects and their locations that exist at time t_i), is clustered using a typical density-based spatial clustering algorithm like DBSCAN [7], to identify dense groups of objects in S_i that are close to each other and the density of the group meets the density constraints of the clustering algorithm, i.e. the minimum number of objects in an object's neighborhood, $MinPts$, and the maximum distance for two objects to be directly density-reachable, e, according to DBSCAN's parameters. Technically, a convoy discovery algorithm is tuned by three parameters: k, m (as defined in flocks above), and e. Compared to flocks, convoys actually differ in that the circular neighborhood is replaced by the notion of density-connection. Convoy variations include *groups* [17] and *evolving groups* [15].

A *swarm* [19] is a collection of moving objects with cardinality of at least m, that are part of the same density-based cluster, defined by a reachability distance threshold e, for at least k (not necessarily consecutive) timepoints. Moreover, comparing the clusters themselves, the population is not required to remain unchanged but at least one cluster containing all objects should be discovered. Note that the trajectory of each object in-between these timepoints, is not under any constraint. Technically, a swarm discovery algorithm is tuned by the same three parameters, k, m, and e, as in the cases of moving clusters and convoys above, with the main difference being that swarms do not require the set of at least k timestamps to be consecutive.

Further related work includes the following. A *moving cluster* [14] is a sequence of clusters $c_1, \ldots, c_k$, such that for each timestamp t_i, clusters c_i and c_{i+1} share a sufficient number of common objects. Intuitively, if the two spatial clusters at two consecutive snapshots have a large percentage of common objects then they are considered a moving cluster between these two timestamps. A *moving micro cluster* [18] is a group of objects that are not only close to each other at the current time, but they are also expected to move together in the near future; techniques for maintaining clusters of moving objects by considering the clusters of the current and near-future positions are proposed in [11]; [6] presents a taxonomy/classification of movement patterns along a set of dimensions that reveal their behavior (and commonalities); [3] demonstrates the shortcomings of the Jaccard (J) measure when it is used for assessing the significance of co-occurrences among spatiotemporal instances with highly different spatiotemporal evolution characteristics and presents two extended novel measures (J^+ and J^*) that address the problems linked to the J measure; [5] studies a regional semantic trajectory pattern mining problem, aiming at identifying all the regional sequential patterns in semantic trajectories.

Most related to our work, [16] defines various mobility behaviors around the idea of the flock pattern; in particular, the Relative Motion (REMO) model and a respective language are proposed in order to express a number of collective mobility patterns under a unified representation. [23] proposes, among others, gpattern and crosspattern, two generic query operators implemented and validated in the *Secondo MOD system* [1], which express groups of moving objects that follow similar motion and mutually interact together, respectively (mobility behaviors, such as flocking, convergence, and leadership can be simulated through these operators).

With respect to related work, our method handles closeness of moving objects in a unified way under a graph-based approach, being able to simulate the most popular patterns (i.e. flocks and convoys) in an online mode.

3 Problem Formulation

An informal *group pattern* definition could be: "a large enough amount of objects moving along paths close to each other for a certain time". These objects could vary from animals (e.g. wolves, birds, lions, etc.) to human transportation means (e.g. cars, airplanes and vessels). Discovering these patterns can give us an insight regarding the behavior of these moving objects, for instance on hunting (wolves, lions), migration (birds), traffic monitoring (cars) and fishing pressure (fishing vessels). In this paper we aim at handling group pattern discovery in a uniform way, where "closeness" is formulated in graph-based terminology.

3.1 Problem definition

Definition 1. *(Evolving Cluster). Given: a set T of moving objects, where the trajectory of each object consists of r pairs (p_i, t_i), a minimum cardinality threshold c, a maximum distance threshold θ, and a minimum time duration threshold d, an Evolving Cluster $\langle C, t_{start}, t_{end}, tp\rangle$ is a subset $C \in T$ of the moving objects' population, $|C| \geq c$, which appeared at time point t_{start} and remained alive until time point t_{end} (with $t_{end}{-}t_{start} \geq d$) during the lifetime $[t_{start}, t_{end}]$ of which the participating moving objects were spatially connected with respect to distance θ and cluster type tp.*

The term "spatially connected" is used on purpose in the above definition, since the structure of our method accounts for a number of different clustering methodologies. In this study, we use both spherical and density-based clustering in order to mine flock and convoy-like patterns, respectively. In particular, for each time point, let us consider the mapping of the points of the moving objects' trajectories (that are active at that time point) in a connectivity graph $G(V, E)$, where vertex $v \in V$ represents a point and edge $e \in E$ represents a pair of points if and only if their distance is less than the given threshold θ; *Cliques* in this graph correspond to spherical-like clusters whereas *Maximal Connected Subgraphs (MCS)* in this graph correspond to density-connected clusters. Cliques (maximal connected subgraphs) that remain alive for an adequate period of time are evolving clusters, according to the above definition, that resemble flock (convoy, respectively) patterns. This concept is better illustrated in Figure 1.

According to Figure 1, sets $C_1 = \{a, b, c, d\}$ and $C_2 = \{a, b, c, d, e, f\}$ form a Clique and an MCS, that remain active for three time points $t_1, \ldots, t_3$, while $C_3 = \{a, b, c\}$ and $C_4 = \{d, e, f\}$ form a Clique and an MCS, that remain active during all four time points $t_1, \ldots, t_4$. Assuming thresholds e.g. $c = 3$ and $d = 3$, we have discovered three *Evolving Clusters*, the spherical-like $\langle C_1, t_1, t_3, 1\rangle$ and $\langle C_3, t_1, t_4, 1\rangle$, and the density-connected $\langle C_2, t_1, t_3, 2\rangle$ and $\langle C_4, t_1, t_4, 2\rangle$, where cluster type 1(2) corresponds to Clique (MCS, respectively). This example illustrates that two evolving clusters can be overlapping with respect to their population.

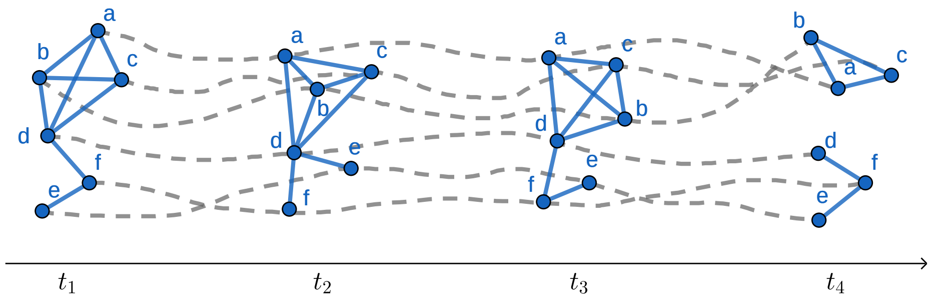


Fig. 1: An example of six objects moving at four consecutive time points and the respective connectivity graphs.

3.2 What is special about maritime data

It is well known that sensor-based information is sensitive to errors due to device malfunctioning. Therefore, a necessary step before performing data analytics tasks is that of pre-processing. A necessary clarification is that since the vessels' locations are recorded in angular (lat/lon) coordinates, we use the Haversine formula as it takes in account the data points' geodesic properties.

In general, pre-processing of GPS-based location data includes data cleansing (noise elimination, location smoothing, etc.) as well as data transformation tasks necessary for the analysis that will follow (fixed rate resampling, trajectory segmentation, etc.) [21]. A typical data preprocessing workflow consists of the following steps:

1. Data Cleansing:
   a. Remove time-based duplicate records;
   b. Remove position-based outliers (i.e. invalid speed, acceleration, etc.);
2. Data transformation
   a. Create Trips from vessels' locations;
   b. (Optional) Perform fixed-rate resampling on Trips;

In particular for Step 2a and in order to organize vessels' locations in trips, a popular approach (in case the ports are given as points instead of polygons) is to create a circle with radius ρ around each port's location in order to approximate their geometry and then, detect port entry and exit points for each vessel trajectory (Port-based Segmentation).

Then, for each produced segment, we may detect pairs of points with temporal difference greater than a given threshold (Temporal-based Segmentation). These pairs signify the transition from the current to the next *Trip*.

The segmentation due to the above steps, may result in a very low number of points. Because they do not offer any significant information, we decide to filter out these particular *Trips* (in particular, those consisting of less than 3 points).

Depending on their connection with ports, vessels' trips can be classified in 4 classes:

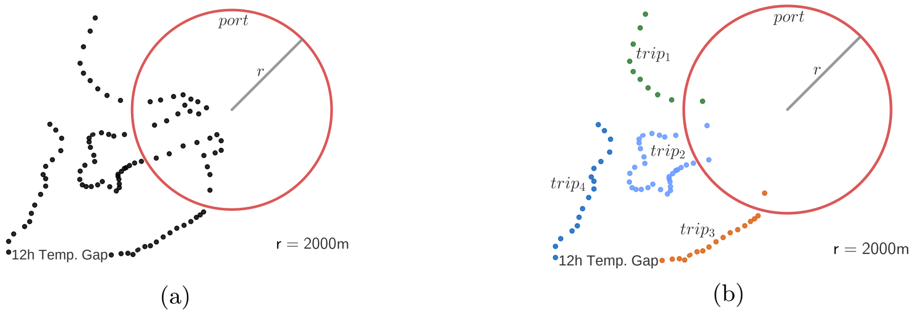


Fig. 2: A sample trajectory: (a) before; and (b) after trajectory segmentation into trips.

- Class C1 — trips that start and end at a port;
- Class C2 — trips that start at a port and end at open sea;
- Class C3 — trips that start at open sea and end at a port;
- Class C4 — trips that start and end at open sea;

The aforementioned methodology is illustrated in Figure 2, where the raw location information is compared with a port's location, hence a vessel trajectory is segmented into trips of Class C1 (e.g. $trip_2$ in Figure 2(b), Class C2 ($trip_3$), Class C3 ($trip_1$), and Class C4 ($trip_4$).

Given that a vessel traffic dataset consists of GPS points that are sampled whenever the captain of each vessel enables the AIS transmitter, it is obvious that there is no form of consistency regarding the time intervals between points. For example, it is easily observable that a vessel is highly likely to stop transmitting for a considerable amount of time if that vessel is inactive, e.g being stationary on a port. As a result, while also keeping in mind that several techniques used for future location prediction as well as group pattern mining need or benefit substantially by a stable rate of sampling and, by extension, a temporal alignment, a fixed-rate resampling technique [8] is used to achieve the consistency needed; see Figure 3 for an illustration of the above discussion.

4 The EvolvingClusters algorithm

In this section we present an algorithm, called *EvolvingClusters*, in order to detect and extract group patterns from raw GPS data points. This algorithm is fully modular, mining clusters with respect to the spatial clustering restrictions stated in the previous sections and then by applying the temporal restrictions, can fetch different types of group patterns simultaneously (in our case, *Cliques* and *MCS*). Due to the fact that we only compare our pattern history with the current time-slice, the algorithm can be connected to a data stream, thus having an online nature.

Algorithm 1 presents the algorithm's corpus. In particular it discovers evolving clusters in a trajectory dataset D, where moving objects' locations arrive

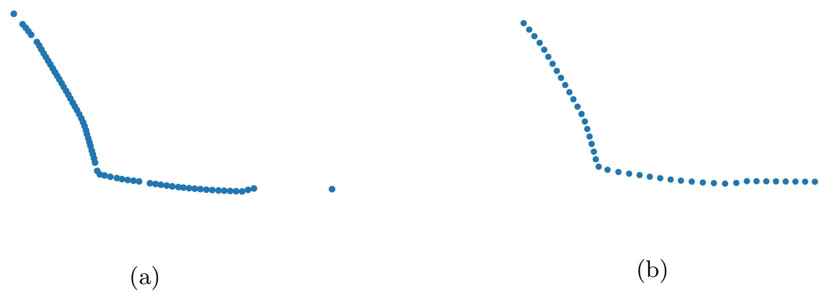


Fig. 3: A sample vessel trip: (a) before; and (b) after resampling (with 1 min. fixed sampling rate).

Algorithm 1: EvolvingClusters. An online algorithm capable of mining the Group Patterns as discussed in the previous subsections

```
   Input: A Dataset $D = \{T_1, T_2, \ldots, T_n\}$ of Time-slices $T_i$ consisting of objects'
          timestamped locations $(p_j, t_i)$, Distance Threshold $\theta$, Time Threshold
          $t$, Cardinality Threshold $c$
   Output: A list of all the mined patterns $MinedPatterns$
 1 $ActivePatterns, ClosedPatterns \leftarrow []$
 2 for Time-slice $T$ in $D$ do
 3     $C_{Cliques}, C_{MCS} \leftarrow GeospatialClustering(T, \theta, c)$
 4     for CurrentClusters in $\{C_{Cliques}, C_{MCS}\}$ do
 5         if $ActivePatterns == \emptyset$ then
 6             $ActivePatterns \leftarrow CurrentClusters$
 7         else if $CurrentClusters == \emptyset$ then
 8             $ClosedPatterns \leftarrow \{ActivePattern \in ActivePatterns :$
                 $ActivePattern.end - ActivePattern.start \geq t\}$
 9         else
10             $ActivePatterns, InactivePatterns \leftarrow$
                 $FindPatterns(CurrentClusters, ActivePatterns, \theta)$
11             $ClosedPatterns \leftarrow \{InactivePattern \in InactivePatterns :$
                 $InactivePattern.end - InactivePattern.start \geq t\}$
12         end
13         output
             $\{Pattern \in ActivePatterns : Pattern.end - Pattern.start \geq t\}$
14     end
15 end
```

at predefined timepoints (e.g. every 60 sec.) or, in other words, at a fixed (and aligned amongst all objects) sampling rate.

In the following paragraphs we provide a thorough explanation regarding its operation. Algorithm 1 is responsible of using the results provided by Algorithm 3 in a sequential manner. Essentially Algorithm 1 uses the results of Algorithm 2 and decides if the available data in the form of $ActivePatterns$ (patterns previously mined) and $CurrentClusters$ (clusters formed based on the location of moving objects at the current time-slice) are eligible to be used as input to Algorithm 3. If not (either set is empty), the algorithm either moves the clusters currently active to the $ActivePatterns$ set (if $ActivePatterns = \emptyset$) or moves all the patterns that satisfy the thresholds given from $ActivePatterns$ to $ClosedPatterns$ (if $CurrentClusters = \emptyset$).

Algorithm 2: Geospatial Clustering. Clusters GPS Points given a Time-slice

```
Input: Time-slice $T = \{p_1, p_2, \ldots, p_n\}$ of coordinate points, Distance
       threshold $\theta$, Cardinality threshold $c$
Output: Clusters of the Time-slice's Points $C_{Cliques}, C_{MCS}$
1 $DistanceMatrix \leftarrow PairwiseDistance(T,\ metric = \text{“}Haversine\ Distance\text{”})$
2 $Pairs \leftarrow \{(p_i, p_j) : DistanceMatrix(p_i, p_j) < \theta\}$
3 $G \leftarrow \text{Graph}(edges = pairs)$
4 $C_{Cliques} \leftarrow \{C \in G.Cliques() : |C| \geq c\}$
5 $C_{MCS} \leftarrow \{C \in G.MaximalConnectedSubgraphs() : |C| \geq c\}$
6 return $C_{Cliques}, C_{MCS}$
```

Algorithm 3 takes all the following *cases* into consideration: (for pattern C_{t_i} at time t_i and $C_{t_{i+1}}$ at time t_{i+1})

1. The patterns are identical ($C_{t_i} = C_{t_{i+1}}$)
2. The patterns have no common objects ($C_{t_i} \cap C_{t_{i+1}} = \emptyset$)
3. The pattern C_{t_i} is a subset of $C_{t_{i+1}}$ ($C_{t_i} \subset C_{t_{i+1}}$)
4. The pattern $C_{t_{i+1}}$ is a subset of C_{t_i} ($C_{t_{i+1}} \subset C_{t_i}$)
5. The patterns contain some common objects ($C_{t_i} \cap C_{t_{i+1}} \neq \emptyset$, $C_{t_i} \cap C_{t_{i+1}} \subset C_{t_i}, C_{t_{i+1}}$)

Therefore, the algorithm operates as follows:

– For every pair of consecutive (with respect to time) pattens, if the cardinality of their intersection is greater than c, add it to the $ActivePatterns$ set (lines: 4–7).
– For every pattern in $C_{t_{i+1}}$, if the list of its intersections with all of the patterns in C_{t_i} doesn't contain the pattern, add it to the $ActivePatterns$ set as a new pattern (lines: 8–9).
– For every pattern in C_{t_i}, if it is not part of the $ActivePatterns$ set, add it to the $InactivePatterns$ set (line: 11).

– Replace each group of duplicate patterns in the *ActivePatterns* set, with a single record of each pattern and substitute its starting and ending timestamps with the oldest starting and newest ending timestamps available in the duplicate group (lines: 12–17).

We observe that in all *cases* the pattern that ought to be maintained through time is the intersection of C_{t_i} and $C_{t_{i+1}}$. *Cases 2 and 3* require some extra attention. Regarding *case 2*, the intersection is an empty set. As a result, $C_{t_{i+1}}$ should be maintained and added to the *ActivePatterns* set. *Case 3* dictates that both the new superset and the previous pattern should be maintained since they both exist at the same time as part of $C_{t_{i+1}}$.

Algorithm 3: FindPatterns. Compares the current with the closed clusters in order to determine their evolution

```
   Input: Consecutive datasets $D_{left}, D_{right}$, Cardinality threshold $c$
   Output: Mined patterns $ActivePatterns, InactivePatterns$
 1 $ActivePatterns \leftarrow []$
 2 for $Pattern\ P_R$ in $D_{right}$ do
 3     $IntersectionList \leftarrow \{\}$
 4     for $Pattern\ P_L$ in $D_{left}$ do
 5         if $|P_R \cap P_L| \geq c$ then
 6             $ActivePatterns.append([[P_R \cap P_L, P_L.start, P_R.end]])$
 7     end
 8     if $IntersectionList = \emptyset$ then
 9         $ActivePatterns \leftarrow P_R$
10 end
11 $InactivePatterns \leftarrow \{pattern \in P_L : pattern \notin ActivePatterns\}$
12 for $Pattern\ P_{active}$ in $ActivePatterns$ do
13     $DuplicatePatterns \leftarrow [pattern_A, pattern_B \in P_{active} : (pattern_A =$
        $pattern_B) \wedge (pattern_A \neq pattern_B)]$
14     if $|DuplicatePatterns| \neq 0$ then
15         $P_{active}.start \leftarrow min(DuplicatePatterns.start)$
16         $P_{active}.end \leftarrow max(DuplicatePatterns.end)$
17 end
18 return $ActivePatterns, InactivePatterns$
```

Based on Figure 1 the proposed algorithm for $c = 3, t = 3$ would mine the patterns $C_1 = \{a, b, c, d\}$, $C_2 = \{a, b, c, d, e, f\}$, $C_3 = \{a, b, c\}$ and $C_4 = \{d, e, f\}$ as follows:

– t_1: Clique $\langle C_1, t_1, t_1, 1\rangle$ and MCS $\langle C_2, t_1, t_1, 2\rangle$ mined (Output: $\emptyset$);
– t_2: Clique $\langle C_1, t_1, t_2, 1\rangle$ and MCS $\langle C_2, t_1, t_2, 2\rangle$ mined (Output: $\emptyset$);
– t_3: Clique $\langle C_1, t_1, t_3, 1\rangle$ and MCS $\langle C_2, t_1, t_3, 2\rangle$ mined (Output: $\{C_1, C_2\}$);
– t_4: Clique $\langle C_3, t_1, t_4, 1\rangle$ and MCS $\langle C_4, t_1, t_4, 2\rangle$ mined (Output: $\{C_3, C_4\}$).

For timestamps t_1 through t_3, C_1 and C_2 are mined and maintained. During timestamp t_4, two new patterns are found (C_3 and C_4), however both new patterns are present during t_3 as *subsets* of C_1 and C_2 respectively. Thus they get to keep the starting timestamp of their respective *past* supersets.

5 Experimental Study

In this section we prepare the dataset that the algorithm will be tested on and present some preliminary results regarding its effectiveness.

5.1 Dataset preparation

In our study, we use a publicly available maritime dataset, called Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance [22], which contains information on maritime traffic in France. The dataset ranges in time and space as follows:

- temporal range: October 1st, 2015 to March 31st, 2016 (6 months);
- spatial range: latitude in [45.00, 51.00], longitude in [-10.00, 0.00] (Celtic sea, the Channel and Bay of Biscay).

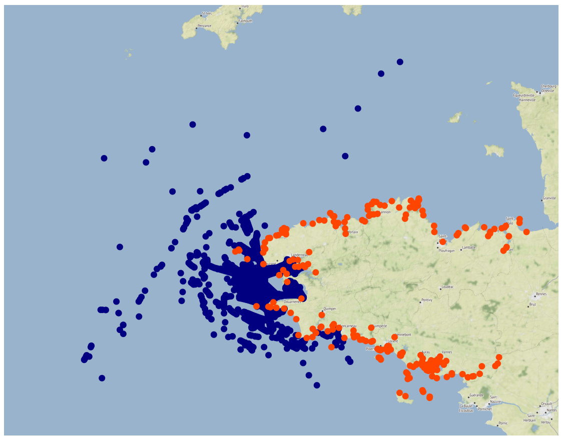


Fig. 4: A snapshot from the Brest dataset: sample of AIS positions on March 1st, 2016 (blue dots) and ports of interest (red dots).

A map visualization of (a part of) the dataset is illustrated in Figure 4. The original dataset contains three classes of information: *vessel-dynamic* (i.e., related to the vessels' movement), *vessel-static* (i.e., related to the vessels' identity), and *geo-related* data (locations of ports, environmental information, etc.). For the purposes of our study, we exploit on the entire vessel-dynamic and vessel-static information available while from the third class we are only interested in the locations of ports, information which is essential for the analysis we design to perform. In particular:

- The *vessel-dynamic* data contains approximately 19 million records. Each record corresponds to an AIS signal and includes the vessel identity (mmsi), its position (lon, lat), the timestamp this position was recorded (ts) as well as other mobility-related information provided by vessel's sensors (speed, course, heading, etc.).
- The *vessel-static* data contains information about vessel registration, such as the vessel's identity (mmsi), radio frequency (call sign), name, type, size, etc.
- The *geo-related* data we used in our study contains 222 records; each record corresponds to a port along with its name and location (point geometry).

Due to step 2a (recall the preprocessing steps of Section 3.2), with port radius set at 2km ($\approx$ 1.08 n.m.) and temporal threshold at 12 hrs., trajectory segmentation yields 9,545,789 data points organized in 24,159 trips from 3,279 vessels (segments with very few data points - i.e. less than 3 - are discarded).

#Records	Number of AIS Records.	9,545,789
#Vessels	Total number of vessels.	3279
#Trips	Total number of trips.	24,159
#Trips Class C1	Total number of trips that started and ended at a port.	11,690
#Trips Class C2	Total number of trips that started at a port and ended at open sea.	2580
#Trips Class C3	Total number of trips that started at open sea and ended at a port.	1849
#Trips Class C4	Total number of trips that started and ended at open sea.	8040

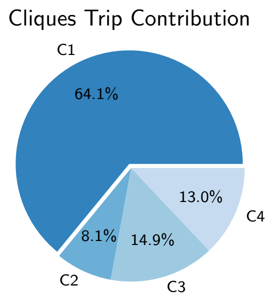
Table 1: Statistics of the dataset after the pre-processing step.

5.2 Preliminary results

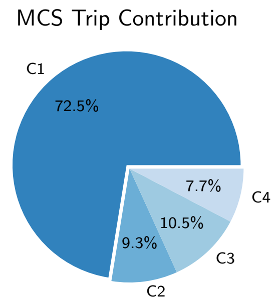
Having processed our dataset using the methodology presented in Section 3.2, we tested our algorithm on a wide range of values for each parameter, namely:

- Cardinality Threshold (c): $3, 5, 8, 12$. Default: 5 vessels
- Temporal Threshold (t): $15, 30, 45, 60$. Default: 15 minutes
- Distance Threshold (θ): $0.25, 0.5, 0.75, 1, 1.25$. Default: 1 nautical mile

Figure 5 illustrates the average percentage of trip classes C1 - C4 in the mined group patterns (using the default parameters). In either pattern type, we observe that C1 is the most dominant class (having more than 60% participation), which is reasonable since the same participation appears – more or less – at trip level (see Table 1). On the other hand, C4 presents an interesting behaviour: although its percentage at trip level is about 30%, this percentage falls down to 13% within cliques and 7.7% within MCSs. Comparing the two pattern types (cliques and MCSs) with each other, we observe that cliques appear to be formed more frequently than MCSs when we focus on C3 or C4 while the opposite is the case when we focus on C1 or C2. These findings may trigger domain experts to take a deeper look and reach insightful conclusions.



(g)



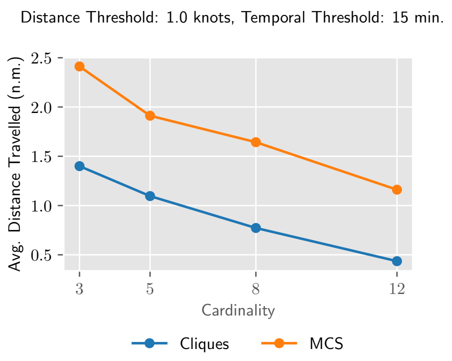
(h)

Fig. 5: Trip contribution on mined (g) Cliques (h) MCS.

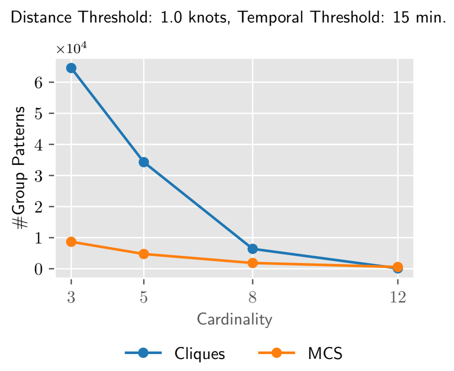
Figure 6 illustrates the change of *average distance travelled* (*#group patterns*, respectively) with respect to one of the algorithm's parameters, while the others are fixed to their respective default values. It can be observed that as c increases, both types of group patterns decline both in their respective average distance travelled and their cardinality (Figures 6a and 6b, respectively), while on the other hand, as θ increases, the opposite can be seen (Figures 6c and 6d, respectively). Moreover, as t increases, we observe a steady rise in the average distance travelled for both patterns but at the cost of having fewer patterns (Figures 6e and 6f, respectively). Furthermore, it is shown that *Cliques* are quite sensitive with respect to their thresholds while *MCS* as less sensitive, showing a more steady growth/decline (Figures 6b,d and f). Last but not least, as illustrated by Figures 6a,c and e, a linear-like correlation can be observed between the thresholds c, t, θ and the average distance travelled.

6 Conclusions and Future Work

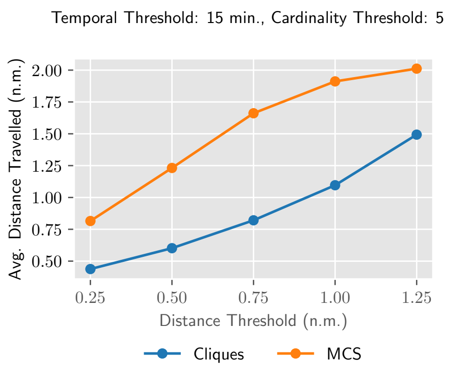
In this paper, we proposed a unified online group pattern mining algorithm, called EvolvingClusters, which is used to discover collective movement behaviour (like flocks and convoys) by monitoring the activity of multiple clusters through time and space. The algorithm is graph-based in the sense that it maintains evolving *Cliques* and *Maximal Connected Subgraphs (MCS)*, thus simulating spherical and density-based evolving clusters. Our study on a large real-world maritime traffic dataset demonstrates the efficiency and effectiveness of the proposed algorithm. The results show that our method is capable of detecting a



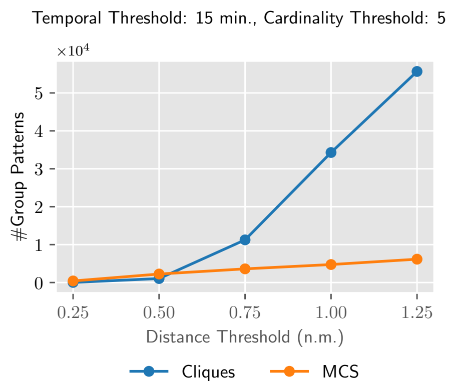
(a)



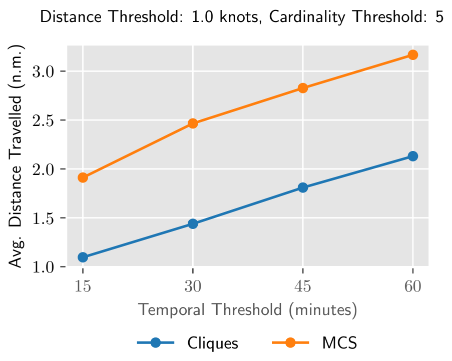
(b)



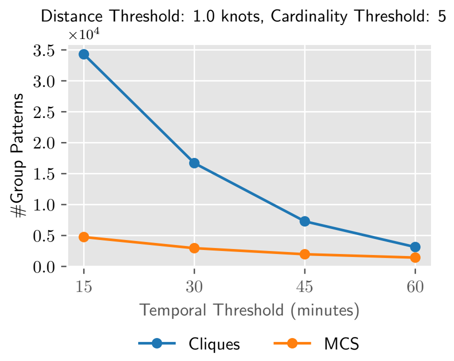
(c)



(d)



(e)



(f)

Fig. 6: Statistics on mined group patterns: cardinality, distance and temporal threshold vs. (a,c,e) avg. distance travelled and (b,d,f) #group patterns.

large amount of group patterns in the given dataset. Thus, based on the potential applications, some of which were mentioned above, as well as the quality of the results produced, we believe that *EvolvingClusters* can be a valuable tool for researchers and practitioners alike.

In the near future we aim to test and evaluate *EvolvingClusters* against other state-of-the-art techniques, using other types of mobility data, such as aviation and public transportation data. Based on our assumptions, the algorithm should function at the same quality level no matter the data type used, since its approach does not make use of any other apriori form of knowledge like road grids or hot-paths. Another set of experiments that we would like to conduct in the near future is using data with different sampling rates as input for *EvolvingClusters*. If the results appear to be in the same quality level with those produced from a dataset with a much higher sampling rate as input, we would be certain that the value of the algorithm is not tied to the sampling rate of the given data. Our long-term plans involve around the creation of a framework that will use the information that is mined using *EvolvingClusters* to classify moving objects into different classes based on their behaviour. By extracting as much information as possible from the available data and combining a well trained classifier with a well defined set of groups with similar behaviour, we want to create a system able to model and – if possible – predict a set of suspicious activities that might consist a violation of law or a possible criminal activity.

Acknowledgments

This work was partially supported by the Greek Ministry of Development and Investment, General Secretariat of Research and Technology, under the Operational Programme Competitiveness, Entrepreneurship and Innovation 2014-2020 (grant T1EDK-03268, i4Sea).

References

1. de Almeida, V.T., Güting, R.H., Behr, T.: Querying moving objects in SECONDO. In: MDM. p. 47. IEEE Computer Society (2006)
2. Andrienko, G.L., Andrienko, N.V., Bak, P., Keim, D.A., Wrobel, S.: Visual Analytics of Movement. Springer (2013)
3. Aydin, B., Küçük, A., Angryk, R.A., Martens, P.C.: Measuring the significance of spatiotemporal co-occurrences. ACM Trans. Spatial Algorithms and Systems **3**(3), 9:1–9:35 (2017)
4. Benkert, M., Gudmundsson, J., Hübner, F., Wolle, T.: Reporting flock patterns. Comput. Geom. **41**(3), 111–125 (2008)
5. Choi, D., Pei, J., Heinis, T.: Efficient mining of regional movement patterns in semantic trajectories. PVLDB **10**(13), 2073–2084 (2017)
6. Dodge, S., Weibel, R., Lautenschütz, A.: Towards a taxonomy of movement patterns. Information Visualization **7**(3-4), 240–252 (2008)
7. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD. pp. 226–231. AAAI Press (1996)

8. Georgiou, H.: 1-pass fixed-rate linear resampler in matlab/octave (2017)
9. Giannotti, F., Pedreschi, D. (eds.): Mobility, Data Mining and Privacy - Geographic Knowledge Discovery. Springer (2008)
10. Gudmundsson, J., van Kreveld, M.J.: Computing longest duration flocks in trajectory data. In: GIS. pp. 35–42. ACM (2006)
11. Jensen, C.S., Lin, D., Ooi, B.C.: Continuous clustering of moving objects. IEEE Trans. Knowl. Data Eng. **19**(9), 1161–1174 (2007)
12. Jeung, H., Shen, H.T., Zhou, X.: Convoy queries in spatio-temporal databases. In: ICDE. pp. 1457–1459. IEEE Computer Society (2008)
13. Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., Shen, H.T.: Discovery of convoys in trajectory databases. PVLDB **1**(1), 1068–1080 (2008)
14. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. In: SSTD. Lecture Notes in Computer Science, vol. 3633, pp. 364–381. Springer (2005)
15. Lan, R., Yu, Y., Cao, L., Song, P., Wang, Y.: Discovering evolving moving object groups from massive-scale trajectory streams. In: MDM. pp. 256–265. IEEE Computer Society (2017)
16. Laube, P., Imfeld, S., Weibel, R.: Discovering relative motion patterns in groups of moving point objects. International Journal of Geographical Information Science **19**(6), 639–668 (2005)
17. Li, X., Ceikute, V., Jensen, C.S., Tan, K.: Effective online group discovery in trajectory databases. IEEE Trans. Knowl. Data Eng. **25**(12), 2752–2766 (2013)
18. Li, Y., Han, J., Yang, J.: Clustering moving objects. In: KDD. pp. 617–622. ACM (2004)
19. Li, Z., Ding, B., Han, J., Kays, R.: Swarm: Mining relaxed temporal moving object clusters. PVLDB **3**(1), 723–734 (2010)
20. Orakzai, F., Calders, T., Pedersen, T.B.: k/2-hop: Fast mining of convoy patterns with effective pruning. PVLDB **12**(9), 948–960 (2019)
21. Pelekis, N., Theodoridis, Y.: Mobility Data Management and Exploration. Springer (2014)
22. Ray, C., Dreo, R., Camossi, E., Jousselme, A.L., Iphar, C.: Heterogeneous integrated dataset for maritime intelligence, surveillance, and reconnaissance. Data in Brief (2019)
23. Sakr, M.A., Güting, R.H.: Group spatiotemporal pattern queries. GeoInformatica **18**(4), 699–746 (2014)
24. Tang, L.A., Zheng, Y., Yuan, J., Han, J., Leung, A., Hung, C., Peng, W.: On discovery of traveling companions from streaming trajectories. In: ICDE. pp. 186–197. IEEE Computer Society (2012)
25. Vieira, M.R., Bakalov, P., Tsotras, V.J.: On-line discovery of flock patterns in spatio-temporal data. In: GIS. pp. 286–295. ACM (2009)
26. Zheng, K., Zheng, Y., Yuan, N.J., Shang, S., Zhou, X.: Online discovery of gathering patterns over trajectories. IEEE Trans. Knowl. Data Eng. **26**(8), 1974–1988 (2014)
27. Zheng, Y.: Trajectory data mining: An overview. ACM TIST **6**(3), 29:1–29:41 (2015)