

Colossal Trajectory Mining

Semantic Co-movement Pattern Mining (Discussion Paper)

Chiara Forresi¹, Matteo Francia^{1,*}, Enrico Gallinucci¹, Matteo Golfarelli¹ and Manuele Pasini¹

¹DISI – University of Bologna, Cesena, Italy

Abstract

Spatio-temporal mobility patterns are at the core of strategic applications such as urban planning and monitoring. Depending on the strength of spatio-temporal constraints, different mobility patterns can be defined. While existing approaches work well in the extraction of groups of objects sharing fine-grained paths, the huge volume of large-scale data asks for coarse-grained solutions. Colossal Trajectory Mining (CTM) efficiently extracts heterogeneous mobility patterns out of a multidimensional space that, along with space and time dimensions, can consider additional trajectory features (e.g., means of transport or activity) to characterize behavioral mobility patterns. The algorithm is natively designed in a distributed fashion, and the experimental evaluation shows its scalability with respect to the involved features and the cardinality of the trajectory dataset.

Keywords

Trajectory mining, Mobility patterns, Frequent itemset mining

1. Introduction

The spreading of Internet of Things and mobile devices [1] stimulated the rise of applications based on mobility patterns, such as urban mobility and traffic planning [2], and moving objects (MOs) profiling and linking [3] where objects moving in similar locations can share interests or relationships. Since different applications require identifying different mobility behaviors, researchers tailored a plethora of specific patterns. However, the need for a *unifying* analytic framework is well-understood and debated in the literature [4, 5].

Generally speaking, a *mobility pattern* captures behaviors that frequently occur among MOs; each MO follows a trajectory that can be partially or completely shared with others. Mobility patterns can be classified by the strength of the constraints on spatial and temporal proximity. All patterns require spatial proximity to be satisfied for a sufficient length of the trajectories. Conversely, temporal proximity is not always mandatory: a mobility pattern can be interesting even if the same path has been traveled at different times. A co-location [6] is a group of objects located at the same points at any time (e.g., customers frequenting the same shops on different days). Similarly, a flow [7] is a co-location group where the path is contiguous (e.g., individuals moving through adjacent road segments). A swarm [8] is a group of objects moving within the

SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy

*Corresponding author.

✉ chiara.forresi@unibo.it (C. Forresi); m.francia@unibo.it (M. Francia); enrico.gallinucci@unibo.it (E. Gallinucci); matteo.golfarelli@unibo.it (M. Golfarelli); manuele.pasini@unibo.it (M. Pasini)

ORCID 0000-0001-5652-2455 (C. Forresi); 0000-0002-0805-1051 (M. Francia); 0000-0002-0931-4255 (E. Gallinucci); 0000-0002-0437-0725 (M. Golfarelli); 0009-0005-7023-6742 (M. Pasini)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

same points at the same, possibly non-consecutive, time instants (e.g., individuals attending the same sport events each week). Similarly, a convoy [9] is a swarm group sharing (at least some) consecutive timestamps (e.g., individuals sharing a means of transport).

Most of the approaches in the literature focus on a specific pattern and a few unifying approaches have been introduced. [10] and [11] target small groups of objects sharing fine-grained paths, however, the huge volume of large-scale data (e.g., hundreds of thousands of trajectories spanning the entire USA) asks for coarse-grained solutions. For instance, if our goal is to extract the groups of people flying across countries (e.g., USA has 50 states and 328 million inhabitants) or moving across city neighborhoods (e.g., Milan has 88 neighborhoods and 3 million inhabitants), meaningful groups are the ones in the order of hundreds/thousands of individuals. Indeed, extracting groups of 10 people given an average domestic flight of 150 people would return $\binom{150}{10} = 1.2 \cdot 10^{15}$ groups.

In this paper, we describe Colossal Trajectory Mining (CTM), previously introduced in [12], an approach that generalizes many of the previous approaches while overcoming their limitations and integration effort. CTM (i) is a unifying framework for the extraction of heterogeneous mobility patterns; (ii) is suitable for applications working on datasets with a huge number of (long) trajectories traveling across limited spatial regions (i.e., #trajectories \gg #regions); (iii) characterizes MOs through a tessellation including spatial and temporal features as well as additional features that enable the comprehension of semantic mobility behaviors [13] (e.g., characterizing mobility behaviors by means of transport or activity); and (iv) does not require similarity computation between trajectories, preventing the design of burdensome metrics.

The remainder of the paper is organized as follows. In Section 2, we position and compare our approach with respect to the related literature. In Section 3, we describe CTM. In Section 4, we assess the effectiveness and efficiency of CTM by leveraging both real-world and synthetic case studies. Finally, we summarize our approach and future research directions in Section 5.

2. Related work

Frequent Itemset Mining uncovers co-occurrences among the items in a transaction dataset [14]. FIM has been also applied to trajectory data to find routes [15, 16] or regions [17] frequently trafficked by MOs. This is an *orthogonal* problem to extracting groups of objects moving together.

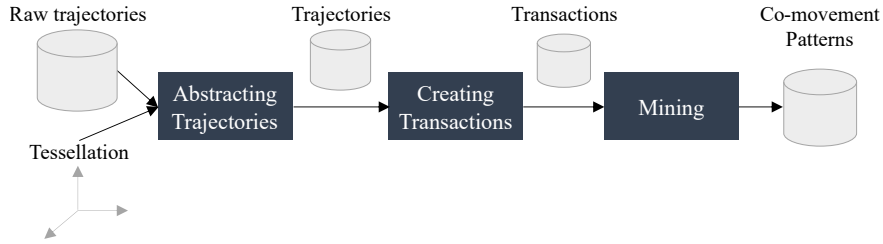
Colossal itemset mining is a branch of FIM that computes FI in highly-dimensional datasets; for instance, a dataset with 30 transactions/rows/instances each with 10^7 items/columns [18]. Carpenter [19] generates frequent closed itemsets in a depth-first centralized fashion. In [18], the authors distribute the branches of the depth-first exploration over distributed executors. However, (i) this results in highly unbalanced partitions, and (ii) a centralization mechanism must be iteratively applied to discard redundant branches (i.e., patterns are redundantly generated). While other algorithms for colossal itemset mining have been recently proposed [19, 20, 21], *none* address the extraction of constrained co-movement patterns.

Clustering groups trajectories such that intra-group similarity is maximized and inter-group similarity is minimized. Since “classic” clustering algorithms *do not* enforce a definition of co-movement patterns (e.g., cardinality, time span, cohesion, spatial and/or temporal contiguity), flock [22], convoy [9], and swarm [23] patterns have been formalized. As swarm [23] groups

Table 1

Excerpt of the comparison with existing co-movement patterns approaches

Features	Co-location	Flow	Convoy	Swarm	Distributed	Contributions
Any	✓	✓	✓	✓	✓	CTM
Space	-	✓	-	-	-	[24], [25], [26]
Space	✓	-	-	-	✓	[27]
Space, Time	-	-	-	✓	-	[28], [8]
Space, Time	-	-	-	✓	✓	[29], [30]
Space, Time	-	-	✓	-	-	[23], [31], [32]
Space, Time	-	-	✓	-	✓	[33], [34]
Space, Time	-	-	✓	✓	✓	[11]
Space, Time	-	✓	-	-	-	[7], [2], [35]

**Figure 1:** Overview of CTM.

objects moving in spatial proximity for a given amount of possibly-non-contiguous time, classical clustering algorithms can be referred to as swarms if no minimal duration is enforced.

Table 1 summarizes the comparison with existing approaches.

3. Mining semantic co-movement patterns

Our approach is based on the following steps (Figure 1).

3.1. Abstracting trajectories

We consider a dataset of raw trajectories, where each trajectory point is labeled with a set of features. Mandatory features are those needed to define spatial or spatio-temporal locations (e.g., latitude, longitude, and timestamp), but other semantic features might be available.

Definition 1 (Raw trajectory and feature). A raw trajectory P is a sequence of points $(p_1, \dots, p_{|P|})$ generated by a MO. The feature space is a set of features $F = \{f_1, \dots, f_{|F|}\}$ that is collected for each point.

Raw trajectories are mapped to a multidimensional tessellation, a composition of multidimensional tiles (from now on, simply *tiles*) of *any* shape – with no overlaps and no gaps – that cover a multidimensional space; the tessellation *is not* necessarily a regular grid. Each dimension of the tessellation corresponds to a feature describing the raw trajectory points: common features are space and time, but additional features can be added to specialize each trajectory point (e.g.,

whether an individual is moving by car or by bicycle). This allows grouping trajectories on semantic and behavioral concepts (e.g., to capture groups of individuals moving across city neighborhoods with different means of transport).

Definition 2 (Tessellation). We call tessellation a multidimensional partitioning $S = \{s_1, \dots, s_m\}$ of the feature space F . Each tile of the tessellation is identified by an id and is characterized by an interval/set of values for each continuous/nominal feature in F . For each feature f , the function $dist_f(s_i, s_j)$ computes the distance between two tiles on the tessellation. Two tiles $s_i, s_j \in S$ are adjacent ($s_i \cong_f s_j$) if $\forall s_z \in S$ it is $dist_f(s_i, s_j) \leq dist_f(s_i, s_z) + dist_f(s_z, s_j)$.

In our implementation, the distance function is

$$dist_f(s_i, s_j) = \begin{cases} f \text{ is ordinal} & geod(s_i, s_j) \\ f \text{ is nominal} \wedge s_i = s_j & 0 \\ f \text{ is nominal} \wedge s_i \neq s_j & \infty \end{cases} \quad (1)$$

where $geod(s_i, s_j)$ is the geodesic distance [36] computed on the tessellation, which is the number of tiles along the shortest path of neighboring tiles connecting s_i and s_j .

Definition 3 (Tile connection). Two tiles s_i, s_j in the tessellation S are connected ($s_i \leftrightarrow_f s_j$) if there exists a path of adjacent tiles in S connecting them in f .

A trajectory is an abstraction of a raw trajectory at the grain defined by the tessellation.

Definition 4 (Trajectory). Given a raw trajectory P and a tessellation S , we define the trajectory T corresponding to P in S as the sequence of tiles $(s_1, \dots, s_{|T|})$ such that a tile s is added to T if at least a point $p \in P$ is in s . A point p is in the tile s if, for each feature in F , the values of the feature for s contain the corresponding feature value characterizing p .

Noticeably, the tessellation: (i) defines the level of the analysis (CTM transparently allows the extraction of patterns at neighborhood/city/country scales) and (ii) compresses trajectories since a trajectory *moves through* a tile if at least one of its points belongs to the tile (a single tile instance is added to T if consecutive points fall in that tile, thus $|P| \geq |T|$).

3.2. Creating transactions

To formalize CTM as a colossal itemset mining problem, we introduce *transactions* and *items*.

Definition 5 (Item, itemset, and transaction). Given a trajectory dataset \mathcal{T} and a tessellation S , each trajectory represents an item, and a set I of trajectories is an itemset. We define the transaction for a tile $s \in S$ as the itemset containing all the trajectories having at least a point in s . \mathcal{Q} , the set of all transactions, is a transaction dataset.

As transactions are *sets* of items, the data is further compressed: if a tile is traversed more than once by a trajectory, the transaction (tile) will contain the item (trajectory) only once.

Since we are looking for trajectories sharing feature values, we need to identify the itemsets contained in a large number of transactions. This property is captured by the support function.

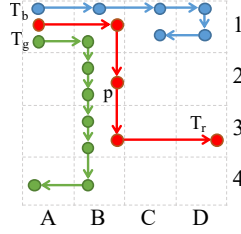


Figure 2: Trajectories moving through a tessellation.

Definition 6 (Support). Given a transaction dataset \mathcal{Q} , the support $sup(I) \subseteq \mathcal{Q}$ of an itemset I is the set of transactions containing I .

Definition 7 (Frequent and closed itemsets). An itemset is I frequent (FI) if $|sup(I)| \geq mSup$, where $mSup$ is the minimum number of transactions to consider the itemset as frequent. A frequent itemset is closed (FCI) if there exists no superset with the same support.

FCIs provide a lossless compression of FIs [37] (i.e., the output is non-redundant and the complete set of FIs is recoverable) which are exponential in the number of trajectories/items [14]. Working with FCIs rather than FIs simplifies data analysis [38].

Example 1 (Trajectory and transaction). With reference to Figure 2, T_b, T_g , and T_r are trajectories while Q_{A1} and Q_{B2} are transactions that correspond to tiles A1 and B2.

$$T_b = (A1, B1, C1, D1, C1), T_g = (A1, B1, B2, B3, B4, A4),$$

$$T_r = (A1, B1, B2, B3, D3), Q_{A1} = \{T_b, T_g, T_r\}, Q_{B2} = \{T_g, T_r\}$$

The maximum number of transactions depends on the number of tiles (i.e., $|\mathcal{Q}| = |S|$). Modeling trajectories as items and tiles as transactions makes our frequent itemset approach a *colossal* one since the number of tiles (i.e., transactions) is typically in the order of magnitudes smaller than the number of trajectories (i.e., items). This is a fair assumption to make: for instance, Milan has 88 neighborhoods with over $3 \cdot 10^6$ inhabitants (i.e., potential MOs). Obviously, our assumption is no more true when a very fine tessellation is adopted. For example, the Milan metropolitan area spans about $1500km^2$, corresponding to $1.5 \cdot 10^5$ uniform tiles with side $100m$ and $1.5 \cdot 10^7$ uniform tiles with side $10m$.

3.3. Mining co-movement patterns

We retrieve the sets of trajectories satisfying minimum cardinality (i.e., the number of trajectories), minimum support (i.e., the length of the shared path), and spatio-temporal constraints. Different mobility patterns can be obtained by specializing such constraints.

Definition 8 (Co-movement pattern). A co-movement pattern I is a FCI s.t. $|I| \geq mCrd$, where $mCrd$ is the minimum number of trajectories to consider a FCI as a co-movement pattern.

Table 2

Characterization of co-movement patterns

Pattern	Feature	Shape constraints
Co-loc.	f^{sp}	None
Flow	f^{sp}	$\exists S' \subseteq \text{sup}(I)$ s.t. $\forall s_i, s_j \in S', s_i \leftrightarrow_{f^{sp}} s_j \wedge S' \geq mSup$
Swarm	f^{sp}, f^{tm}	None
Convoy	f^{sp}, f^{tm}	$\exists S' \subseteq \text{sup}(I)$ s.t. $\forall s_i, s_j \in S', s_i \leftrightarrow_{f^{tm}} s_j \wedge S' \geq mSup$

This “basic” co-movement pattern can be specialized depending on the involved features. Co-location and flow are defined over a spatial feature (f^{sp} ; i.e., they are required to happen in the same spatial tile), while swarm and convoy require both spatial (f^{sp}) and temporal (f^{tm}) features (i.e., they are required to happen in the same spatio-temporal tiles). In other words, while for swarm and convoy it is necessary to be in the same spatio-temporal tile, co-location and flow only require to be in the same spatial tile (e.g., same location even if at different times).

Although the simplest representation of the space and time features is a regular binning of their absolute values, CTM allows adopting abstractions richer in semantics as long as these determine a tessellation (i.e., a partitioning) of space and time. Characterizing co-movement patterns with additional features means imposing additional constraints, which we call *behavioral* constraints. Note that behavioral constraints are more expressive than filtering trajectories based on a specific feature value since they further characterize objects that *behaves similarly* while moving in space and time; for instance, behavioral features are highly important in the linkage/anonymization of mobility data [39]. In CTM, behavioral constraints are transparently enforced by simply extending the input tessellation with additional features (see Definition 4, the tile directly models the features in the tessellation). More formally, two or more trajectories *share* a tile s if they have at least a point in s . Then, the support of an itemset I (a set of trajectories), includes all and only the transactions (tiles) shared by the trajectories. Note that the behavioral constraints must be computed jointly with the spatio-temporal ones, and not before/after running CTM.

When the number of items is much larger than the number of transactions, searching for frequent itemsets by enumerating all itemsets with an *Apriori-like strategy* [14] can be unfeasible or very inefficient. In this case, it is convenient to adopt a *row-enumeration-like strategy* [19]. CTM extracts co-movement patterns through a breadth-first enumeration of closed itemsets: starting from single transactions, CTM progressively intersects them with further transactions. Closed itemsets associated with a larger set of transactions are characterized by lower cardinalities and larger supports (intuitively, fewer trajectories sharing more tiles). The enumeration process can be represented as an enumeration tree, where each node corresponds to a distinct closed itemset. Naively enumerating the whole tree is inefficient, thus we exploit several pruning mechanisms to limit the enumerated portion of the tree. We conceive CTM as a parallel and big-data approach for co-movement pattern mining. Specifically, CTM: (i) adopts a breadth-first enumeration approach to fully exploit task parallelization and workload balancing; (ii) adopts local pruning criteria to avoid centralized checks that would limit parallelization; (iii) adopts spatio-temporal pruning criteria that have been specifically devised for trajectories-related patterns; (iv) broadcasts the transaction dataset to locally compute the itemset support. For the sake of space, the details of the enumeration process are omitted and are available in

Table 3
KPIs by dataset and pattern type

Type	$ S $	Patterns	Enum.	Time (s)	Shape check	Card. check	Red. check
Flow	88	$9.7 \cdot 10^4$	$2.1 \cdot 10^7$	31	53%	82%	30%
Co-location	88	$2.3 \cdot 10^5$	$2.6 \cdot 10^7$	44	31%	82%	31%
Convoy ¹	528	0	$5.5 \cdot 10^8$	180	1%	99%	90%
Swarm	528	124	$3.0 \cdot 10^8$	127	8.2%	98%	70%

the extended version [12].

Note that approaching this problem as a colossal itemset mining rather than a typical clustering one (i) avoids the computation of similarities that would make computational complexity explode for large datasets; and (ii) exploits monotonicity properties (e.g., as the generation process proceeds, the cardinality of trajectory groups decreases while the length of the path shared by trajectories in the same group increases) to filter out invalid co-movement patterns without generating them all.

4. Evaluation

The approach has been implemented in Spark and is available at <https://github.com/big-unibo/ctm>. All tests run on a cluster of 10 nodes, each equipped with an 8-core i7 CPU@3.60GHz and 16 GB of RAM and interconnected by Gigabit Ethernet. We tested CTM on several use cases and scalability tests, but for the sake of space we only report the results on a real-world dataset.

Milan is a real trajectory dataset that contains trajectories from $6 \cdot 10^6$ MOs (i.e., individuals) from the Milan metropolitan area (around $200km^2$). Trajectories are sparse in time since they represent inhabitants as well as travelers over three months. The dataset has been collected during the urban mobility analysis project “*La città intorno*” (<https://lacittaintorno.fondazionecripi.it/>) that aims to understand the mobility patterns of inhabitants living in suburban neighborhoods and to rank neighborhoods by their attractiveness in order to understand how to allocate economic resources for requalification. The attractiveness of a neighborhood is defined as the percentage of co-movement patterns passing through that neighborhood. To fulfill the analysis, we initially define a tessellation where the spatial feature represents the 88 neighborhoods in Milan and the temporal feature represents a relative dimension that partitions absolute timestamps into six bins, such as night (from 0 to 3) and morning (from 8 to 11); overall $|S| = 88 \cdot 6 = 528$ tiles. Then, together with domain experts, we set relevant values for $mCrd$ (100) and $mSup$ (7). Table 3 reports the outcomes for all co-movement pattern types. Table 4 shows the results of our attractiveness analysis using swarm patterns, highlighting the need for higher requalification in “Lodi - Corvetto”, “Padova”, and “Adriano”; the most attractive neighborhoods are the ones closest to the city center².

Finally, we tested CTM against SPARE [11] (the only big-data approach to the extraction of *generic* co-movement patterns), and PFP Growth [40] (a big data approach to Frequent Itemset Mining) on two synthetic datasets Oldenburg and Hermoupolis. For the sake of space, the

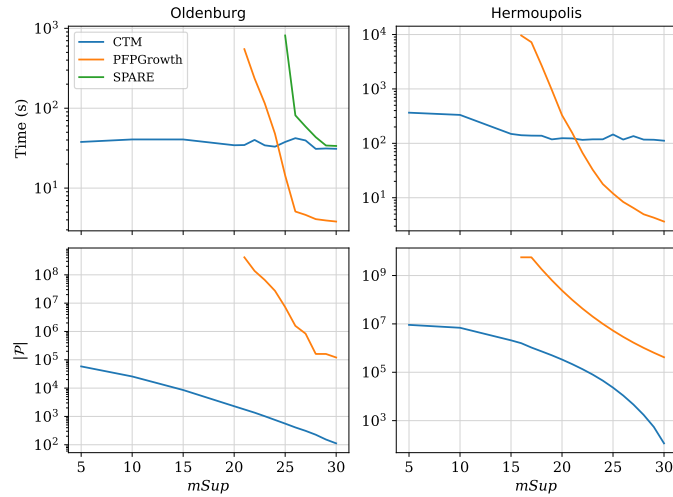
¹Due to the sparsity in time, no convoy pattern is returned in the Milan dataset.

²By filtering tiles on the time bin, it is possible to characterize how attractiveness changes during the day.

Table 4

An excerpt of attractiveness for neighborhoods in Milan

Neighborhood	Attractiveness
Brera	83%
Duomo	83%
Lodi - Corvetto	4%
Padova	3%
Adriano	1%

**Figure 3:** Comparing CTM, SPARE, and PFGrowth in terms of computational time (top) and co-movement patterns (bottom).

details of the scalability testing process are omitted and are available in the extended version [12]. Overall, CTM scalability proves to be better than the others for big groups of trajectories.

5. Conclusion

CTM is a big-data approach to extract spatial and spatio-temporal mobility patterns possibly enriched by additional trajectory features that characterize behavioral mobility patterns. With respect to the existing literature, CTM is general-purpose as it provides a unifying approach to extract different pattern types and is particularly suited for applications characterized by a high number of trajectories to be analyzed on a feature space with limited cardinality (e.g., to capture the daily commuting of a citizen through different neighborhoods, rather than analyzing her detailed path at the single street level). As new research directions, we plan to: (i) introduce a definition of *group cohesion* to further prune mobility patterns based on the shared tiles, (ii) investigate how the extracted mobility patterns can be summarized in a more succinct representation, (iii) investigate how gridding affects the stability of co-movement patterns, and (iv) consider a unifying extraction of mobility patterns from streaming trajectory data in order to apply CTM to online location-based systems.

References

- [1] G. Vitali, M. Francia, M. Golfarelli, M. Canavari, Crop management with the IoT: An interdisciplinary survey, *Agronomy* 11 (2021) 181. doi:10.3390/agronomy11010181.
- [2] D. Kumar, H. Wu, S. Rajasegarar, C. Leckie, S. Krishnaswamy, M. Palaniswami, Fast and scalable big data trajectory clustering for understanding urban mobility, *IEEE Trans. Intelligent Transportation Systems* 19 (2018) 3709–3722. doi:10.1109/tits.2018.2854775.
- [3] M. Francia, E. Gallinucci, M. Golfarelli, N. Santolini, Dart: De-anonymization of personal gazetteers through social trajectories, *J. of Inf. Security and Applications* 55 (2020) 102634. doi:10.1016/j.jisa.2020.102634.
- [4] X. Ding, L. Chen, Y. Gao, C. S. Jensen, H. Bao, Ultraman: a unified platform for big trajectory data management and analytics, in: *Proc. of VLDB*, volume 11, 2018, pp. 787–799. doi:10.14778/3192965.3192970.
- [5] M. M. Kwakye, Conceptual model and design of semantic trajectory data warehouse, *Int. J. Data Warehous. Min.* 16 (2020) 108–131. doi:10.4018/ijdwm.2020070106.
- [6] X. Bao, J. Lu, T. Gu, L. Chang, Z. Xu, L. Wang, Mining non-redundant co-location patterns, *IEEE Trans. Neural Networks and Learning Systems* 33 (2022) 6613–6626. doi:10.1109/tnnls.2021.3082628.
- [7] B. Han, L. Liu, E. Omiecinski, Road-network aware trajectory clustering: Integrating locality, flow, and density, *IEEE Trans. Mob. Comput.* 14 (2015) 416–429.
- [8] Z. Li, B. Ding, J. Han, R. Kays, Swarm: mining relaxed temporal moving object clusters, in: *Proc. of VLDB*, volume 3, 2010, pp. 723–734. doi:10.14778/1920841.1920934.
- [9] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, H. T. Shen, Discovery of convoys in trajectory databases, in: *Proc. of VLDB*, volume 1, VLDB Endowment, 2008, pp. 1068–1080. doi:10.14778/1453856.1453971.
- [10] N. Phan, P. Poncelet, M. Teisseire, All in one: Mining multiple movement patterns, *Int. J. of Inf. Technology and Decision Making* 15 (2016) 1115–1156. doi:10.1142/s0219622016500280.
- [11] Q. Fan, D. Zhang, H. Wu, K. Tan, A general and parallel platform for mining co-movement patterns over large-scale trajectories, in: *Proc. of VLDB*, volume 10, 2016, pp. 313–324. doi:10.14778/3025111.3025114.
- [12] M. Francia, E. Gallinucci, M. Golfarelli, Colossal trajectory mining: A unifying approach to mine behavioral mobility patterns, *Expert Syst. Appl.* 238 (2024) 122055. doi:10.1016/j.eswa.2023.122055.
- [13] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, K. Aberer, Semantic trajectories: Mobility data computation and annotation, *ACM Transactions on Intelligent Systems and Technology* 4 (2013) 1–38. doi:10.1145/2483669.2483682.
- [14] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: *Proc. of VLDB*, volume 25, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, p. 487–499. doi:10.1145/235968.233311.
- [15] M. Qiu, D. Pi, Mining frequent trajectory patterns in road network based on similar trajectory, in: *Proc. of IDEAL*, Springer, Yangzhou, China, 2016, pp. 46–57. doi:10.1007/978-3-319-46257-8_6.
- [16] Z. Fu, Z. Tian, Y. Xu, K. Zhou, Mining frequent route patterns based on personal trajectory

- abstraction, *IEEE Access* 5 (2017) 11352–11363. doi:10.1109/access.2017.2712703.
- [17] L. Zheng, D. Xia, X. Zhao, L. Tan, H. Li, L. Chen, W. Liu, Spatial–temporal travel pattern mining using massive taxi trajectory data, *Physica A: Statistical Mechanics and its Applications* 501 (2018) 24–41.
- [18] D. Apiletti, E. Baralis, T. Cerquitelli, P. Garza, F. Pulvirenti, P. Michiardi, A parallel mapreduce algorithm to efficiently support itemset mining on high dimensional data, *Big Data Research* 10 (2017) 53–69. doi:10.1016/j.bdr.2017.10.004.
- [19] F. Pan, G. Cong, A. K. H. Tung, J. Yang, M. J. Zaki, Carpenter: finding closed patterns in long biological datasets, in: *Proc. of KDD, ACM, Washington, DC, USA, 2003*, pp. 637–642. doi:10.1145/956750.956832.
- [20] F. A. M. Zaki, N. F. Zulkurnain, RARE: mining colossal closed itemset in high dimensional data, *Knowl.-Based Systems* 161 (2018) 1–11. doi:10.1016/j.knosys.2018.07.025.
- [21] M. K. Vanahalli, N. Patil, An efficient parallel row enumerated algorithm for mining frequent colossal closed itemsets from high dimensional datasets, *Inf. Sci.* 496 (2019) 343–362. doi:10.1016/j.ins.2018.08.009.
- [22] J. Gudmundsson, M. J. van Kreveld, Computing longest duration flocks in trajectory data, in: *14th ACM Int. Symposium on Geographic Inf. Systems, ACM, Arlington, Virginia, USA, 2006*, pp. 35–42. doi:10.1145/1183471.1183479.
- [23] H. H. Aung, K.-L. Tan, Discovery of evolving convoys, in: M. Gertz, B. Ludäscher (Eds.), *Scientific and Statistical Database Management, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010*, pp. 196–213. doi:10.1007/978-3-642-13818-8_16.
- [24] T. L. C. da Silva, F. Lettich, J. A. F. de Macêdo, K. Zeitouni, M. A. Casanova, Online clustering of trajectories in road networks, in: *Proc. of MDM, IEEE, Versailles, France, 2020*, pp. 99–108. doi:10.1109/MDM48529.2020.00031.
- [25] P. Yang, L. Wang, X. Wang, L. Zhou, SCPM-CR: A novel method for spatial co-location pattern mining with coupling relation consideration, in: *Proc. of ICDE, IEEE, 2022*, pp. 1503–1504. doi:10.1109/icde53745.2022.00128.
- [26] A. Tritsarolis, G.-S. Theodoropoulos, Y. Theodoridis, Online discovery of co-movement patterns in mobility data, *International Journal of Geographical Information Science* 35 (2021) 819–845.
- [27] J. C. Fonseca-Galindo, G. de Castro Surita, J. M. Neto, C. L. de Castro, A. P. Lemos, A multi-agent system for solving the dynamic capacitated vehicle routing problem with stochastic customers using trajectory data mining, *Expert Systems with Applications* 195 (2022) 116602. doi:10.1016/j.eswa.2022.116602.
- [28] J. Lee, J. Han, K. Whang, Trajectory clustering: a partition-and-group framework, in: *Proc. of SIGMOD, ACM, Beijing, China, 2007*, pp. 593–604. doi:10.1145/1247480.1247546.
- [29] C. Hu, X. Kang, N. Luo, Q. Zhao, Parallel clustering of big data of spatio-temporal trajectory, in: *11th Int. Conf. on Natural Computation, ICNC 2015, Zhangjiajie, China, August 15-17, 2015, 2015*, pp. 769–774. doi:10.1109/icnc.2015.7378088.
- [30] P. Tampakis, N. Pelekis, C. Doulkeridis, Y. Theodoridis, Scalable distributed subtrajectory clustering, in: *2019 IEEE Int. Conf. on Big Data, IEEE, Los Angeles, CA, USA, 2019*, pp. 950–959. doi:10.1109/bigdata47090.2019.9005563.
- [31] F. Orakzai, T. Calders, T. B. Pedersen, k/2-hop: Fast mining of convoy patterns with effective pruning, in: *Proc. of VLDB, volume 12, 2019*, pp. 948–960. doi:10.14778/

3329772.3329773.

- [32] Y. Liu, H. Dai, B. Li, J. Li, G. Yang, J. Wang, ECMA: An efficient convoy mining algorithm for moving objects, in: Proc. of CIKM, ACM, Queensland, Australia, 2021, pp. 1089–1098. doi:10.1145/3459637.3482255.
- [33] F. Orakzai, T. B. Pedersen, T. Calders, Distributed mining of convoys in large scale datasets, *GeoInformatica* 25 (2021) 353–396. doi:10.1007/s10707-020-00431-w.
- [34] A. Tritsarolis, E. Chondrodima, P. Tampakis, A. Pikrakis, Y. Theodoridis, Predicting co-movement patterns in mobility data, *GeoInformatica* (2022) 1–23. doi:10.1007/s10707-022-00478-x.
- [35] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, X. Qin, Fast large-scale trajectory clustering, in: Proc. of VLDB, volume 13, VLDB Endowment, 2019, pp. 29–42. doi:10.14778/3357377.3357380.
- [36] J. Bouttier, P. Di Francesco, E. Guitter, Geodesic distance in planar graphs, *Nuclear physics B* 663 (2003) 535–567. doi:10.1016/s0550-3213(03)00355-9.
- [37] J. Pei, J. Han, R. Mao, CLOSET: an efficient algorithm for mining frequent closed itemsets, in: Proc. of SIGMOD, ACM, Dallas, Texas, USA, 2000, pp. 21–30. doi:10.1145/956750.956779.
- [38] M. Francia, M. Golfarelli, S. Rizzi, Summarization and visualization of multi-level and multi-dimensional itemsets, *Inf. Sciences* 520 (2020) 63–85. doi:10.1016/j.ins.2020.02.006.
- [39] F. Jin, W. Hua, M. Francia, P. Chao, M. E. Orłowska, X. Zhou, A survey and experimental study on privacy-preserving trajectory data publishing, *IEEE Trans. Knowl. Data Eng.* 35 (2023) 5577–5596. doi:10.1109/TKDE.2022.3174204.
- [40] H. Li, Y. Wang, D. Zhang, M. Zhang, E. Y. Chang, Pfp: parallel fp-growth for query recommendation, in: Proc. of RecSys, ACM, New York, NY, USA, 2008, pp. 107–114. doi:10.1145/1454008.1454027.