PFLOCK Report

Andres Calderon

University of California, Riverside

August 20, 2019

Remarks of Chen et al. (2019)

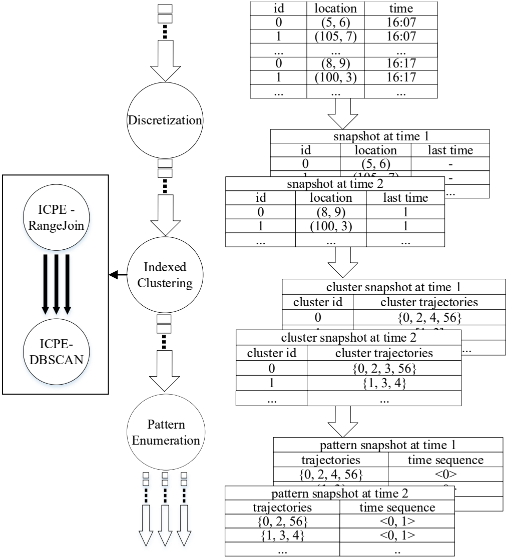
Real-time Distributed Co-Movement Pattern Detection on Streaming Trajectories (Chen et al., 2019)

- Explore a general co-movement pattern definition (following Fan et al, 2016) based on 5 constrains:
    1. Closeness: control spatial proximity.
    2. Significance (M): control minimum number of objects.
    3. Duration (K): control how long objects move together.
    4. Consecutiveness (L): Minimum length of consecutive *segments*.
    5. Connection (G): Maximum length of gaps between *segments*.

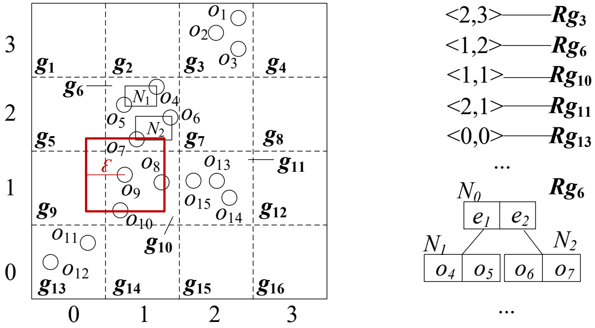
Processing flow

1. First, It focus on transforming the input on discretized snapshots (time instants).
    - It uses window operations and time synchronization to organize locations happening at the same time.
2. Then, It focus on finding spatial cluster:
    - It uses closeness (ε) and significance (M) to run DBSCAN and find cluster at each snapshot.
3. Finally, It focus on enumerating patterns in the temporal domain:
    - It uses duration(K), consecutiveness (L) and connection (G) to mine set of clusters that fill those constrains.

Indexed Clustering and Pattern Enumeration

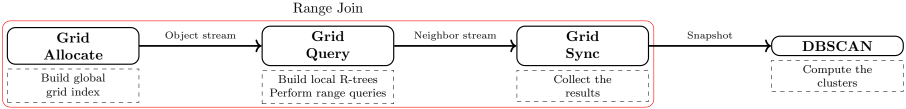


Indexed Clustering



(a) Grid index (b) R-trees for all grid cells

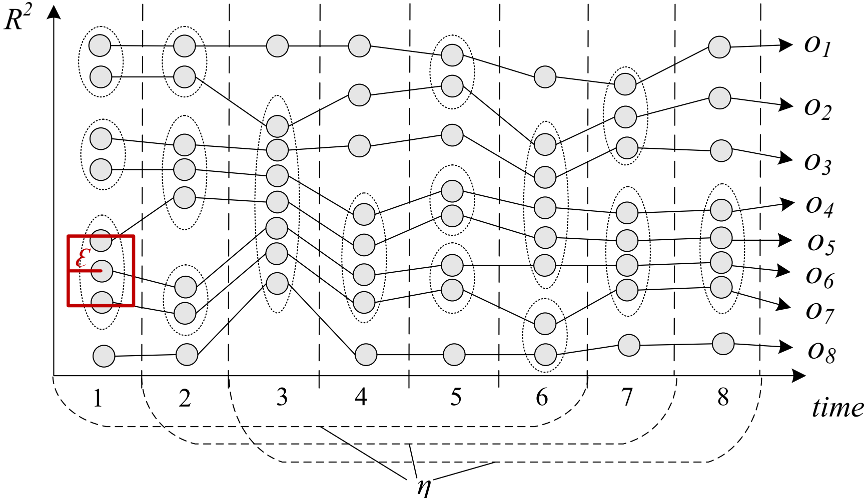
Figure 4: Example of a GR-index



Main diferences with our approach

- DBSCAN finds variable shape clusters. Flocks demands disks with fixed diameter which introduce a large number of redundant/duplicate candidates.
- DBSCAN queries only input locations to find core and distance-reachable points. Finding disk locations for flocks is more complex (twice the number of pairs).
- DBSCAN is run at Snapshot level, they claim Range Join prune enough points and partitions are no needed. “In ICPE framework, we achieve the parallelism by clustering snapshots separately.” It depends in dataset size and parameters.

Pattern Enumeration



Partitions on temporal domain

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
subtask 1 for o_1	$\{o_2\}$	$\{o_2\}$	$\emptyset$	$\emptyset$	$\{o_2\}$	$\emptyset$	$\{o_2,o_3\}$	$\emptyset$
subtask 2 for o_2	$\emptyset$	$\emptyset$	$\{o_3,o_4,o_5,o_6,o_7,o_8\}$	$\emptyset$	$\emptyset$	$\{o_3,o_4,o_5,o_6\}$	$\{o_3\}$	$\emptyset$
subtask 3 for o_3	$\{o_4\}$	$\{o_4,o_5\}$	$\{o_4,o_5,o_6,o_7,o_8\}$	$\emptyset$	$\emptyset$	$\{o_4,o_5,o_6\}$	$\emptyset$	$\emptyset$
subtask 4 for o_4	$\emptyset$	$\{o_5\}$	$\{o_5,o_6,o_7,o_8\}$	$\{o_5,o_6,o_7\}$	$\{o_5\}$	$\{o_5,o_6\}$	$\{o_5,o_6,o_7\}$	$\{o_5,o_6,o_7\}$
subtask 5 for o_5	$\{o_6,o_7\}$	$\emptyset$	$\{o_6,o_7,o_8\}$	$\{o_6,o_7\}$	$\emptyset$	$\{o_6\}$	$\{o_6,o_7\}$	$\{o_6,o_7\}$
subtask 6 for o_6	$\{o_7\}$	$\{o_7\}$	$\{o_7,o_8\}$	$\{o_7\}$	$\{o_7\}$	$\emptyset$	$\{o_7\}$	$\{o_7\}$
subtask 7 for o_7	$\emptyset$	$\emptyset$	$\{o_8\}$	$\emptyset$	$\emptyset$	$\{o_8\}$	$\emptyset$	$\emptyset$
subtask 8 for o_8	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

η

Figure 7: Example of Id-based Partitioning for Fig. 2

Fan et al. states and proves $\eta = (\lceil \frac{K}{L} - 1 \rceil) \times (G - 1) + K + L - 1$

Pattern verification

1. Baseline:
   - For each $P_t(o)$ at time t, it enumerate all possible combinations $o \cup P_t(o)$.
   - Then, it find the valid time sequence for each combination in the subsequent η snapshots.
   - i.e. for $P_2(o_3) = \{o_4, o_5\}$, it looks if $\{o_3, o_4\}, \{o_3, o_5\}, \{o_3, o_4, o_5\}$ appear in the following snapshots.

Bit compression improvements

2 Fixed Length Bit Compression Method:

time	3	4	5	6	7	8	
o_5	1	1	1	1	1	1	√
o_6	1	1	0	1	1	1	√
o_7	1	1	0	0	1	1	√
o_8	1	0	0	0	0	0	×
$\{o_5, o_6\}$	1	1	0	1	1	1	√
$\{o_5, o_7\}$	1	1	0	0	1	1	√
$\{o_6, o_7\}$	1	1	0	0	1	1	√
$\{o_5, o_6, o_7\}$	1	1	0	0	1	1	√

Figure 8: Bit Compression on $P_3(o_4)$

Bit compression improvements

3 Variable Length Bit Compression Method:

time 2

time	2	3	4	5	6	7
o_5	1	1	1	1	1	1

$\eta = 6$

time 3

time	3	4	5	6	7	8
o_5	1	1	1	1	1	1
o_6	1	1	0	1	1	1
o_7	1	1	0	0	1	1
o_8	1	0	0	0	0	0

...

(a) fixed length bit compression for each time

time	2	3	4	5	6	7	8		*st*	*et*
o_5	1	1	1	1	1	1	1	√	2	8

time	3	4	5	6	7	8		*st*	*et*
o_6	1	1	0	1	1	1	√	3	8

time	3	4	5	6	7	8		*st*	*et*
o_7	1	1	0	0	1	1	√	3	8

time	3	4	
o_8	1	0	×

(b) variable length bit compression for all times

Figure 9: Bit Compression for Subtask of o_4 in Fig. 2

Experimental Evaluation

Table 2: Datasets Used in our Experiments

Attributes	GeoLife	Taxi	Brinkhoff
# trajectories	18,670	20,151	10,000
# locations	24,876,978	189,419,934	23,906,131
# snapshots	92,645	502,559	97,241
Storage Size	1.5G	14G	1.7G

Locations per snapshot: GeoLife = 269, Taxi = 377, Brinkhoff = 246

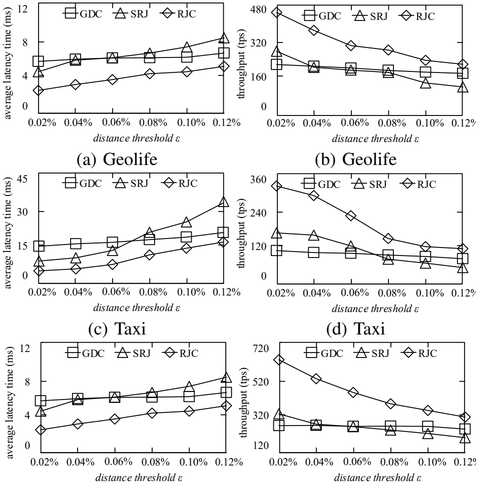
Experimental Evaluation

Table 3: Parameter Ranges and Default Values

Parameter	**Range**
grid cell width l_g	0.2%, 0.4%, **0.8%**, 1.6%, 3.2%, 6.4%
distance threshold ϵ	0.02%, **0.04%**, 0.06%, 0.08%, 0.10%, 0.12%
min objects M	5, 10, **15**, 20, 25
min duration K	120, 150, **180**, 210, 240
min local duration L	10, 20, **30**, 40, 50
max gap G	10, 20, **30**, 40, 50
ratio of objects O_r	10%, 20%, 40%, 60%, 80%, **100%**
machine number N	1, 2, 4, 6, 8, **10**

l_g and ϵ are based on "the maximal distance of the whole dataset" (?)

Experimental Evaluation



(a) Geolife (b) Geolife

(c) Taxi (d) Taxi

(e) Brinkhoff (f) Brinkhoff

GDC: Grid-based DBSCAN Clustering, SRJ: state-of-the-art distributed range join.

Experimental Evaluation

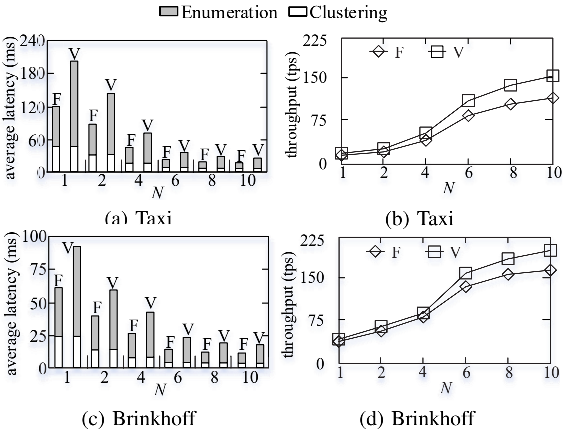


Figure 14: Pattern Detection Performance vs. N

24 cores and 64Gb RAM per node