Table 1: Symbols and Description

Notation	Description
$r = (l, t)$	the GPS record with location $l = (x, y)$ and time t
$o = \langle r_1, r_2, ... \rangle$	the streaming trajectory
T	the time sequence
$T[i]$	the i-th element in T
$\|T\|$	the number of elements in T
$max(T)$	the last time in T
T_l	the last time segment in T
ϵ	the distance threshold
$minPts$	the minimal number of points to form a dense region used in density-based clustering DBSCAN
M	the significance constraint
K	the duration constraint
L	the consecutive constraint
G	the connection constraint
$CP(M, K, L, G)$	the co-movement pattern w.r.t. M, K, L, and G
$RJ(O, \epsilon)$	the range join query for a set O with ϵ
$RQ(o, \epsilon)$	the range query for a query object o with ϵ
S_t	the snapshot at time t
l_g	the grid cell width
key	the key for a grid cell g
$B[o_i]$ or $B[O]$	the bit string for the trajectory o_i or the set O

Storm[3], Samza[4], and Flink[5] support this type of processing. Next, with **mini-batch semantics**, in-coming records that have arrived within the last few seconds are batched and then processed in a single mini-batch. Spark Streaming[6] supports this type of processing. We choose Flink, because it is a typical stream processing platform, and because it offers both efficiency and reliability. Nonetheless, our methods and techniques (e.g., GR-index, bit-compression, and candidate-based enumeration) are generic and hence can be easily adapted to other distributed stream processing platforms.

3. BACKGROUND

In this section, we introduce in turn the notion of co-movement pattern, DBSCAN, and range join. Table 1 summarizes the symbols used frequently throughout the paper.

3.1 Co-Movement Pattern

A GPS record is a pair $r = (l, t)$, where l is a location and t is a time value. A sequence $o = \langle r_1, r_2, ..., r_n \rangle$ of GPS records that capture a particular trip make up a trajectory.

Following an existing trajectory pattern detection approach [10], we first discretize the timestamps in trajectories. The discretization maps the real clock times to indices of the time intervals during which they occurred. For instance, assume that we partition the time line into intervals of duration 5s and that the start time is 13:00:20 UTC. Then the time series ⟨13:00:21 UTC, 13:00:24 UTC, 13:00:28 UTC, 13:00:32 UTC, 13:00:42 UTC⟩ is discretized into ⟨0, 0, 1, 2, 4⟩. This example discretization causes (i) a sequence where 0 appears twice, and (ii) that has a misleading gap. To avoid such problems, it is important to choose the duration used for discretization carefully. The duration cannot be too large or too small. In our experiments, the interval duration is set to 1s or 5s depending on sampling rates of the datasets used. Next, we give the definition of a discretized time sequence.

DEFINITION 1. **(Time Sequence)**. *Let $\mathbb{T} = \{1, 2, ..., \mathbb{N}\}$ be a discretized temporal dimension. A time sequence T is defined as a sequence of elements from $\mathbb{T}$, i.e., $T = \langle t_1, t_2, ..., t_m \rangle$, where (i) t_i $(1 \leq i \leq m) \in \mathbb{T}$, and (ii) $t_i > t_j$ iff $i > j$ $(t_i \in T, t_j \in T)$.*

[3]http://storm.apache.org/
[4]http://samza.apache.org/
[5]http://flink.apache.org/
[6]http://spark.incubator.apache.org/

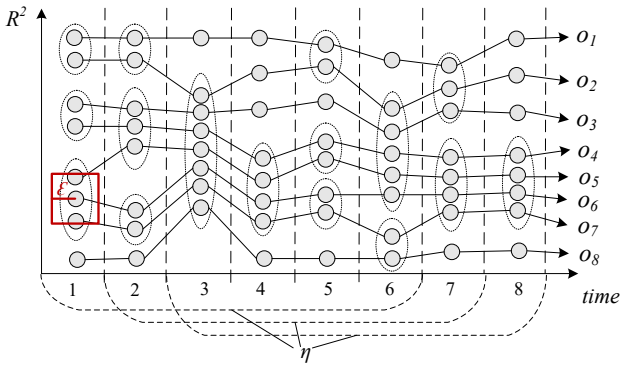

Figure 2: Example of Co-movement Patterns

A time sequence T is consecutive if $\forall 1 \leq i < |T|$ $(T[i+1] = T[i] + 1)$. We call a consecutive sequence T a segment. For example, $T_1 = \langle 1, 2, 3, 4 \rangle$ and $T_2 = \langle 1, 2, 4, 5 \rangle$ are two sequences, where T_1 is a segment while T_2 is not, because time 3 is missing. Based on the definition of sequence, we define the notions of L-consecutive and G-connected. Here, L-consecutive is used to control the lengths of segments, while G-connected is employed to control the lengths of gaps between consecutive segments.

DEFINITION 2. **(L-consecutive)**. *Let T_i $(i \leq i \leq m)$ be segments with $|T_i| \geq L$. Then sequence $T = \cup T_i$ is L-consecutive.*

DEFINITION 3. **(G-connected)**. *A sequence T is G-connected if the gap between any neighboring times is at most G, i.e., $\forall 1 \leq i \leq |T| - 1$ $(T[i+1] - T[i] \leq G)$, where $T[i]$ denotes the i-th element in T.*

For instance, $T = \langle 1, 2, 4, 5, 6 \rangle$ is 2-consecutive and 2-connected. Specifically, there are two segments $T_1 = \langle 1, 2 \rangle$ and $T_2 = \langle 4, 5, 6 \rangle$ in T, and the length of each segment is no smaller than 2. Thus, T is 2-consecutive. Further, according to Definition 3, $\forall 1 \leq i \leq 4$ $(T[i+1] - T[i] \leq 2)$. Hence, T is 2-connected.

Next, we formalize the definition of co-movement pattern. Co-movement pattern mining detects a group of objects that move together while satisfying five constraints: (1) "*closeness*" that defines the concept of "moving together", (2) "*significance*" that controls the number of the objects that move together, (3) "*duration*" that controls the length of time when objects move together, while (4) "*L-consecutive*" and (5) "*G-connected*" that relax the consecutiveness of "duration". Specifically, the entire time period that objects move together is not necessarily strictly consecutive, as gaps are allowed between consecutive segments. Hence, "L-consecutive" and "G-connected" control the length of each consecutive time segment and the gap between two consecutive time segments, respectively.

DEFINITION 4. **(Co-movement Pattern)**. *Given a set ST of discretized trajectories, a subset O of ST is a co-movement pattern $CP(M, K, L, G)$ if a time sequence T exists such that the following five constraints are satisfied: (i) **closeness:** the locations of trajectories in O belong to the same cluster in every time of T; (ii) **significance:** $|O| \geq M$; (iii) **duration:** $|T| \geq K$; (iv) **consecutiveness:** T is L-consecutive; and (v) **connection:** T is G-connected.*

To provide a concrete definition of the first constraint (i.e., closeness), we choose to rely on density-based clustering as implemented by the popular clustering method DBSCAN (as also done for *convoy* [17]), which is detailed in the next subsection. Considering