MARIO: A Spatio-temporal Data Mining Framework on Google Cloud to Explore Mobility Dynamics from Taxi Trajectories

Shreya Ghosh, Soumya K Ghosh, Rajkumar Buyya

Abstract

With the major advances in location acquisition techniques, deployment of GPS enabled devices and increasing number of mobile users, substantial amount of location traces are generated from different geographical regions. It provides unprecedented opportunities to analyse and derive valuable insights of urban dynamics, specifically, time-dependent mobility patterns and region-specific travel demands. This work proposes an end-to-end mobility association rule mining framework called MARIO, conducive to extract urban mobility dynamics through analysing large taxi trip traces of a city. Our approaches consist of (i) generating mobility-dynamics network by spatio-temporal analysis of taxi-trips, (ii) finding travel demand variations in different functional regions of the urban area, (iii) extracting mobility association rules and (iv) predicting travel demands and traffic dynamics using extracted associative rules. The proposed MARIO framework is implemented in Google Cloud Platform and an extensive set of experiments using real GPS trace dataset of NYC Green and Yellow Taxi trace, Roma Taxi Dataset and San Francisco Taxi Dataset have been carried out to demonstrate the effectiveness of the framework. The performance evaluation result of our proposed method compared to baseline methods is able to predict travel demand with an average MAPE value and execution time reduced by 50%.

Keywords: Trajectory, Spatio-temporal Data Analysis, Association rule mining, Deep Learning, Travel Demand Prediction

1. Introduction

The ever-increasing popularity of wireless communications, sensor technologies and GPS embedded devices have motivated extensive research on analysing the enormous amount of GPS traces and various location-aware applications. The location traces of moving agents (GPS equipped private vehicles, mobile sensors of individuals, public transportation data etc.) are accumulated easily and effectively leading to generation of huge amount of location traces. This increasing availability of human location traces (also may be defined as ‘human trail’) from various sources has opened up interesting research directions like human behavior/activity learning, studying city dynamics, improved route planning, resource allocation and traffic analysis.

This paper aims to discover overall mobility patterns of a city analyzing taxi-traces to build a mobility-knowledge base (associative patterns) which captures the inherent urban-dynamics and utilize the knowledge to another region for better urban planning. Notably, the public transportation vehicles provide predictable patterns as they follow fixed routes and regular time schedules. Further, private vehicles offer fairly expected routes as the traces depict the pattern of the owner (say, commuting from home to work). On the other side, GPS enabled taxis serve diverse transportation needs of a large population and contributes improving urban transportation planning and thus acts as an integral part of intelligent transportation system. However, there exists an well-known dilemma that trajectory data are rich in terms of volume but activity information poor. This gap can be mitigated if contextual information such as functional regions or landmarks of the taken route, activities at stay points are considered instead of analysing raw GPS log (timestamped latitude, longitude data). As the pattern is likely to be dependent on start and destination functional regions of the cities, more specifically on travel purposes, this paper incorporates the ‘POI’ information (ex., industrial, residential etc.) of the trips along with the spatio-temporal attributes. In summary, our framework generates mobility knowledge base by extracting prevalent mobility association rules from city taxi-trips and predicts possible travel demand spikes.

Motivating example: Figure 1 shows a motivating example for our MARIO framework. Each POI (point-of-interest) or functional region depicts travel demand pattern in different temporal scales. Various mobility events effect other neighboring regions in a temporal sequence and subsequently it helps to predict travel demand and travel time efficiently. Our framework generates mobility knowledge base by extracting prevalent mobility association rules from city taxi-trips and predicts possible travel demand spikes. For a given city, the proposed MARIO framework may explore the rules like (R1): When the region $A_1$ of a city experiences traffic congestion, then with 80% probability the traffic density of region $A_2$ will be higher after 6 time-period and consequently high travel-demands in the spatial-neighborhood region of $A_2$. $R_1$ reflects both the spatial

Corresponding author

*Email addresses: shreya.cst@gmail.com (Shreya Ghosh), skg@slitkgp.sc.in (Soumya K Ghosh), rbuyya@unimelb.edu.au (Rajkumar Buyya)
and temporal neighbourhood traffic states and the associative flow-patterns among different regions of the city. The mobility association rule mining problem becomes quite challenging for several spatio-temporal resolutions, i.e., an infrequent itemset for the entire region or large time-interval may be a frequent item within a part of the region or a relatively small time-quantum. Furthermore, mobility state of one region effects the neighboring regions in a temporal sequence. Therefore, there is a strong need to learn the traffic-flow inter-dependencies of the regions in different temporal scales and partition the city-network into smaller regions.

Contributions: To address the above mentioned challenges and issues, we propose mobility rule miner MARIO framework which involves

1. generating mobility dynamics network by segmenting the large city road network and sub-graph decomposition to discover traffic-flow in different regions of the city
2. proposing a variant of Apriori algorithm utilizing frequency tree and deep learning architecture to extract prevalent mobility rules by analysing spatio-temporal neighbourhood effects
3. extracting mobility rules, namely, region-specific mobility association rules, i.e. mobility dynamics (crowd behaviour, traffic flow and travel-demand) of different functional regions of a city, taxi-trip specific or individualized mobility patterns to capture the overall urban dynamics
4. predicting travel demands effectively and timely manner in different regions of the city

In summary, we have implemented an end-to-end trajectory analysis service over Google Cloud Platform (GCP) by creating interfaces with varied cloud services and vCPUs to explore the mobility dynamics of a city. The remainder of this article is organized as follows. Section 2 presents a review of recent studies in this direction. A few preliminary concepts of the work and different modules of the framework are presented in section 3 and 4 respectively. Section 5 depicts the experimental set-up and detailed discussion and evaluations of the proposed method. Finally, section 6 summarizes the study and discusses some avenues of future research direction.

2. Related Work

Understanding and discovering mobility patterns facilitate better urban planning and traffic resource management. In this section, we brief the existing works on spatio-temporal association rule mining and applications of mobility trace analysis.

In short, spatial data mining aims to extract patterns which are previously unknown but potentially useful. Spatial association rule mining [3] discovers the frequent co-occurrences of spatial predicates, such as adjacent_to, nearby and objects, namely, highway and house. There are two strategies followed by the existing works to extract spatio-temporal patterns: transaction-based strategy and transaction-free strategy. In transaction-based strategy, spatio-temporal data is transformed into a format of transactions such that traditional association rule mining technique can be deployed. These transactions are created by partitioning the study regions into different sub-regions, where the sub-regions are transactions and features in these sub-regions correspond to items [4, 5]. In the transaction-free strategy, spatial points are considered as Boolean spatial features [5] i.e., the presence or absence of spatial phenomena. Another strand of research in this domain focuses on spatio-temporal events, such as state-based events, location-based events and change-based events [7]. Location-based events [8] are represented by spatial locations, where several geographical events, such as vehicle collision, typhoon, flood etc. have occurred. Spatio-temporal association pattern reveals the connection between two such events. However, the temporal dynamics (temporal relation between two such correlated events) of the events can not be extracted from the existing approaches. We have not found any consensus regarding the definition of spatio-temporal events and further, no such existing works delineate events related to mobility behaviours of people or vehicles. There are very few works which explore associative patterns in human movement traces. Yang et al. [9] present a framework to extract association rules from individual mobility traces. For instance, they extract rules like “In 70% of the days, person X visits POI Y; or visits shopping mall once in a week”. In our previous works [10] and [11], we present the association rule mining technique from individuals’ trajectory traces, however, these approaches are not suitable for extracting aggregate movement patterns city-wide. In our present work, we study the mobility patterns thoroughly and detail the mobility event definition to deploy the mobility rule miner (MARIO) framework. All of the existing works mainly focus on relationships among diversity of spatial data-types [14, 18] and finding spatial autocorrelation [21]. A variant of Apriori algorithm named T-Apriori is proposed to discover association rules in interval based data in [22]. There is existing work in protecting the privacy of

Figure 1: Example scenario: Different mobility events, like road-closure, high travel demand effects other regions.
users’ data [23]. The work considers location semantic diversity and randomness of query behavior to protect the trajectory privacy. Cao et al. [24] present a classification model for location prediction based on user check-in patterns. Rosen Ivanov [25] presents a novel algorithm for real-time GPS track simplification, which in turn helps outdoor navigation of visually impaired people. There are a broad range of applications on GPS data - a systematic survey [26] on human mobility discusses the mobility features and prediction techniques. There are also works on effective forecasting of traffic flows [27], [28]. The authors [27] propose a deep learning approach to forecast crowd flows in different regions of a city using several factors such as weather and intra-region traffic. A multi-task deep learning framework is proposed in [28] where both the node flow and edge flow are predicted. The authors consider the fact that flow at a node and transitions between nodes (edge flow) are dependent to each other. However, this work does not explore the spatial and temporal neighborhood effects of such crowd flows. Zhang et al. [29] formulate a new problem of mining spatial co-evolving patterns from several geo-sensory data. The authors explore different sensor-records such as air-quality, bike/vehicle data; and finds out co-evolving patterns by assembling the individual sensors’ patterns into a single pattern. Our problem set-up is quite different from this work, where instead of finding co-evolving patterns from several sensory information, the patterns are extracted and analysed only from GPS trajectories. Discovering the spatio-temporal dynamics is one of the most challenging issue in our mobility analysis work. It is useful to predict the traffic condition apriori to build an effective traffic recommendation system. Most of the studies [12, 13] model the geographical phenomenon (air-pollution or typhoons) as simple point-events. However, in our framework, dynamic characteristics of the moving agents and their interrelations need to be extracted efficiently.

Table 1 summarizes the features of MARIO and other most relevant existing works. To the best of our knowledge, there are few works on spatio-temporal association rule mining but none have reported for extracting mobility association rules from taxi-trips. Further, the indexing scheme of MARIO, mobility-dynamics network based on traffic flow dynamics and analysing spatio-temporal neighbourhood by deep LSTM architecture are novel propositions. In summary, MARIO provides an end-to-end framework to model mobility dynamics, extract mobility association rules and predict travel demand efficiently.

### 3. MARIO Framework: Workflow

Figure 2 depicts the workflow of the MARIO framework which discovers varied spatio-temporal association rules of taxi-trips. The problem definition of this paper can be summarized as:

- **Given a spatial region r and the taxi-trips (Ty) of r, extract mobility association rules (MARs).** The mobility association rules (MARs) are extracted from the mobility database by deploying proposed mobility-rule miner framework.

- **Given r and Ty of r, generate the mobility dynamics network to represent the overall urban dynamics.** This network captures the overall mobility dynamics of the geographical region by analysing varied sources of movement data.

- **Predict the travel demands in different regions of the city effectively.** Finally, based on the historical movement log and spatio-temporal analysis, the travel demand in different regions of a city is computed.

The framework has three main modules: (i) **trajectory trace pre-processing module**: takes the mobility database consisting taxi-trips of a region (r) and road-network of r as input and generates the item-set or transactions from the mobility data by following application-specific mobility rule templates, (ii) **Spatio-temporal analysis module**: analyses taxi-trip database to discover correlation among different entities, such as time, place and moving objects (vehicles, people etc.) and (iii) **Mobility...**
association rule mining module: shows how mobility association rules can be extracted to understand the urban dynamics for better traffic resource management.

We consider a set of moving agents $M$, say, taxis, set of POIs in a 2D space or region $r$. Each moving agent $m \in M$ is associated with sequences of location information $(x, y)$. In this work, we study taxi-trajectory dataset to explore varied movement patterns in a city region. As shown in Table 2, the acquired taxi-trips cover divergent types of trip-information, such as Type I stores the pick-up and drop-off locations along with passenger count, trip distances etc whereas GPS sequences of complete trip are stored in Type II trace database. These taxi-trip database facilitate different interesting patterns, namely, correlations of pick-up and drop-off locations in different time-instances, frequent-path followed by the trajectories, and even, taxi-specific (or taxi-driver specific) movement behaviours. To start with, few preliminaries are defined as follows.

1. **GPS Log or Trace or Trajectory:** GPS log is a collection of time stamped GPS points $P = \{p_1, p_2, \ldots, p_n\}$. Each GPS point $p_i \in P$ contains latitude ($\text{Lat}_i$), longitude ($\text{Lng}_i$) and timestamp $(t_i)$ of the moving agent $[1]$. In this paper GPS log, trace or trajectory are used interchangeably.

   **Geotagged Point:** Each GPS point is associated with most appropriate land use information. Here, each GPS point $p_i$ contains $(p_i, \text{place})$ along with latitude, longitude and timestamp information $[30]$. For example, $p_i$ is associated with *residential building* or the nearest landmark of $p_j$ is *supermarket*.

2. **Road Network:** Road network of a region is represented by a directed graph $R = (V_R, E_R)$ where $V_R$ represents all intersecting points of the road-segments, i.e., either starting or ending points of each such road-fragment and $E_R$ is the set of edges or roads in the map. It may be noted that in real-life scenario, direction of the roads $E_R = \{v_{R_i} \rightarrow v_{R_j} \mid (v_{R_i}, v_{R_j}) \in V_R \}$ may depend on particular time-slot of a day and the variation of direction based on temporal slot has been considered in the road network graph construction.

3. **Trajectory Slider:** It contains the route followed by the taxi within a specific distance ($D$) and time threshold ($T$). Trajectory Slider contains a set of GPS points or traces within a specific distance and time-window boundary. TrajSlider $= \{(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)\}$ iff $\text{max}(\text{dist}((x, y), (j, k))) \leq D \land (x,y)$ and $(j,k)$ pair $\in P$ and $\text{diff}(t_j, t_k) \leq T$

4. **Taxi-Trip Trajectory:** We define two types of taxi trajectories, $T_{Y_1} = \{< p_{t_1}, t_{p_1}, < p_{t_2}, t_{p_2}, \ldots, < p_{t_n}, t_{p_n} >, (C), D, F, T \}$ and $T_{Y_2} = \{< p_{t_1}, t_{p_1}, < p_{t_2}, t_{p_2}, \ldots, < p_{t_n}, t_{p_n} >, \ldots, < p_{t_{n+1}}, t_{p_{n+1}} >, \ldots, < p_{t_D}, t_{p_D} >, \ldots \}$, where $T_1$ consists of start and end point of the taxi-trip. $T_2$ consists of all GPS points of the trajectory segments between the start and end point (Table 2).

Mobility traces offer huge opportunities to discover multi-states information, such as past, present and future states (travel demand, traffic congestion etc.) of a city road-network. For example, knowing the recurrent effect of traffic congestion of neighborhood region allows one to avoid the probable congested re-
Table 2: Taxi-trip Description

<table>
<thead>
<tr>
<th>Taxi-trace type</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type I</strong></td>
<td>GPS: &lt; pickup, drop off location, timestamp &gt;</td>
</tr>
<tr>
<td>* Pattern: Travel demand</td>
<td>Others: Passenger count, Trip distance, Fare amount</td>
</tr>
<tr>
<td><strong>Type II</strong></td>
<td>GPS: &lt; lat, lon, time, &gt; …</td>
</tr>
<tr>
<td>* Pattern: Frequent path</td>
<td>Others: Running status (Vacant, Occupy)</td>
</tr>
</tbody>
</table>

regions apriori. To be more specific, it is important to capture the evolution of events in neighboring regions over time and space.

Association rule mining techniques are utilized to discover the unknown dependencies among the data-items to reflect the correlations among varied item-sets. In this paper, we aim to discover spatio-temporal association rules, or to be more specific, mobility association rules from taxi traces of a geographical region. Typically, it depicts that if a spatio-temporal event takes place, a resultant event is likely to occur within a defined spatio-temporal neighborhood. The challenges in finding mobility association rules are following:

- Unlike traditional relational databases, mobility information are not explicitly encoded as transactions but are rather embedded within the spatial framework of the geo-referenced data. Therefore, it is required to generate transactions from the spatio-temporal dataset such that association rule mining techniques can be deployed to discover mobility association rules.

- Conventional association rule mining works with categorical data and not with numeric data such as metric distance.

- To extract spatio-temporal association rules, huge amount of data needs to be analyzed in varied spatial and temporal scales. Therefore a computationally efficient approach is required to generate the effective association rules in timely manner.

In this direction, we propose a novel framework which is capable to discover the spatio-temporal interrelationships from mobility traces and utilize the association rules to predict travel demand.

3.1. Mobility-Flow: Spatio-temporal event

Spatio-temporal database captures events which represent the chronologically ordered instances of timestamp-geometry pair. Typically, geometries are used to represent space and can be depicted by set of grid cells or spatial region, such as polygon. The instances are application-specific and illustrate the type of the spatio-temporal event. For example, “Forest fire always occurs at region $R_1$ prior to the occurrence of haze in nearby region $R_2$” - where Forest fire and Haze are two instances of spatio-temporal events occurring at two different spatial locations ($R_1$ and $R_2$) and temporal sequence is represented by prior. The definitions used in this paper are defined as follows:

1. **Definition 1. Spatio-temporal event:** A spatio-temporal event or simply event, denoted as, $e(r,t,v)$, represents a spatial feature $e$ (namely, rain, haze) in the spatial region $r$ at time $t$ with an intensity value $v$.

   - $r$ represents co-ordinates (latitude, longitude) or a polygon geometry (denoted by bounding-box) or nodes of a spatial network

   - $v$ represents measurement of $e$, which may be estimated by specific unit (10mm of precipitation) or linguistic variables (high, medium).

   - $t$ represents either a particular time-stamp (1529496160 epoch) or a time-interval pair ($[t_i,t_{i+1}]$)

According to our problem set-up, we consider $e$ as a mobility event which helps to model traffic-flow and travel demand of a region $r$ in a given time-interval $t$.

2. **Definition 2. Mobility event:** A mobility event is a specific type of spatio-temporal event, denoted as, $M(r,t,v)$, illustrates movement dynamics of vehicles in a region $r$. In this work, we consider three types of mobility events:

   - Pick-up event: The number of pick-ups or starting of trips is $v_{pu}$ from the source region $r$ in the time-interval $t$

   - Drop-off event: The number of drop-offs or completion of trips is $v_{do}$ in the destination region $r$ in the time-interval $t$

   - Moving: The count of trajectory-segments of vehicles passing region $r$ in time $t$ is $v_{mn}

3. **Definition 3. Spatio-temporal Association Rule (SAR):** Spatio-temporal association rule is defined as $SAR = (r_i,[t_s,t_b]) \Rightarrow (r_j,[t_s,t_{j+}])$, with $s$ support and $c$ confidence, where objects appearing in region $r_i$ during time-interval $[t_s,t_b]$ either

   (i) appear in region $r_j$ for the first time or by (at or before time)

   (ii) be in region $r_j$ during the $[t_s,t_{j+}]$ time interval

   Our paper primarily considers the movement behaviours of taxis and aims to analyze the patterns by mining mobility events. We define mobility association rule as follows:

4. **Definition 4. Mobility Association Rule (MAR):** Mobility association rule is expressed as: $MAR = M_1(r_i, \Gamma = [t_s,t_b], v_i) \Rightarrow M_2(r_j, \Gamma + \gamma, v_j)$, which depicts that objects (such as taxis) contributing in a mobility event, $M_1$ also partake another mobility event $M_2$ after $\gamma$ timestamp of the former events’ time-interval. There are nine possible correlations or direct MARs from three mobility events (pick-up, drop-off and moving). Few such MARs along with their notations are given below.

   - Notation: $M_{pick-up}(r_i, \Gamma, v_i) \Rightarrow M_{drop-off}(r_j, \Gamma + \gamma, v_j)$ : Objects participating in a pick-up mobility event in
a region \( r_i \) in the time-interval \( \Gamma \) subsequently completes the trip in region \( r_j \) within the time-window \( [I_0 + \gamma, I_0 + \gamma] \). This type of mobility association rules help to correlate two spatial-regions, namely, source \( (r_i) \) and destination \( (r_j) \) within a specific time-window. It may be noted that \( \gamma \) is essentially different for different trips, however, the goal of our work is to extract such \( \gamma \) value so that the support and continuity of the rules maximize.

- Notation: \( M_{\text{moving}}(r_i, \Gamma, v_i) \Rightarrow M_{\text{moving}}(r_j, \Gamma + \gamma, v_j) \): Objects passing through region \( r_i \) in the time-interval \( \Gamma \) also pass through region \( r_j \) within the time-window \( [I_0 + \gamma, I_0 + \gamma] \). This MAR discovers frequent paths in different time-stamp. The entities, namely, \( r_i, r_j \) of the rule vary with different time-windows and given a specific time-interval and time-window our approach is capable to extract varied correlated regions having greater support and confidence than the minimum threshold.

- Notation: \( M_{\text{drop-off}}(r_i, \Gamma, v_i) \Rightarrow M_{\text{pick-up}}(r_j, \Gamma + \gamma, v_j) \): Objects completing trips in region \( r_i \) in the time-interval \( \Gamma \) start trips from region \( r_j \) within the time-window \( [I_0 + \gamma, I_0 + \gamma] \). This rule assists the drivers to find out next probable pick-up regions and subsequently helps to reduce the waiting time.

The above mentioned MARs can be directly extracted by mining taxi-trip database. However, to predict the travel demands efficiently and understand the overall mobility dynamics of a city-region, it is required to discover derived mobility association rule. The derived mobility association rule’s templates will be discussed in the later section of the paper.

**5. Definition 5. Travel demand:** Travel demand \( (d) \) of a region \( r_i \) at time interval \( \Gamma \) is the aggregated count of pick-up mobility event in \( r_i \) at \( \Gamma \).

\[
f(r_i, \Gamma, [t_n, t_b]) = \sum_{v \in \mathbb{V}} f(\Gamma, [t_n, t_b]) = \sum_{v \in \mathbb{V}} M_{\text{pick-up}}(r_i, t, v = v)\]

(1)

Travel demand may also include the number of trip requests from the citizens in a region. Since only taxi-trips are available as experimental purpose, we consider the travel demand of a region as the number of pick-ups in that region at a given temporal scale. However, the proposed approach is scalable enough to incorporate this value as a parameter along with the count of pick-up events in case it is available.

**4. MARIO Framework: Algorithms and Implementation**

MARIO framework exploits varied type of taxi-trip related information and utilizes the attributes to come up with the mobility rule template. It analyzes the spatial neighborhood effects of travel demand variation using Long Short Term Memory (LSTM) architecture where the temporal quantitative relationships of passenger count at different places are utilized as input feed. Moreover, we propose variant of temporal-apriori algorithm to discover the mobility association rules.

**4.1. Trajectory trace pre-processing**

This work uses GPS trajectory data of taxis in cities. Two different types of taxi-trace (ref. Table 2) have been utilized in this purpose. This GPS trace only includes \( <\text{latitude, longitude} > ; <\text{time}>\) of pick-up and drop-off locations (trace type I) or \( <\text{latitude, longitude} > \) after a particular time-interval (say, 30 secs) of the complete trip (trace type II). Prior to analyzing the GPS log, appending other semantic information, such as underlying road-network structure or nearby landmark is important for discovering intrinsic patterns of movement behaviour of a geographical region.

**4.1.1. Enriching taxi-trip log with semantic information**

Primarily in this step, GPS error is removed followed by appending semantic information, namely, geo-tagged points, road-network structure. The GPS traces of the taxis of the mobility database are segmented where each such fragments are denoted as trajectory-segment or \( T.S\text{eg} \). Initially, the pre-processing guarantees that all GPS points are strictly ordered on timestamps in the \( T.S\text{eg} \). For example, if there are more than one GPS point with same timestamp, then only the first occurrence of the GPS point is kept and rest points are removed.

\[
p_1(x_1, y_1, t_1) > \cdots > p_n(x_n, y_n, t_n) \text{ and } T.S\text{eg} = \{p_1, \ldots, p_n\}
\]

(2)

Here, GPS log is fragmented such that each \( T.S\text{eg} \) represents a single taxi-trip, i.e., \( p_1 \) and \( p_n \) are pick-up and drop-off points of a taxi-trip. Then, GPS errors, which are relatively large, in scale of 100 meters (due to GPS capturing device failure or poor satellite signals) are removed deploying Kalman filtering [33] technique. To address the issue of different sampling rates of various GPS-trackers of the taxis the analysis is carried out by segmenting the trips into varied trajectory sliders which store path followed by the taxis bounded by a specific distance and time thresholds. In the next step, geo-tagged information for each pick-up and drop-off points are extracted and appended to enrich the semantic information of the raw traces. To augment the nearest landuse information, iterative reverse geocoding (IRG) [40] is used. Additionally, we maintain a POI-database where \( <\text{latitude, longitude, POI}> \) are stored for any subsequent computation.

Finally, road network structure of the region is extracted from OpenStreetMap\(^1\) to semantically enrich the mobility traces. This information contributes in the semantic enrichment process by appending geometric information of the road network (such as length and width of the road-segments), the connectivity and continuity (like, intersection of roads) of the road network. Since we aim to utilize the MARIO framework in both urban

\(^1\)OpenStreetMap: https://www.openstreetmap.org/
(dense road-segments) and non-urban regions, an efficient map-matching process with high-accuracy is required. It may be noted that our framework works in two phases:

**Phase-I:** Firstly, it generates the knowledge base of movement dynamics from historical mobility records by discovering the inherent patterns,

**Phase-II:** It applies the extracted knowledge from the first phase to predict travel demand in different places. For the first phase, map-matching is an one-time phenomenon and could be performed offline. However extracting the position of the vehicle in the road network in phase two requires online-strategy to reduce pre-processing time. To map the raw taxi-trips onto the digital map we have used AntMapper matching which utilizes both topological information and global similarity measurement and provides result within a relatively short running time. The map-matching for phase-II of the framework exploits the advantages of mobility-index and map-matched trajectory data of phase-I.

To complement the missing values, two-step procedure is followed in the MARIO framework. In the first step, map-matching process is deployed. The map-matching process maps the GPS points into the underlying road-structure. The features of the road-segments: length, width, connectivity and continuity (like, interaction of roads) are extracted in this step. The map-matching algorithm named AntMapper assigns each GPS point to the appropriate road-segment. In the next step, for more refinement of the GPS points, all the GPS points are smoothened using Kalman filtering technique to reduce the error of the measurement/ data collection. Using kalman filtering, in case the outlier/ missing value is detected, it is replaced by other intermediate points as returned by the filtering technique. This step takes the output of the previous step (map-matching process), and finds out intermediate points to complement the missing values. The road-network and POI information are stored in Cloud Big Query Storage of Google Cloud platform. The map-matching and semantic enrichment algorithms are implemented using the compute engine, which calls Google Place API service and stores the data in a database [Oracle Spatial and Graph] with spatial extension.

4.1.2. Mobility association rule (MAR) template and GPS transactional database construction

The definitions of spatio-temporal event, association rule and mobility association rule are discussed earlier in section 3.1. In this section, we explore varied types of derived MAR templates and the construction procedure of GPS transactional database. The key-idea of exploring such MAR template is to facilitate a framework capable to effectively model mobility dynamics and discover interesting movement patterns. Given a database of mobility traces, we aim to define a language L conducive to express spatio-temporal properties of the entities in terms of mobility events. The intuitive meaning of any association rule $X \rightarrow Y$ is that transactions containing set of $X$ items tend to contain set $Y$ of items. Similarly finding GPS records containing set of spatio-temporal entities $X$ and $Y$ as well, is the main idea of discovering mobility association rules. The language $L$ is complex and there may be huge number of possible sentences of this language. In this work, we are interested to mine travel-demand specific patterns. Therefore, a set of specific rules (or sentences), which are a sub-set of $L$ are considered.

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeVal(t)</td>
<td>Timestamp (time-interval or time-slot of a day)</td>
</tr>
<tr>
<td>POI(p)</td>
<td>point-of-interest of ‘p’ type</td>
</tr>
<tr>
<td>locTraversal(dis, S, D)</td>
<td>Traveled dis distance from $S$ source to $D$ destination</td>
</tr>
<tr>
<td>timeTraversal(t, S, D)</td>
<td>Time spent $t$ to travel from $S$ source to $D$ destination</td>
</tr>
<tr>
<td>Traveld(c, r, t)</td>
<td>Travel demand (number of pick-ups) in a region $r$ at time-interval $t$ is $c$</td>
</tr>
<tr>
<td>footPrint(c, r, t)</td>
<td>Number of GPS footprints in a region $r$ at time-interval $r$ is $c$</td>
</tr>
</tbody>
</table>

To represent mobility association rule template, we primarily introduce few predicates [refer Table 2] with parameters which help to construct the basic structure of the rule-templates. It may be noted that all three mobility-events (pick-up, drop-off and moving) are also considered as predicates which take region, time and count of the event-instances as parameters. Based on the set of predicates ($Pre$) we form derived mobility association rules with spatio-temporal variables (region: r, time: t) and possible annotations. The basic syntax of such rule is:

$$\varphi_1, \varphi_2 \ldots \varphi_n := Pre(r, t) \ s.t. \ \varphi_1 \implies \varphi_2|\varphi_1 \wedge \ldots \varphi_i \implies \varphi_2 \wedge \ldots \varphi_j \implies \varphi_2 \vee \ldots \varphi_j$$

(3)

Given such rules, a selection function finds out how many transactions or records in the mobility database satisfy the implies condition. This selection function measures the support and confidence of the rule and if the count is above the threshold values of support and confidence then that particular rule is included for mapping movement dynamics of the region.

The city dynamics can be modelled by (i) exploring travel demands in different regions and in different time-slots, (ii) how the travel demand of one region impacts the GPS footprints of other regions, and (iii) movement behaviour of taxi-drivers based on region and time. In other words, we aim to reflect peoples’ movement dynamics through the above-mentioned factors. Four generic rule-templates have been designed, and we argue that the rule template is complete considering our application to model city-dynamics. Table 4 represents varied types of mobility association rules that help to model the overall mobility dynamics, such as travel demand, traffic flow, individual drivers’ movement patterns in a region. MAR is the template of direct mobility association rule which can be extracted from three mobility events directly. There are nine possible MARs under rule template MAR

We have extracted and illustrated such MARs with real-life data in the experimental section. Travel demand variation at different POIs can be extracted from
In this regard, the measurements of such MARs need to be quantified. It is obvious that support and confidence calculations of traditional association rules do not hold for this case. These measurements need to be extended in spatio-temporal domain. The spatial coverage of a rule defines the sum of the area referenced in the predicates of the rule. Temporal coverage of a rule represents the time window for which rules must be valid. The temporal coverage of different mobility rules are different. We deploy a scaling for both temporal and spatial coverage of the rules which is represented by the ratio of spatial or temporal validity of the rule and spatial and temporal information of the complete study period. Huang et al. [6] define participation index to measure the strength or frequency of a co-location pattern. MARIO utilizes similar measure to define the spatio-temporal support and confidence. Given a mobility association rule $MAR(\varphi_i \Rightarrow \varphi_j)$ containing spatio-temporal features $f_1, \ldots, f_k$, the following measures are described.

- Spatio-temporal Support ($\varphi_i(r, t_i) \Rightarrow \varphi_j(r, t_j)$): It is the scaled spatial coverage and the total length of the time-intervals in the rules.

$$STsupp = \frac{|\{f_1, \ldots, f_k\}|}{|\{f_1, \ldots, f_k\}|} \times \frac{\sum_{i=1}^{\infty} \frac{|\varphi_i|}{N \times T}}{N \times T} \quad (4)$$

where $|\{f_1, \ldots, f_k\}|$ represents the number of transactions in the database containing the spatio-temporal features and the next term denotes the temporal scaling parameter where $N$ is the total time-scale of the study period.

- Spatio-temporal Confidence ($\varphi_i(r, t_i) \Rightarrow \varphi_j(r, t_j)$): It is measured as the conditional probability of the predicate $\varphi_j$ is true given that $\varphi_i$ is already true.

$$STconf = \frac{STsupp(\varphi_i(r, t_i) \Rightarrow \varphi_j(r, t_j))}{STsupp(\varphi_i(r, t_i))} \quad (5)$$


<table>
<thead>
<tr>
<th>Rule Id</th>
<th>Rule template</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAR1</td>
<td>$M_1(t_1, t_2, t_3) \Rightarrow M_2(t_2, t_3, t_4)$</td>
</tr>
<tr>
<td>MAR2</td>
<td>$\text{timeVal}(t) \land \text{POI}(p) \Rightarrow \text{travelD}(c, p, t)$</td>
</tr>
<tr>
<td>MAR3</td>
<td>$\text{travelD}(c, r, t) \Rightarrow \text{footPrint}(c, r, t)$</td>
</tr>
<tr>
<td>MAR4</td>
<td>$\text{locTraversals}(dis, S, D) \Rightarrow \text{context} \land \text{timeVal}(t)$</td>
</tr>
</tbody>
</table>

MAR2. It may be noted that rules are fully dependent on space and temporal values, i.e., variation in region and time change the count (or in other word, support and confidence measure) of the records which satisfy that rule. Therefore the proposed framework works in two ways, (i) extracts frequent patterns of the records which satisfy that rule. Therefore the proposed algorithm can be deployed. From the conventional definitions of item-set and transaction, we know that, $I = \{I_1, I_2, \ldots, I_n\}$ set of items and transactional database contains $T = \{T_1, T_2, \ldots, T_m\}$ set of transactions, where each transaction $T \subseteq I$. In this work, from the raw GPS log, first we construct transactional database where each transaction $T$ consists of $<\text{tld}, \text{spatialIn}, \text{event}, \text{temporalS pan}, \text{context}>$ tuple. It may be noted that we have considered spatial database (Oracle Spatial and Graph) with spatial data-types, such as point, polygon, polyline etc. Therefore spatialIn field stores the spatial information, namely, polygon for a region and polyline for movement trace. Table 5 shows an example of such transactional database, where the last column represents additional information, such as $|C| - 2$: count of passengers, $F_t - 50$: fare amount or $O$: occupy status. All the GPS records are converted to such spatial database transactions and further, mobility index is generated on such records.

<table>
<thead>
<tr>
<th>tld</th>
<th>spatialIn</th>
<th>event</th>
<th>temporalSpan</th>
<th>context</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ij}$</td>
<td>point_geon</td>
<td>pick-up</td>
<td>08.00-08.05</td>
<td>$</td>
</tr>
<tr>
<td>$T_{ij}$</td>
<td>point_geon</td>
<td>drop-off</td>
<td>08.40-08.50</td>
<td>$F_t - 50$</td>
</tr>
<tr>
<td>$T_{ij}$</td>
<td>polyline_geon</td>
<td>moving</td>
<td>10.00-10.25</td>
<td>$O$</td>
</tr>
</tbody>
</table>
From the next level, sequences of the trajectory-segments are maintained in different levels along with the place-id (type and location), frequency of visit and time-slots. The hashing technique considers the spatial-locality information, i.e., the nearby places are stored in consecutive buckets. The following pairing function is used as the hash-function:

$$H(p_x, p_y) = (p_x + p_y)(p_x + p_y + 1)/2 + p_y$$  \hspace{1cm} (7)

Frequency plays an important role in the hashing scheme. When a new entry comes in the storage, a search function is deployed and if similar entry is already present, then frequency is incremented and the new entry is inserted in the list. This frequency attribute helps to minimize the redundant entries and also can be used as a feature to find out most followed trajectory sequence.

MARIO deploys the k-level temporal hashing scheme to store trajectory-sequences of a region into k layers. This hashing scheme is beneficial for extracting movement information efficiently and in timely manner. For example, the most followed route of a region can be discovered by analyzing the k-sequences and frequency. On the other side, traffic states of a place can be explored by interpreting the GPS footprints of the taxi-ids in different time-slots. We have implemented this schema using BigTable and BigQuery of Google Cloud Platform (GCP) as storage platform. Cloud DataProc of GCP is used to maintain the index and insertion of records in the storage. Cloud Spanner of GCP is used to store these information which supports horizontal scalability.

4.2. Spatio-temporal analysis

In this section, we discuss about movement dynamics in the study region and how to model the variation of mobility events in varied space and temporal dimension. We have presented how mobility events of one region effect other regions after a particular time-interval. We have introduced trajectory graph to model and store varied mobility features in different temporal scale.

4.2.1. Trajectory graph

Trajectory graph is a labeled directed graph where the properties of the graph change as a function of time. The graph evolves by appending or deleting nodes and links over time. This graph helps to characterize information flow in a network, namely, travel demand variation or mobility events on the underlying road-network structure of the region. Formally, trajectory graph is defined as $G = (V, E, \Upsilon, \tau)$ where $V$ represents the set of vertices or nodes and $E \subseteq V \times V$ is the set of edges. The other two parameters $\Upsilon$ and $\tau$ depict node labeling function and edge labeling function over time respectively. The time-evolving nature of the graph is conceptually expressed by a series of directed graphs $G_{t_1}(V_{t_1}, E_{t_1}, \Upsilon_{t_1}, \tau_{t_1}), \ldots, G_{t_T}(V_{t_T}, E_{t_T}, \Upsilon_{t_T}, \tau_{t_T})$ at different time-instances $t_1, \ldots, T$. Based on the labels assigned by $\Upsilon$ and $\tau$ there are three possible types of trajectory-graphs in our analysis. The node labeling function assigns
$V \rightarrow \Sigma$ a label to each node in the graph from an alphabet $\Sigma$. The labels illustrate the properties of the nodes in the graph. The node properties may represent static information (such as $G_1$ : intersection of roads or $G_2$ : POI-placements do not change with time) or may vary in different temporal scales (such as $G_3$ : mobility-events). The trajectory graph with mobility events as nodes is constructed as follows:

- Sources and destinations of the taxi trips are converted to nodes of the graph and labeled as $<\text{Pick – up}, t_i>$ and $<\text{Drop – off}, t_j>$ respectively.
- If the traces are available from source to destination of the trip, then after $\zeta$ time-interval one GPS point (p) is selected from the trace and $p$ is converted to a node of the graph with label $<\text{Moving}, (t_i + \zeta)>$.
- The edges of the graph are generated from taxi trip’s nodes and ordered on increasing timestamp. The label of the edges are the time-differences between two nodes $<t_{i+1} - t_i>$.  

The detailed procedures to construct trajectory graph, namely, mobility event graph are noted in algorithm 1. The tabular representation of trajectory graph is shown in Table 6. For road graph construction, we deploy a nested adjacency list structure to store the direction of the roads in different time-intervals. It may be noted that two different road graph structures have been maintained for weekdays and weekends. It is observed that in most cities traffic directions vary typically in weekdays and weekends. The algorithm 1 constructs the mobility-event graph from the historical movement traces of a region. Let us assume, the number of taxis in the dataset is $M$. The average number of trips by each taxi is $n$. Therefore, it takes $O(M \times n)$ to find out the pick-up and drop-off points in the complete dataset. [line no. 3 – 10] Next, the algorithm finds out the GPS points between pick-up and drop-off points of the complete path. Let us assume, the average length of the taxi is: $L$. The average time-duration of the trip: $dur$. The time-offset parameter is: $\zeta$. Therefore, the number of intermediate (moving) points $= (dur/\zeta)$. The time complexity to generate the moving edges and nodes (line no 11 – 26) is $= O(n \times L \times \frac{dur}{\zeta})$. Therefore, the overall time-complexity of the algorithm 1 is $= O(\sum_{k=0}^{M}(n \times L \times \frac{dur}{\zeta})) + n$.

4.2.2. Modelling travel demand variation

Discovering the effects of a mobility phenomenon, such as traffic blockage and road closure, on a different region after specific time-interval is the most challenging task in travel demand analysis. For example, “If there is a road closure in region 1, then region 2 experiences higher travel demand in a particular time-interval”. Here, region 1 is effecting the mobility patterns of region 2 and prior prediction of such effects may help in effective resource management. Therefore, the main objective is to find out such relations among spatial regions from the historical movement log. Next, the framework aims to find out the temporal relations among the events’ effect, i.e., “traffic congestion of region 1 is propagated to region 2 after $\Delta$ time period” - where the range of $\Delta$ value needs to be discovered. In this work, the regions which are effected by a mobility event at

<table>
<thead>
<tr>
<th>ID</th>
<th>Node(V)</th>
<th>Edge (E)</th>
<th>$T$</th>
<th>$\tau$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>Intersection of roads</td>
<td>Road segments</td>
<td>Location information (latitude, longitude)</td>
<td>Direction</td>
</tr>
<tr>
<td>$G_2$</td>
<td>POI</td>
<td>Road segments</td>
<td>POI-type information</td>
<td>Direction</td>
</tr>
<tr>
<td>$G_3$</td>
<td>&lt;Mobility event, location, timeInterval&gt;</td>
<td>Sequence (After/Before relation)</td>
<td>Types of mobility events</td>
<td>Time taken from one event to another</td>
</tr>
</tbody>
</table>

Algorithm 1 Trajectory graph (Mobility event graph) construction

**Input:** Taxi-trip trajectory ($T_{Y_1}, T_{Y_2}$) log

**Output:** Mobility event graph $G_3(V, E, T, \tau)$

1: function Generate$G_3(T_Y)$ where $T_Y$: taxi trajectory log with pick-up and drop-off points, $T_{Y_2}$: taxi trajectory log with all intermediate points
2: $G_3([V], [E]) \leftarrow \text{NULL}$ $\triangleright$ 2-D matrix where each row, column is of $<\text{id, location, label}>$ data-type
3: for $i = 1$ to length($T$) do
4: Create_node($G_1, V, \text{left} (\text{id, location, label})$
5: $T : G_3.V[|id|] \leftarrow p \triangleright p$ - pick-up
6: $G_3.V[p][\text{label}] \leftarrow T[|t|].\text{pick-up}$
7: Create_node($G_3, V, \text{left} (\text{id, label})$
8: $T : G_3.V[|id|] \leftarrow d \triangleright d$ - drop-off
9: $G_3.V[d][\text{label}] \leftarrow T[|t|].\text{drop-off}$
10: end for
11: for $j = 1$ to $i$ do
12: $t \leftarrow V_j.[\text{label}$
13: for $k = 1$ to length($T[|j|]$) do
14: if length($T[|k|]$) $> 2$ then $\triangleright$ $T_{Y_2}$ trajectory trace
15: $m \leftarrow 1; l \leftarrow \text{length}(T[|k|])$
16: while $m! = l$ do
17: $p \leftarrow$ Select GPS point after $(t + \zeta)$ time – interval from ($T[|k|]$)
18: Create_node($G_1, V_j, \text{left} (\text{id, label})$
19: $T : G_3.V[|id|] \leftarrow m \triangleright m$ - moving
20: $G_3.V[p][\text{label}] \leftarrow p.\text{timestamp}$
21: $t \leftarrow p.\text{timestamp}$
22: $\tau : \text{CreateEdge} \leftarrow (V_i[p], V_i[m])$
23: end while
24: end if
25: end for
26: end for
27: end function
other regions are termed as \textit{neighbour}.

Typically, spatial neighbors are formed with points of a grid or nodes of a graph those are \textit{close} to one another. The \textit{neighborhood} definition largely depends on the denotation of \textit{close}, where close generally represent adjacent or within some threshold range of distance. However, in our analysis, distance measure is not well-suited for analysing the underlying correlations among varied regions. Here, we define \textit{neighborhood function} and few related terms.

\textbf{Region}: A region $R = (V', E')$ is a sub-graph of the trajectory graph $(G)$, where $V' \subseteq V$ represents the nodes (intersection point or POI) and $E' \subseteq E$ depicts the roads between two such nodes. For the ease of visualization any region is denoted by a rectangular bounding box.

\textbf{Source}: A region is denoted \textit{Source} $(S)$ if the count of moving objects entering $(n_e)$ in the region is much lesser than the objects leaving $(n_l)$ in the region in a particular time-slot.

\begin{equation}
|n_l - n_e| \geq \text{min}_\text{thresh} \tag{8}
\end{equation}

\textbf{Sink}: A region is denoted \textit{Sink} $(S')$ if the count of moving objects entering $(n_e)$ in the region is greater than the objects leaving $(n_l)$ in the region in a particular time-slot.

\begin{equation}
|n_l - n_e| \leq \text{min}_\text{thresh} \tag{9}
\end{equation}

The pick-up and drop-off events are considered while counting $n_e$ and $n_l$.

\textbf{Flow distance}: Flow distance $(F)$ between two regions $(r$ and $r')$ is defined as a time-series arraylist of $(n_l, n_e)$ pairs:

\begin{equation}
F(r, r')[t_i, t_j] = \{|n_l - n_e|_{[t_i, t_j]}, |n_l - n_e|_{[t_i, t_j]}\} \tag{10}
\end{equation}

In the next step, MARIO explores the travel demand variation of different regions based on functional area and time-slots. In section 4.2.1 we describe the method of augmenting geo-tagged information with the GPS log. The functional area or geo-tagged information is crucial to model the travel demand, since citizens mostly follow similar temporal rules for their social activities. The complete study region is divided into 12 social functional areas, \textit{residential}, \textit{commercial}, \textit{government organization}, \textit{industrial region}, \textit{school or university}, \textit{entertainment area}, \textit{area of historical interests}, \textit{religious place}, \textit{tourist spots}, \textit{railway}, \textit{airport region} and \textit{highway region}. Each day is divided into 15 minutes slot where slot 0 starts from 0800 in the morning. The study-area is divided into $n = 12$ functional regions where each region is considered as the bounding box of a sub-graph of the trajectory graph. The number of pick – up, drop-off and moving mobility events in region $R_i$ during different time-slots $(t_0, \ldots, t_n)$ of a day (say, $L_i$) are represented as:

\begin{equation}
P^k_{t_i} = (p^k_{t_i, 1}, p^k_{t_i, 2}, \ldots, p^k_{t_i, n}) \tag{11}
\end{equation}

\begin{equation}
D^k_{t_i} = (D^k_{t_i, 1}, D^k_{t_i, 2}, \ldots, D^k_{t_i, n})
\end{equation}

\begin{equation}
M^k_{t_i} = (M^k_{t_i, 1}, M^k_{t_i, 2}, \ldots, M^k_{t_i, n})
\end{equation}

Next, for each region 3 matrices are formed for all GPS logs $(L_1, \ldots, L_d)$ of $d$ days.

\begin{equation}
M = (M_s, M_p, \ldots, M_t) = \left[\begin{array}{cccc}
M^s_{t_1} & M^s_{t_2} & \cdots & M^s_{t_d} \\
M^p_{t_1} & M^p_{t_2} & \cdots & M^p_{t_d} \\
M^t_{t_1} & M^t_{t_2} & \cdots & M^t_{t_d}
\end{array}\right] \tag{12}
\end{equation}

where $M$ shows the distributions of moving taxis in $n$ regions for a particular time-slot $(t)$ of $d$ days. Similarly, other two matrices namely pick-up distribution $(P)$ and drop-off distribution $(D)$ are formed. Next, we deploy \textit{autoregressive integrated moving average} to find out the values of pick-up, drop-off and moving events in different time-slots from the historical movement traces. Although the method can effectively model non-linear tim-series, however, the impact or effect of other regions are not considered. Therefore, in the next step, MARIO aims to find out effect of one region to other from spatio-temporal context.

Two regions $(r$ and $r')$ are neighbors of each other in a time-slot $T = [t_i, t_{i+1}]$ if the flow distance between two regions are within a threshold and the regions are edge-reachable in $T$. Edge-reachable defines that no other neighboring region in the same time-slot $T$ is present in the connected path of $r$ and $r'$. The edge-reachable returns whether two regions are neighbor of each other in a given time-slot. Typically, it checks whether the flow-distance and time-differences are within the threshold range.

\begin{equation}
f_{\text{neighbor}}(r, r', t_i, t_j) = \begin{cases} 
1 & \text{if } (t_j - t_i) < t_h \land \text{edge-reachable}(r, r') = 1 \\
0 & \text{otherwise}
\end{cases} \tag{13}
\end{equation}

$\text{edge-reachable}$ returns whether two regions are neighbor of each other from the connected path of the present. Figure 5 shows the snapshot of LSTM network, which maps an input sequence to an output sequence by computing the network activations in different time-instances.

\footnote{To avoid the ambiguity of the definition of region, the paper uses $R$ as the underlying network and $r$ as the bounding box (polygon geometry) of any geographical area.}
The basic equations of a typical LSTM architecture are as follows:

\[
\begin{align*}
    i_t &= \sigma(w_i[h_{t-1}, x_t] + b_i); \\
    f_t &= \sigma(w_f[h_{t-1}, x_t] + b_f); \\
    o_t &= \sigma(w_o[h_{t-1}, x_t] + b_o); \\
    c_t &= f_t \cdot c_{t-1} + i_t \cdot \tanh(w_c[h_{t-1}, x_t] + b_c); \\
    h_t &= o_t \cdot \tanh(c_t)
\end{align*}
\]  

where input gate, output gate and forget gates are represented by \(i, f\) and \(o\). The scalar product between two vectors are represented by \(\cdot\) whereas \(\sigma(\cdot)\) denotes the logistic sigmoid function. The activation vectors for each cell is \(c\) and \(m\) are memory blocks. \(w_i\) and \(b_i\) are the weight matrices for neurons and bias vectors for respective blocks. MARIO uses \(\tanh\) for cell input, output activation functions and \(\text{softmax}\) as network activation function.

The input of the LSTM architecture is region along with mobility states and road-network structure. MARIO aims to model the mobility states jointly with road network structure and POI-placements since the correlations with other regions inherently depend on these aspects. Sak et al. \cite{37} demonstrates that two-layer deep LSTM architecture outperforms other existing methods where each layer has a linear recurrent projection layer. MARIO uses this two-layer deep architecture to find out the correlated regions. The epoch (time step) is varied from 10 to 200. The batch size is 32. The input dimension is 10x10 out the correlated regions. The epoch (time step) is varied from 10 to 200. The batch size is 32. The input dimension is 10x10 (meter) and the output dimension is list of regions, where each region such that the sub-graph within each grid has nearly similar cardinality (number of edges). The grid information help to correlate the location and other contextual information, such as POI placement, mobility events etc. Next, road graph is placed on the created grids. In level 2, mobility event graph is created based on the spatial transactional database containing taxi-trips. Finally, level 3 consists of the unusual mobility phenomenon, such as traffic congestion (when velocity of the vehicles are less than a threshold value), traffic blockage (velocity is nearly zero) or road-closure and travel-demand changes. This level specifically helps to understand the real-time movement dynamics of a region while level 2 provides frequent movement patterns and travel demands in various time-slot. The major steps are as follows:

- **Grid construction**: The study area is divided into \(n\) grids such that differences of \(|E| \ast \sum_{e \in E} \text{length}(e)\) in grids are within a threshold value. We start the partitioning task in a top-down fashion, where the complete road graph is divided into \(n\) connected sub-graphs.

- **Aggregated mobility event graph construction**: This process takes the mobility event graph as input and deploying a label propagation clustering generate region-clusters having same mobility events.

- **Mobility phenomenon**: We have considered mobility phenomena: traffic congestion, traffic blockage and travel demand variation. First, from the level 2 the matrix is created with the features, GPS footprints in each edge of the grids, travel-demand in the regions and average velocity of the vehicles in the road-segments.

Where \(Ne(n)\) is the neighbors of node \(n\). This process converts the sparse road-network into vector embedding for encoding layer. The deep learning architecture is implemented using Google Tensorflow\cite{3} The implementation is done on the top of the Google App Engine, including DataStore and Task Queues. Two Cloud SQL instances are created, where one is executed from Google App engine. The other instance has the database access permission. To add storage capacity, automatic storage increase is enabled.
Algorithm 2 provides the procedures of grid construction by graph partitioning. Firstly it computes the geo-hash codes of the study-region and divides the complete region into several uniform grids. Then it analyses the cardinality of road-network graph within each such grids and aims to minimize the differences between cardinalities of the sub-graphs. Finally, the algorithm outputs the geo-hash codes of generated grids having more or less similar cardinality of graph information. Let us assume, the number of edges and nodes of the road-graph (R) is $|E|$ and $|V|$ respectively. Time taken to partition the region into $n$ uniform grid: $O(n)$ Time to compute the geo-hash code of the region: $O(c_1 \times n)$, where $c_1$ is a constant Time to compute the cardinality (size of edge-set) of all such grid: $O(c_2 \times (|V| + |E|))$, where $c_2$ is a constant (since the number of edges of all $n$ grids will be equal to the number of edges of the road-graph) Time to find out the differences of edge-set is: $O(n \times |E|/n) = O(n)$ Let us assume the optimal number of partitioned graph is: $a$ and $a \leq n$, The number of edge set in each such partitioned graph is: $b$ and $a \times b \leq |E|$. Therefore, the overall time-complexity of the algorithm 2 is $O(n + c_1 \times n + c_2 \times (|V| + |E|) + n + a \times b) = O(n + c_1 \times n + c_2 \times (|V| + |E|) + n + |E|)$. Since, $a \leq n$ and $a \times b \leq |E|$, the worst-case time complexity is $O(c_3 \times n + c_4 \times (V + 2E))$.

4.3. Mobility rule mining

In this section, we demonstrate the process of extracting mobility association rules from the taxi-trips by deploying a variant of temporal apriori [22]. Furthermore, we aim to predict travel demand in different places based on the extracted mobility rules. The mobility miner method consists of two major steps, namely, capturing the prevalent candidate items of the rules in different spatio-temporal resolution and extracting the rules having support and confidence greater than the threshold values. A M-flow or mobility-flow represent any sub-graph of level 3 of mobility dynamics network, i.e., the effects of mobility phenomenon to other neighboring region. With the help of mobility rules, we aim to analyze the spatio-temporal neighborhood effects and predict the mobility characteristics of the regions along with predicted travel demand. Here frequency tree is utilized and the GPS trace of different time-intervals have been used as the input feed.

**Spatio-temporal property 1: M-flow follows Apriori Property: Any sub-set of infrequent spatio-temporal event-set (sequential pattern) is also infrequent**

To this end, we can formulate the mobility-rule extraction problem as: Given a database ($\mathcal{D}$) of spatio-temporal events, a specific location and time window, a mobility rule template
Algorithm 2 Grid construction by graph partitioning

Input: Road graph \((R(V,E))\), min, max
Output: List of Grids \((\text{Grid}[\text{geo} \cup \text{hash}])\)

1: \text{function} \text{GridConstruct}(R(V,E), \text{min, max}) \quad \triangleright \text{Where } R \text{ is the road-graph of the study region, min and max represent the minimum and maximum number of grids}
2: \text{geo} \cup \text{hash} \leftarrow \text{generate geo} \cup \text{hash code}(R)
3: \text{subG}[i] \leftarrow \text{Uniform - grid(geo} \cup \text{hash, min)}; n \leftarrow \text{min}
4: \text{diffL}[i] \leftarrow \sum_{\text{subG}[i]} \text{connectedComps(subG)} \times \sum_{\text{subG}[i]} ||\text{E}[\text{subG}] - \text{E}[\text{subG}]||
5: \text{while } n \leq \text{max do}
6: \quad m \leftarrow \text{compute min diff}_n \quad \triangleright \text{Solution using simplex method}
7: \quad \text{for } i = 1 \text{ to } m \text{ do}
8: \quad \quad G_i \leftarrow \text{mergeSubgraphs(subG, i)}
9: \quad \quad \text{diffL}[i] \leftarrow \sum_{G_i} \text{connectedComps}(G_i) \times \sum_{G_i} ||\text{E}[G_i] - \text{E}[G_i]]||
10: \quad \text{diff}_n \leftarrow \text{diffL;} \text{subG} \leftarrow G_i
11: \quad \text{end for}
12: \text{end while}
13: \text{Grid(geo} \cup \text{hash} \leftarrow \text{compute geo} \cup \text{hash(subG)}
14: \text{end function}

\(\text{MAR}\), and a minimum support threshold, the problem of extraction of mobility rules is equivalent to discovering set of frequent item-sets among a set of items in a transactional database. Extracting spatio-temporal mobility rules is not a trivial process because space and time can not be analysed in same dimension, time is an increasing sequence of small temporal-quantum while such sequence is not present in 2D space.

We consider another situation: Spatio-temporal property 2: An infrequent item-set of M-flow for a time-range \(T\) and spatial region \(R\), may be frequent item-set for a smaller region \(R - \Delta\) and a time-interval \(T - \delta\). It depicts that numerical attributes (trip-length, trip-duration etc.) depend on the spatial and temporal coverage of the rules.

Formally, the **mobility association rule** is defined as \(e_1 \land e_2 \land c_1 \rightarrow d_1\), where a conjunction of literals \(e_1, l_1, c_1\), i.e., edge information, location information and other contextual information are associated with a specific travel demand. A tuple \(t\) satisfies a rule if and only if it satisfies every possible literals (space, time) in the rule. If \(t\) satisfies the rule, travel demand of the region at the particular time can be represented by the value of \(d_1\). This leads to the former research question “Using taxi-trajectories, how mobility dynamics of a region can be captured and represented as mobility-association rules and help to predict the travel demand?” The extracted mobility association rules of a region will help to understand the travel demands accurately and predict the traffic flow. The next process involves extracting mobility flow of a region by analysing taxi-trips. The first step of the proposed method is to discover all frequent spatio-temporal event-sets followed by finding correlated events analysing spatio-temporal neighborhood of the events.

The key observation is any temporal relationship between the antecedent and consequent of the rules must follow higher order sequences. Hence, any infrequent item-set within the same temporal resolution but in a reverse order is discarded a priori. An **Iterative Deepening** based enumeration method has been proposed where different item-sets are inserted based on the temporal occurrences. In the next step, a **frequency tree** has been generated, where,

- Each node at height \(k\) stores sequence of event-sets, and possible travel-demand at different time-resolution
- The travel demand is initialized as 1 in the beginning phase
- After each iteration, the frequency of the events are modified and different granularity of spatio-temporal information are maintained [to satisfy spatio-temporal property 2]

The rule mining algorithm is a variant of T-Apriori method which consists of a number of passes depending on the time-slots. The algorithm finds out i-itemsets (i.e., itemsets with i items having at least the minimum support) at pass \(i\). It generates the set of candidate of i-itemsets and computes the count by scanning the database. Finally it finds out the MARs by inspecting the spatio-temporal support of all the candidate itemsets. The algorithm is terminated when no large itemset is discovered after a pass. Furthermore, the MAR template is used as an input feed. The algorithm only finds out the item-sets which are present in the rule-template. It significantly reduces the search time. The algorithm works in a bottom-up fashion where small temporal scale is used in the first phase followed by grouping larger time-slots given that spatio-temporal support and confidence are larger than the minimum threshold. Given the four rule-templates, MARIO extracts MAR\(_1\) type rules by analyzing the spatio-temporal neighborhood method proposed in section 4.2.3 and deploying temporal apriori method. MAR\(_2\) and MAR\(_3\) type rules are extracted by computing \(M, P\) and \(D\) matrices [see section 4.2.2] followed by executing temporal apriori algorithm. Finally, the individualized movement patterns MAR\(_4\) can be extracted by grouping unique taxi-ids and then performing rule mining process.

Our proposed rule mining algorithm differs from T-Apriori \(^4\) which is used to discover temporal pattern for interval-based events. However, the impacts of one regions’ mobility events can not be incorporated using only T-Apriori. MARIO exploits the extracted spatial-neighborhood effects of mobility events while extracting the mobility rules. The patterns of effecting other regions’ mobility states are extracted by deep LSTM architecture and associated with the rule mining algorithm.

5. Performance Evaluation

All experimental evaluations are carried out on VM instance of Google Cloud Platform\(^4\) having 4 vCPUs, 15GB memory
and Ubuntu 16.04, Linux as the OS. The algorithms are implemented in Python, R and all the experiments are performed on three real datasets of taxi trajectories.

5.1. Dataset

The dataset (Type I) is collected by the NYC Taxi and Limousine Commission (TLC). It contains trip records from all trips completed in Yellow and Green taxis in NYC from 2009 to present, and all trips in for-hire vehicles (FHV) from 2015 to present. It contains over 750 millions taxi-trips and the storage of over 190GB. We have also evaluated MARIO with Roma taxi traces (Type II). The Roma dataset contains mobility traces of taxi cabs in Rome, Italy consisting GPS coordinates of approximately 320 taxis collected over 30 days. The SF dataset contains mobility traces of taxi cabs in San Francisco, USA consisting GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay Area. Both the Rome and SF dataset contain GPS points of the taxi-trips in a particular time-interval (10-15 secs). As depicted before, two types of taxi-trip data have been analysed (see Table 2), where type I dataset has only pick-up and drop-off points, and GPS traces of complete trip have been logged in the type II dataset. The missing values of logging occur in type II dataset. It is observed that among 320 taxis, there are 6 taxis with less than 5 days trip-history. These 6 taxis are eliminated from the data-set. Amongst other 314 taxis, the locations are logged in 12secs offset. Total number of trajectory segments are 44,286. The number of trajectory segments consisting missing values is 2795, which is around 6% of the complete trajectory dataset of Rome. On the other-side, the total number of trajectory-segments in SF dataset is 56,920 and the missing values are found in 5467 segments, which is around 10% of the SF dataset. These missing values are replaced by intermediate points by map-matching and Kalman filtering techniques.

To depict the effectiveness of the proposed framework, we have carried out the experimental evaluations in two aspects: evaluating the system performance of MARIO framework and extracting varied mobility rules, predicting travel demand in different part of the city network.

5.2. Performance Evaluation of MARIO: Indexing scheme

One of the major contributions of MARIO is facilitating an end-to-end framework which is conducive to resolve user travel related queries. We propose k-level temporal hash based schema to store the huge amount of data and reduce the information extracting time. To depict the efficacy of the storage along with indexing scheme of MARIO, we compare it with five baseline methods namely R-tree, extended historical R-tree (HR+ tree), Scalable and Efficient Trajectory Index (SETI), TrajStore and R2-D2. A multiversion structure is proposed in HR+ tree where entries of different timestamps are placed in the same node leading to reduction of space. All of these methods are implemented and tested with NYC taxi-trip dataset. Chakka et al. propose a two-level index structure to decouple the index of spatial and temporal information. Trajstore maintains an optimal index and dynamically co-locates and compresses spatially and temporally adjacent segments on disk. A grid based index is proposed in R2-D2 where the area of interest is divided into a set of rectangular cells with fixed size and trajectories are indexed in the cells they pass.

The comparisons have been carried out for index-size, and query-time. We have considered R-query and T-query and report the average query-response time for all methods. Figures demonstrate the evaluation results of MARIO with other baseline methods. It has been shown that MARIO has less query execution time compared to other methods. Here, the proposed k-level temporal hashing scheme has outperformed other baselines in a huge margin (almost 50% less execution...
time in average). The key reason is that the trip-sequences of a region are stored into k-temporal levels and in consecutive buckets following the hash-function based on latitude and longitude information. It helps to extract the range and T-query in an efficient manner compared to other methods. Furthermore, with the increasing data size, MARIO maintains a reasonable rate of index size, since it avoids to maintain any duplicate entry using the frequency attribute of the hashing-scheme. In summary, although there are several indexing methods for spatio-temporal data-set, however they fall short in maintaining large mobility database and providing timely access of trajectory information.

5.3. Extracted Mobility Rules, Measurements

Table 7 shows a subset of extracted mobility rules along with average scaled spatio-temporal support and confidence. The minimum threshold of spatio-temporal support and confidence are set to 0.10 and 0.75 respectively from [0,1] range. The number of extracted mobility rules in this range are 120, 356 and 288 for NYC, ROME and SF respectively. The average spatio-temporal confidence of these rules lie in the range of (0.78 – 0.84), (0.82 – 0.88) and (0.85 – 0.89) respectively for the three study-areas.

It may be observed two types of mobility rules have been extracted. The former rules (MR1 – MR5) are based on travel demand in different POIs at varied timestamps while other rules are generated depending on mobility behaviours of taxi-drivers. For example, it has been noted that few taxi-drivers start their trips at early morning and typically travel in the residential and commercial area, while another set of taxi-trips are mainly from airport region to residential region and of large time-duration. Variation of trip-distance, taxi-fare along with varied timestamps have been reflected by the extracted mobility rules. These rules then are utilized to predict travel demand in different regions. Apart from the rules shown in Table 7, the other types of rules considered in MARIO (rule template MARt) are neighborhood effects. These type of rules are the highest numbers of rules that have been extracted. Although this can not be shown in the tabular format however these rules significantly help to predict travel demand and GPS footprints in different regions.
of the city.

Based on the trip’s pattern, taxi-drivers may be grouped into three different categories, (i) \( G_1 \) : Starts trip early in the morning slot and typically moves around residential area, academic area and railway or bus stops. The number of trips is fairly large in a day. (ii) \( G_2 \) : Trips are initiated in late afternoon from commercial and entertainment places. There are also fair amount of trips from areas of historic interests or tourist places. (iii) \( G_3 \) : These taxi-trips mainly cover movements from airport area and railway stations and bus-stop. The trip-duration is large and subsequently less number of rides are covered by the taxi-drivers.

\[ M \text{-Rule} \]

<table>
<thead>
<tr>
<th>M-Rule</th>
<th>S</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>( MR_1 ): TimeStamp(( T_1 \land T_3 )) \land Weekday ( \Rightarrow ) travelDemand(High, ( R_1 ))</td>
<td>0.28</td>
<td>0.867</td>
</tr>
<tr>
<td>( MR_2 ): TimeStamp(( T_1 \land T_2 )) \land Weekday ( \Rightarrow ) travelDemand(High, ( R_3 ))</td>
<td>0.21</td>
<td>0.785</td>
</tr>
<tr>
<td>( MR_3 ): TimeStamp(( T_3 \land T_4 )) \land Weekday ( \Rightarrow ) travelDemand(High, ( R_2 ))</td>
<td>0.182</td>
<td>0.843</td>
</tr>
<tr>
<td>( MR_4 ): TimeStamp(( T_1 \land T_2 \land T_3 )) \land Weekend ( \Rightarrow ) travelDemand(High, ( R_4 ))</td>
<td>0.145</td>
<td>0.874</td>
</tr>
<tr>
<td>( MR_5 ): travelDemand(High, ( R_1 )) \land travelDemand(High, ( R_3 )) \land travelDemand(High, ( R_2 )) \land travelDemand(High, ( R_4 )) ( \Rightarrow ) travelDemand(High, ( R_5 ))</td>
<td>0.128</td>
<td>0.902</td>
</tr>
</tbody>
</table>

\[ MR_6 \]: Trip – duration\( (\text{small}) \) \land TimeStamp(\( T_1 \) to \( T_3 \)) \( \Rightarrow \) noOfTrips\( (\text{high}) \) \land Region(\( R_1 \), \( R_3 \), \( R_5 \))

\[ MR_7 \]: Trip – duration\( (\text{small}) \) \land TimeStamp(\( T_4 \) to \( T_3 \)) \( \Rightarrow \) noOfTrips\( (\text{high}) \) \land Region(\( R_2 \))

\[ MR_8 \]: Trip – duration\( (\text{large}) \) \land TimeStamp(\( T_1 \) to \( T_4 \)) \( \Rightarrow \) noOfTrips\( (\text{small}) \) \land Region(\( R_3 \), \( R_6 \))

\[ MR_{10} \]: Trip – duration\( (\text{large}) \) \land TimeStamp(\( T_1 \) to \( T_3 \)) \( \Rightarrow \) noOfTrips\( (\text{large}) \) \land Amount\( (\text{Medium}) \) \land Region(\( R_1 \), \( R_2 \), \( R_3 \))

\[ MR_{11} \]: Trip – duration\( (\text{large}) \) \land TimeStamp(\( T_4 \) to \( T_3 \)) \( \Rightarrow \) noOfTrips\( (\text{large}) \) \land Region(\( R_6 \), \( R_4 \))

\[ MR_{12} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_4 \), \( R_6 \))

\[ MR_{13} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_1 \), \( R_4 \), \( R_6 \))

\[ MR_{14} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_1 \), \( R_2 \), \( R_3 \), \( R_6 \))

\[ MR_{15} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_1 \), \( R_2 \), \( R_3 \), \( R_4 \), \( R_5 \), \( R_6 \))

\[ MR_{16} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_1 \), \( R_2 \), \( R_3 \), \( R_4 \), \( R_5 \), \( R_6 \), \( R_7 \), \( R_8 \))

\[ MR_{17} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_1 \), \( R_2 \), \( R_3 \), \( R_4 \), \( R_5 \), \( R_6 \), \( R_7 \), \( R_8 \), \( R_9 \), \( R_{10} \))

\[ MR_{18} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_1 \), \( R_2 \), \( R_3 \), \( R_4 \), \( R_5 \), \( R_6 \), \( R_7 \), \( R_8 \), \( R_9 \), \( R_{10} \), \( R_{11} \))

\[ MR_{19} \]: noOfTrips\( (\text{medium}) \) \land Amount\( (\text{High}) \) \land Region(\( R_1 \), \( R_2 \), \( R_3 \), \( R_4 \), \( R_5 \), \( R_6 \), \( R_7 \), \( R_8 \), \( R_9 \), \( R_{10} \), \( R_{11} \), \( R_{12} \))

of the city.

Table 8: RMSE Comparison (travel demand [passenger count + number of originating trips]) with baselines

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HA</th>
<th>GBDT</th>
<th>SVM</th>
<th>Linear Regression</th>
<th>ARIMA</th>
<th>VAR</th>
<th>ST-ANN</th>
<th>DeepST</th>
<th>ST-ResNet</th>
<th>MARIO</th>
</tr>
</thead>
</table>

Table 9: RMSE Comparison of moving mobility events with baseline methods

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HA</th>
<th>GBDT</th>
<th>SVM</th>
<th>Linear Regression</th>
<th>ARIMA</th>
<th>VAR</th>
<th>ST-ANN</th>
<th>DeepST</th>
<th>ST-ResNet</th>
<th>MARIO</th>
</tr>
</thead>
</table>
strate the RMSE values for travel demand and moving mobility events for MARIO and five baselines. Figure 8 and Figure 9 show the MAPE values for moving and pick-up events. It has been observed that MARIO achieves a lowest 7.016 and 0.1251 RMSE and MAPE respectively. Amongst all the baselines based on deep-learning, DeepST and ST-ResNet show better performances. Next, we manually select 10 days from all three datasets when specific events (accident, road-blockage, crowd due to social event) occurred and evaluate the travel demand in different places of the cities. The results of the deep learning models are shown in fig. 10 and fig. 11. It is observed that MARIO performs significantly better than other deep learning models in these 10 days when any event occurs. The key reason of this performance is that MARIO not only models the spatial and temporal travel demand patterns, it is also capable to model the variation of travel demand occurring due to some events.

MARIO provides significant reduction in RMSE and MAPE over the baseline methods. It may be noted that the existing methods fall short especially in predicting moving events, however MARIO provides significantly better results both in pick-up and mobility event prediction.

5.5. Discussions

1. This work aims to extract mobility association rules from taxi-trips of a city. These rules consist of underlying mobility patterns of a city road network, namely, travel demand of a region (say, residential or commercial) in different time-slots of a day, taxi-drivers’ movement behaviours and finally how the variation of travel demand affects other regions. These rules are effective for early prediction of travel demand spikes such that several ridesharing companies, individual taxi-drivers can schedule the trips and as well as to provide a better transportation to citizens.

2. The key challenges to extract mobility association rules are two-folds, (i) Firstly the statistical analysis to predict travel demand in near future is time and computation extensive, since it requires a huge amount of taxi trips to be analyzed. (ii) Further, this analysis may fall short to extract underlying dynamics of the mobility features. Mobility association rules are extracted from historical log and utilized for demand prediction without the need to analyze enormous amount of taxi-trips. Moreover, the rules are capable to reflect mobility features of a typical city network. Further, the indexing scheme of MARIO has outperformed other baseline methods in terms of quick access of time-series data [refer Figure 6 and Figure 7].

3. Categorization of taxi-drivers has been done based on their mobility behaviours. For example, few taxi drivers cover only small (trip-length) trips, while others take long trips from airport to residential area and vice versa. These varied mobility behaviours also effect the total amount of payment in a day. These analysis may help to show which types of trips in a given region and timestamps are useful to reduce vacant time (no passenger) of a taxi.

4. The spatio-temporal neighboring effect is crucial to predict any variation of travel demand and re-route the trips. Our proposed method finds out the interrelationships of neighboring areas using a deep LSTM architecture [refer Figure 4] and thus outperforms other existing methods to predict travel demand and variation of travel demands. The proposed framework has achieved better RMSE value, MAPE values [refer Figure 8 and Figure 9] for predicting total passenger count, number of originating trips in a region and how travel demand variation of one region impacts other neighboring regions.

6. Conclusions and Future Work

Exploring city dynamics from mobility traces facilitates several location based services such as traffic resource management and improved route planning. The mobility association rules play an important role to summarize such travel patterns, variation of travel demands and finally modelling the overall movement dynamics. However, extracting mobility association rules are challenging for time and extensive computation complexity. In this work, we present a mobility-rule miner framework named MARIO which is capable to extract spatio-temporal mobility association rules from taxi-trip dataset. Furthermore, with the help of the proposed methods, we discover mobility patterns and finally predict the spatiotemporal distribution of travel-demands for different functional regions of a city. The mobility association rules predict the travel demand of different regions of a city which may improve taxi drivers’ profits and passengers’ travel experience. MARIO is not only limited to find interesting patterns from trajectory database, but incorporates how the effect of any mobility event evolve over longer periods and on different spatial scale. The experimental evaluations on three real-life taxi traces yield the efficacy of MARIO. We strongly believe that our framework can be utilized as an end-to-end system to model mobility dynamics of a city and subsequently predicting travel demands. MARIO has outperformed other existing methods for predicting travel demands in terms of RMSE and MAPE measures significantly.

In future, we will extend MARIO by incorporating other contextual information such as weather information to model the effects of these features on overall mobility dynamics. Furthermore, we will aim to develop an intelligent trip planner using the MARIO framework for resolving users’ queries. It has been shown that the proposed approach may facilitate the development of a context-aware trip recommendation system.

References
