

Mobility Data Warehousing and Mining

Gerasimos Marketos
Supervised by: Yannis Theodoridis
Information Systems Laboratory
Department of Informatics
University of Piraeus, Greece
<http://infolab.cs.unipi.gr>
marketos@unipi.gr

ABSTRACT

The usage of location aware devices, such as mobile phones and GPS-enabled devices, is widely spread nowadays, allowing access to large spatiotemporal datasets. The space-time nature of this kind of data results in the generation of huge amounts of mobility data and imposes new challenges regarding the analytical tools that can be used for transforming raw data to knowledge. In our research, we investigate the extension of Data Warehousing and data mining technology so as to be applicable on mobility data. In this paper, we present the, so far, developed framework for analyzing mobility data and some preliminary results.

1. INTRODUCTION

The flow of data generated from low-cost modern sensing technologies and wireless telecommunication devices enables novel research fields related to the management of this new kind of data and the implementation of appropriate analytics for knowledge extraction. The analysis of such mobility data raises opportunities for discovering behavioral patterns that can be exploited in applications like mobile marketing, traffic management etc.

Online analytical processing (OLAP) and data mining (DM) techniques can be employed in order to convert this vast amount of raw data into useful knowledge. Their application on conventional data has been extensively studied during the last decade. The high volume of generated mobility data arises the challenge of applying analytical techniques on such data. In order to achieve this aim, we have to take into consideration the complex nature of spatiotemporal data and thus to extend appropriately the two aforementioned techniques to handle them in an efficient way.

Towards this direction, we provide two motivation scenarios. Firstly, let us consider an advertising company which is interested in analyzing mobility data in different areas of a city so as to decide upon road advertisements (placed on panels on the roads). They are interested in analyzing the demographical profiles of the people visiting different urban areas of the city at different time zones of the day so as to decide about the proper sequence of advertisements that will appear on the panels at different time

periods. This knowledge will enable them to execute more focused marketing campaigns and apply a more effective strategy.

Indicatively, a Trajectory Data Warehouse (TDW) can serve this aim by analyzing various measures such as the number of moving objects in different urban areas, the average speed of vehicles, the ups and downs of vehicles' speed as well as useful insights, like discovering popular movements.

Secondly, trying to understand, manage and predict the traffic phenomenon in a city is both interesting and useful. For instance, city authorities, by studying the traffic flow, would be able to improve traffic conditions, to react effectively in case of some traffic problems and to arrange the construction of new roads, the extension of existing ones, and the placement of traffic lights.

The above targets can be served by analyzing traffic data so as to monitor the traffic flow and thus to discover traffic related patterns. These patterns can be expressed through relationships among the road segments of the city network. In other words, we aim to discover, by using aggregated mobility data, how the traffic flows in this network, the road segments that contribute to the flow and how this happens.

In order to realize the two above scenarios, but also many others, we work on a framework for Mobility Data Warehousing and Mining that takes into consideration the complete flow of tasks required for the development of a TDW and the application of trajectory-inspired mining algorithms so as to extract traffic patterns.

The rest of the paper is organized as follows. Section 2 presents the related work in the area of trajectory warehousing and mining. Section 3 constitutes the core of the paper, where we present the different components of the framework we have developed. Conclusions are outlined in Section 4.

2. RELATED WORK

2.1 Warehousing spatial and mobility data

The pioneering work by Han et al. [6] introduces the concept of spatial data warehousing (SDW). The authors extend the idea of cube dimensions so as to include spatial and non-spatial ones, and of cube measures so as to represent space regions and/or calculate numerical data. In [15], spatial OLAP operators are studied. One step further from modeling a SDW is modeling a TDW.

Trajectory warehousing [17] is in its infancy but we can distinguish three major research directions on this field: modeling, aggregation and indexing. From a modeling perspective, the definition of hierarchies in the spatial dimension introduces issues that should be addressed. The spatial dimension may include not explicitly defined hierarchies [7]. Thus, multiple aggregation

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Database Endowment. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires a fee and/or special permissions from the publisher, ACM.

VLDB '09 PhD Workshop, August 24, 2009, Lyon, France.

Copyright 2009 VLDB Endowment, ACM 000-0-00000-000-0/00/00.

paths are possible and they should be taken into consideration during OLAP operations. Tao and Papadias [22] propose the integration of spatial and temporal dimensions and present appropriate data structures that integrate spatiotemporal indexing with pre-aggregation. Choi et al. [2] try to overcome the limitations of multi-tree structures by introducing a new index structure that combines the benefits of Quadrees and Grid files. However, the above frameworks focus on calculating simple measures (e.g. count customers).

Furthermore, an attempt to model and maintain a TDW is presented in [14] where a simple data cube consisting of spatial / temporal dimensions and numeric measures concerning trajectories is defined. In our research, we investigate efficient solutions to support complex measures and to define the complete flow of processes in a TDW.

2.2 Mining patterns from mobility data

In [11], a distributed traffic stream mining system is proposed: the central server performs the mining tasks and ships the discovered patterns back to the sensors, whereas the sensors monitor whether the incoming traffic violates the patterns extracted from the historical data. This work emphasizes on the description of the distributed traffic stream system, rather on the discovery of traffic related patterns.

Also, relative to our research is the work by [10] for the discovery of hot routes (sequences of road segments with heavy traffic) in a road network. The authors propose a density-based algorithm, called FlowScan, which cluster road segments based on the density of the common traffic they share. The algorithm, however, requires the trajectories of the objects that move within the network, thus cannot be applied in our problem settings (as we already mentioned and will be further explained in Section 3.4, we assume aggregated mobility data and not the trajectories of each object).

A line of research relevant to our work is that of *spatiotemporal trajectory clustering* that aims at grouping trajectories of moving objects into groups of similar trajectories.

Lee et al. [9] propose a partition-and-group framework for trajectory clustering. Similar line segments are grouped into a cluster using a density based clustering method. For each cluster, the representative trajectory is discovered which is defined as the trajectory describing the overall movement of the trajectory partitions that belong to the same cluster. This work concerns the trajectories of the moving objects, free movement and no some predefined network like, in our case, the road network.

Giannotti et al. [4] propose the notion of trajectory patterns (T-patterns) and introduce appropriate trajectory mining algorithms for their discovery. Trajectory patterns represent sequences of spatial areas of interest that are temporally related. Such areas of interest can be predefined by the user or they can be discovered in a dynamic way using some density-based algorithm.

Kalnis et al. [8] introduce the notion of moving clusters for discovering groups of objects that move close to each other for a long time interval. However, their method requires the IDs of the objects and considers unconstrained environments.

Also, relevant to our work is the work on *change detection*. For example, the MONIC framework has been proposed [20] for modeling and detecting transitions between clusters discovered at

consequent time points. However, their method relies on cluster members (IDs of the objects), thus cannot be directly applied to our problem settings.

Nakata and Takeuchi [21] employ probe-car data for collecting traffic information concerning much larger areas than by traditional fixed sensors. They model traffic time as time series and they apply the Auto Regression Model after removing periodic patterns. However, in this work spatial information is not taken into consideration.

3. RESEARCH AGENDA & PRELIMINARY RESULTS

Our proposed framework for Mobility Data Warehousing and Mining (MDWM) consists of various components (actually, KDD steps) which are illustrated in Figure 1. Below, we present these components accompanied by our contributions:

- First, sampled positions received by GPS-enabled devices need to be converted into trajectory data and to be stored in a MOD; to this end, we propose a *trajectory reconstruction* technique that transforms sequences of raw sample points into meaningful trajectories.
- Second, the TDW is to be fed with aggregate trajectory data; to achieve it we propose two alternative solutions: a (index-based) *cell-oriented* and a (non-index-based) *trajectory-oriented* ETL process.
- Third, aggregation capabilities over measures are offered for OLAP purposes. The peculiarity with trajectory data is that a trajectory might span multiple base cells (the so called *distinct count problem* [23]). This causes *aggregation hindrances* in OLAP operations. We provide approximation solutions for this problem, which turn out to perform effectively.
- Fourth, our framework provides mining capabilities over mobility data (generated from vehicles) that are stored in MOD. We focus on the detection of traffic patterns and we propose algorithms for the detection of traffic relationships between the different road segments of a city network.

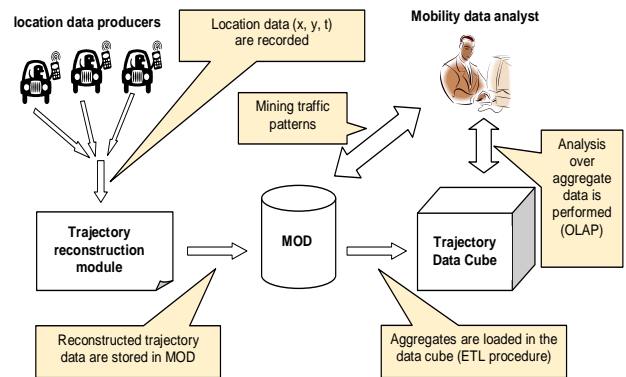


Figure 1. The architecture of our MDWM framework.

3.1 From raw locations to trajectories: the trajectory reconstruction problem

As already discussed, collected raw data represent time-stamped geographical locations (Figure 2a). Apart from storing these raw

data in the MOD, we are also interested in reconstructing trajectories (Figure 2b). The so-called *trajectory reconstruction* task is not a straightforward procedure. Having in mind that raw points arrive in bulk sets, we need a filter that decides if the new series of data is to be *appended* to an existing trajectory or not.

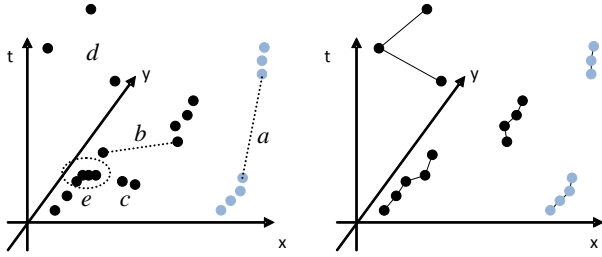


Figure 2. a) raw locations, b) reconstructed trajectories.

In [12], we proposed a method for determining different trajectories. The proposed trajectory reconstruction algorithm employs the idea of a filter based on appropriate parameters. The input of the algorithm includes raw data points (i.e., time-stamped positions) along with object-id, and a list containing the partial trajectories processed so far by the *trajectory reconstruction module*; these partial trajectories are composed by several of the most recent trajectory points, depending on the values of these parameters. Due to the fact that the notion of trajectory cannot be the same in every application, we define the following generic trajectory reconstruction parameters:

- *Temporal gap between trajectories* gap_{time} : the maximum allowed time interval between two consecutive time-stamped positions of the same trajectory for a single moving object (case a in Figure 2a).
- *Spatial gap between trajectories* gap_{space} : the maximum allowed distance in 2D plane between two consecutive time-stamped positions of the same trajectory (case b in Figure 2a).
- *Maximum speed* V_{max} : the maximum allowed speed of a moving object. When a new time-stamped location of object o_i is received, it is checked with respect to the last known position of that object, and the corresponding instant speed is calculated. If it exceeds V_{max} , this location is considered as noise and (temporarily) it is not considered in the trajectory reconstruction process (however, it is kept separately as it may turn out to be useful again – see the parameter that follows) (case c in Figure 2a).
- *Maximum noise duration* $noise_{max}$: the maximum duration of a noisy part of a trajectory. For example, consider an application recording positions of pedestrians where the maximum speed set for a pedestrian is $V_{max} = 3$ m/sec. When he/she picks up a transportation mean (e.g., a bus), the recorded instant speed will exceed V_{max} , flagging the positions on the bus as noise. The maximum noise length parameter stands for supporting this scenario: when the duration of this sequence of ‘noise’ exceeds $noise_{max}$, a new trajectory containing all these positions is created (case d in Figure 2a).
- *Tolerance distance* D_{tol} : the tolerance of the transmitted time-stamped positions. In other words, it is the maximum distance between two consecutive time-stamped positions of the same object in order for the object to be considered as stationary (case e in Figure 2a).

Figure 3 illustrates the efficiency of our trajectory reconstruction technique. It is clear that our algorithm performs linear with the size of the input dataset (and allows the processing of the full dataset in about 2 min). Furthermore, the average processing rate is almost stable ($\sim 50K$ records/sec). The complete evaluation study can be found in [12].

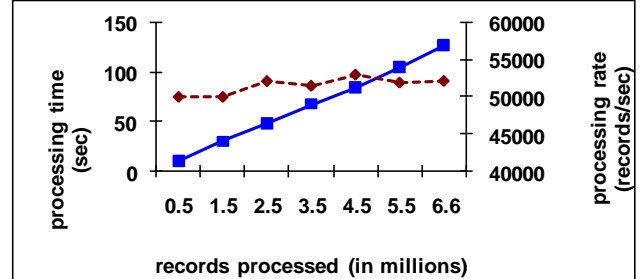


Figure 3. Performance of trajectory reconstruction (solid line: processing time; dotted line: processing rate)

Our ongoing work on this topic explores intelligent ways to automatically extract proper values of trajectory reconstruction parameters according to a number of characteristics of datasets. Furthermore, we are interested in extending this technique so as to be able to identify different movement types (pedestrian, bicycle, motorbike, car, truck etc) so as to apply customized trajectory reconstruction.

3.2 Trajectory data cube design and the ETL process

In [17], we investigated the prerequisites and the constraints imposed when describing the design of a TDW from a user perspective (i.e. conceptual model), as well as when describing the final application as a system in a platform-independent tool (i.e. logical model).

Following the multidimensional model [1], a data cube for trajectories consists of a fact table containing keys to dimension tables and a number of appropriate measures. Dimension tables might have several attributes in order to build multiple hierarchies so as to support OLAP analysis whereas measures could be trajectory-oriented (e.g., number of trajectories, number of objects, average speed, etc.). For each dimension we define a finest level of granularity which refers to the detail of the data stored in the fact table.

Definitely, a TDW should include a *spatial* and a *temporal dimension* describing geography and time, respectively. Another dimension regarding *conventional* information about moving objects (including demographical information, such as gender, age, occupation etc.) could be considered as well.

Based on the above, we highlight the following dimensions and measures (the corresponding star schema is illustrated in Figure 4) [12]:

- *Geography*: the spatial dimension (SPACE_DIM) allows us to define spatial hierarchies. Handling geography at the finest level of granularity could include (as alternative solutions) a simple grid, a road network or even coverage of the space with respect to the mobile cell network. According to the first alternative, the space is divided in explicitly defined (usually, rectangular) areas.

- *Time*: the temporal dimension (TIME_DIM) defines temporal hierarchies. Time dimension has been extensively studied in the data warehousing literature [1]. At the finest level of granularity, we assume user-defined time intervals.
- *User Profile*: the thematic dimension (OBJECT_PROFILE_DIM) refers to demographic and technographic information.

Apart from keys to dimension tables, the fact table also contains a set of measures including aggregate information. The measures considered in the TDW schema of Figure 4 include the *number of distinct trajectories* (COUNT_TRAJECTORIES), the *number of distinct users* (COUNT_USERS), the *average traveled distance* (AVG_DISTANCE_TRAVELED), the *average travel duration* (AVG_TRAVEL_DURATION), the *average speed* (AVG_SPEED) and the *average acceleration* in absolute values (AVG_ABS_ACCELER), for a particular group of people moving in a specific spatial area during a specific time period.

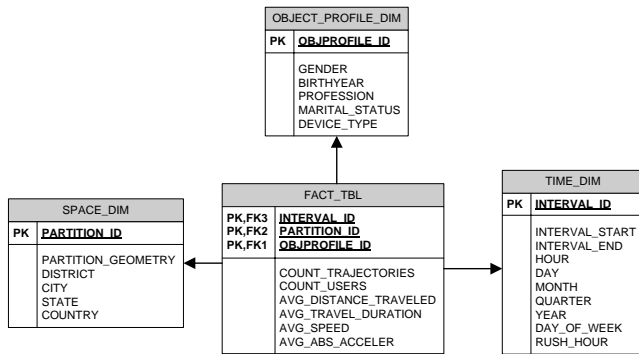


Figure 4. An example of TDW.

After defining the schema of the TDW, we have to consider ETL issues: load trajectories stored in the MOD and feed the TDW. Loading data into the dimension tables is straightforward; however, this is far more complex for the fact table. In particular, the main task is to fill in the measures with the appropriate numeric values for each of the base cells that are identified by the foreign keys of the fact table.

As already mentioned, in order to calculate the measures of the data cube, we have to extract the portions of the trajectories that fit into the base cells of the cube. We proposed two alternative solutions to this problem: (i) a cell-oriented and (ii) a trajectory-oriented approach in [12].

According to the *cell-oriented approach* (COA), we search for the trajectory portions that lie within the base cells. First, we search for the portions of trajectories under the concurrent constraint that they reside inside a spatiotemporal cell C . The efficiency of the above described COA solution depends on the effective computation of the parts of the moving object trajectories that reside in the spatiotemporal cells. This step is actually a spatiotemporal range query that returns not only the identifiers but also the portions of trajectories that satisfy the range constraints. To efficiently support this trajectory-based query processing requirement, we employ the TB-tree [18], a state-of-the-art index for trajectories that can efficiently support trajectory query processing.

According to the *trajectory-oriented approach* (TOA), we discover the spatiotemporal cells where each trajectory resides in. In order to avoid checking all cells, we use a rough approximation

of the trajectory, its Minimum Bounding Rectangle (MBR), and we exploit the fact that the granularity of cells is fixed in order to detect (possibly) involved cells in constant time. Then, we identify the portions of the trajectory that fits into each of those cells.

For the evaluation of the ETL process we compared the performance of the TOA vs. the index-based COA approaches. Both approaches were implemented in Hermes, a prototype MOD engine [16]. We used a large real dataset: a part of the e-Courier dataset [3] consisting of 6.67 millions of raw location records (a file of 504 Mb, in total), that represent the movement of 84 couriers moving in London (covered area 66,800 km²) during a one month period with a 10 sec sample rate. For all the experiments we used a PC with 1 Gb RAM and P4 3 GHz CPU.

We used two different granularities to partition the spatial and the temporal hierarchies; a spatial grid of equally sized squares of 10×10 Km² (100×100 Km², respectively) and a time interval of one (six, respectively) hours. The results of the four cases are illustrated in Figure 5, where it is clear that the choice of a particular method is a trade-off between the selected granularity level and the number of trajectories.

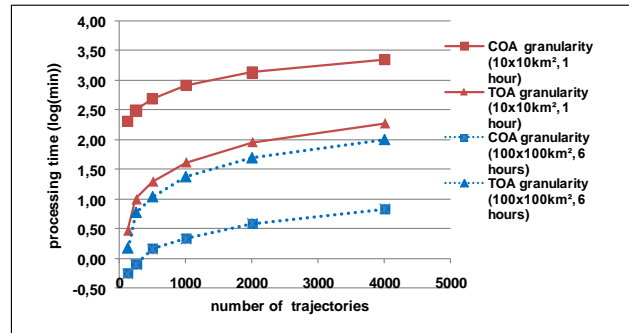


Figure 5. Comparison of alternative ETL processes

Our ongoing work on this topic includes the design of a trajectory data cube that will be more flexible in the sense of taking into consideration different semantic definitions of trajectories [19]. Let us consider, e.g., the case of a tracked user traveling from home to work in the morning, from work to the supermarket in the afternoon and, after a while, back to home. Different application scenarios may consider a different number of trajectories in the above example. Our approach proposed in [12] considers a specific semantic definition of trajectories that was fixed during the reconstruction stage. Hence, to achieve flexibility, we revisit basic components of the data warehouse (fact table, dimensions, materialization etc) so as to build a system suitable for ad-hoc analysis on trajectory data.

3.3 Trajectory-oriented OLAP and the distinct count problem

During the ETL process, measures can be computed in an accurate way by executing MOD queries based on the formulas provided in [12]. However, once the fact table has been fed, the trajectory and user identifiers are not maintained and only aggregate information is stored inside the TDW.

The aggregate functions to obtain super-aggregates for the main measures presented in Figure 4 are classified as holistic [5] and as such they require the MOD data to compute super-aggregates in all levels of dimensions. This is due to the fact that COUNT_USERS,

COUNT_TRAJECTORIES and, as a consequence, the other measures defined in terms of COUNT_TRAJECTORIES (e.g. the AVG measures) are subject to the distinct count problem [23]: if an object remains in the query region for several timestamps during the query interval, instead of counting this object once, it is counted multiple times in the result.

Notice that once a technique for rolling-up the COUNT_TRAJECTORIES measure is devised, it is straightforward to define a roll-up operation for the AVG measures. In fact the latter can be implemented as the sum of the corresponding auxiliary measures divided by the result of the roll-up of COUNT_TRAJECTORIES. As such, diminishing the calculations in the numerator, hereafter, we focus on the (denominator) number of distinct trajectories (COUNT_TRAJECTORIES); COUNT_USERS is handled in a similar way.

In order to implement a roll-up operation over this measure, a first solution is to define a distributive aggregate function [5] which simply obtains the super-aggregate of a cell C by summing up the measures COUNT_TRAJECTORIES in the base cells composing C . Following the proposal in [14], an alternative solution is to define an algebraic aggregate function. The idea is to store in the base cells a tuple of auxiliary measures that will help us to correct the errors caused due to the duplicates when rolling-up. Hence, for the base cell $C_{(x,y),t,p}$ we store:

- $C_{(x,y),t,p} \cdot \text{Traj}$: the number of distinct trajectories of profile p intersecting the cell.
- $C_{(x,y),t,p} \cdot \text{cross-x}$: the number of distinct trajectories of profile p crossing the *spatial* border between $C_{(x-1,y),t,p}$ (adjacent cell along with x - axis) and $C_{(x,y),t,p}$
- $C_{(x,y),t,p} \cdot \text{cross-y}$: the number of distinct trajectories of profile p crossing the *spatial* border between $C_{(x,y-1),t,p}$ (adjacent cell along with y - axis) and $C_{(x,y),t,p}$
- $C_{(x,y),t,p} \cdot \text{cross-t}$: the number of distinct trajectories of profile p crossing the *temporal* border between $C_{(x,y),t-1,p}$ (adjacent cell along with t - axis) and $C_{(x,y),t,p}$

Let $C_{(x',y'),t',p'}$ be a cell consisting of the union of two adjacent cells with respect to a spatial/temporal dimension, for example $C_{(x',y'),t',p'} = C_{(x,y),t,p} \cup C_{(x+1,y),t,p}$ (when aggregating along x - axis). In order to compute the super-aggregate corresponding to $C_{(x',y'),t',p'}$, we proceed as follows:

$$C_{(x',y'),t',p'} \cdot \text{Traj} = C_{(x,y),t,p} \cdot \text{Traj} + C_{(x+1,y),t,p} \cdot \text{Traj} - C_{(x+1,y),t,p} \cdot \text{cross-x}$$

The computation of $C_{(x',y'),t',p'} \cdot \text{Traj}$ can be thought of as an application of the well-known Inclusion/Exclusion principle for sets: $|A \cup B| = |A| + |B| - |A \cap B|$. However, it is worth noticing that the agility of a trajectory affects the error in the roll-up computation. Due to space limitations, we omit the experimental study of this approach, which can be found in [12].

We plan to extend the TDW so as to include both numerical and movement based measures. An example of such a measure is the so-called *typical trajectory* (e.g. [4], [9]) that describes the trend of movement within a cell. This is a rather challenging problem as it is not straightforward to derive the typical trajectory of a cell based on the typical trajectories of its sub-cells.

3.4 Trajectory-inspired mining: discovering traffic patterns

Detecting traffic relationships between the different road segments is an interesting problem. We consider a road network

modeled as a directed graph $G = (V, E)$ where the set V of vertices indicates locations (e.g. shopping centers, workplaces, crossings) and the set E of edges corresponds to direct connections (i.e., road segments) between them. We assume that aggregated mobility data are available and more specifically: for each edge, the traffic volume at adjacent time periods. These data can be derived either from sensors that are placed along the network and transmit traffic information at adjacent time periods or by GPS data that are map matched on edges and are aggregated at a specific temporal granularity. Both situations drive to time series that can be further analyzed so as to discover relationships among the different edges/road segments of the network.

The traffic series of a network edge e is defined as a time ordered sequence of traffic measurements: $TS = (v_i, t_i)$, where v_i is the number of cars crossing e during $[t_i, t_i + \Delta t]$. The parameter Δt is the transmission rate of the sensor and is common for all sensors in the network.

In [13], we defined various relationships between the edges of the network graph during specific time periods: *traffic propagation* from one edge to some other edge, *split* of traffic from one edge into multiple edges, *merge* of traffic from multiple edges into a single edge. We worked on efficient methods for the discovery of such traffic relationships which are based on appropriate distance measures defined on the time series of edges.

Let e_1, e_2 be two network edges and let $TS_1 = \{(v_{1i}, t_i)\}$, $TS_2 = \{(v_{2i}, t_i)\}$ be their corresponding traffic (time) series, $t_i \in [t_s, t_e]$. In [13] we proposed a distance between two traffic edges e_1, e_2 is given as a weighted combination of their corresponding value based, shape based and structure based distances:

$$\text{dis}(e_1, e_2) = a \cdot \text{dis}_{\text{shape}}(e_1, e_2) + b \cdot \text{dis}_{\text{struct}}(e_1, e_2) + c \cdot \text{dis}_{\text{value}}(e_1, e_2)$$

where $\text{dis}_{\text{value}}(e_1, e_2)$ is the *value based distance* between e_1 and e_2 which is given by the Euclidean distance of their corresponding traffic series (TS_1, TS_2), $\text{dis}_{\text{struct}}(e_1, e_2)$ is the *shape based distance* between e_1 and e_2 which is given by the Euclidean distance of their corresponding normalized (to avoid differences in baselines, scales) traffic series (TS'_1, TS'_2). Finally, $\text{dis}_{\text{shape}}(e_1, e_2)$ is the *structure based distance* between two traffic edges e_1 and e_2 and equals to the minimum number of edges between end(e_1) and start(e_2). For each application, we can instantiate the weights a, b, c according to the measure(s) on which we wish to emphasize.

If we consider the three measures separately, we realise that each measure further filters the initial set of traffic edges. More specifically, the *shape* based measure returns groups of edges with similar traffic shape, the *structure* measure looks further for neighbour edges, and finally the *value* measure further restricts the result set by searching also for value based similar traffic edges. This rationale provides a hierarchy of traffic flow organised in three levels: the level of similar traffic shape edges (L1), the level of similar traffic shape edges that are also nearby in the graph network (L2) and finally, the level of similar traffic value edges (L3).

To detect such a hierarchy, in [13] we proposed a divisive hierarchical clustering algorithm. The distance measure is that of equation of $\text{dis}(e_1, e_2)$, which combines the three notions of distance between traffic edges. The algorithm works as follows: Initially all traffic edges are placed in one cluster. At each step of the algorithm, a cluster is further split into subclusters according to the following three steps:

- Step 1 [Edges of similar shape]: A cluster is split into subclusters based on the shape similarity of its traffic edge members. This process is continued until a split is caused by the next distance measure, the structure based distance. At the end of this step, the clusters contain edges with similar traffic shape.
- Step 2 [Nearby edges]: The clusters generated by the previous step are further split based on the structural distance measure until a split is caused by the traffic values distance. At this moment, the clusters contain neighbouring traffic edges with similar shape.
- Step 3 [Edges of similar values]: The clusters generated by the previous step are further split based on the similar values distance. At the end of the execution, the clusters contain neighbouring edges with similar values, and similar shape as well.

Our ongoing work on this topic includes the enhancing of the above approach so as to discover time focused relationships (at specific periods). To achieve this, we redefine the *value based distance* as the absolute distance of their corresponding traffic series at these periods. Changes apply also to the *shape based distance*, where we evaluate different time series comparison techniques (DTW, correlation-coefficient etc). Finally, we extend the relationships so as to include traffic sink to an edge and traffic source from an edge.

4. CONCLUSIONS

In this paper, we provided a brief outline of the framework we propose for efficient and effective Mobility Data Warehousing and Mining. We described its different components and provided some preliminary results as well as hints about ongoing work per topic.

5. ACKNOWLEDGMENTS

Special thanks to the coauthors collaborated to our joint works. Research partially supported by the EU FP6-14915 IST/FET Project GeoPKDD (Geographic Privacy-aware Knowledge Discovery and Delivery) and by a PENED'2003 grant funded by the General Secretariat for Research and Technology of the Greek Ministry of Development.

6. REFERENCES

- [1] Agarwal, S., Agrawal, R., Deshpande, P., Gupta, A., Naughton, J., Ramakrishnan, R., and Sarawagi, S. On the computation of multidimensional aggregates. *Proc. VLDB*, 1996.
- [2] Choi, W., Kwon, D., and Lee, S. Spatio-temporal data warehouses using an adaptive cell-based approach. *DKE*, 59, 1 (Oct. 2006), 189-207.
- [3] eCourier.co.uk dataset, <http://api.ecourier.co.uk/>. (URL valid on June 20, 2009).
- [4] Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. Trajectory pattern mining. *Proc. KDD*, 2007.
- [5] Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *DMKD*, 1, 1 (Mar. 1997), 29-53.
- [6] Han, J., Stefanovic, N., and Koperski, K. Selective Materialization: An Efficient Method for Spatial Data Cube Construction. *Proc. PAKDD*, 1998.
- [7] Jensen, C.S., Kligys, A., Pedersen, T.B., Dyreson, C.E., and Timko, I. Multidimensional data modeling for location-based services. *VLDBJ*, 13 (Jan. 2004), 1–21.
- [8] Kalnis, P., Mamoulis, N., and Bakiras, S. On discovering moving clusters in spatio-temporal data. *Proc. SSTD*, 2005.
- [9] Lee, J., Han, J., and Whang, K. Trajectory Clustering: A Partition-and-Group Framework. *Proc. SIGMOD*, 2007.
- [10] Li, X., Han, J., Lee, J.-G., and Gonzalez, H. Traffic density-based discovery of hot routes in road networks. *Proc. SSTD*, 2007.
- [11] Liu, Y., Choudhary, A.N., Zhou, J., and Khokhar, A.A. A scalable distributed stream mining system for highway traffic data. *Proc. PKDD*, 2006.
- [12] Marketos, G., Frentzos, E., Ntoutsis, I., Pelekis, N., Raffaeta, A., and Theodoridis, Y. Building Real World Trajectory Warehouses. *Proc. MobiDE*, 2008.
- [13] Ntoutsis, I., Mitsou, N., and Marketos, G. Traffic mining in a road-network: How does the traffic flow? *IJBIDM*, 3, 1, (Apr. 2008), 82-98.
- [14] Orlando, S., Orsini, R., Raffaetà, A., Roncato, A., and Silvestri, C. Trajectory Data Warehouses: Design and Implementation Issues. *JCSE*, 1, 2 (Dec. 2007), 211-232.
- [15] Papadias, D., Kalnis, P., Zhang, J., and Tao, Y. Efficient OLAP Operations in Spatial Data Warehouses. *Proc. SSTD*, 2001.
- [16] Pelekis, N., Frentzos, E., Giatrakos, N. and Theodoridis, Y. HERMES: Aggregative LBS via a Trajectory DB Engine. *Proc. SIGMOD*, 2008.
- [17] Pelekis, N., Raffaetà, A., Damiani, M.-L., Vangenot, C., Marketos, G., Frentzos, E., Ntoutsis, I., and Theodoridis, Y. Towards Trajectory Data Warehouses. Chapter in *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*. Springer-Verlag. 2008.
- [18] Pfoser, D., Jensen, C.S., and Theodoridis, Y. Novel Approaches to the Indexing of Moving Object Trajectories. *Proc. VLDB*, 2000.
- [19] Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., and Vangenot, C. 2008. A conceptual view on trajectories. *DKE*, 65, 1 (Apr. 2008), 126-146.
- [20] Spiliopoulou, M., Ntoutsis, I., Theodoridis, Y., and Schult, R. Monic: modeling and monitoring cluster transitions. *Proc. KDD*, 2006.
- [21] Nakata, T., and Takeuchi, J. Mining traffic data from probe-car system for travel time prediction. *Proc. KDD*, 2004.
- [22] Tao, T., and Papadias, D. Historical Spatio-Temporal Aggregation. *Proc. TODS*, 2005.
- [23] Tao, Y., Kollios, G., Considine, J., Li, F., and Papadias, D. Spatio-Temporal Aggregation Using Sketches. *Proc. ICDE*, 2004.