International Journal of Computing and Business Research (IJCBR)
ISSN (Online) : 2229-6166
International Manuscript ID : 22296166V7I2201701
Volume 7 Issue 2 July – August 2017

Trajectory Data Warehouse and Satellite Based Technologies

Halgare Nanasaheb Mahadev
Research Scholar
Kalinga University, Chhattisgarh, India

Dr. Ali Akbar Bagwan
Research Supervisor
Kalinga University, Chhattisgarh, India

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 trajectory data and imposes new challenges regarding their efficient management. To address this need, the traditional database technology has been extended into Moving Object Databases (MODs) that handle modeling, indexing and query processing issues for trajectories. Moreover, the analysis of such trajectory data raises opportunities for discovering behavioral patterns that can be exploited in applications like traffic management and service accessibility. Online analytical processing (OLAP) and data mining (DM) techniques have been employed in order to convert this vast amount of raw data into useful knowledge. Indicatively, the variable 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 can be analyzed in a Trajectory Data Warehouse (TDW).

Keywords: Trajectory Data Warehouse, TDW, Design Perspectives in TDW

Introduction
The pioneering work by Han et al. 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. One step further from modeling a SDW is modeling a TDW. The motivation here is to transform raw trajectories to valuable information that can be utilized for decision making purposes in ubiquitous applications, such as mobile marketing, location-based services and traffic control management. Trajectory warehousing 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. Thus, multiple aggregation paths are possible and they should be taken into consideration during OLAP operations. Tao and Papadias propose the integration of spatial and temporal dimensions and present appropriate data structures that integrate spatiotemporal indexing with pre-aggregation. Choi et al. try to overcome the limitations of multi-tree structures by introducing a new index structure that combines the benefits of Quadtrees and Grid files. However, the above frameworks focus on calculating simple measures (e.g. count customers). Very recently, an attempt to model and maintain a TDW is presented in , where a simple data cube consisting of spatial / temporal dimensions and numeric measures concerning trajectories, is defined. The distinguishing features of our work are:

i) the presence of a preprocessing phase dealing with the explicit construction of the trajectories, which are then stored into a MOD that offers powerful and efficient operations for the manipulation of such data;
ii) the proposal of alternative ETL processes, a procedure underestimated so far in related work; and
iii) the solutions proposed on the challenging issue of measure aggregation which occurs due to the trajectory oriented cube model.

We emphasize that this work does not aim at proposing yet another TDW model. Instead, we provide efficient solutions to support the complete flow of processes in a TDW, from trajectory reconstruction to trajectory-oriented OLAP.

Reconstructing trajectories

As already discussed, collected raw data represent time-stamped geographical locations. Apart from storing these raw data in the MOD, we are also interested in reconstructing trajectories. 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.

In this work, we assume this filter to be part of a trajectory reconstruction manager, along with a simple method for determining different trajectories, which applies it on raw positions. 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 gaptime: the maximum allowed time interval between two consecutive time-stamped positions of the same trajectory for a single moving object. As such, any time-stamped position of object oi, received after more than gaptime time units from its last recorded position, will cause a new trajectory of the same object to be created.

Spatial gap between trajectories gapspace: the maximum allowed distance in 2D plane between two consecutive time-stamped positions of the same trajectory. As such, any time-stamped position of object oi, with distance from the last recorded position of this object greater than gapspace, will cause a new trajectory to be created for oi.

Maximum speed Vmax: the maximum allowed speed of a moving object. It is used in order to determine whether a reported time-stamped position must be considered as noise and consequently discarded from the output trajectory. When a new time-stamped location of object oi 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 Vmax, 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).

 Maximum noise duration noisemax: the maximum duration of a noisy part of a trajectory. Any sequence of noisy time-stamped positions of the same object will result in a new trajectory given that its duration exceeds noisemax. For example, consider an application recording positions of pedestrians where the maximum speed set for a pedestrian is Vmax = 3 m/sec. When he/she picks up a transportation mean (e.g., a bus), the recorded instant speed will exceed

Vmax, 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 noisemax, a new trajectory containing all these positions is created Tolerance distance Dtol: the tolerance of the transmitted timestamped 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. When a new time-stamped location of object oi is received, it is checked with respect to the last known position of that object, and if the distance of the two locations is smaller than Dtol, it is considered redundant and consequently discarded.

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 manager; these partial trajectories are composed by several of the most recent trajectory points, depending on the values of the algorithm parameters.

```
Algorithm TRC-Reconstruction
(PartialTrajectories List, P Point, OId ObjectId)
IF NOT PartialTrajectories.Contains(OId) THEN
    CTRC=New TRC;
    CTRC.AddPoint(P);
    PartialTrajectories.Add(CTRC);
ELSE
    CTRC=PartialTrajectories(OId);
    IF Distance(CTRC.LastPoint,P)<= DTOL THEN
        IF P.T – CTRC.LastPoint.T > gapTime THEN
            Report CTRC.LastPoint;
            CTRC.Id=CTRC.Id+1;
            CTRC.AddPoint(P);
        ENDIF
    ELSEIF Speed(CTRC.LastPoint,P)> Vmax THEN
        IF P.T – CTRC.LastPoint.T > noisemax THEN
            Report CTRC.Noise;
        ELSE
            CTRC.AddNoise(P);
        ENDIF
    ELSEIF Distance(CTRC.LastPoint,P)> gapspace THEN
        Report CTRC.LastPoint;
        CTRC.Id=CTRC.Id+1;
        CTRC.AddPoint(P);
    ELSEIF P.T – CTRC.LastPoint.T > gapTime THEN
        Report CTRC.LastPoint;
        CTRC.Id=CTRC.Id+1;
        CTRC.AddPoint(P);
    ELSE
        CTRC.AddPoint(P);
    ENDIF
ENDIF
```

Conclusions

In this paper, we propose solutions for the efficient and effective development of trajectory warehouses. To the best of our knowledge, this is the first work that supports all the required steps for building a TDW, from trajectory reconstruction and MOD loading, to data cube feeding and aggregating over summary information. More specifically, we proposed techniques for the solution of the trajectory reconstruction problem, for supporting ETL of trajectory data, and for addressing the problem of measure aggregation, giving particular attention to the distinct count problem. Our approach has been experimentally tested in a large real dataset and has been shown to be efficient. As part of our future work, we plan to examine new measures for the trajectory warehouse, specifically suited for trajectories. An example of such a measure is the so-called typical trajectory (e.g. ,) 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. In a similar line of research, other interesting measures could be investigated like the average direction measure of the trajectories within a cell. Finally, we plan to explore the analytical capabilities of the proposed framework by applying DM techniques over the aggregated data stored in the TDW.

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):189-207, 2006. eCourier.co.uk dataset, http://api.ecourier.co.uk/..

[3] Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. Trajectory pattern mining. Proc. KDD, 2007.

[4] Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. Data cube: A relational aggregation operator generalizing groub-by, crosstab and sub-totals. DMKD, 1(1):29-54, 1997.

[5] Güting, R.H., and Schneider, M. Moving Object Databases, Morgan Kaufman Publishers. 2005.

[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, The VLDB Journal, 13:1–21, 2004.

[8] Lee, J., Han, J., and Whang, K. Trajectory Clustering: A Partition-and-Group Framework. Proc. SIGMOD, 2007.

[9] Orlando, S., Orsini, R., Raffaetà, A., Roncato, A., and Silvestri, C. Spatio-Temporal Aggregations in Trajectory Data Warehouses. Proc. DaWaK, 2007.