Amnesic Online Synopses for Moving Objects

Michalis Potamias, Kostas Patroumpas, and Timos Sellis
School of Electrical and Computer Engineering
National Technical University of Athens, Hellas
{mpotamias, kpatro, timos}@dbnet.ece.ntua.gr

ABSTRACT

We present a hierarchical tree structure for online maintenance of time-decaying synopses over streaming data. We exemplify such an amnesic behavior over streams of locations taken from numerous moving objects in order to obtain reliable trajectory approximations as well as affordable estimates regarding distinct count spatiotemporal queries.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications

General Terms: Algorithms, Management, Performance

Keywords: Compression, Data Streams, Moving Objects, Trajectories, Continuous Queries

1. INTRODUCTION

Recent advances in telecommunications and geopositioning facilities enable online collection of locations from many moving sources (e.g., humans, vehicles). Thus, streams of positional updates are created, particularly useful in monitoring applications ranging from traffic surveillance to environmental protection. Apart from ever-increasing storage requirements, this rapidly accumulating data should be processed in a timely fashion for providing real-time response to multiple continuous user requests.

It is apparent that the significance of each isolated location is inherently *time-decaying*, since any recorded position of an object will be soon outdated by forthcoming ones. This motivates the need for an *amnesic* treatment of the positional updates gathered for each object: we argue that the older a data item gets, the coarser its representation could become in a progressive fashion, implying that greater precision should be reserved for the most recent positions. With respect to approximation of one-dimensional time series, a wide range of amnesic functions has been identified in [2], useful in controlling the amount of error tolerated for every single point in the time series. In addition, we think that different levels of abstraction are inherent in semantics related to multi-scale representation of spatiotemporal features, allowing progressive refinements of their evolution at additional cost in storage consumption and processing time.

In this work, we present the *multiple-granularity* AmTree framework that accepts streaming items and maintains summaries over *hierarchically* organized levels of precision, essentially realizing an amnesic behavior over stream portions.

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.
ACM 1-59593-433-2/06/0011.

This time-aware scheme distantly resembles SWAT [1], but it is considerably more generic, since it is not intrinsically bound to wavelet transform of scalar stream values. In contrast, AmTree can even deal with multi-dimensional points with user-specified approximation functions.

The proposed mechanism can efficiently handle streaming locations from moving objects and retain a compressed outline of their entire trajectories, always preserving *contiguity* among successive segments for each individual object. In conjunction with FM sketches, AmTree can further be used in spatiotemporal aggregation for providing good-quality estimates to distinct count queries over locations of moving objects.

2. THE AMNESIC TREE FRAMEWORK

The general structure of an AmTree is illustrated in Figure 1. For ease of presentation, we opt for a summarization scheme that manipulates pairs of items at every level of the amnesic tree structure, so we assume that a *time granule* at each level spans two granules half its size at the level beneath. However, AmTree is much more general and can be easily calibrated to work with a varying number of nodes (i.e., time granules) at each level.

Except for the root, each level i of the tree consists of a right (R_i) and a left node (L_i). At the lowest (0^{th}) level, node R_0 accepts data with reference to the finest time granularity (e.g., seconds), which characterizes every timestamp attached to incoming tuples. Each node at the i^{th} level contains information about twice as many timestamps as a node at the $(i-1)^{th}$ level. Hence, a node at level i contains information characterizing 2^i timestamps.

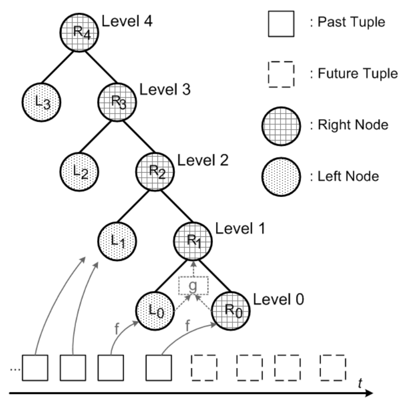

Figure 1: Basic structure of AmTree.

We consider a user-defined mapping f that is applied over the batch of tuples with current timestamp value τ, and transforms them into a single tuple that can become the content of a tree node (operation $new(\tau)$). As illustrated in Figure 1, the resulting content is assigned to node R_0, while the previous content of R_0 is shifted to node L_0 (operation $shift(i)$). As time goes by and new data comes in, the contents of each level are combined using a function g and propagated higher up in the tree, retaining less detail (operation $merge(i)$). Note that node R_0 is the only entry point to the synopsis maintained by the AmTree.

When this structure consumes streaming items, node updates are performed in a bottom-up fashion. It can be easily shown that each level i is updated every 2^i timestamps, where $i = 0, 1, 2, \ldots$ Each time, the update procedure reaches a maximum level M of the tree, which depends on the timestamp value of the incoming item. Therefore, AmTree updates can be carried out online in $O(1)$ amortized time per tuple with only logarithmic requirements in memory storage.

3. APPLICATION ON MOVING OBJECTS

We believe that AmTree is best suited for summarizing streams of sequential features, i.e., time series that must retain contiguity among their consecutive elements. This is exactly the case of streaming locations collected from moving objects. Next, we describe amnesic synopses concerning singleton trajectories, as well as a sketch-based variant of AmTree for computing spatiotemporal aggregates.

3.1 Maintaining Trajectory Synopses

As an alternative representation to a time series of points, the trajectory of a moving object can be represented with a polyline composed of consecutive *displacements*. Every such line segment connects a pair of successive point locations recorded for this object, eventually providing a continuous, though approximate, trace of its movement.

With respect to compressing a single trajectory, we suggest an instantiation of AmTree that manipulates all successive displacement tuples recorded for this object. In direct correspondence to the generic AmTree functionality, mapping f converts each current object position into a displacement tuple taking into account its previous location. This displacement is then inserted into the R_0 node of AmTree, possibly triggering further updates at higher levels. When the contents of level i must be merged to produce a coarser representation, a simple *concatenation* function g is used to combine the successive displacements stored in nodes L_i and R_i. After eliminating the common articulation point of the two original segments, a single line segment is produced and then stored in node R_{i+1}. As a result, endpoints of all displacements stored in AmTree nodes correspond to original positional updates, while displacements remain connected to each other at every level. Evidently, an amnesic behavior is achieved for trajectory segments through levels of gradually less detail in this bottom-up tree maintenance.

As long as consecutive displacements are preserved, the movement of a particular object can be properly reconstructed by choosing points in descending temporal order, starting from its most recent position and going steadily backwards in time. Any trajectory reconstruction process can be gradually refined by combining information from multiple levels and nodes of the tree, leading to a *multi-resolution approximation* for a given trajectory.

3.2 Answering Distinct Count Queries

Next, we present a summarization technique that is able to provide unbiased estimates for the number of objects that move in an area of interest during a specified time interval. When each object must be counted only once, the problem is known as *distinct counting* [4].

We consider a regular decomposition of the 2-d Euclidean plane into equal-area cells, which are used to maintain a simplified spatial reference of moving objects instead of their actual locations. To accommodate temporal extents, we make use of a hierarchical AmTree. Thus, each cell points to an AmTree, which maintains gradually aging information concerning the number of moving objects inside that cell. Query-oriented compression is achieved using FM_PCSA sketches. Each node of an AmTree corresponds to m bitmap vectors utilized by the FM_PCSA sketch. Hence, we avoid enumeration of objects, as we are satisfied with an acceptable estimate of their distinct count given by the sketching algorithm.

In order to estimate the number of distinct objects moving within a given area α during a time interval $\Delta\tau$, we first identify the grid cells that completely cover region α. Those cells determine the group of qualifying AmTree structures that maintain the aggregates. For each such tree, we need to locate the set of nodes that overlap time period $\Delta\tau$ specified by the query; these nodes are identical for each qualifying tree. By taking the union of the sketches attached to these nodes (i.e., an OR operation over the respective bitmaps), we finally compute an approximate answer to our query.

4. DISCUSSION

We conducted a series of experiments confirming the approximation quality and processing performance achieved by our approach. It was verified that recent trajectory segments always remain more accurate, while overall error largely depends on the temporal extent of the query in the past. Even for heavily compressed trajectories, accuracy is proven quite satisfactory for answering spatiotemporal range queries. With respect to distinct count queries, it was observed that a finer grid partitioning incurs more processing time at the expense of increased accuracy.

We believe that the suggested AmTree framework is a modular amnesic structure with exponential decay characteristics, especially tailored to cope with streaming positional updates. As further explained in [3], instead of just computing aggregate statistics, our particular concern is to maintain age-biased synopses that are able to provide reduced approximations. We plan to adjust amnesic behavior with user-specified aging patterns and study more complex schemes of multiple, concurrently updated AmTrees.

5. REFERENCES

[1] A. Bulut and A.K. Singh. SWAT: Hierarchical Stream Summarization in Large Networks. In *ICDE*, pp. 303-314, March 2003.

[2] T. Palpanas, M. Vlachos, E. Keogh, D. Gunopulos, and W. Truppel. Online Amnesic Approximation of Streaming Time Series. In *ICDE*, pp. 338-349, March 2004.

[3] M. Potamias, K. Patroumpas, and T. Sellis. Online Amnesic Summarization of Streaming Locations. Technical Report, NTUA, June 2006.

[4] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-Temporal Aggregation Using Sketches. In *ICDE*, pp. 214-226, March 2004.