SPATIO-TEMPORAL DATA MINING FROM MOVING OBJECT TRAJECTORIES

Yoshiharu Ishikawa

Information Technology Center, Nagoya University
ishikawa@itc.nagoya-u.ac.jp

ABSTRACT

We review our current research on spatio-temporal data mining from moving object trajectories. With the recent progress of spatial information technologies and communication technologies, it has become easier to track positions of a large number of moving objects in real-time. Our research objective is to extract mobility statistics for their effective utilization in an efficient manner.

1. INTRODUCTION

Owing to the development of mobile computing technologies, GPS and positioning devices, and wireless network facilities, tracking the positions of moving objects [1] has become relatively easier today. The collected trajectory data can be used to monitor the behavior of moving objects and to analyze their movement patterns. The latter activity is often called *mobility analysis*. Based on mobility analysis, we can observe an overall movement tendency and predict future movement patterns. Analysis of movement patterns are also important for query processing in spatio-temporal databases that store trajectories of moving objects.

Our approach is based on the *Markov chain model* [2]. In mobility analysis, the Markov chain model is used to describe a spatio-temporal movement tendency between spatial areas under the assumption that the movement patterns can be described by Markov chains. The model has been used in the analysis of various kinds of movement data such as car traffic and demographic transition. However, to our knowledge, the utilization of the model has not been investigated in the context of large-scale spatio-temporal data mining.

2. THE MARKOV CHAIN MOBILITY MODEL

Consider that a target two-dimensional space is partitioned into spatial *regions* as shown in Fig. 1. Each dimension is equally divided in 2^P ranges such that $R = 2^{2P}$ regions in total. The figure shows the case of $P = 2$. We call a partitioning shown in the figure as *level-P partitioning*. For each region, a $2P$ bit region number that obeys the *Z-ordering method* is assigned. The figure shows that object A located in region 9 at $t = \tau$ moves to region 12 at $t = \tau + 1$ and then moves to region 6 at $t = \tau + 2$.

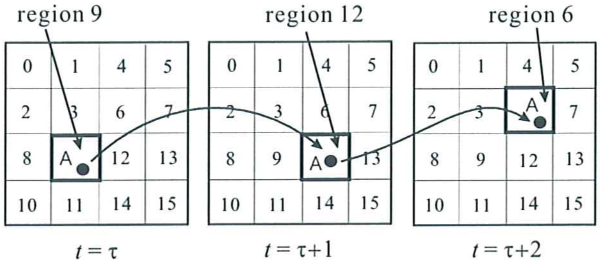

Fig. 1. Notion of Markov chain model

Suppose that another moving object B located in region 9 moves to region 12 in unit time later. Consider that we want to know the probability that object B moves to region 6 next; we denote the probability by $\Pr(6|9, 12)$. If we assume the transition between spatial regions obeys the *Markov chain model* [2], we can say that the probability is a second-order *Markov transition probability*. The idea can be easily generalized to an order-n Markov transition probability $\Pr(r_n|r_0, \ldots, r_{n-1})$. That is, the probability of an object, that traveled to regions $r_0, r_1, \ldots, r_{n-1}$ in this order at unit time will next visit region r_n.

The problem we tacke here is effecitive summarization of a large number of trajectories. Since trajectory data is often delivered continually as a form of *data stream*, we need to develop an efficient processing method which effectively works in real-time. Our approach is to construct a *mobility histogram* that approximately represents the distribution of movement patterns to reduce the data size and the computational cost in the analysis. In the next section, we describe its logical representation.

3. LOGICAL MOBILITY HISTOGRAM

We employ a *data cube* [3] as the logical representation of a mobility histogram. To represent order-n Markov chain-based statistics, a histogram is constructed as an $(n + 1)$-dimensional data cube. Figure 2 shows a sample of a data cube for $n = 2$ and $P = 1$. Since the two-dimensional target space is partitioned into $R = 2^{2P} = 4$ spatial regions, the data cube contains $R^{n+1} = 64$ cells. For each dimension of the data cube, steps 0, 1, and 2 corresponds to each step of a second-order Markov chain. For instance, when the sequence

$1 \rightarrow 1 \rightarrow 2$ is received from the transition sequence stream, the corresponding cell value is incremented.

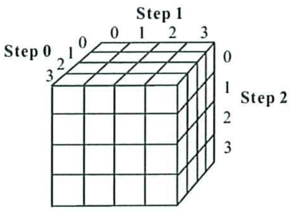

Fig. 2. Logical mobility histogram

When a statistics estimation query is given, the constructed mobility histogram is utilized. For example, the probability that an object that has moved from region 1 to region 2 and then moves to region 4 is calculated as $\Pr(4|1, 2) = val(1, 2, 4)/val(1, 2, *)$, where $val(1, 2, 4)$ is the value of the cube cell $(1, 2, 4)$ and $val(1, 2, *) = \sum_{i=0}^{2^P-1} val(1, 2, i)$.

4. PHYSICAL MOBILITY HISTOGRAM

The direct implementation of the logical data cube representation has a huge overhead. To reduce the size and enable dynamic maintenance of the histogram structure, we propose a tree-based physical representation.

Suppose that input transition sequences are based on level-M partitioning such as $2^{(M)} \rightarrow 10^{(M)} \rightarrow 12^{(M)}$, where the notation $r^{(M)}$ is used to specify the partitioning level of region r explicitly. Each node in a histogram tree has zero to four child nodes for the corresponding transition sequences. For example, consider an order-2 transition sequence $r_0^{(M)} \rightarrow r_1^{(M)} \rightarrow r_2^{(M)}$ is inserted in a tree, where $r_0^{(M)}, r_1^{(M)}$, and $r_2^{(M)}$ are region numbers. We call $r_0^{(M)}, r_1^{(M)}$, and $r_2^{(M)}$ step-0 region, step-1 region, and step-2 region, respectively. The sequence is processed as follows:

1. Start from the root of the tree.
2. Translate $r_0^{(M)}$ into a binary number, then extract its first two-bits. Depending on the value, that is, `00` ($= 0^{(1)}$), `01` ($= 1^{(1)}$), `10` ($= 2^{(1)}$), or `11` ($= 3^{(1)}$), follow the corresponding edge and then visit the child node. If the child node is not instantiated, create the node and an edge from the parent (root) to the node.
3. Extract the first two bits from $r_1^{(M)}$ and follow the corresponding edge.
4. Process $r_2^{(M)}$ in a similar manner. Steps 2 to 4 correspond to transitions in level-1 coarse resolution.
5. Use the next two bits from $r_0^{(M)}$, $r_1^{(M)}$, and $r_2^{(M)}$, respectively, to traverse following each edge. This step corresponds to level-2 partitioning.
6. Repeat these steps until all bits are consumed.

Note that there is a *counter* in each node; when we visit a node we increment its count.

Figure 3 shows an example to add a transition sequence $3^{(2)} \rightarrow 6^{(2)} \rightarrow 12^{(2)}$ for a tree with $M = 2$. A dotted edge indicates that a corresponding transition sequence has not yet arrived so that the edge is not allocated.

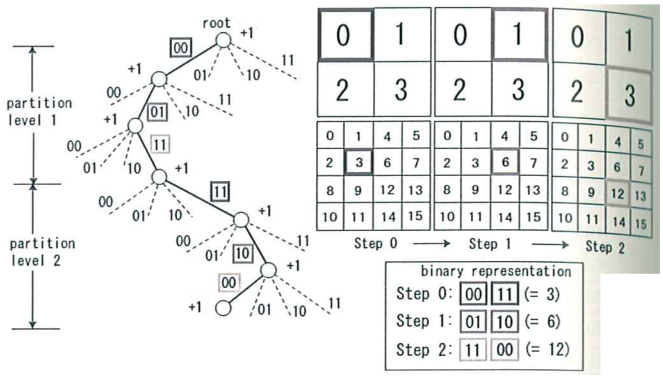

Fig. 3. Physical histogram structure

In addition to the above naive physical histogram structure, we have developed an approximated histogram construction method which achieves quite compact histogram representation with small degradation in precision. Further detail of our mobility histogram method and related spatio-temporal data mining research are covered in [4, 5, 6].

5. REFERENCES

[1] R. H. Güting and M. Schneider, *Moving Objects Databases*, Morgan Kaufmann, 2005.

[2] G. J. G. Upton and B. Fingleton, *Spatial Data Analysis by Example, Volume II: Categorical and Directional Data*, John Wiley & Sons, 1989.

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2nd edition, 2006.

[4] Y. Ishikawa, Y. Machida, and H. Kitagawa, "A dynamic mobility histogram construction method based on Markov chains," in *Proc. 18th Inte. Conf. on Scientific and Statistical Database Management (SSDBM 2006)*. July 2006, pp. 359–368, IEEE CS Press.

[5] Y. Ishikawa, Y. Machida, and H. Kitagawa, "A dynamic mobility histogram construction method based on Markov chains," *IEICE Trans. on Information and Systems*, Feb. 2007, (to appear, in Japanese).

[6] Y. Ishikawa, Y. Tsukamoto, and H. Kitagawa, "Extracting mobility statistics from indexed spatio-temporal databases," in *Proceedings of the 2nd Workshop on Spatio-Temporal Database Management (STDBM'04)*, Aug. 2004, pp. 9–16.