Combining Clustering with Moving Sequential Pattern Mining: A Novel and Efficient Technique*

Shuai Ma, Shiwei Tang, Dongqing Yang, Tengjiao Wang, and Jinqiang Han

Department of Computer Science, Peking University, Beijing 100871, China
{mashuai,tjwang,jqhan}@db.pku.edu.cn
{tsw,dqyang}@pku.edu.cn

Abstract. Sequential pattern mining is a well-studied problem. In the context of mobile computing, moving sequential patterns that reflects the moving behavior of mobile users attracted researchers' interests recently. In this paper a novel and efficient technique is proposed to mine moving sequential patterns. Firstly the idea of clustering is introduced to process the original moving histories into moving sequences as a preprocessing step. Then an efficient algorithm called PrefixTree is presented to mine the moving sequences. Performance study shows that PrefixTree outperforms LM algorithm, which is revised to mine moving sequences, in mining large moving sequence databases.

1 Introduction

Mining moving sequential patterns has great significance for effective and efficient location management in wireless communication systems. The problem of mining moving sequential patterns is a special case of mining traditional sequential patterns with the extension of support. There are mainly four differences between mining conventional sequential patterns and moving sequential patterns. Firstly, if two items are consecutive in a moving sequence α, and α is a subsequence of β, those two items must be consecutive in β. This is because we care about what the next move is for a mobile user in mining moving sequential patterns. Secondly, in mining moving sequential patterns the support considers the number of occurrences in a moving sequence that helps a more reasonable pattern discovery, so the support of a moving sequence is the sum of the number of occurrence in all the moving sequences of the whole moving sequence database. Thirdly, the Apriori property plays an important role for efficient candidate pruning in mining traditional sequential patterns. For example, suppose <ABC> is a frequent length-3 sequence, and then all the length-2 subsequences {<AB>, <AC>, <BC>} must be frequent in mining sequential patterns. In mining moving sequential patterns <AC> may not be frequent. This is because a

* Supported by the National High Technology Development 863 Program of China under Grant No. 2002AA4Z3440; the National Grand Fundamental Research 973 Program of China under Grant No. G1999032705.

H. Dai, R. Srikant, and C. Zhang (Eds.): PAKDD 2004, LNAI 3056, pp. 419–423, 2004.

mobile user can only move into a neighboring cell in a wireless system and items must be consecutive in mining moving sequential patterns. In addition, <AC> is not a subsequence of <ABC> any more in mining moving sequential patterns and that any subsequence of a frequent moving sequence must be frequent is still fulfilled from that meaning, which is called Pseudo-Apriori property. The last difference is that a moving sequence is an order list of items, but not an order list of itemsets, where each item is a cell id.

Wen-Chih Peng et al. presented a data-mining algorithm, which involves mining for user moving patterns in a mobile computing environment in [1]. Moving pattern mining is based on a roundtrip model [2], and their LM algorithm selects an initial location S, which is either VLR or HLR whose geography area contains the homes of the mobile users. Suppose a mobile user goes to a strange place for one month or longer, the method in [1] cannot find the proper moving pattern to characterize the mobile user. A more general method should not give any assumption of the start point of a moving pattern. Basically, algorithm LM is a variant one from GSP [3]. The Apriori-based methods can efficiently prune candidate sequence patterns based on Aprior property, but in moving sequential pattern mining we cannot prune candidate sequences efficiently because the moving sequential pattern only preserves Pseudo-Apriori property. In the meanwhile Apriori-based algorithms still encounter problem when a sequence database is large and/or when sequential patterns to be mined are numerous and/or long [4].

Time factor is considered for personal paging area design in [5]. G. Das et al. present a clustering based method to discretize a times series in [6]. Time is also a very important factor in mining moving sequential patterns. In this paper, firstly the idea of clustering is also introduced into the mining of moving sequential patterns to discretize the time attribute of the moving histories, and the moving histories are transformed into moving sequences based on the clustering result. Then based on the idea of projection and Pseudo-Apriori property, an efficient moving sequential pattern mining algorithm called PrefixTree is proposed, which can effectively represent candidate frequent moving sequences with a key tree structure of prefix trees. In addition, the wireless network topology based optimization approach is also presented to improve the efficiency of the PrefixTree algorithm.

The rest of the paper is organized as follows. Data preprocessing and the PrefixTree algorithm are given in section 2. Section 3 gives the experimental results from different viewpoints. Discussion is made in section 4.

2 Mining Moving Sequential Patterns

User moving history is the moving logs of a mobile user, which is an ordered (c, t) list where c is the cell ID and t is the time when the mobile user reaches cell c. Let MH=<(c_1, t_1), (c_2, t_2), (c_3, t_3), …(c_n, t_n)> be a moving history, and MH means the mobile user enters c_1 at t_1, leaves c_1 and enters c_2 at t_2, leaves c_2 and enters c_3 at t_3…and finally enters c_n at t_n. Each (c_i, t_i)∈MH ($1 \leq i \leq n$) is called an element of MH. The time difference between two consecutive elements in a moving history reflects a mobile users' moving speed (or sojourn time in a cell). If the difference is high, it shows the mobile user moves at a relatively low speed; otherwise, if the difference is low, it shows the mobile user moves at a relatively high speed. The idea of using clustering to discretize time is that if a mobile user possesses regular moving behavior, then he often moves on the same set of paths, and the arrival time to each point of the paths is similar.

The clustering algorithm CURD [7] is used to discretize the time attribute of the moving histories. For each cell c in the cell set, all the elements in the moving history database D is collected, denoted by $ES(c)=\{(c, t)|\exists(c, t)\in MH_i \text{ and } MH_i\in D\}$. Then the CURD algorithm is used to cluster the element set ES(c), where Euclidean distance on time t is used as the similarity function between two elements in ES(c). This is a clustering problem in one-dimension space. After the clustering processing we have a clustering result as $\{(c, T_s, T_e)|T_s, T_e\in T \text{ and } T_s\leq T_e\}$(T is the time domain), which means the mobile user often enters cell c at the period $[T_s, T_e]$.

The main idea of data transformation is to replace the MH elements with the corresponding clusters that they belong to. The transformed moving history is called a moving sequence, and the transformed moving history database is called moving sequence database. Each moving history can be transformed into one and only one moving sequence because the clustering method guarantees that any MH element belongs to one and only one cluster.

The PrefixTree algorithm only need scan the database three times, and the key idea of PrefixTree is the use of the prefix trees. Prefix tree is a compact representation of candidate moving sequential patterns. The root is the frequent item, and is defined at depth one. Each node contains three attributes: one is the item, one is the count attribute which means the support of the item, and the last one is the flag indicating whether the node is traversed. The items of a node's children are all contained in its candidate consecutive items. In the first two scans PrefixTree generates the frequent itmes, frequent length-2 moving sequential patterns and CCIs of each frequent item, and the prefix trees are constructed in the third scan. For each item C_i, we call the items that may appear just after it candidate consecutive items. It is easy to know that only the items after C_i in a moving sequence may be the consecutive items of C_i, denoted by $CCI(C_i)$. And for any item $C_j\in CCI(C_i)$, length-2 moving sequence $<C_iC_j>$ is frequent. It is easy for us to generate the moving sequential patterns based on the prefix trees. Every moving sequence from the root node to the leaf node is a candidate frequent moving sequences. Scanning all the prefix trees once can generate all the moving sequential patterns. The support of each node decreases with the depth increase, so a new frequent moving sequence is generated when we traverse the prefix trees from the root to the leaves when encountering a node whose count is less than the support threshold.

The extension of support is considered when generating the prefix trees, which can be done by generating projected sequences. Another novelty of the PrefixTree algorithm is that it doesn't generate projected physical files, which is different from traditional projection-based sequential algorithms [4] and is a time-cost work. Due to the room reason, we cannot give too many details here.

As pointed in [8], the initial candidate set generation, especially for the length-2 frequent patterns, is the key issue to improve the performance of data mining. Fortunately, sometimes the apriori of the wireless network topology can be got. From the apriori the neighboring cells for each cell can be known in advance. It has been pointed out that for any two consecutive items in a moving sequence, one item must be the other one's neighboring cell or itself. The number of neighbor cells is often small, for example it is two in one-dimension model, and six in two-dimension hexagonal model and eight in two-dimension mesh model, and is relative small even in graph model [9]. So based on this apriori the number of the candidate length-2 moving sequential patterns is decreased much, and then the frequent length-2 moving sequential patterns can be efficiently generated.

3 Experimental Results and Performance Study

All experiments are performed on a 1.7GHz Pentium 4 PC machine with 512M main memory and 60G hard disk, running Microsoft Windows 2000 Professional. All the methods are implemented using JBuilder 6.0. The synthetic dataset used in our experiments comes from SUMATRA (Stanford University Mobile Activity TRAces, which is available at http://www-db.stanford.edu/sumatra/). BALI-2: Bay Area Location Information (real-time) dataset records the mobile users' moving and calling activities in a day. The mobile user averagely moves 7.2 times in a day in 90 zones, so the number of items is 90 and the average length of the moving sequences is 8.2. BALI-2 contains about 40,000 moving sequences, which are used for our experiments.

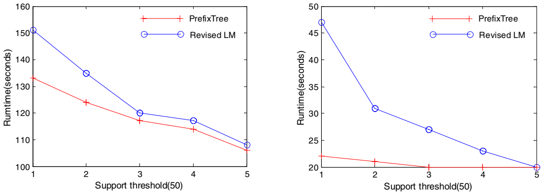

Fig. 1. Performance study with varying support threshold, where the left part and the right part are without and with optimization respectively

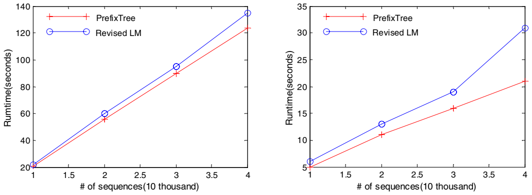

Fig. 2. Performance study with varying number of moving sequences, where the left part and the right part are without and with optimization respectively

Fig.1 and Fig.2 show that the PrefixTree algorithm, which needs only three scans of the moving sequence database, is more efficient and scalable than the Revised LM algorithm, which needs multiple scans of the moving sequence database limited with the length of the longest moving sequential pattern. In addition, the optimization based on the wireless network topology improves much the efficiency of mining processing. Let |D| is the number of moving sequences in database D, and L is the length of the longest moving sequential pattern. Let reading a moving sequence in a data file costs 1 unit of I/O. The I/O cost of the Revised LM algorithm is equal to L(|D|); the I/O cost of PrefixTree is equal to 3(|D|). From the I/O costs analysis we could get a coarse conclusion that the PrefixTree algorithm is more efficient than the Revised LM algorithm if L is bigger than 3. The above simple I/O analysis of the PrefixTree algorithm and the Revised LM algorithm gives an evience of the PrefixTree algorithm's efficiency showed in the experimental results.

4 Discussion

In this paper the idea of clustering method is introduced to discretize the time attribute in moving histories. And then a novel and efficient method, called PrefixTree, is proposed to mine the moving sequences. Its main idea is to generate projected sequences and construct the prefix trees based on candidate consecutive items. It is highly desirable because in most cases the user tends to try a few minimum supports before being satisfied with the result. Another valuable function of the PrefixTree algorithm is supporting parameter tuning, which means the prefix trees with a higher support threshold can be generated directly from the prefix trees with a smaller one.

References

1. Wen-Chih Peng, Ming-Syan Chen. Mining User Moving Patterns for Personal Data Allocation in a Mobile Computing System. Proc. of ICPP, pp. 573-580, 2000.
2. N. Shivakumar, J. Jannink, and J. Widom. Per-user Profile Replication in Mobile Environments: Algorithms Analysis and Simulation Result. ACM/Baltzer Journal of Mobile Networks and Applications, vol. 2, no. 2, pp. 129-140, 1997.
3. Ramakrishnan Srikant, Rakesh Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. Proc. of EDBT, pp. 3-17, 1996.
4. J. Pei, J. Han, B. Mortazavi-Asl et al. PrefixSpan: Mining Sequential Patterns Efficiently by PrefixProjected Pattern Growth. Proc. of ICDE, pp. 215-224, 2001.
5. Hsiao-Kuang Wu, Ming-Hui Jin, Jorng-Tzong Horng. Personal Paging Area Design Based On Mobiles Moving Behaviors. Proc. of INFOCOM, pp. 21-30, 2001.
6. G. Das, K. I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule Discovery from Time Series. Proc. of KDD, pp. 16-22, 1998.
7. Shuai Ma, Tengjiao Wang, Shiwei Tang et al. A New Fast Clustering Algorithm Based on Reference and Density. Proc. of WAIM, pp. 214-225, 2003.
8. Jong Soo Park, Ming-Syan Chen, Philip S. Yu. Using a Hash-Based Method with Transaction Trimming for Mining Association Rules. IEEE TKDE, vol. 9, no. 5, pp. 813-825, 1997.
9. Vincent W.S. Wong, Victor C. M. Leung. Location Management for Next Generation Personal Communication Networks. IEEE Network, vol. 14, no. 5, pp. 18-24, 2000.