PrefixUnion: Mining Traversal Patterns Efficiently in Virtual Environments

Shao-Shin Hung, Ting-Chia Kuo, and Damon Shing-Min Liu

Department of Computer Science and Information Engineering,
National Chung Cheng University,
Chiayi, Taiwan 621, Republic of China
{hss, ktc91,damon}@cs.ccu.edu.tw

Abstract. Sequential pattern mining is an important data mining problem with broad applications. Especially, it is also an interesting problem in virtual environments. In this paper, we propose a projection-based, sequential pattern-growth approach, called *PrefixUnion*. Meanwhile, we also introduce the relationships among transactions, views and objects. According to these relationships, we suggest two mining criteria — *inter-pattern growth* and *intra-pattern growth*, which utilize these characteristics to offer ordered growth and reduced projected database. As a result, the large-scale VRML models could be accessed more efficiently, allowing for a real-time walk-through in the scene.

1 Introduction

The interactive walkthrough system provides a virtual environment with complex 3D models [1, 3, 4]. On the other side, sequential pattern mining is one of the main topics in data mining methods [2, 6, 7, 8]. In this paper, we propose a mining mechanism based on *inter-pattern growth* and *intra-pattern growth*. These two pattern growth criteria are used to minimize useless pattern growth by finding these patterns, whose projected-patterns are the same, and letting them union beforehand. This results in less access times and much better performance.

The rest of this paper is organized as follows. In Section 2, the related works are given. In Section 3, the mining problem of virtual environment sequential patterns is introduced along with the notation that is used throughout the paper. The *PrefixUnion* mining algorithm is suggested in Section 4. To evaluate the efficiency of the *PrefixUnion* algorithm, our experimental results are presented in Section 5. Finally, we conclude our study in Section 6.

2 Related Works

2.1 Sequential Patterns Mining

Sequential pattern mining problem was first introduced in [6]. With the motivation of avoiding or substantially reducing the expensive candidate generation and pruning, the *FreeSpan* [9] and *PrefixSpan* [10] were proposed. On the other side, they still

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, pp. 830 – 833, 2005.

have some non-trivial costs. One is that the full length original sequence must be retained in each projected database because a pattern can be generated by any subsequence combination. This leads to many duplicated sequences involved. The other is that the growth of a subsequence is explored at any split point in a candidate sequence resulting in several possible new subsequences.

3 Problem Formulation

3.1 Notations

In this section, we introduce the terms used in our problem and mining algorithm. Let $\Sigma = \{l_1, l_2, ..., l_m\}$ be a set of *m* literals called *objects* (also called *items*) [1, 6]. A *view v* is denoted by $v = < \chi_1, \chi_2, ..., \chi_k >$, is an unordered list of objects such that each objects $\chi_i \in \Sigma$. The *view v* is defined as whatever the user stays and observes during the processing of virtual environments. A *sequence S*, denoted by $\{v_1, v_2, ..., v_n\}$, is an ordered list of *n* views. Let the database *D* be a set of sequences (also called transactions). Each sequence records each user's traversal path in walkthrough system. A sequence $\beta = < \beta_1, \beta_2, ..., \beta_\kappa >$ is a *subsequence* (or is called *contained*) of sequence $\alpha = < \alpha_1, \alpha_2, ..., \alpha_n >$ if there exists $1 \le i_1 < i_2 < ... < i_\kappa \le n$ such that $\beta_1 \subseteq \alpha_1, \beta_2 \subseteq \alpha_2, ..., \beta_k \subseteq \alpha_k$ holds. The *support* of a sequence *p* in the sequence database *D* is defined as the number of the sequences which contain this pattern *p*. A *frequent sequence* is a sequence whose support is equal to or more than the user defined threshold (also called *min_support*). A *frequent pattern* is a *maximal sequence* that is *frequent*. Finally, let *P* be a set of all frequent patterns in *D*.

4 *PrefixUnion Mining Algorithm*

In order to realize our pattern-growth approach, we will define two kinds of pattern-growth types. One is the *intra-view growth* – the growth of pattern is bounded by the

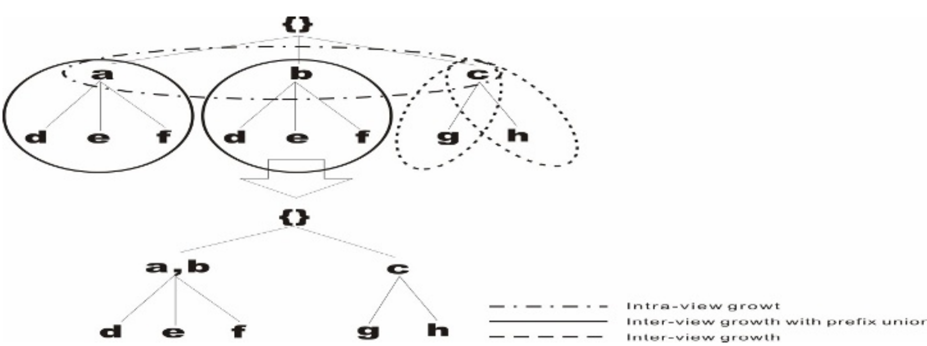

Fig. 1. Scenario of differences among intra-view growth, inter-view growth and inter-view growth with *PrefixUnion*

view boundary. The other is the *inter-view growth* – the growth of pattern is to select an object in next view. The pattern-growth algorithm is based on recursively constructing the patterns. The *α-projected* sub-database is defined as the set of subsequences in the database which the *suffixes* of the sequences have the common *prefix* *α*. In order to demonstrate this concept, these relationships are shown in the Figure 1.

In Figure 1, objects *a*, *b*, and *c* are contained in the same view (i.e., the intra-view growth). Object *a* and its siblings belong to the inter-view growth. Object *b* also has the same case. Object c and its siblings is another different example of inter-view growth (i.e. *c*---*g* and *c*---*h*). Only the object *a* and object *b* can be applied by *PrefixUnion* approach. After the processing, both objects *a* and *b* are merged. By this *PrefixUnion* property, the search space is reduced sharply in each step. This property improves much better performance, especially in the presence of small minimum support threshold.

5 Experimental Results

In this section, the effectiveness of the proposed mining algorithm is investigated. The data set is about 437 MB of 1,256 objects in the system. The traversal path consist of approximately 10 ~ 15 views from one end to the other end.

In the Figure 2, our algorithm outperforms the pattern growth method. If the locations of any two objects are too far in the virtual environments, it is naturally assumed that there are no relationships between the objects. In other words, the relationships are limited and restricted compared to the IBM dataset.

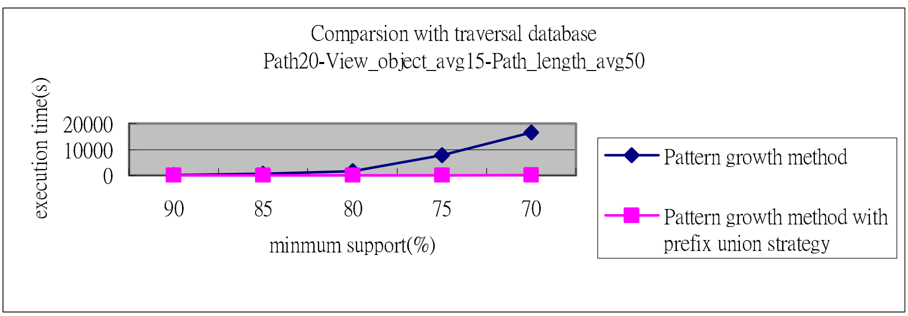

Fig. 2. Comparison of traditional pattern growth mining algorithm and our *PrefixUnion* mining algorithm in our virtual environments

6 Conclusions and Future Work

We have extended the applications of mining. With properties of intra-view pattern growth and inter-view pattern growth added, it is more precise and useful for us to discover the frequent traversal patterns. Besides, we also consider how to efficiently mining the necessary patterns in order to speed up the computations.

References

[1] Daniel G. Aliaga, and Anselmo Lastra, "Automatic Image Placement to Provide a Guaranteed Frame Rate", *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 307-316, 1999.

[2] Ming-Syan Chen, Jong Soo Park, and Philip S. Yu, "Efficient Data Mining for Path Traversal Patterns", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10, Issue 2, pp. 209-221, 1998.

[3] Y. Nakamura, S. ABE, Y. Ohsawa, and M. Sakauchi, "A Balanced Hierarchical Data Structure Multidimensional Dada with Efficient Dynamic Characteristic**",** *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 682-694, 1993.

[4] Y. Nakamura and T. Tamada, "An Efficient 3D Object Management and Interactive Walkthrough for the 3D Facility Management System**",** *Proc. IECON'94*, Vol. 2, pp. 1937-1941, 1994.

[5] T. Morzy, M. Wojciechowski, and M. Zakrzewicz, "Pattern-Oriented Hierarchical Mining**",** *Proc. of the 3rd East European Conference on Advances in Databases and Information Systems (ADBIS'99)*, Maribor, Slovenia, LNCS 1691, Springer-Verlag, 1999.

[6] Rakesh Agrawal, Tomasz Imielinski and Arun N. Swami, "Mining Association Rules between Sets of Items in Large Databases"**,** *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp.207-216, 1993.

[7] Rakesh Agrawal, and R. Srikant, "Mining Sequential Patterns", *Proceedings of the 1995 International Conference on Data Engineering (ICDE'95)*, pp.3-14, 1995.

[8] Rakesh Agrawal, and R. Srikant, "Mining Sequential Patterns: Generalizations and Performance Improvements", *Proceeding Fifth International Conference on Extending Database Technology (EDBT'96),* pp.3 -17, Mar 1996.

[9] J.pei, J.han, B. M-Asl, J. Wang, H. pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining", *Proceedings of the 2000 ACM SIGKDD International Conference Knowledge in Database (KDD'90)*, pp. 355-359, August 2000.

[10] J.pei, J.han, B. M-Asl, J. Wang, H. pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Proceedings of the 2001 International Conference Data Engineering (ICDE'01), pp. 215-224, April 2001.