Binary Sequences and Association Graphs for Fast Detection of Sequential Patterns

Selim Mimaroglu*, Dan A. Simovici**

* Bahcesehir University,Istanbul, Turkey, selim.mimaroglu@gmail.com
**University of Massachusetts Boston, Massachusetts 02125, USA, dsim@cs.umb.edu

Abstract. We develop an efficient algorithm for detecting frequent patterns that occur in sequence databases under certain constraints. By combining the use of bit vector representations of sequence databases with association graphs we achieve superior time and low memory usage based on a considerable reduction of the number of candidate patterns.

1 Introduction

Mining sequential patterns was originally proposed in Agrawal and Srikant (1995), where three algorithms, (*AprioriAll*, *AprioriSome*, and *DynamicSome*) were introduced. *PrefixSpan*, based on the prefix projection idea, was introduced in Pei et al. (2001). SPADE Zaki (2001) performs space efficient joins on prefix-based equivalence classes. *PRISM* Gouda et al. (2007), uses prime number encoding for support counting. A related but distinct problem (discussed in Mannila et al. (1997)) is finding frequent episodes in very long sequences. *SPAM* Ayres et al. (2002) finds sequential patterns using a bitmap representation. An extension of SPAM, which incorporates gap and regular expression constraints was achieved in Ho et al. (2005). The GSP algorithm Srikant and Agrawal (1996) is similar to *AprioriAll*; additionally it can handle three types of constraints: minimum and maximum gap between consecutive elements of a sequence (referred to as *min_gap* and *max_gap*), and window size between rows. When *min_gap* $= 0$, *max_gap* $= \infty$, and *window_size* $= 0$, the sequential patterns found by GSP are the classical sequential patterns as introduced in Agrawal and Srikant (1995). The algorithm *cSPADE* Zaki (2000) introduces similar constraints, and it is implemented on top of *SPADE*. *SPIRIT* Garofalakis et al. (1999) is more general than both *GSP* and *cSPADE* as it deals with regular expression constraints.

In this note we describe SPAG, an algorithm that combines the dual use of bit vector representations of sequence databases with association graphs to achieve superior performance in identifying patterns in sequences.

2 Apriori Frameworks on Sequence Sets

We refer the reader to Simovici and Djeraba (2008) for mathematical concepts and notations. Let I be a set of items, and let $\mathbf{Seq}(I)$ be the set of sequences of items of I. We consider a a graded poset $(P, \leq, h)$, where $P \subseteq \mathbf{Seq}(I)$, and $h : P \longrightarrow \mathbb{N}$, referred to as the *set of patterns*, and a *data set* $\mathcal{D}$ defined as a sequence of sequences, $\mathcal{D} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subseteq \mathbf{Seq}(\mathbf{Seq}(I))$. A *sequence Apriori framework* is a triple $((P, \leq, h), \mathcal{D}, \sigma)$, where σ is a relation between patterns and data, such that $\mathbf{t} \leq \mathbf{t}'$ and $(\mathbf{t}', \mathbf{s}) \in \sigma$ implies $(\mathbf{t}, \mathbf{s}) \in \sigma$.