Chapter 95
Approaches for Pattern Discovery Using Sequential Data Mining

Manish Gupta
University of Illinois at Urbana-Champaign, USA

Jiawei Han
University of Illinois at Urbana-Champaign, USA

ABSTRACT

In this chapter we first introduce sequence data. We then discuss different approaches for mining of patterns from sequence data, studied in literature. Apriori based methods and the pattern growth methods are the earliest and the most influential methods for sequential pattern mining. There is also a vertical format based method which works on a dual representation of the sequence database. Work has also been done for mining patterns with constraints, mining closed patterns, mining patterns from multi-dimensional databases, mining closed repetitive gapped subsequences, and other forms of sequential pattern mining. Some works also focus on mining incremental patterns and mining from stream data. We present at least one method of each of these types and discuss their advantages and disadvantages. We conclude with a summary of the work.

INTRODUCTION

What is Sequence Data?

Sequence data is omnipresent. Customer shopping sequences, medical treatment data, and data related to natural disasters, science and engineering processes data, stocks and markets data, telephone calling patterns, weblog click streams, program execution sequences, DNA sequences and gene expression and structures data are some examples of sequence data.

DOI: 10.4018/978-1-4666-2455-9.ch095

Notations and Terminology

Let $I = \{i_1, i_2, i_3 \ldots i_n\}$ be a set of items. An item-set X is a subset of items i.e. $X \subseteq I$. A sequence is an ordered list of item-sets (also called elements or

events). Items within an element are unordered and we would list them alphabetically. An item can occur at most once in an element of a sequence, but can occur multiple times in different elements of a sequence. The number of instances of items in a sequence is called the length of the sequence. A sequence with length l is called an l-sequence. E.g., s=<a(ce)(bd)(bcde)f(dg)> is a sequence which consists of 7 distinct items and 6 elements. Length of the sequence is 12.

A group of sequences stored with their identifiers is called a sequence database. We say that a sequence s is a subsequence of t, if s is a "projection" of t, derived by deleting elements and/or items from t. E.g. <a(c)(bd)f> is a subsequence of s. Further, sequence s is a δ-distance subsequence of t if there exist integers $j_1 < j_2 < \ldots < j_n$ such that $s_1 \subseteq t_{j1}$, $s_2 \subseteq t_{j2} \ldots s_n \subseteq t_{jn}$ and $j_k\text{-}j_{k-1} \leq \delta$ for each k = 2, 3... n. That is, occurrences of adjacent elements of s within t are not separated by more than δ elements.

What is Sequential Pattern Mining?

Given a pattern p, support of the sequence pattern p is the number of sequences in the database containing the pattern p. A pattern with support greater than the support threshold min_sup is called a frequent pattern or a frequent sequential pattern. A sequential pattern of length l is called an l-pattern. Sequential pattern mining is the task of finding the complete set of frequent subsequences given a set of sequences. A huge number of possible sequential patterns are hidden in databases.

A sequential pattern mining algorithm should:

A. find the complete set of patterns, when possible, satisfying the minimum support (frequency) threshold,
B. be highly efficient, scalable, involving only a small number of database scans
C. be able to incorporate various kinds of user-specific constraints.

APPROACHES FOR SEQUENTIAL PATTERN MINING

Apriori-Based Method (GSP: Generalized Sequential Patterns) (Srikant & Agrawal, 1996)

The Apriori property of sequences states that, if a sequence S is not frequent, then none of the super-sequences of S can be frequent. E.g, <hb> is infrequent implies that its super-sequences like <hab> and <(ah)b> would be infrequent too.

The GSP algorithm finds all the length-1 candidates (using one database scan) and orders them with respect to their support ignoring ones for which support < min_sup. Then for each level (i.e., sequences of length-k), the algorithm scans database to collect support count for each candidate sequence and generates candidate length-(k+1) sequences from length-k frequent sequences using Apriori. This is repeated until no frequent sequence or no candidate can be found.

Consider the database as shown in Figure 1. Our problem is to find all frequent sequences, given min_sup=2.

As shown in Figure 2, using Apriori one needs to generate just 51 length-2 candidates, while without Apriori property, 8*8+8*7/2=92 candidates would need to be generated. For this example, Apriori would perform 5 database scans, pruning away candidates with support less than

Figure 1. Database

Database

Seq Id	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Length-1 Patterns

Cand	Seq
<a>	3
<b>	5
<c>	4
<d>	3
<e>	3
<f>	2
<g>	1
<h>	1