

Efficient Mining of Temporally Annotated Sequences

Fosca Giannotti Mirco Nanni
ISTI - CNR, Pisa, Italy
{f.giannotti, m.nanni}@isti.cnr.it

Dino Pedreschi
C.S. Dep., Univ. of Pisa, Italy
pedre@di.unipi.it

Abstract

Sequential patterns mining received much attention in recent years, thanks to its various potential application domains. A large part of them represent data as collections of time-stamped itemsets, e.g., customers' purchases, logged web accesses, etc. Most approaches to sequence mining focus on *sequentiality* of data, using time-stamps only to order items and, in some cases, to constrain the temporal gap between items. In this paper, we propose an efficient algorithm for computing (*temporally-*)*annotated sequential patterns*, i.e., sequential patterns where each transition is annotated with a *typical* transition time derived from the source data. The algorithm adopts a prefix-projection approach to mine candidate sequences, and it is tightly integrated with an annotation mining process that associates sequences with temporal annotations. The pruning capabilities of the two steps sum together, yielding significant improvements in performances, as demonstrated by a set of experiments performed on synthetic datasets.

1 Introduction

Frequent Sequential Pattern mining (FSP) deals with the extraction of frequent sequences of events from datasets of transactions; those, in turn, are time-stamped sequences of events (or sets of events) observed in some business contexts: customer transactions, patient medical observations, web sessions, trajectories of objects moving among locations.

As we observe in the related work section, time in FSP is used as a user-specified constraint to the purpose of either preprocessing the input data into ordered sequences of (sets of) events, or as a pruning mechanism to shrink the pattern search space and make computation more efficient. In either cases, time is forgotten in the output of FSP. For this reason, in our previous work [4] we introduced a form of sequential patterns annotated with temporal information representing typical transition times between the events in a frequent sequence. Such a pattern is called *Temporally-Annotated Sequence*, *TAS* in short.

In principle, this form of pattern is useful in several contexts: (i) in web log analysis, different categories of

users (experienced vs. novice, interested vs. uninterested, robots vs. humans) might react in similar ways to some pages — i.e., they follow similar sequences of web access — but with different reaction times; (ii) in medicine, reaction times to patients' symptoms, drug assumptions and reactions to treatments are a key information.

In all these cases, enforcing fixed time constraints on the mined sequences is not a solution. It is desirable that typical transition times, when they exist, emerge from the input data.

The contributions of this paper are the following:

1. We provide a new algorithm for mining frequent *TAS*, that is efficient and correct and complete w.r.t. the formal definition of *TAS*— whereas the algorithm given in [4] provides approximate solutions.
2. We propose a new way for concisely representing sets of frequent *TAS*'s, making them readable for the user.
3. We provide an empirical study of the performances of our algorithm, focusing on the overall computational cost and on some of the central and most interesting sub-tasks.

The paper is organized as follows: Section 2 provides an overview of related work and background information; Section 3 briefly summarizes the formal definition of the *TAS* mining problem; Section 4 describes in detail the proposed algorithm, and then Section 5 provides an empirical evaluation of the system. Finally, Section 6 closes the paper with some conclusive remarks.

2 Background and related work

In this section we summarize a few works related to the topic of this paper, and will introduce some relevant basic concepts and related works on sequential pattern mining, clustering and estimation of probability distributions.

2.1 Sequence mining. The *frequent sequential pattern* (FSP) problem is defined over a database of sequences D , where each element of each sequence is a

