

Chapter 3.20

MILPRIT*: A Constraint-Based Algorithm for Mining Temporal Relational Patterns

Sandra de Amo

Universidade Federal de Uberlândia, Brazil

Waldecir P. Junior

Universidade Federal de Uberlândia, Brazil

Arnaud Giacometti

Université François Rabelais de Tours, France

ABSTRACT

In this article, we consider a new kind of temporal pattern where both interval and punctual time representation are considered. These patterns, which we call temporal point-interval patterns, aim at capturing how events taking place during different time periods or at different time instants relate to each other. The datasets where these kinds of patterns may appear are temporal relational databases whose relations contain point or interval timestamps. We use a simple extension of Allen's Temporal Interval Logic as a formalism for specifying these temporal patterns. We also present the algorithm MILPRIT for mining temporal point-interval patterns, which uses variants of the classical levelwise search algorithms.*

In addition, MILPRIT allows a broad spectrum of constraints to be incorporated into the mining process. An extensive set of experiments of MILPRIT* executed over synthetic and real data is presented, showing its effectiveness for mining temporal relational patterns.*

INTRODUCTION AND MOTIVATION

The problem of discovering sequential patterns in temporal data has been studied extensively in the past years (Pei et al., 2004; Srikant & Agrawal, 1996; Zaki, 2001), and its importance is fully justified by the great number of potential application domains where mining sequential patterns appears as a crucial issue, such as financial

market (evolution of stock market shares quotations), retailing (evolution of clients purchases), medicine (evolution of patient symptoms), local weather forecast, telecommunication (sequences of alarms output by network switches), and so forth. Most of these patterns are specified by formalisms, which are, to some extent, reducible to Propositional Linear Temporal Logic, where time is represented by points in a straight line. For instance, let us consider a classical sequential pattern of the form $s = \langle \{a, b\}, \{c, d\} \rangle$ (where $\{a, b\}$ and $\{c, d\}$ are sets of items purchased by a client. This sequential pattern can be expressed in the Propositional Linear Temporal Logic by the formula $P_a \wedge P_b \wedge \Diamond (P_c \wedge P_d)$, where for each $i \in \{a, b, c, d\}$, P_i is a propositional symbol standing for “client buys item i .” The symbol \Diamond stands for the temporal operator *sometimes in the future* (for a comprehensive survey on Linear Temporal Logic (Emerson, 1990)). The need for a more expressive kind of temporal patterns arises, for instance, when modeling Unix users’ behavior (Jacobs & Blockeel, 2001). Consider, for instance¹, the following sequence of commands related to latex users: *ls, vi paper.tex, latex paper.tex, dvips paper.dvi, lpr paper.ps*. This sequence of commands can be represented as a sequence of relational (or first-order) atoms of the form: *ls, vi(paper:tex), latex(paper:tex), dvips(paper:dvi), lpr(paper:ps)*. Within such a database of sequences of relational atoms, it would be possible to discover that the relational sequence pattern *vi(X), latex(X)* is frequent. Notice that this pattern tells us that the sequence of commands (predicates) *vi, latex* is frequently requested by users. In order to specify this kind of sequential patterns, we need a more powerful logical formalism, the First-Order Linear Temporal Logic, since the elements involved in the patterns include both predicates and their parameters.

In all the previous examples illustrating the propositional and relational settings of the sequential pattern mining problem, the events are instantaneous; that is, the time when they happen

is represented as a point (instant) in a straight line. The sequence of events corresponds to a sequence of instants when these events take place. However, there are many situations where events have a certain *duration*, and thus, the underlying time is measured in terms of *intervals* instead of points. In this article, we propose a new temporal pattern involving a hybrid representation for time and a new algorithm (MILPRIT*) for mining them. These patterns, which we call *temporal point-interval patterns* (or pi-patterns for short), aim at capturing how events taking place in time intervals or time instants relate to each other. For instance, (1) in a medical application, we could be interested in discovering if patients who take some medicine X during a certain period of time P , and in some moment m in P undergo a stomach surgery, will present the symptom Z during a period of time beginning *right after* the surgery and *finishing* as soon as they stop taking the medicine X ; (2) in an agricultural application, we could be interested in discovering if the use of some organic fertilizer during a period of time has an effect on the way a plant grows *during* and *after* the fertilizer application. The following example illustrates the medical application in more detail.

Example 1 (Running Example): *Let us consider the database schema $\mathbf{R} = \{ Patient, Med, Symp, Hist \}$ and the relational database instance D over \mathbf{R} illustrated in Figure 1. Notice that the following behavior is verified by 50% of the patients (two patients out of four in the Patient relation verified it; namely, Paul and Sarah): the patient takes penicillin during a certain period of time e ; during a period of time f following his taking the medicine, he feels dizzy and undergoes a stomach surgery someday t during f .*

Temporal pi-patterns are specified as a set of atomic first-order formulae where time is represented by an interval-time variable or by a point-time variable, together with a set of temporal predicates $\{ before, meets, overlaps, dur-$

19 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/milprit-constraint-based-algorithm-mining/7966

Related Content

Using Ontology Languages for Conceptual Modeling

Palash Bera, Anna Krasnoperova and Yair Wand (2010). *Journal of Database Management* (pp. 1-28).
www.irma-international.org/article/using-ontology-languages-conceptual-modeling/39114

An Approach for Evolution-Driven Method Engineering

Jolita Ralyte, Colette Rolland and Mohamed Ben Ayed (2005). *Information Modeling Methods and Methodologies: Advanced Topics in Database Research* (pp. 80-101).
www.irma-international.org/chapter/approach-evolution-driven-method-engineering/23010

Optimizing Semi-Stream CACHEJOIN for Near-Real- Time Data Warehousing

M. Asif Naeem, Erum Mehmood, M. G. Abbas Malik and Noreen Jamil (2020). *Journal of Database Management* (pp. 20-37).
www.irma-international.org/article/optimizing-semi-stream-cachejoin-for-near-real-time-data-warehousing/245298

Principles on Symbolic Data Analysis

Héctor Oscar Nigro and Sandra Elizabeth González Císaro (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends* (pp. 74-81).
www.irma-international.org/chapter/principles-symbolic-data-analysis/20690

Data Warehouse Design to Support Customer Relationship Management Analyses

Colleen Cunningham, Il-Yeol Song and Peter P. Chen (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 702-724).
www.irma-international.org/chapter/data-warehouse-design-support-customer/7939