Computers & Graphics 38 (2014) 38–50

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on Visual Analytics

Mind the time: Unleashing temporal aspects in pattern discovery

T. Lammarsch [a,*], W. Aigner [a], A. Bertone [b], S. Miksch [a], A. Rind [a]

[a] Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria
[b] Institute for Cartography, Dresden University of Technology, Germany

ARTICLE INFO

Article history:
Received 23 June 2013
Received in revised form
7 October 2013
Accepted 8 October 2013
Available online 17 October 2013

Keywords:
Visual analytics
KDD
Temporal Data Mining
Data mining
Time-oriented data
Pattern finding
Interactive visualization

ABSTRACT

Temporal Data Mining is a core concept of Knowledge Discovery in Databases handling time-oriented data. State-of-the-art methods are capable of preserving the temporal order of events as well as the temporal intervals in between. The temporal characteristics of the events themselves, however, can likely lead to numerous uninteresting patterns found by current approaches. We present a new definition of the temporal characteristics of events and enhance related work for pattern finding by utilizing temporal relations, like *meets*, *starts*, or *during*, instead of just intervals between events. These prerequisites result in MEMuRY, a new procedure for Temporal Data Mining that preserves and mines additional time-oriented information. Our procedure is supported by SAPPERLOT, an interactive visual interface for exploring the patterns. Furthermore, we illustrate the efficiency of our procedure presenting a benchmark of the procedure's run-time behavior. A usage scenario shows how the procedure can provide new insights.

1. Introduction

Data Mining is a central part of Knowledge Discovery in Databases (KDD). A very important data type is time, and one of the most successful approaches for Temporal Data Mining (TDM) is the search for temporal patterns. Methods for temporal pattern finding include clustering, classification, and association rules. Their main goal is disclosing local structures of interest [1]. Several state-of-the-art methods adopt the concepts of events, which are tuples of a time interval (a temporal primitive with an extent [2]) and a set of conditions (e.g., 10 cars are passing a road between 2 am and 3 am). These methods consider patterns as combinations of events (e.g., after hours with 30 cars, there are often hours with 40 cars). Fig. 1a demonstrates patterns mined by the state-of-the-art approach MuTIny [3] (see below for more details about that approach). We formally define all necessary terms in Section 3.

The task of finding interesting patterns is important in several domains, which is demonstrated by the range of applications covered in related work (Section 2). In this paper, we focus on examples from traffic data analysis, but data from medicine, retail, and other domains are all conceivable. Wong et al. [4] show that similar tasks arise in different domains and how the same Visual Analytics (VA) methods can be used to deal with them.

Pattern finding usually requires complex parameterization. Many approaches require the conditions to be defined in advance (called Apriori algorithms). These algorithms often produce large numbers of patterns that need to be explored. VA can support parameterization, exploration, and iterative re-parameterization by intertwining the approach with interactive visual interfaces [5]. Starting from early work in pattern finding [6], research has increasingly focused on the temporal aspect of patterns. Current methods, like the MuTIny approach [3], are capable of preserving the temporal order of events as well as the intervals in between.[1]

A weakness of current methods is that they only consider events of a fixed length which is usually predetermined by the raster interval[2] of the source data. The most advanced ones can deal with flexible time intervals between events, but they have to be regular multiples of the raster, and the event lengths are still fixed. If the conditions are met for a longer interval than the raster size, two consecutive events of the same type are found.

Consider the following example: a road is used more frequently on weekdays than on weekends. Traffic might be significantly lower on Saturday and Sunday (Fig. 1a). If the traffic dataset has one value for each day, then each event will also have a length of

[1] According to Aigner et al. [2], what is often called "interval" in related work from TDM should be distinguished between "indeterminate interval" after pattern finding and "indeterminate span" during parameterization. In this paper, we will just use the term "interval" for both cases.

[2] A raster is "a fragmentation of time without gaps consisting of raster intervals (usually with same lengths). A raster interval is a unit of time that constitutes a raster: "hour", "day", "week", or "30"" [7].

* Corresponding author.
E-mail addresses: lammarsch@ifs.tuwien.ac.at (T. Lammarsch), aigner@ifs.tuwien.ac.at (W. Aigner), alessio.bertone@tu-dresden.de (A. Bertone), miksch@ifs.tuwien.ac.at (S. Miksch), rind@ifs.tuwien.ac.at (A. Rind).

0097-8493/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cag.2013.10.007

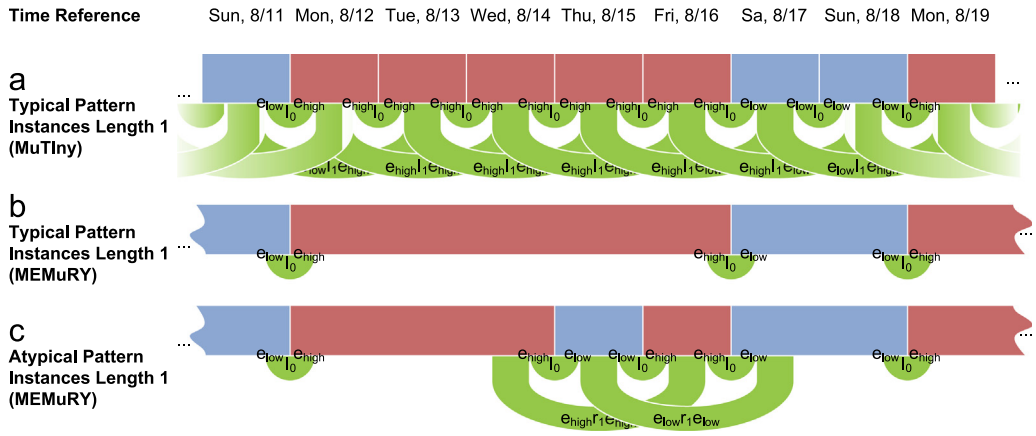

Fig. 1. Examples for patterns generated with (a) the MuTIny approach [3] and (b,c) with the MEMuRY procedure from this paper. (a) Even for few event and interval types, MuTIny [3] finds a vast number of patterns. (b) There are considerably less patterns found by MEMuRY, when looking for events that start one day after other events end, no patterns are found in this example at all. (c) If something unusual happens, like a low value on a Thursday, there are more patterns than for regular weeks.

one day. Based on the events of day-length, approaches tend to find patterns based on the business week. For five consecutive days, there is high traffic, and for two consecutive days there is low traffic. These sequences alternate. The main pattern found in this example is high traffic for one day (as the approaches only support the raster interval) followed by high traffic for one day. The next frequent pattern is low traffic for one day followed by low traffic for one day (slightly more frequent than high–low and low–high because of "long weekends"), and so on. The important information gained is strongly cluttered by unimportant information. When more intervals are looked for, e.g., one day in between, the number of patterns further increases. A similar problem can happen when the same dataset is looked at using an hour raster: during day hours, there often is much more fluctuation. At night, the traffic is continuously low. As a result, existing approaches hide patterns among the daily fluctuations behind a dominant pattern of low traffic follows low traffic. Such patterns are interesting, but by talking to domain experts, we found out that they are already well-known. An interview with a domain expert working in analysis of time-oriented data and scheduling tasks like shift-planning (Section 6.4) confirmed our own assessment: It is too difficult to apply state-of-the-art methods in practice if they generate too many patterns that stem from well-known effects of social time. The reason is that those patterns clutter the results when searching for new and unknown patterns. They make the list of results too large, so that it is hard to compute and even harder to visualize and understand.

Related work mostly relies on filtering out the less important patterns using the well-known concept of support (Section 2). In this paper, we give values for support as the fraction of *number of times a certain pattern exists/number of total patterns*. We also call this the frequency of a pattern, and it is unit-less. Only patterns with a certain frequency (often a high one) are considered. As we explained above, this approach is hard to parameterize if frequent patterns are not necessarily interesting, most likely because they are well-known to domain experts. Bertone et al. [5] propose including time information (like, "weekend") in the event definition, so that it is easier to prune weekend events followed by weekend events. However, this approach still does not provide events that cover a whole weekend and users end up with many more different event and pattern types.

Another solution proposed for this problem stems from data simplification. These methods transform raw data, which usually is given in the form of numerical data values to events and, thus, help to keep the complexity low [8–11]. According to the temporal aspects, these methods can simplify time by rasterization (converting the data to a given raster). If there are too many uninteresting patterns from several consecutive events of the same type, for example, during night hours or weekends, rastering the data to days can solve that problem, but possibly interesting patterns on finer granularities that exist during other times become hidden. Moreover, if a day raster is used to solve problems with night hours, the weekend problems arise more prominently. A fine raster size is needed to prevent information loss. Aggregating according to domain knowledge is better, so aggregating to blocks of eight hour length can even out the night/day issues while still giving some information. However, some information might still be hidden, and in real-world data, we also found that sometimes, the same effect just started one hour earlier or later. These hours would have been aggregated into the wrong block. As a result, we use the part of data simplification that simplifies the data dimensions into univariate events. For these events, only properties of nominal data are required. Aggregation over time is not an integrated part of our procedure, but we analyze its applicability in preprocessing in Section 6.

To deal with the shortcomings of existing pattern finding methods without resorting to aggregation over time, we presented a new procedure called Multi Event-length Multi Relation discoverY (MEMuRY) at the EuroVA 2013 workshop in Leipzig, Germany [12]. This paper is an extended version of our workshop paper. MEMuRY's basic idea is to reduce the number of unimportant patterns while retaining the important ones. This is done by reducing the number of events in a way that omits repetitive patterns from effects like weekends and night hours, which are usually the problematic cases. This is done by adapting the event length dynamically. This event length is preserved in the resulting pattern data (Fig. 1b), and can later be investigated visually, like in the arc view (Section 6, Fig. 5).

In Fig. 1b, the weekend is still found, but represented by considerably less patterns. Therefore, these patterns do not clutter the possible case that something unusual happens, for example a Thursday with low traffic, as shown in Fig. 1c. We will further discuss the effect over the course of this publication, after our procedure has been explained in detail.

To gain the maximal benefit from this new procedure, it has to be integrated in an interactive visual interface, as the limitations of related work can be greatly reduced, but not fully solved by automated methods. We explain the details about this in Section 4. Visualizations of output from our new procedure provide more insights than visualizations of output from related work (Section 6), as they are less cluttered.