Knowledge-Based Systems 94 (2016) 23–32

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Pattern discovery via constraint programming

Jian Xu [a], Liang Tang [b], Chunqiu Zeng [b], Tao Li [b,c,*]

[a] School of Computer Science and Engineering, Nanjing University of Science& Technology, China
[b] School of Computing and Information Sciences, Florida International University, United States
[c] School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, China

ARTICLE INFO

Article history:
Received 17 April 2015
Revised 23 October 2015
Accepted 30 October 2015
Available online 2 December 2015

Keywords:
Temporal dependency
Lag interval
Event mining

ABSTRACT

Pattern discovery is one of the most fundamental problems in data mining. Various patterns with their discovering algorithms are proposed in different applications and domains. There is still a great demand for defining new meaningful patterns with new requirements since every application has its unique characteristics. Existing studies propose new query languages to describe these ad-hoc patterns. However, most of them focus on small variations of frequent item sets and association rules. Many meaningful patterns in other domains, such as temporal and spatial patterns, are not covered. This paper proposes a constraint based view for pattern discovery without introducing new languages, where the patterns are described by a collection of constraints given at run time. In this view, a pattern discovery problem is seen as a constraint satisfaction problem. This view provides a general framework for universal pattern discovery. Many previously known patterns can be regarded as different variations derived from this general framework with different constraints. Two generic algorithms are proposed for solving the constraint satisfaction problem. Empirical evaluation on two well-studied patterns shows that (1) the time cost of one generic algorithm is close to that of those specialized mining algorithms, and (2) the space cost of the generic algorithm increases linearly according to the input data volume. Two other case studies also demonstrate the effectiveness of this constraint based view for solving new problems in new scenarios.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Pattern discovery is an important approach to investigate the hidden characteristics of data. A vast number of patterns are proposed and applied in the data analysis for business, bioinformatics, security and other domains [32,41]. But these patterns may not be able to meet the requirements of new applications since every application has its uniqueness [38]. To handle the ad-hoc pattern discovery problem, existing approaches regard every pattern as a query and develop new querying languages to support the query [6,10,14,17,21,25,28,29,35,37,44]. Some of these queries are in form of constraints [3–5,18,25], so executing them we are dealing with a constraint programming problem [39]. Two main limitations exist in these previous studies: (1) new operators and statements are often introduced to the well-established querying languages (e.g., SQL and logic query) to make them complex; (2) most of the studies still only focus on mining frequent item sets and association rules (or with small variations), but many complex patterns, such as temporal and spatial patterns, are not investigated.

In this paper, we present a new constraint based view for pattern discovery without introducing new querying languages. Patterns are defined by a set of constraints that are given by users at run time. Each constraint states a relation of related data subsets. The relation is formulated by an inequality with respect to the aggregated values of these data subsets. These related data subsets are specified using predefined relational queries with some unknown variables. The desired patterns are represented by the unknown variables. In this view, the pattern discovery problem is to find the feasible values of the unknown variables such that the data subsets returned by the relational queries satisfying the given constraints. All relational queries are only expressed by the traditional relational algebra. This view provides a general framework for universal pattern discovery. Many previously known patterns can be seen as different variations derived from this general framework with different constraints. Thus the framework also provides an elegant basis to establish the connections between patterns while highlighting their differences. To demonstrate the generality of our work, in this paper, we present various real-world patterns by using this view, which include frequent item set [41], temporal dependency [43], co-location pattern [19], spatio-temporal co-occurrence pattern [8], emerging

Categories and Subject Descriptors: H.2.8 [Database Management], Database Application-Data Mining.

* Corresponding author at: School of Computing and Information Sciences, Florida International University, United States. Tel.: +1 305 348 6036; fax: 1 305 348 3549.
E-mail address: taoli@cs.fiu.edu (T. Li).

http://dx.doi.org/10.1016/j.knosys.2015.10.031
0950-7051/© 2015 Elsevier B.V. All rights reserved.

pattern [11], iceberg cube [13], bursty tweet phrase, and a fraud pattern in health care.

The contributions of this paper are summarized:

- Proposing a new constraint based view for pattern discovery problem without introducing new query languages and demonstrating its generality by presenting various real-world patterns in different applications.
- Developing two generic algorithms for solving the pattern discovery problem in the proposed view.
- Conducting experiments on two well-studied patterns and two case studies to evaluate the performances of the generic algorithms and the effectiveness of the proposed view.

The rest of the paper is organized as follows: In Section 2 first we discuss the constraint representations of various existing patterns and then propose a unified view for pattern discovery. In Section 3, we introduce two generic algorithms for solving the unified view. In Section 4, we present our empirical studies based on two existing patterns and two new patterns. In Section 5, we summarize the state-of-the-art research studies which are related to ad-hoc pattern discovery problems. Finally, in Section 6, our conclusions and future works are presented.

2. Constraint representation

In this section, we present several existing patterns by using inequality constraints and then introduce a unified constraint based view for pattern discovery problem. The relational algebra is used in these constraints. The common operators and notations in relational algebra are summarized in Table 1.

2.1. Representations for existing patterns

In this subsection, we first introduce many previously known patterns, including frequent patterns, dependent patterns, co-location patterns, and emerging patterns. Admittedly, there are a few recently-proposed patterns, such as correlation patterns between time series and events [24], partially ordered patterns, and jumping patterns [12,22,32,36] are not considered here.

2.1.1. Frequent item set

$R(tid, item)$ is a relational table of a set of transactions, where *tid* is transaction identifier, *item* is an item that is contained by the transaction *tid*, $tid = 1, \ldots, n$, $item = 1, \ldots, d$, n is the number of transactions, and d is the number of item types. Given a minimum support min_{sup}, finding the k-frequent item sets (i.e., k-itemsets whose supports are no less than min_{sup}) is equivalent to solving the inequality:

$$F_{c(tid^{(1)})}(\sigma_{item^{(1)}=x_1 \wedge \cdots \wedge item^{(k)}=x_k}(R^{(1)} \bowtie_\phi \cdots \bowtie_\phi R^{(k)})) \geq n \cdot min_{sup},$$

where $x_1, \ldots, x_k$ are unknown variables, $x_1, \ldots, x_k \in \{1, \ldots, d\}$, and F_c is the distinct *COUNT* aggregation operator (see Table 1). Here ϕ is the join condition, for any two relations $R^{(j)}$ and $R^{(l)}$, $l, j = 1, \ldots, k$,

$$\phi : tid^{(j)} = tid^{(l)} \wedge item^{(j)} < item^{(l)}.$$

The k-frequent item sets are the solutions for this inequality, i.e., $\{x_1, x_2, \ldots, x_k\}$. To obtain all frequent item sets, we can enumerate $k = 1, \ldots, d$.

2.1.2. Temporal dependency

$R(type, t, id)$ is a relational table of an event sequence, where each data tuple is an observation, *type* is the event type, t is the time stamp of this observation, and *id* is the unique observation identifier. Note that $id = 1, \ldots, n$, where n is the number of observations. Given a minimum support threshold min_{sup} and a minimum confidence threshold min_{conf}, finding the pair-wise temporal dependencies (i.e., dependencies between a pair of event sequences satisfying the thresholds) [43] is equivalent to solving the inequalities:

$$F_{c(id^{(1)})}(\sigma_{type^{(1)}=x_1 \wedge type^{(2)}=x_2}(R^{(1)} \bowtie_\phi R^{(2)})) \geq n \cdot min_{sup},$$

$$F_{c(id^{(2)})}(\sigma_{type^{(1)}=x_1 \wedge type^{(2)}=x_2}(R^{(1)} \bowtie_\phi R^{(2)})) \geq n \cdot min_{sup},$$

$$F_{c(id^{(1)})}(\sigma_{type^{(1)}=x_1 \wedge type^{(2)}=x_2}(R^{(1)} \bowtie_\phi R^{(2)})) \geq F_{c(id)}(\sigma_{type=x_1}(R)) \cdot min_{conf},$$

$$F_{c(id^{(2)})}(\sigma_{type^{(1)}=x_1 \wedge type^{(2)}=x_2}(R^{(1)} \bowtie_\phi R^{(2)})) \geq F_{c(id)}(\sigma_{type=x_2}(R)) \cdot min_{conf},$$

where x_1 and x_2 are the unknown variables, x_1 and x_2 belong to the set of the event types. ϕ is the join condition,

$$\phi : 0 \leq t^{(2)} - t^{(1)} \leq w,$$

w is the predefined time window size [26]. The discovered temporal dependencies are the solutions of these inequalities, i.e,. $\{x_1, x_2\}$ or $x_1 \rightarrow_w x_2$ [43].

2.1.3. Co-location pattern

Let $R(id, feature, loc)$ be a relational table of a spatial data set, where each data tuple is an observation, *id* is the unique observation identifier, *feature* is the feature type, and *loc* is the geo-location. Given a minimum *participation index* min_{prev} [19], finding the co-location patterns is equivalent to solving the following relational inequalities:

$$F_{c(id^{(1)})}(\sigma_{feature^{(1)}=x_1 \wedge \cdots \wedge feature^{(k)}=x_k}(R^{(1)} \bowtie_\phi \cdots \bowtie_\phi R^{(k)})) \geq min_{prev} \cdot F_{c(id)}(\sigma_{feature=x_1}(R)),$$

$$\ldots$$

$$F_{c(id^{(k)})}(\sigma_{feature^{(1)}=x_1 \wedge \cdots \wedge feature^{(k)}=x_k}(R^{(1)} \bowtie_\phi \cdots \bowtie_\phi R^{(k)})) \geq min_{prev} \cdot F_{c(id)}(\sigma_{feature=x_k}(R)),$$

where $x_1, \ldots, x_k$ are the unknown variables, $x_1, \ldots, x_k$ belong to the set of features, $k = 1, \ldots, d$, and d is the number of distinct features. Here ϕ is the join condition, for two relations $R^{(j)}$ and $R^{(l)}$, $l, j = 1, \ldots, k$, and k is the number of features in the pattern.

$$\phi : loc^{(j)} \text{ and } loc^{(l)} \text{ are neighbors.}$$

ϕ can be computed by their Euclidean distance with a threshold. The discovered co-location patterns are the solutions of these inequalities, i.e., $\{x_1, \ldots, x_k\}$.

Table 1
Notations.

Notations	Description
$\sigma_\varphi(R)$	Selection, where φ is the selection condition and expressed as a propositional formula, R is a relation.
$\pi_{a_1,\ldots,a_k}(R)$	Projection, where $a_1, \ldots, a_k$ are projection attributes of relation R.
$V(a)$	The value set of attribute a.
$\bowtie_\phi$	Join, where ϕ is the join condition and expressed as a propositional formula.
$\bowtie$	Natural join.
$R_1 \times R_2$	Cross product of two relations, R_1 and R_2.
$F_{(a_f)}(R)$	Aggregation, where a_f is the aggregate attribute, a_f is an attribute of R, R is a relation.
F_c	Distinct *COUNT* aggregation.
$R^{(i)}$	A copy of R, where each attribute a in R is also renamed to $a^{(i)}$ in $R^{(i)}$, i is the copy number.

Download English Version:

https://daneshyari.com/en/article/6862336

Download Persian Version:

https://daneshyari.com/article/6862336

Daneshyari.com