Constraint-Based Association Rule Mining

Carson Kai-Sang Leung
The University of Manitoba, Canada

INTRODUCTION

The problem of association rule mining was introduced in 1993 (Agrawal et al., 1993). Since then, it has been the subject of numerous studies. Most of these studies focused on either performance issues or functionality issues. The former considered *how* to compute association rules efficiently, whereas the latter considered *what* kinds of rules to compute. Examples of the former include the Apriori-based mining framework (Agrawal & Srikant, 1994), its performance enhancements (Park et al., 1997; Leung et al., 2002), and the tree-based mining framework (Han et al., 2000); examples of the latter include extensions of the initial notion of association rules to other rules such as dependence rules (Silverstein et al., 1998) and ratio rules (Korn et al., 1998). In general, most of these studies basically considered the data mining exercise in isolation. They did not explore how data mining can interact with the human user, which is a key component in the broader picture of knowledge discovery in databases. Hence, they provided little or no support for user focus. Consequently, the user usually needs to wait for a long period of time to get numerous association rules, out of which only a small fraction may be interesting to the user. In other words, the user often incurs a high computational cost that is disproportionate to what he wants to get. This calls for *constraint-based association rule mining*.

BACKGROUND

Intuitively, *constraint-based association rule mining* aims to develop a systematic method by which the user can find important association among items in a database of transactions. By doing so, the user can then figure out how the presence of some *interesting* items (i.e., items that are interesting to the user) implies the presence of other interesting items in a transaction. To elaborate, many retailers, such as supermarkets, carry a large number of items. Progress in bar-code technology has made it possible to record items purchased on a per-transaction basis. Each customer purchases one or more items in a given transaction. Types and quantities of different items can vary significantly among transactions and/or customers. Given a database of sales transactions, constraint-based association rule mining helps discover important relationships between the different interesting items so that retailers can learn how the presence of some interesting items in a transaction relates to the presence of other interesting items in the same transaction. The discovered rules reveal the buying patterns in consumer behaviour. These rules are useful in making decisions in applications such as customer targeting, shelving, and sales promotions. Although we describe this problem in the context of the shoppers' market basket application, constraint-based association rule mining is also useful in many other applications such as finding important relationships from financial time series, Web click streams, and biomedical records. When compared with its traditional unconstrained counterpart, constraint-based association rule mining allows the user to express his interest via the use of constraints. By exploiting some nice properties of these constraints, the user can efficiently find association rules that are interesting to him.

More formally, the problem of *constraint-based association rule mining* can be described as follows. Given a database of transactions, each transaction corresponds to a set of items (also known as an *itemset*) that appear together (say, merchandise items that are purchased together by a customer in a single visit to a checkout counter). Constraint-based association rule mining generally consists of two key steps. First, it finds interesting frequent itemsets (i.e., frequent itemsets that satisfy user-specified constraints) from the database of transactions. An itemset is frequent if its frequency exceeds or equals the user-specified minimum frequency threshold. Then, it uses these interesting frequent itemsets to form association rules that satisfy user-specified constraints. Typically, rules are of the form "A➔C" such that both A (which represents the antecedent of the rule) and C (which represents the consequent of the rule) are interesting frequent itemsets.

MAIN FOCUS

Constraint-based association rule mining generally aims to mine association rules that satisfy user-specified constraints, where the antecedent and the consequent of the rules are frequent itemsets that satisfy user-specified constraints. It has several advantages over its traditional unconstrained counterpart. First, it provides *user flexibility* so that the user is able to express his interest by specifying various types of constraints. Second, it leads to *system optimization* so that the computational cost for rules is proportionate to what the user wants to get. Note that, on the surface, it may appear that constraint checking would incur extra computation. However, the constraints can be pushed deep inside the mining process through the exploitation of their nice properties, and thus reducing computation. In the following, we describe *what* types of constraints have been proposed and we also discuss *how* the properties of constraints can be exploited to efficiently find association rules that satisfy the constraints.

Types of Constraints

The user-specified constraints can be categorized according to their types or according to their properties. For constraint-based association rule mining, the user can specify various types of constraints—which include knowledge-type constraints, data constraints, dimension constraints, level constraints, interestingness constraints, and rule constraints (Han & Kamber, 2006). Let us give a brief overview of these types of user-specified constraints as follows:

- **Knowledge-type constraints** allow the user to specify what type of knowledge (e.g., association, correlation, causality) to be discovered.
- **Data constraints** allow the user to specify what set of data (e.g., sales transactions, financial time series, Web click streams, biomedical records) to be used in the mining process.
- **Dimension constraints** allow the user to specify how many dimensions of data to be used when forming rules. By specifying dimension constraints, the user could express his interest of finding one-dimensional rules (e.g., "buy(milk) ➔ buy(bread)" that involves only one dimension "buy"), two-dimensional rules (e.g., "occupation(student) ➔ buy(textbook)" that relates two dimensions "occupation" and "buy"), or multi-dimensional rules.
- **Level constraints** allow the user to specify how many levels of the concept hierarchy to be used in the mining process. By specifying level constraints, the user could express his interest of finding single-level rules (e.g., "milk➔bread" that involves only a single level of the concept hierarchy) or multi-level rules (e.g., "dairy product ➔ Brand-X white bread" that involves multiple levels as (1) milk is a dairy product and (2) the consequent of this rule is a brand of white bread, which in turn is a kind of bread).
- **Interestingness constraints** allow the user to specify what statistical measure (e.g., support, confidence, lift) or thresholds to be applied when computing the interestingness of rules.
- **Rule constraints** allow the user to specify what forms of rules (e.g., what items to be included in or excluded from the rules) to be mined.

Over the past decade, several specific constraints have been proposed for constraint-based association rule mining. The following are some notable examples of rule constraints. For instance, Srikant et al. (1997) considered *item constraints*, which allow the user to impose a Boolean expression over the presence or absence of items in the association rules. The item constraint "(jackets AND shoes) OR (shirts AND (NOT hiking boots))" expresses the user interest of finding rules that either contain jackets and shoes or contain shirts but not hiking boots.

Lakshmanan, Ng, and their colleagues (Ng et al., 1998; Lakshmanan et al., 1999, 2003) proposed a constraint-based association rule mining framework, within which the user can specify a rich set of rule constraints. These constraints include SQL-style aggregate constraints and non-aggregate constraints like domain constraints. *SQL-style aggregate constraints* are of the form "*agg*(S.*attribute*) θ *constant*", where *agg* is an SQL-style aggregate function (e.g., min, max, sum, avg) and θ is a Boolean comparison operator (e.g., =, ≠, <, ≤, ≥, >). For example, the aggregate constraint "min(S.Price) ≥ 20" expresses that the minimum price of all items in an itemset S is at least $20. *Domain constraints* are non-aggregate constraints, and they can be of the following forms: (1) "S.*attribute* θ *constant*", where θ is a Boolean comparison operator; (2) "*constant* ∈ S.*attribute*"; (3) "*constant* ∉ S.*attribute*";