Mining Association Rules from XML Data

Daniele Braga[1], Alessandro Campi[1], Mika Klemettinen[2], and PierLuca Lanzi[3]

[1] Politecnico di Milano, Dipartimento di Elettronica e Informazione,
P.za L. da Vinci 32,I-20133 Milano, Italy
{braga,campi}@elet.polimi.it
[2] Nokia Research Center, P.O.Box 407, FIN-00045 Nokia Group, Finland
mika.klemettinen@nokia.com
[3] Artificial Intelligence and Robotic Laboratory
Politecnico di Milano, Dipartimento di Elettronica e Informazione
pierluca.lanzi@polimi.it

Abstract. The eXtensible Markup Language (XML) rapidly emerged as a standard for representing and exchanging information. The fast-growing amount of available XML data sets a pressing need for languages and tools to manage collections of XML documents, as well as to *mine interesting information* out of them. Although the data mining community has not yet rushed into the use of XML, there have been some proposals to exploit XML. However, in practice these proposals mainly rely on more or less traditional relational databases with an XML interface. In this paper, we introduce association rules from native XML documents and discuss the new challenges and opportunities that this topic sets to the data mining community. More specifically, we introduce an *extension of XQuery* for mining association rules. This extension is used throughout the paper to better define association rule mining within XML and to emphasize its implications in the XML context.

1 Introduction

Knowledge discovery in databases (KDD) deals with the extraction of *interesting* knowledge from large amounts of data usually stored in databases or data warehouses. This knowledge can be represented in many different ways such as clusters, decision trees, decision rules, etc. Among the others, association rules [4], proved effective in discovering interesting relations in massive amounts of data.

During the recent years, we have seen the dramatic development of the eXtensible Markup Language (XML) [19] as a standard for representing and exchanging information. Accordingly, there is a pressing need for languages and tools to manage collections of XML documents and to *extract interesting knowledge* from XML documents. At the moment, the use of XML within the data mining community is still quite limited. There are some proposals to exploit XML within the knowledge discovery tasks, but most of them still rely on the traditional relational framework with an XML interface. However, the pressure for data mining tools for native XML data is rapidly increasing.

Y. Kambayashi, W. Winiwarter, M. Arikawa (Eds.): DaWaK 2002, LNCS 2454, pp. 21–30, 2002.

In this paper, we introduce association rules from *native* XML documents. This extension arises nontrivial problems, related to the hierarchical nature of the XML data model. Consequently, most of the common and well-known abstractions of the relational framework need to be adapted to and redefined in the specific XML context.

2 Association Rules

Association rules (ARs) were first introduced in [3] to analyze customer habits in retail databases; up-to-date definitions can be found in [8]. An association rule is an implication of the form $X \Rightarrow Y$, where the rule *body* X and *head* Y are subsets of the set $\mathcal{I}$ of *items* ($\mathcal{I} = \{I_1, \ldots, I_n\}$) within a set of *transactions* $\mathcal{D}$. A rule $X \Rightarrow Y$ states that the transactions T ($T \in \mathcal{D}$) that contain the items in X ($X \subset T$) are *likely* to contain also the items in Y ($Y \subset T$). ARs are usually characterized by two statistical measures: *support*, which measures the percentage of transactions that contain the items in both X and Y, and *confidence*, which measures the percentage of transactions that contain the items in X and also contain the items in Y. More formally, given the function $\mathit{freq}(X, \mathcal{D})$, denoting the percentage of transactions in $\mathcal{D}$ which contains X, we define:

$$\mathit{support}(X \Rightarrow Y) = \mathit{freq}(X \cup Y, \mathcal{D})$$

$$\mathit{confidence}(X \Rightarrow Y) = \mathit{freq}(X \cup Y, \mathcal{D}) / \mathit{freq}(X, \mathcal{D})$$

Suppose there is an AR "*bread,butter* $\Rightarrow$ *milk*" with confidence 0.9 and support 0.05. The rule states that customers who buy *bread* and *butter*, also buy *milk* in 90% of the cases and that this holds in 5% of the transactions. The problem of mining ARs from a set of transactions $\mathcal{D}$ consists of generating all the ARs that have support and confidence greater than two user-defined thresholds: minimum support (*minsupp*) and minimum confidence (*minconf*).

To help the specification of complex AR mining tasks, a number of query languages have been proposed (see [8] for a review). In particular, [13] introduced the `MINE RULE` operator, an extension of SQL specifically designed for modeling the problem of mining ARs from relational databases. The `MINE RULE` operator captures the high-level semantics of AR mining tasks and allows the specification of complex mining tasks with an intuitive SQL-like syntax. For instance, consider the table in Figure 1, contains the purchase records from a clothing store. The transaction column (`tr`) contains an identifier of the transaction; the other columns correspond to the `customer` identifier, the type of purchased `item`, the `date` of the purchase, the unitary `price`, and the purchase quantity (`qty`). Suppose that we want to mine ARs with a minimum support of 0.1, a minimum confidence of 0.2, at most four items in the body, and exactly one item in the head. Using `MINE RULE`, this task is formalized by the following statement: