Towards Generic Pattern Mining*

Mohammed J. Zaki**, Nagender Parimi, Nilanjana De, Feng Gao, Benjarath Phoophakdee, Joe Urban, Vineet Chaoji, Mohammad Al Hasan, and Saeed Salem

Computer Science Department,
Rensselaer Polytechnic Institute, Troy NY 12180

Abstract. Frequent Pattern Mining (FPM) is a very powerful paradigm for mining informative and useful patterns in massive, complex datasets. In this paper we propose the Data Mining Template Library, a collection of generic containers and algorithms for FPM, as well as persistency and database management classes. DMTL provides a systematic solution to a whole class of common FPM tasks like itemset, sequence, tree and graph mining. DMTL is extensible, scalable, and high-performance for rapid response on massive datasets. Our experiments show that DMTL is competitive with special purpose algorithms designed for a particular pattern type, especially as database sizes increase.

1 Introduction

Frequent Pattern Mining (FPM) is a very powerful paradigm which encompasses an entire class of data mining tasks. The specific tasks encompassed by FPM include the mining of increasingly complex and informative patterns, in complex structured and unstructured relational datasets, such as: Itemsets or co-occurrences [1] (transactional, unordered data), Sequences [2, 29] (temporal or positional data, as in text mining, bioinformatics), Tree patterns [30, 3] (XML/semistructured data), and Graph patterns [12, 16, 26, 27] (complex relational data, bioinformatics). Figure 1 shows examples of these different types of patterns; in a generic sense a pattern denotes links/relationships between several objects of interest. The objects are denoted as nodes, and the links as edges. Patterns can have multiple labels, denoting various attributes, on both the nodes and edges.

The current practice in frequent pattern mining basically falls into the paradigm of incremental algorithm improvement and solutions to very specific problems. While there exist tools like MLC++ [15], which provides a collection of algorithms for classification, and Weka [25], which is a general purpose

* This work was supported by NSF Grant EIA-0103708 under the KD-D program, NSF CAREER Award IIS-0092978, and DOE Early Career PI Award DE-FG02-02ER25538.

** We thank Paolo Palmerini and Jeevan Pathuri for their work on an early version of DMTL.

B. Ganter and R. Godin (Eds.): ICFCA 2005, LNAI 3403, pp. 1–20, 2005.

Java library of different data mining algorithms including itemset mining, these systems do not have an unifying theme or framework, there is little database support, and scalability to massive datasets is questionable. Moreover, these tools are not designed for handling complex pattern types like trees and graphs.

Our work seeks to address all of the above limitations. In this paper we describe Data Mining Template Library (DMTL), a generic collection of algorithms and persistent data structures, which follow a generic programming paradigm[4]. DMTL provides a systematic solution for the whole class of pattern mining tasks in massive, relational datasets. The main contributions of DMTL are as follows:

- Isolation of generic containers which hold various pattern types from the actual mining algorithms which operate upon them. We define generic data structures to handle various pattern types like itemsets, sequences, trees and graphs, and outline the design and implementation of generic data mining algorithms for FPM, such as depth-first and breadth-first search.
- Persistent data structures for supporting efficient pattern frequency computations using a tightly coupled database (DBMS) approach.
- Native support for both vertical and horizontal database formats for highly efficient mining.
- Developing the motivation to look for unifying themes such as right-most pattern extension and depth-first search in FPM algorithms. We believe this shall facilitate the design of a single generic algorithm applicable across a wide spectrum of patterns.

One of the main attractions of a generic paradigm is that the generic algorithms for mining are guaranteed to work for **any** pattern type. Each pattern is characterized by inherent properties that it satisfies, and the generic algorithm exploits these properties to perform the mining task efficiently. We conduct several experiments to show the scalability and efficiency of DMTL for different pattern types like itemsets, sequences, trees and graphs. Our results indicate that DMTL is competitive with the special purpose algorithms designed for a particular pattern type, especially with increasing database sizes.

2 Preliminaries

The problem of mining frequent patterns can be stated as follows: Let $\mathcal{N} = \{x_1, x_2, \ldots, x_{n_v}\}$ be a set of n_v distinct nodes or vertices. A pair of nodes (x_i, x_j) is called en edge. Let $\mathcal{L} = \{l_1, l_2, \ldots, l_{n_l}\}$, be a set of n_l distinct labels. Let $L_n : \mathcal{N} \rightarrow \mathcal{L}$, be a node labeling function that maps a node to its label $L_n(x_i) = l_i$, and let $L_e : \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{L}$ be an edge labeling function, that maps an edge to its label $L_e(x_i, x_j) = l_k$.

It is intuitive to represent a *pattern* P as a graph (P_V, P_E), with labeled vertex set $P_V \subseteq \mathcal{N}$ and labeled edge set $P_E = \{(x_i, x_j) \mid x_i, x_j \in P_V\}$. The number of nodes in a pattern P is called its *size*. A pattern of size k is called a k-pattern, and the class of frequent k-patterns is referred to as F_k. In some applications P is a symmetric relation, i.e., $(x_i, x_j) \equiv (x_j, x_i)$ (undirected edges),