25th International Symposium on Computer and Information Sciences (ISCIS 2010),
Organized Session on Discovery Science (invited), London, UK, 22-24 September 2010 (to appear).

Efficient Algorithms for Discovering Frequent and Maximal Substructures from Large Semistructured Data

Hiroki Arimura

Division of Computer Science, Hokkaido University
N14, W9, Sapporo 060-0814, Japan
Tel: +81-11-706-7678, Fax: +81-11-706-7680
E-mail: arim@ist.hokudai.ac.jp

Abstract. In this paper, we review recent advances in efficient algorithms for *semi-structured data mining*, that is, discovery of rules and patterns from structured data such as sets, sequences, trees, and graphs. After introducing basic definitions and problems, We present efficent algorithms for frequent and maximal pattern mining for classes of sets, sequences, and trees. In particular, we explain general techniques, called the *rightmost expansion* and *PPC-extension*, which are powerful tools for designing efficient algorithms. We also give examples of applications of semi-structured data mining to real world data.

1 Introduction

Data mining. By rapid progress of high-speed networks and large-scale storage technologies in 1990s, a huge amount of electronic data has been available on computers and databases distributed over the Internet. *Knowledge Discovery in Databases* or *Data Mining* [2] is a formal study on efficient methods for discovering interesting rules or patterns in these massive electronic data. The study of data mining started since the early 1990s, quickly expanded in thoery and practice in the late 1990s, and became one of the major branches of computer science and data engineering. Although data mining has its roots in machine learning, statistics, the current data mining technologies focus on efficiency and scalability of mining algorithms as well as identificaition of unknown rules and patterns.

Semi-structured data. Massive electronic data of new types, called *semi-structured data*, have been emerged in the late 1990s [1]. The largest example of semi-structured data is the World Wide Web (WWW), which is the collection of Web pages and XML documents on the Internet, which is sometimes reffered to as the largest collection of knowledge that the human being ever had. Hence, there exist demands for efficient algorithms to extract useful knowledge from these semi-structured data.

Traditionally, data mining mainly deals with well-structured data, e.g., transaction databases or relational databases, which have table-like structures. On the

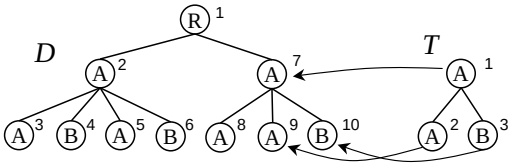

Fig. 1. A data tree D and a pattern tree T on the set $\mathcal{L} = \{A, B\}$ of labels

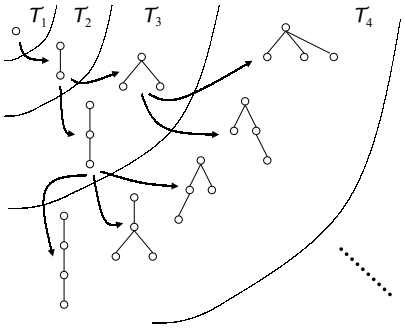

Fig. 2. A search graph for (unlabeled) ordered trees

other hand, these semi-structured data are (i) *huge*, (ii) *heterogeneous* collections of (iii) *weakly-structured* data that do not have rigid structures. Thus, we cannot directly apply these traditional data mining technologies to semi-structured data. For this reason, semi-structured data mining has been extensively studied since 2000.

In this paper, we present efficient semistuructured data mining algorithms for discovering rules and patterns from structured data such as sequence, trees, and graphs. Especially, we describe basic techniques, called *rightmost expansion* and *PPC-extension*, for designing efficient algorithms for frequent and maximal pattern discovery from such semi-structured data.

2 Efficient Frequent Pattern Mining Algorithms

2.1 Frequent Ordered Tree Mining

Tree mining is to find all subtrees appearing more than a specified number of times in a given tree-structured data. We presented an efficient algorithm FREQT [3] that finds all frequent ordered tree paterns in a given tree database. The key is efficient enumeration of labeled ordered trees [3, 25].

In tree mining, data and patterns are modeled by *labeled ordered trees* as shown in Fig. 1. An *ordered tree* over a label alphabet $\Sigma = \{A, B, \ldots\}$ is a rooted tree T where each node x is labeled with a symbol $lab_T(x)$ from Σ, and the order of siblings matters. We denote by V_T and $root_T$ the node set and the root of T, respectively. We denote by $\mathcal{OT}$ and $\mathcal{UT}$ the classes of labeled ordered trees and unordered trees. For ordered trees P and T, we say P matches Q ($P \sqsubseteq Q$) if there exists a *matching function* $\phi : V_P \to V_T$ from P to T that satisfies the following conditions (i) – (iv): (i) ϕ is one-to-one; (ii) ϕ preserves the parent-child relation; (iii) ϕ preserves the sibling relation; (iv) ϕ preserves the node label. Intuitively, P matches T if P is a substructure of T. Then, the node $y = \phi(root_P)$ is called an *occurrence* of P in T. We denote by $\Phi(P, T)$ the set of all matching functions from P to T.

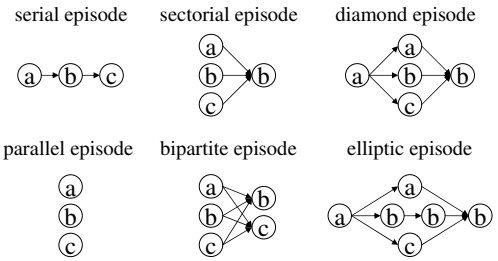

Fig. 3. Examples of subclasses of episodes

Problem. (frequent tree mining) Given an input collection $\mathcal{T} = \{T_1, \ldots, T_m\} \subseteq \mathcal{OT}$ of ordered trees and a nonnegative integers $0 \leq \sigma \leq |T|$ called a *minimum frequency threshold*, find all *frequent* ordered trees $P \in \mathcal{OT}$ appearing in $\mathcal{T}$ with frequency $freq(P, \mathcal{T}) = |\{\ \phi(root_P) : \phi \in \Phi(P, T)\ \}| \geq \sigma$.

A basic idea of the algorithm is to build a spanning tree on the search space of frequent ordered tree patterns, called an *enumeration tree* $\mathcal{E}$ (Fig. 2). By using $\mathcal{E}$, we can systematically enumerate all the distinct ordered tree patterns without duplicates by starting from the empty tree $\perp$ of size 0 and by expanding (or *growing*) an already generated tree of size $k-1$ (a *parent tree*) by attaching a new node to yield a larger tree of size k (a *child tree*) for every $k \geq 1$.

However, a straightforward implementation of this idea leads exponential number of the duplication for one tree. To avoid duplicates, we developed a technique called the *rightmost expansion* ([3, 25]), where attachment of a new node is restricted to only the rightward positions on the rightmost branch of the parent tree. We extended FREQT for frequent unordered tree mining by canonical tree technique [4].

2.2 Frequent Sequence Episode Mining

It is one of the important tasks in data mining to discover frequent patterns from time-related data. Mannila et al. [21] introduced the episode mining to discover frequent episodes in an event sequence. An *episode* is an acyclic labeled digraphs (DAGs) as shown in Fig. 3, where labels correspond to events and arcs represent a temporal precedent-subsequent relation in an event sequence. Classes of episodes are rich representation of temporal relationship in time-series data. Furthermore, we can use additional constraints formulated by a sliding window of a fixed time width.

Mannila et al. [21] presented efficient algorithms for mining classes of *parallel and serial episodes*, which are sets and linear chains of events, respectively. They also considered a mining of general episodes that have DAG structures. Unfortunately, its complexity is rather high due to the inherent computational hardness of subgraph matching. To overcome this difficulty, we presented efficent episode

mining algorithms for subclasses of episodes such as *sectoria*, *diamond*, *elliptic*, and *bipartite episodes* [19, 18]. (Fig. 3). All of these algorithms have polynomial delay and space complexities, and thus they find all frequent episodes in polynomial time per episode with small memory footprint.

3 Efficient Maximal Pattern Mining Algorithms

Maximal Pattern Discovery. Maximal pattern discovery (or *closed* pattern discovery) is one of the most important topics in recent studies of data mining. Assuming a class of patterns and associated partial ordering over patterns indicating a generalization or subsumption order, a *maximal pattern* is such a pattern that is maximal with respect to the subsumption ordering (or the generalization relation) among an equivalence class of patterns having the same set of occurrences in a database.

For some known classes of patterns, such as itemsets and sequence motifs [2], maximal patterns enjoy a nice property that maximal patterns are uniquely determined in each equivalence class of patterns w.r.t. a given database. Also, it is known that the number of frequent maximal patterns is much smaller than that of frequent patterns on most realworld datasets, while the frequent maximal patterns still contain the complete information of the frequency of all frequent patterns. Thus, the complete set of maximal patterns give a compact representation for all frequent patterns. Maximal pattern discovery is useful to increase the performance and the comprehensivity of data mining.

Depth-first Maximal Pattern Discovery algorithms. For maximal pattern discovery, we have developed the following efficient algorithms for finding all maximal patterns from a given collection of data.

- LCM (*Linear-time Closed Itemset Miner*) for maximal sets [24]. (Fig. 4)
- MaxMotif (*Maximal Motif Miner*) for mining maximal sequences [5].
- CloAtt (*Closed Atribute Tree Miner*) for mining maximal trees [6].
- MaxGeo (*Maximal Geometric Subgraph Miner*) for mining maximal geometric graphs [9].
- MaxPicture for mining maximal 2-dimensional subpictures [7].

All algorithms adopt depth-first search strategy unlike the previous maximal pattern algorithms, and are light-weight high-speed mining algorithms that operate in polynomial time per pattern and in polynomial space with respect to the input size only and in dependent of the number of output maximal patterns. For the purpose, we developed as a basic technique for maximal pattern discovery, the PPC-extension (prefix-preserving extension) technique. Fig 4 shows the search structure of PPC-extension in LCM algorithm. For details, see [24].

Recently, we succeeded to give a uniform algorithmic framework [11] for constructing polynomial delay polynomial space algorithms for maximal pattern mining by generalizing the above results including mining closed sequences, graphs, and pictures.

1	1 2 3 4 5 6
2	2 3 5
3	2 5
4	1 2 4 5 6
5	2 4
6	1 4 6
7	3 4 6

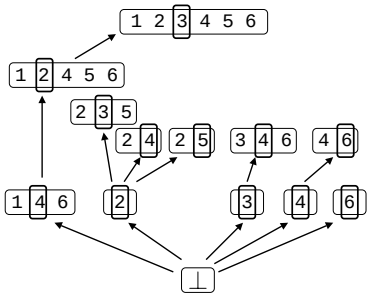

Fig. 4. A transaction database $\mathcal{T}$ on items $\Sigma = \{1, 2, 3, 4, 5, 6\}$ (left), where each row represents a record. All maximal (closed) item sets generated (right), where each arrow indicates a generation of a child from a parent by the PPC-extension.

4 Conlusion

In this talk, we reviewed efficient mining algorithms for large semi-structured data. Finally, we mention applications of semi-structured data mining. Frequent tree miners and optimized tree miners, such as FREQT and OPTT are used to apply standard statistical machine learning techniques, such as *support vector machines* (SVM) and *statistical modeling* to tree and graph structured data [23, 22]. They are also used for the *tree/graph boosting* by extending Boosting algorithms, such as ADABOOST [15], to tree data. We also applied a set of sequential episode mining algorithms to bio-medical data mining, e.g. [19, 20], to extract a collection of episodes representing interaction patterns among a set of antibiotics and bacteria, such as *replacements of bacteria*, in bacterial culture data obtained in the real clinical record data. Further applications will be an interesting future problem.

Acknowledgment. The results presented in this talk are obtained in the joint works with Takeaki Uno, Shin-ichi Nakano, Shin-ich Minato, Tatsuya Asai, Takashi Katoh, and Kouichi Hirata. The author would like to express sincere thanks to them.

References

1. S. Abiteboul, P. Buneman, D. Suciu, *Data on the Web*, Morgan Kaufmann, 2000.
2. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo, Fast discovery of association rules, *Advances in Knowledge Discovery and Data Mining, Chapter 12*, AAAI Press / The MIT Press, 1996.
3. T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto, S. Arikawa, Efficient Substructure Discovery from Large Semi-structured Data, *Proc. SDM'02*, 2002.
4. T. Asai, H. Arimura, T. Uno, S. Nakano, Discovering frequent substructures in large unordered trees, Discovery Science 2003, LNCS 2843, 47–61, 2003.
5. H. Arimura, T. Uno, An efficient polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence, *Journal of Combinatorial Optimization*, 13, 243–262, 2006.

6. H. Arimura, T. Uno, An output-polynomial time algorithm for mining frequent closed attribute trees, *Proc. ILP'05*, LNAI 3625, 1–19, August 2005.
7. H. Arimura and T. Uno, A polynomial space and polynomial delay algorithm for enumerating maximal two-dimensional patterns with wildcards, *Technical Report*, TCS-TR-A-06-19, DCS, Hokkaido Univ., 18 July 2006.
8. H. Arimura, Efficient algorithms for mining frequent and closed patterns from semi-structured data (invited talk), *Proc. PAKDD'08, LNAI 5012*, 2–13, 2008.
9. H. Arimura, T. Uno and S. Shimozono, Time and space efficient discovery of maximal geometric graphs, *Proc. Discovery Science 2007*, LNAI 4755, 42–55, 2007.
10. Hiroki Arimura and Takeaki Uno, Mining Maximal Flexible Patterns in a Sequence, *Proc. LLLL'07*, LNAI 4914, 2008.
11. H. Arimura and Takeaki Uno, Polynomial-delay and polynomial-space algorithms for mining closed sequences, graphs, and pictures in accessible set systems, *Proc. the 9th SIAM Int'l Conf. on Data Mining (SDM2009)*, 1087-1098, 2009.
12. T. Asai, H. Arimura, K. Abe, S. Kawasoe, S. Arikawa, Online algorithms for mining semi-structured data stream, *Proc. ICDM'02*, IEEE, 27–34, 2002.
13. T. Asai, H. Arimura, T. Uno, S. Nakano, Discovering frequent substructures in large unordered trees, *Proc. Discovery Science 2003*,LNAI, Springer, 2003.
14. D. Avis, K. Fukuda, Reverse search for enumeration, *Discrete Applied Mathematics*, 65(1–3), 21–46, 1996.
15. Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci., 55(1): 119-139, 1997.
16. D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen, Data mining, hypergraph transversals, and machine learning, *Proc. PODS'97*, ACM, 209–216, 1997.
17. A. Inokuchi, T. Washio, H. Motoda, Complete mining of frequent patterns from graphs: mining graph data, *Machine Learning*, 50(3), 321–354, 2003.
18. T. Katoh, H. Arimura and K. Hirata, Mining frequent k-partite episodes from event sequences, *Proc. Discovery Science 2009*, LNAI 5808, 136–151, 2009.
19. T. Katoh, H. Arimura and K. Hirata, A polynomial-delay polynomial-space algorithm for extracting frequent diamond episodes from event sequences, *Proc. PAKDD'09*, LNAI 5476, Springer, 172–183, 2009.
20. T. Katoh, K. Hirata, H. Arimura, S. Yokoyama and K. Matsuoka, Extracting sequential episodes representing replacements of bacteria from bacterial culture data, Proc. Complex Medical Engineering 2009, IEEE/ICME, 2009.
21. H. Mannila, H. Toivonen, A. I. Verkamo Discovery of frequent episodes in event sequences, *Data Mining and Knowledge Discovery* 1, 259.289, 1997.
22. S. Morinaga, H. Arimura, T. Ikeda, Y. Sakao, S. Akamine, Key Semantics Extraction by Dependency Tree Mining, Proc. KDD'05, ACM, 666-671, 2005.
23. Koji Tsuda, Taku Kudo, Clustering graphs by weighted substructure mining, ICML 2006, 953–960, 2006.
24. T. Uno, T. Asai, Y. Uchida, H. Arimura, An efficient algorithm for enumerating closed patterns in transaction databases, *Proc. Discovery Science 2004*, LNAI 3245, Springer, 16–30, 2004.
25. M. J. Zaki. Efficiently mining frequent trees in a forest, In *Proc. SIGKDD'02*, ACM, 2002.