Theoretical Computer Science 411 (2010) 2784–2797

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Efficient frequent connected subgraph mining in graphs of bounded tree-width☆

Tamás Horváth [a,b,*], Jan Ramon [c]

[a] Department of Computer Science III, University of Bonn, Germany
[b] Fraunhofer Institute IAIS, Schloss Birlinghoven, Sankt Augustin, Germany
[c] Department of Computer Science, Katholieke Universiteit Leuven, Belgium

ARTICLE INFO

Article history:
Received 24 June 2009
Received in revised form 12 March 2010
Accepted 19 March 2010
Communicated by J. Díaz

Keywords:
Listing algorithms
Frequent patterns
Tree-width
Graph mining
Data mining

ABSTRACT

The frequent connected subgraph mining problem, i.e., the problem of listing all connected graphs that are subgraph isomorphic to at least a certain number of transaction graphs of a database, cannot be solved in output polynomial time in the general case. If, however, the transaction graphs are restricted to forests then the problem becomes tractable. In this paper we generalize the positive result on forests to graphs of *bounded tree-width*. In particular, we show that for this class of transaction graphs, frequent connected subgraphs can be listed in *incremental polynomial time*. Since subgraph isomorphism remains NP-complete for bounded tree-width graphs, the positive complexity result of this paper shows that efficient frequent pattern mining is possible even for computationally hard pattern matching operators.

1. Introduction

During the last decade, *graph mining* developed into a separate field of knowledge discovery in databases, motivated by various practical applications for example in bioinformatics, computational chemistry, and the WWW. A basic task in this field is the *frequent connected subgraph mining (FCSM) problem*: Given a database of labeled graphs, called *transaction* graphs, and some positive integer threshold t, *list* all *connected* graphs that are subgraph isomorphic to at least t transaction graphs. Such frequent connected patterns have successfully been used, for example, in ligand-based virtual screening as features [9].

For arbitrary transaction graphs, the FCSM problem cannot be solved in output polynomial time (if P $\neq$ NP) [13]. While several heuristic methods have been developed for this general problem that proved to be effective on various graph datasets, surprisingly there are only few theoretical results concerning the identification of tractable graph classes. To the best of our knowledge, the only positive (non-trivial) result towards this direction is about forests; the FCSM problem can be solved with polynomial delay, and hence, in incremental polynomial time for forest transaction graphs (see [8] for a survey on tree mining). The exploration of the border between tractable and intractable graph classes is an important theoretical challenge because it could provide useful insights into the problem which could then be exploited in the design of practical algorithms.

In this paper we take a step towards this goal by generalizing the positive result on mining forests in incremental polynomial time to graphs of *bounded tree-width*. *Tree-width* [18] is a measure of tree-likeness of graphs that proved to

☆ A preliminary version of this paper appeared in W. Daelemans, B. Goethals, and K. Morik (Eds.): Machine Learning and Knowledge Discovery in Databases, European Conference (ECML/PKDD), Proceedings, Part I., vol 5211 of LNAI, Springer, Heidelberg, 2008, pp. 520–535.

* Corresponding address: Fraunhofer Institute IAIS, Schloss Birlinghoven, Sankt Augustin, Germany. Tel.: +49 2241 142584.
E-mail addresses: tamas.horvath@iais.fraunhofer.de (T. Horváth), jan.ramon@cs.kuleuven.be (J. Ramon).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.03.030

be a useful property in algorithmic graph theory; several NP-hard problems on graphs become tractable for the class of bounded tree-width graphs. This class is also of practical importance, as it includes many graph classes (see, e.g., [4,6]) that appear in practical applications. For example, the molecular graphs of the vast majority of pharmacological compounds have tree-width at most 3.

We present a levelwise search algorithm listing frequent connected subgraphs in incremental polynomial time if the tree-width of the transaction graphs is bounded by some constant. To avoid redundant pattern generation, we make use of the fact that isomorphism between graphs of bounded tree-width can be decided efficiently [7]. To calculate the support count of candidate patterns, we use a modification of the subgraph isomorphism algorithm developed for graphs of bounded tree-width and *k-log-bounded fragmentation* [12], where the class of k-log-bounded fragmentation graphs properly contains the class of *bounded degree* graphs. This algorithm is based on a fundamental generic algorithm designed for deciding various morphisms between graphs of bounded tree-width and bounded degree [17]. In a nutshell, the main result of [17] is that several graph morphisms, including subgraph isomorphism, can be decided efficiently by a dynamic programming algorithm computing a certain set of polynomially many, polynomial time computable properties (tuples) used to decide the underlying graph morphism if the tree-width and the degree of the graphs are both bounded by some constant.

Since we do not assume any bound on the degree, the number of such properties can be *exponentially* large. We can show, however, that for a given candidate pattern H, it is sufficient to compute only a *polynomially* large subset of these properties; the rest, maybe exponentially large set, can be derived from those of the frequent subgraphs listed before H. To show this result, we utilise the levelwise generation of frequent patterns and the anti-monotonic property of frequency. In this way, the delay can be exponential in the size of the input only after the enumeration of exponentially many frequent patterns. This technique might be of some independent interest and useful to design efficient pattern mining algorithms where straightforward dynamic programming would require exponential space.

We note that subgraph isomorphism remains NP-complete even for connected graphs of bounded tree-width (see, e.g., [17]). A significant consequence of the positive result of this paper is thus immediate to the study of mining frequent patterns: *Efficient frequent pattern mining is possible even for NP-hard pattern matching operators.*

The rest of the paper is organised as follows. In Section 2 we first collect the necessary notions and fix the notations. In Section 3 we present a generic levelwise search algorithm mining frequent connected subgraphs and analyse its computational properties. In Section 4 we adapt this generic algorithm to graphs of bounded tree-width and show that it lists frequent connected subgraphs in incremental polynomial time. Finally, in Section 5, we conclude and discuss an open problem.

2. Preliminaries

In this section we first briefly review some basic concepts and fix the notations used in this paper. We start with some standard definitions from graph theory.

Graphs. An *undirected graph* is a pair (V, E), where V is a finite set of *vertices* and $E \subseteq \{e \subseteq V : |e| = 2\}$ is a set of *edges*. A *labeled undirected graph* is a triple (V, E, λ), where (V, E) is an undirected graph and $\lambda : V \cup E \to \mathbb{N}$ is a function assigning a label to every element of $V \cup E$. Unless otherwise stated, in this paper by graphs we always mean *labeled undirected* graphs and denote the set of vertices, the set of edges, and the labeling function of a graph G by $V(G)$, $E(G)$, and λ_G, respectively.

Let G and G' be graphs. Then G' is a *subgraph* of G, if $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$, and $\lambda_{G'}(x) = \lambda_G(x)$ for every $x \in V(G') \cup E(G')$; it is an *induced subgraph* of G if it is a subgraph of G satisfying $\{u, v\} \in E(G')$ if and only if $\{u, v\} \in E(G)$ for every $u, v \in V(G')$. For a subset $S \subseteq V(G)$, $G[S]$ denotes the induced subgraph of G with vertex set S.

A *path* connecting the vertices $v_1, v_k \in V(G)$ in a graph G is a sequence $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\} \in E(G)$ such that the v_i's are pairwise distinct. A graph is *connected* if there is a path between any pair of its vertices. A *connected component* of a graph G is a maximal subgraph of G that is connected. The set of all connected components of a graph G is denoted by $\mathcal{C}(G)$.

Isomorphism and subgraph isomorphism. Let G_1 and G_2 be graphs. They are *isomorphic* if there is a *bijection* $\varphi : V(G_1) \to V(G_2)$ satisfying

(i) $\{u, v\} \in E(G_1)$ if and only if $\{\varphi(u), \varphi(v)\} \in E(G_2)$ for every $u, v \in V(G_1)$,
(ii) $\lambda_{G_1}(u) = \lambda_{G_2}(\varphi(u))$ for every $u \in V(G_1)$, and
(iii) $\lambda_{G_1}(\{u, v\}) = \lambda_{G_2}(\{\varphi(u), \varphi(v)\})$ for every $\{u, v\} \in E(G_1)$.

For G_1 and G_2 we say that G_1 is *subgraph isomorphic* to G_2, denoted $G_1 \preccurlyeq G_2$, if G_1 is isomorphic to a subgraph of G_2. Deciding whether a graph is subgraph isomorphic to another graph is NP-complete, as it generalizes the Hamiltonian path problem [10].

Tree-width. The notion of tree-width was reintroduced in [18]. It proved to be a useful parameter of graphs in algorithmic graph theory. A *tree-decomposition* of a graph G, denoted $TD(G)$, is a pair $(T, \mathcal{X})$, where T is a rooted unordered tree and $\mathcal{X} = (X_z)_{z \in V(T)}$ is a family of subsets of $V(G)$ satisfying