Frequent Subgraph Mining with "Think like an embedding"-Approach

Nguyen Duc Hieu

May 9, 2016

Contents

1 Introduction

2 Problems of Frequent Subgraph Mining

In this section we will get introduced to Frequent Subgraph Mining (FSM). The goal of data mining is to extract inferences and information from given data. In FSM we treat particular data which are represented by using graphs. For FSM there are two types of problem formulation: **graph transaction based FSM** and **single graph based FSM**. In graph-transaction based FSM the input is a collection of medium-sized graphs called *transactions*.

Let $\boldsymbol{g}$ b a subgraph. g is considered frequent if its *occurrence count* is greater than some predefined threshold. The occurrence count for a subgraph is also referred as its *support* and the threshold is referred to as the *support threshold*. The support of g is being computed by using either *transaction-based counting* or *occurrence-based counting*. Transaction-based counting is only being used for transaction based FSM, while occurrence-based counting can be used for either transaction based FSM or single graph based FSM.

In transaction-based counting the support is defined by the number of graph transactions in that g occurs. It is irrelevant whether g occurs one or multiple times in a particular graph transaction. Thus, given a database $\mathcal{G} = \{G_1, G_2, \ldots, G_T\}$

2.1 Terminologies and Formalism

In this subsection we will go through some definitions in order to obtain a better understanding on the subject FSM. Generally a graph is defined by a set of vertexes and a set of edges which represent the interconnections between the vertexes. The graphs used for FSM are assumed to be *labelled simple graphs*. In the following

Labelled Graph:
Let $G(V, E, L_V, L_E, \varphi)$ be a labelled graph. V is a set of vertexes and $E \subseteq V \times V$ is a set of edges; L_V and L_E are sets of vertex and edge labels; and φ is a label function that defines the mappings $V \to L_V$ and $E \to L_E$.

Subgraph:
Let two labelled graphs be given: $G_1(V_1, E_1, L_{V_1}, L_{E_1}, \varphi_1)$ and $G_2(V_2, E_2, L_{V_2}, L_{E_2}, \varphi_2)$. G_1 is a subgraph of G_2, if G_1 satisfies following points:
(i) $V_1 \subseteq V_2$ and $\forall v \in V_1, \varphi_1(v) = \varphi_2(v)$,
(ii) $E_1 \subseteq E_2$ and $\forall (u, v) \in E_1, \varphi_1(u, v) = \varphi_2(u, v)$.
G_2 is also a supergraph of G_1.

Graph Isomorphism:
A graph $G_1(V_1, E_1, L_{V_1}, L_{E_1}, \varphi_1)$ is isomorphic to another graph $G_2(V_2, E_2, L_{V_2}, L_{E_2}, \varphi_2)$, if and only if a bijection $f : V_{1 \to V_2}$ exists such that:
(i) $\forall u \in V_1, \varphi_1(u) = \varphi_2(f(u))$,
(ii) $\forall (u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$,
(iii) $\forall (u, v) \in E_1, \varphi_1(u, v) = \varphi_2(f(u), f(v))$.
The bijection f is an isomorphism between G_1 and G_2. A graph G_1 is **subgraph isomorphic** to a graph G_2, if and only if there exists a subgraph $g \subseteq G_2$ such that G_1 is isomorphic to g. In this case g is called an **embedding** of G_1 in G_2.

Lattice:
Given a database $\mathcal{G}$, a lattice is a structural form used to model the search space for finding frequent subgraphs, where each vertex represents a connected subgraph of the graph in $\mathcal{G}$. A vertex p is a patent of the vertex q in the lattice, if q is a child of p. All the subgraphs of each graph $G_i \in \mathcal{G}$ which ouccur in the database are present in the lattice and every subgraph occurs only once in it.

2.2 Overview of FSM

In this section we will have an overview of the process of FSM. The methods of FSM can be divided into two categories: (i) Apriori-based approaches and (i) the pattern growth-based approaches. The Apriori-based approach uses a Breadth First Search (BFS) strategy to explore the subgraph lattice of a data set, while the pattern growth-based one uses the Depth First Search (DFS) strategy. The Apriori-based approach has to check all k-level subgraphs before considering *(k+1)*-level subgraphs. The pattern growth-based approach searches for subgraphs by extending each discovered subgraph g which is extended recursively until all frequent subgraphs are discovered.

2.2.1 Canonical Representation

There are several methods of representing a graph structure. Two simple and most commonly used ones are the use of *adjacency matrix* and *adjacency list.*

3 Pregel and MapReduce

4 Arabesque