Simulation-based Approximate Graph Pattern Matching

Xiaoshuang Chen
†*UNSW Sydney*

Motivation

Conventional algorithms for graph pattern matching is based on the subgraph isomorphism, where exact matches are the subgraphs of the data graph that are isomorphic to the query graph. However, there are some drawbacks.

- ❖ Huge computation complexity as subgraph isomorphism test is NP-complete
- ❖ Hard to come up with a query that exactly conforms with the structures in the data graph due to data noises.
- ❖ Too restrictive to capture those reasonable but not exact matches.

Our Solutions

We propose a new simulation-based approximate pattern matching algorithm that is not only efficient to compute, but also capture the reasonable results. Our work contains two parts, namely fractional simulation and top-k results retrieval, shown as follows.

➢ Fractional Simulation

Given node u in the query graph and node v in the data graph, the fractional simulation score between u and v, denoted as FSim(u, v), is calculated by

$$FSim(u,v) = w^{*}\mathrm{I}(u,v) + \frac{w^{+}\sum_{(x,y)\in M(N^{+}(u),N^{+}(v))} FSim(x,y)}{|N^{+}(u)|} + \frac{w^{-}\sum_{(x,y)\in M(N^{-}(u),N^{-}(v))} FSim(x,y)}{|N^{-}(u)|} \quad (1)$$

where $\mathrm{I}(\cdot)$ is an indicator function that is 1 if u and v have the same label, and 0 otherwise; $N^{+}(u)$ and $N^{-}(u)$ indicate u's out-neighbors and in-neighbors, respectively; $|N^{+}(u)|$ and $|N^{-}(u)|$ denote the size of the related node set; and w^{*}, w^{+} and w^{-} are the weighting factors satisfying w^{*}, $w^{+} \geq 0$, $w^{-} \geq 0$ and $w^{*} + w^{+} + w^{-}$; and M returns a set of node pairs, which is defined as follows.

$$M(S_1, S_2) = \{(x,y) | x \in S_1, y = \mathrm{argmax}_{y' \in S_2} FSim(x, y')\} \quad (2)$$

Conclusion: $FSim(u,v) = 1$ **if and only if** there exists a simulation relation between node u and node v.

➢ Top-K Results Retrieval

We then detail how to retrieve the top-k matches in a data graph for a pattern graph. We first define the subgraph matching gain function $Ga(\varphi)$ as:

$$Ga(\varphi) = \sum_{u \in V_q} FSim(u, \varphi(u)) \quad (3)$$

where φ is a matching satisfying φ: $V_q \rightarrow V_G$. Consequently, we can return the subgraphs in the data graph with the top-k highest gain values based on equation (3)

Heuristically, we can retrieve the top-k matches in the following. First, we select some candidates with high simulation scores for each node in the query graph. Then, we can adopt the backtracking algorithm to compute the matched subgraphs (allow mismatching edges). Finally, the matched subgraphs with top-k highest Ga() scores are returned as results.

Existing Solutions and Drawbacks

Approximate subgraph matching is explored to address the drawbacks of the isomorphism-based subgraph pattern matching algorithms. We briefly introduce the approximate subgraph matching algorithms and show their limits in the following.

➢ Edit-Distance-based Algorithms

Intuition: Given a query graph Q, algorithms in this category first enumerate all connected query graphs Q', where Q' is obtained by eliminating certain number (a given threshold) of edges. Then, the algorithms will return all the subgraphs in the data graph G that are isomorphic to Q'.

Drawback: Algorithms in this category are computationally harder than the problem of exact graph pattern matching, given that the problem degenerates to exact matching when the given threshold is equal to 0.

➢ Similarity-based Algorithms

Intuition: The algorithms in this category are mainly based on structure similarity and label similarity. Intuitively, nodes in the query graph Q are matched with the nodes in the data graph with larger similarity values. Typical algorithms include NeMa, VELSET and NAGA.

Drawback: algorithms in this category are generally too costly to scale.

➢ Simulation-based Algorithms

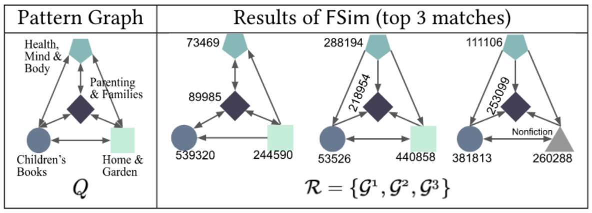
Intuition: The algorithms in this category are to find data nodes as candidates that are simulated by the query nodes (can be calculated in polynomial time), and use the candidates to construct the matched instances.

Drawback: algorithms in this category cannot capture the matches that are nearly (but not exactly) simulated to the query graph.

Experiments

We use the Amazon co-purchasing network to evaluate the performance of the proposed FSim (equation 1) in finding matches. The network contains 554,790 nodes, 1,788,725 edges and 82 node labels. Top-3 results of FSim for query Q (labels are the categories of the books) are shown in Table 1. Note that existing simulation-based algorithms in [1][2] are all fail to retrieve any results as there exists no exact simulation between the query graph and the data graph. In comparison, Fsim is still capable to capture some closely matched results based on Table 1.

Table 1: The results of pattern matching



[1] Ma, S., Cao, Y., Fan, W., Huai, J. and Wo, T., 2014. Strong simulation: Capturing topology in graph pattern matching. *ACM Transactions on Database Systems (TODS)*, *39*(1), pp.1-46.

[2] Song, C., Ge, T., Chen, C. and Wang, J., 2014. Event pattern matching over graph streams. *Proceedings of the VLDB Endowment*, *8*(4), pp.413-424.