Study of Subgraph Isomorphism and Java Implementation of QuickSI algorithm

Ajinkya A Kale (aak9225@rit.edu)
Advisor: Prof. Carlos R. Rivero
Rochester Institute of Technology

Introduction

- Subgraph Isomorphism is a graph containment problem, which finds the exact matching of the query graph in a database.
- Its applications can be found in various domains such as social network, chemistry, data engineering, etc.
- To solve the subgraph isomorphism problem, an efficient algorithm called QuickSI is presented.
- The goal of this capstone project is to devise an Java Implementation of QuickSI algorithm.

Approach

- To perform exact matching of query over data graph, each vertex of query graph should be mapped to vertex of data graph, such that it satisfies all the properties of query graph vertex.
- A query graph is traversed in a particular sequence to find the mapping of each vertex.
- QI Sequence – Randomized sequence
- Effective QI Sequence – optimal search order
- This sequence is generated by finding the spanning tree of query graph.
- Pseudo edge and degree – An extra edge / degree entry is recorded in the sequence.

QI Sequence

QI Sequence is order in which the vertices are processed. Following table is QI Sequence for Fig 1

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Effective QI Sequence

- Prims algorithm is used to find sequence.
- The graph is converted to weighted graph by calculating the average inner support.
- Spanning edges are selected in 4 steps:
  1. Filtering based on avg. Inner support.
  2. Filtering based on induced subgraph count.
  3. Filtering based on Degrees of vertices.
  4. Vertex is randomly chosen if there are more than 1 candidate vertex.

Select Spanning Edge Algorithm

- Select Spanning Edge ($P$, $a'\bar{v}$): $P$ = edge set of query graph
  $a'\bar{v}$ = weighted query graph edge.
- $a$ = edge set from step 1 filter
  $\beta$ = edge set from step 2 filter
  $\gamma$ = edge set from step 3 filter
- $a = \left\{ e \mid e \in P \cap \nabla \left( e' \in P \cap \nabla (w(e) \leq w(e')) \right) \right\}$
- $\beta = \left\{ e \mid e \in P \cap \nabla \left( e' \in P \cap \nabla (\delta(e, v) \leq \delta(e', v')) \right) \right\}$
- $\gamma = \left\{ e \mid e \in P \cap \nabla \left( e' \in P \cap \nabla (\delta(e, v) \leq \delta(e', v')) \right) \right\}$

QuickSI Algorithm

- It uses QI Sequence of the query graph to find the bijection mapping of data graph vertices.
- It maintains two vectors, that records the information about mapped vertices.
- QuickSI is a backtracking algorithm that explores all the possible paths of the data graphs.
- A vertex is candidate vertex if it forms an edge with parent vertex mapping of T1 and has same edge label and vertex label.

Results and Experiments

- The java implementation of QuickSI was tested on AIDS dataset consisting of 1000 query graphs and 1000 data graphs.
- Each query was matched with 1000 data graphs.
- Total matched results were as follows: 10050 mapping solutions of query graphs over all the data graphs.

Conclusion and Future work

- Successful implementation of QuickSI was devised in Java.
- Implemented QI and Effective QI Sequence.
- The search space for candidate vertices can reduced significantly by grouping the vertices based on their label.
- Study of randomized sequence matching and its performance comparison with Effective QI Sequence.

References

- Jinsoo Lee, Wook-Sun Han, Romans Kasperovics, Jeeong-Soon Lee: An In-depth Comparison of Subgraph Isomorphism Algorithms in Graph Databases. PVLDB 6(2): 133-144 (2012).