FG-Index: Towards Verification-Free Query Processing on Graph Databases

Sharanya Jayaraman

COP 5725 - Adavnced Database Systems

March 07, 2013

Paper

- Title:

  FG-Index: Towards Verification-Free Query Processing on Graph Databases

- Authors:

  James Cheng, Yiping Ke, Wilfred Ng, and An Lu;
  Hong Kong University of Science and Technology.

- Conference:

  2007 ACM SIGMOD International Conference on Management of Data; Beijing, China.

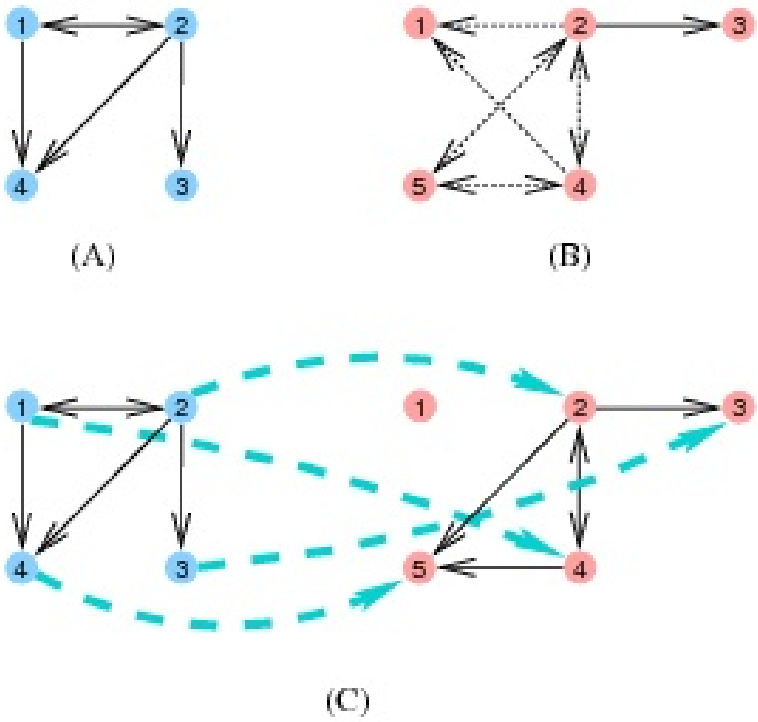
- URL:
  `http://er2004.cse.ust.hk/faculty/wilfred/paper/sigmod07.pdf`

Overview

Problem Definition

Given a Graph Database D = $\{g_1, g_2 \ldots g_N\}$, and a smaller graph q, we retrieve all graphs in D that are supergraphs of q.

The naive approach is to check sequentially, if any g_i has q as the subgraph. This is infeasible, since graph isomorphism is NP-complete.



By indexing some subgraphs, we can filter and verify the candidate set.

Prominent methods include g-Indices, D(k)-Indices, etc.

Problem Definition

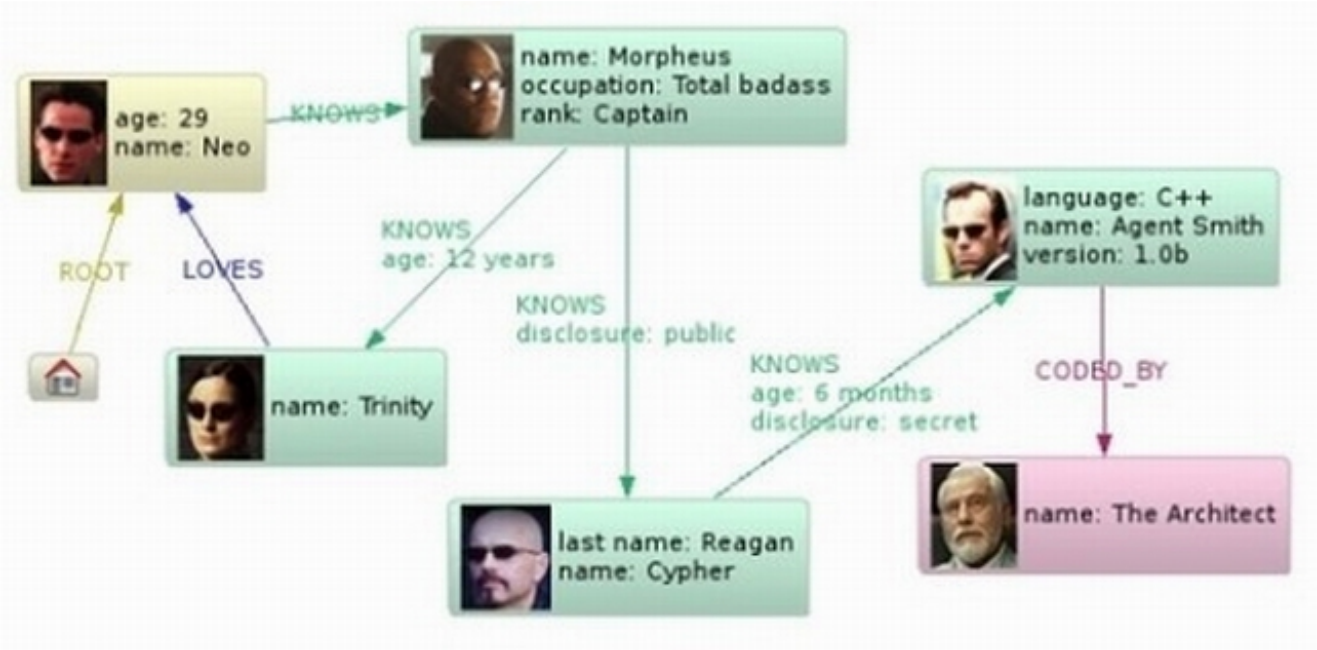
Candidate verification is an expensive step. It involves at least as many isomorphism checks as the size of the answer set.

Can we avoid candidate verification while querying graph databases?

We can use a nested inverted index based on Frequent subGraphs to create an FG-Index.

Applications

- Molecular Structure of Chemical Compounds
- Food Chains, Protein Modeling
- Computer Networks, Social Networks
- ER diagrams, UML diagrams, AI Graphical models
- Automata, Control flow graphs
- Websites, XML documents



Motivation

Why?

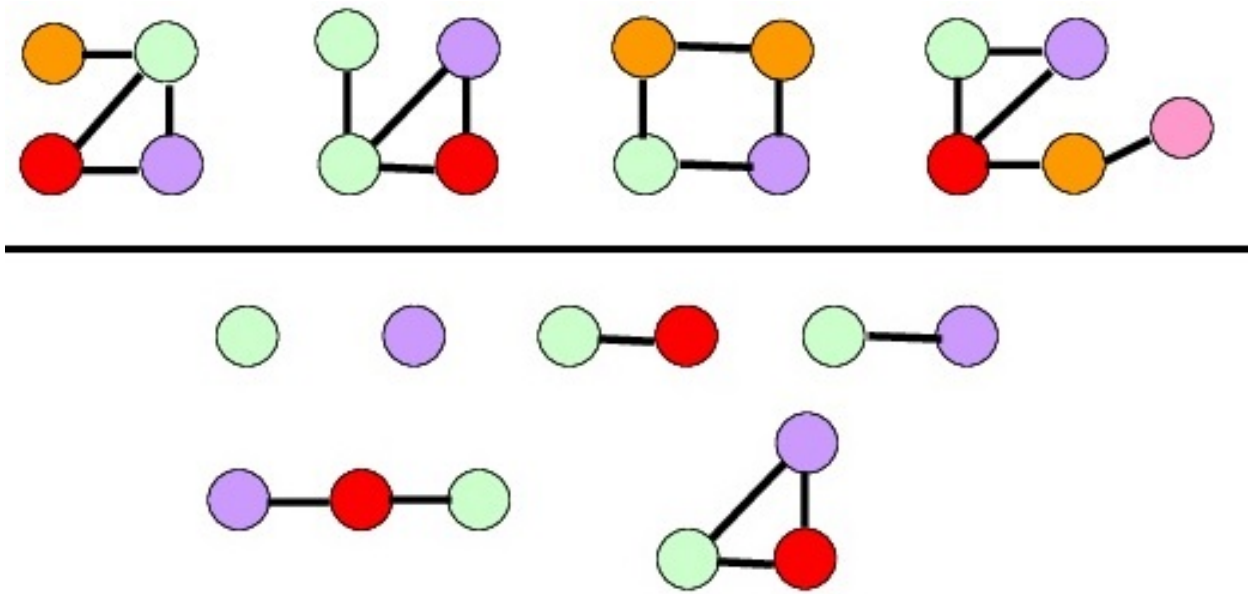
1. Graphs are a convenient way to represent relationships among schema-less data.
2. Graph databases are increasing in size and complexity.
3. No simple way to query for exact matches; indexing is the obvious way forward.
4. Existing indexing methods choose a path based or frequent structure based mechanisms that grow very quickly.

How?

1. Use frequent subgraph mining, a fundamental data mining technique.
2. Compress some similar subgraphs into a supergraph if the index is too big.

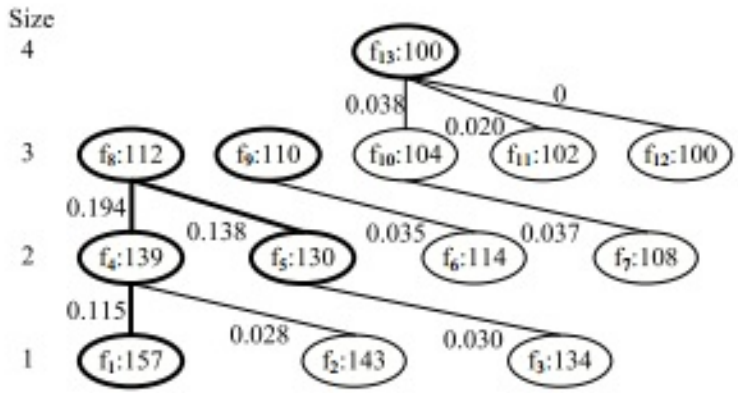
Frequent Subgraph Mining

Extract all subgraphs in a data set, whose occurrence counts are above a threshold.



An FG is a subgraph of at least (σ.|D|) graphs in D. In the example, σ=0.75.

δ-Tolerance Closed Frequent subGraphs (δ-TCFG)

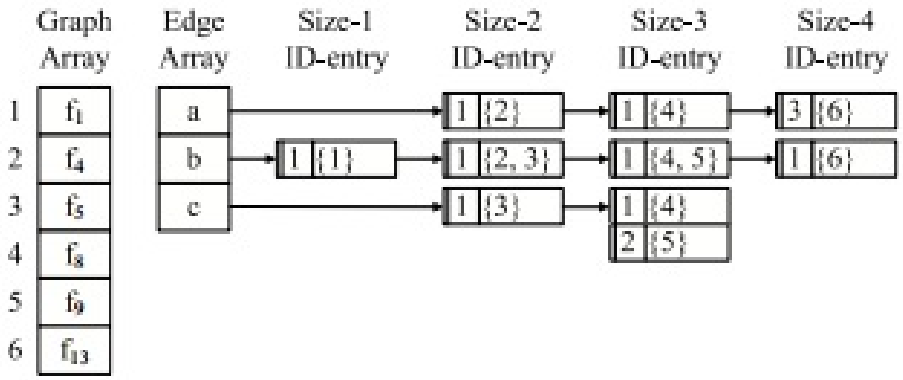


A graph g is a δ-TCFG $\Leftrightarrow$ g $\in$ F and there is no g' $\in$ F such that g'$\supset$ g and freq(g')$\geq$((1δ)freq(g)), where $(0 \leq \delta \leq 1)$

- A Closed Frequency Graph (CFG) is 0-TCFG and a Maximal Frequency Graph is 1-TCFG.
- δ-TCFG's are used to reduce the size of the index by splitting it into 2 levels.

BuildIndex

- FG-index consists of the core FG-index and Edge-index.
- First, a memory-resident inverted-index is constructed on the set τ of δ-TCFG's.
- A disk-resident inverted-index is built on the FGs in the closure of each δ-TCFG.
- Another index, called Edge-index, is built on the set of infrequent distinct edges in D.



FG Query

Frequent subGraph

- Given a query q, we first search it in the core FG-index.
- If q is a δ-TCFG, retrieve q and D_q from the memory-resident index.
- Otherwise, find qs closest δ-TCFG supergraph g's disk-resident index and retrieve D_q.

Infrequent Subgraph

- Get a set of subgraphs S_{core} of q from the core index. Also get a set S_{edge} of infrequent edges from the edge index.
- Compute the intersections to get a candidate set C_q. Perform candidate verification.

Performance

This is a comparison of the perfromace of FG-Index and G-Index on the 43K graph AIDS antiviral screen dataset.

Table 1: Index Construction Performance for *AIDS*

	FG-index ($\sigma = 0.1$)	FG-index ($\sigma = 0.01$)	gIndex
Time (sec)	10.03 (10+0.03)	1085 (627+458)	217
Memory (MB)	2	95	107
$\lvert\mathcal{T}\rvert$ or $\lvert\mathcal{F}_d\rvert$	195	21893	3276

Table 2: Query Performance for *AIDS*

	FG-index ($\sigma = 0.1$)	FG-index ($\sigma = 0.01$)	gIndex
Response Time (sec)	0.031	0.040	0.57
Memory (MB)	2	14	47

Pros and Cons

Pros

1. FG Index performs up to an order of magnitude better on large datasets in comparison with gIndex.
2. Minimization of candidate verification.

Cons

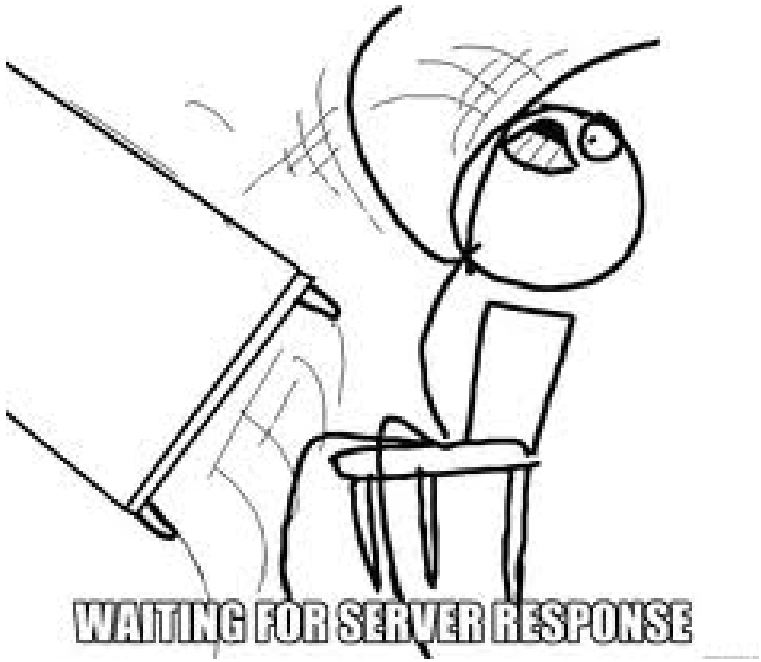
1. Choosing σ and δ carelessly can seriously affect query response times.
2. Index can become large very quickly if graphs are reasonably complex.

Future Work

- Examining the effects of σ and δ in very large graphs.
- Other options include iterative feature mining or using a hybrid index structure for an approximate match.

Take Home

- FG-Indexing increases query performance by minimizing verification in subgraph queries.
- δ-TCFGs can be used to restrict the size of the index in large databases.
- Query response time can be improved by up to an order of magnitude.



Thank You

Questions?