Overview

The paper reviews different types of querying and indexing techniques used in graph databases and provides some directions for future research. It discusses the strengths and weaknesses of each indexing technique. During the discussion, the authors provide a categorization method for indexing techniques based on the query type and whether it uses part or all of the features in the graph, which could be very useful to our term project (developing taxonomy for different indexing).

The paper first introduces some basic concepts that are necessary for understanding graph query and indexing. Types of graph query are emphasized for further classifying indexing methods. There are three types of graph queries:

- subgraph queries
- supergraph queries
- similarity queries

The subgraph queries are used to search whether the graph database contains a certain pattern (subgraph). The supergraph queries are used to search whether the database contains a graph that can be fully incorporated into the input query graph. The similarity queries are used to search graphs that are similar (not necessarily isomorphic) to a given query graph.

The author later introduces the current graph querying framework (with index). Overall, the querying can be divided into three parts:

1. Building: Generates index for database offline
2. Filtering: Uses indexes built from step 1 to eliminate and filter some of the results and generate a potential candidate set.
3. Verification: Checks the result from step 2 one by one and outputs the answer that satisfies the result.

In the next part the paper discusses different types of indexing techniques used in different graph query types (also provides a potential classification methodology for indexing method).

Subgraph query processing indexing method

It contains non-mining-based and mining-based indexing methods[2]**.**

Non mining-based graph indexing: This technique tries to build an index based on the overall database instead of only choosing some selected features, so the comparisons in the filtering process are expensive. But once built, the cost of updating will be less because their performance doesn't rely on the selected features and thus will avoid potential chances of rebuilding indexes.
Examples:

- GDIndex: Indexing using graph decomposition. (in our term project reference list) [3]
- GString, GraphGrep, GCoding, GraphREL [2][4][5][6]

Mining-based indexing: Apply graph mining method to extract some features from the graph database. Inverted index is created for each feature. The more the features are chosen, the better the filtering power (but will cost more space), and vice versa. Generally, it requires less

space compared to non-mining-based graph indexing techniques. However, the quality of index largely depends on the indexing feature chosen, the size and number of indexing features. Also, as the database keeps changing, the quality of the selected features may decrease, forcing it to reconstruct a new index.
Examples:
This category is divided into two sub categories by the paper.

- Graph-based mining: Use subgraph as index
    - GIndex: make use of frequent subgraphs as the basic indexing unit [7]
    - FG-Index: Enhanced version of GIndex. The index stored in memory is built with frequently used subgraphs. The index stored in disk is built with the closure of the frequent graphs. (in our term project reference list) [8]
- Tree-based mining: Use frequent subtrees as the indexing unit.
    - TreePI (in our term project reference list) [9]
    - Tree + △ [10]

Super-graph query processing indexing method

There are very few indexing methods to handle supergraph queries. The author only mentions two.
Examples**:**

- cIndex [11]
- GPTree: For graphs that are dense. [12]

Graph Similarity query processing indexing method

Examples:

- Grafil, PIS, SAGA, TALE - periscope [14], [15], [16]
- Closure tree (in our term project reference list) [13]

Conclusion

In general, the author analyzes types of indexing techniques. The classification shown in the paper can be used for reference by subsequent research on developing index taxonomy. The author also summarized current problems in the graph database area and proposed some research directions in the future.

Reference

[1] Sakr, Sherif, and Ghazi Al-Naymat. "Graph indexing and querying: a review." *International Journal of* Web Information Systems (2010).

[2] Zou, L., Chen, L., Yu, J.X. and Lu, Y. (2008), “A novel spectral coding in a large graph database”, Proceedings of the 11th International Conference on Extending Database Technology (EDBT), Nantes, March 25-29, pp. 181-92.

[3] Williams, D.W., Huan, J. and Wang, W. (2007), “Graph database indexing using structured graph decomposition”, Proceedings of the 23rd International Conference on Data Engineering (ICDE), Istanbul, pp. 976-85.

[4] Jiang, H., Wang, H., Yu, P.S. and Zhou, S. (2007), “GString: a novel approach for efficient search in graph databases”, Proceedings of the 23rd International Conference on Data Engineering (ICDE), Istanbul, April 15-20, pp. 566-75.

[5] Sakr, S. (2009), “GraphREL: a decomposition-based and selectivity-aware relational framework for processing sub-graph queries”, Proceedings of the 14th International Conference on Database Systems for Advanced Applications (DASFAA), Brisbane, April 21-23, pp. 123-37.

[6] Giugno, R. and Shasha, D. (2002), “GraphGrep: a fast and universal method for querying graphs”, IEEE International Conference in Pattern Recognition (ICPR), pp. 112-15.

[7] Yan, X., Yu, P.S. and Han, J. (2004), “Graph indexing: a frequent structure-based approach”, Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, June 13-18, pp. 335-46.

[8] Cheng, J., Ke, Y., Ng, W. and Lu, A. (2007), “FG-index: towards verification-free query processing on graph databases”, Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, June 11-14, pp. 857-72.

[9] Zhang, S., Hu, M. and Yang, J. (2007), “TreePi: a novel graph indexing method”, Proceedings of the 23rd International Conference on Data Engineering (ICDE), Istanbul, 15-20 April, pp. 966-75.

[10] Zhao, P., Yu, J.X. and Yu, P.S. (2007), “Graph indexing: treeþdelta $ graph”, Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), Vienna, pp. 938-49.

[11] Chen, C., Yan, X., Yu, P.S., Han, J., Zhang, D.-Q. and Gu, X. (2007), “Towards graph containment search and indexing”, Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), University of Vienna, September 23-27, pp. 926-37.

[12] Zhang, S., Li, J., Gao, H. and Zou, Z. (2009), “A novel approach for efficient supergraph query processing on graph databases”, Proceedings of the 12th International Conference on Extending Database Technology (EDBT), Saint Petersburg, March 24-26, pp. 204-15.

[13] He, H. and Singh, A.K. (2006), “Closure-tree: an index structure for graph queries”, Proceedings of the 22nd International Conference on Data Engineering (ICDE), Atlanta, GA, April 3-7, pp. 38-52.

[14] Yan, X., Yu, P.S. and Han, J. (2005b), “Substructure similarity search in graph databases”, Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, MD, June 14-16, pp. 766-77.

[15] Yan, X., Zhu, F., Han, J. and Yu, P.S. (2006), “Searching substructures with superimposed distance”, Proceedings of the 22nd International Conference on Data Engineering (ICDE), Washington, DC, pp. 88-98.

[16] Tian, Y., McEachin, R.C., Santos, C., States, D.J. and Patel, J.M. (2007), “SAGA: a subgraph matching tool for biological graphs”, Bioinformatics, Vol. 23 No. 2, pp. 232-9.