Massive Graph Data Mining

February 2, 2015

1 Motivation

Graphs encode the interconnections between entities to model relations between them. They have proven to be an intuitive datastructure in modelling, querying and reasoning about real-life relationships. In websearch the graphs derived from links between webpages are used for link analysis and establishing authority of webpages. In social networks, the graph structure captures interesting insights about the influential nodes, community structures, connectivity and reachability etc. In biological networks people are interested in how diseases spread. In ontologies, they are useful to model taxonomies in the form of type hierarchies. Apart from these scenarios they find applications in road, water networks for shortest path computations and flow problems.

Traditional research has focussed on largely on *static graphs*. However, as one might quickly realize, most of the scenarios enumerated above have a evolutionary nature to them. Social networks, Web graphs, citation networks and even biological networks evolve over time. So there has been a growing need to support data management, querying, mining and analytics tasks over evolving or time-varying graphs. Moreso, to date there is little work on understanding the challenges needed to ingest, manage and query such data in large graph databases. In this proposal we attempt to scope out the problems relating to dynamic graphs and more specifically time-varying graphs with an intent to identify open issues and problems which would be worth investigating.

2 Temporal Graphs

We study dynamic graphs with a specific focus on three dimensions – *data or input models* for representing graphs, support for *primitive operations* in querying and mining dynamic graphs, and finally data management issues pertaining to storing dynamic graphs for supporting *approximate* and *exact queries*.

2.1 Dynamic Graph Input/Data Model

In this section we outline some of the prevalent data models in the literature pertaining to dynamic graphs.

1. **Streaming Model [4] :** The most prevalent and standard data model for dynamic graphs is the *streaming model* which assumes the input to be a sequence edges from a given family of nodes. (cite Muthukrishnan et. al Data Streams: Algorithms and Applications). Here the algorithm sees the entire input and the typical assumption is of a limited working memory.

2. **Dynamic Graph Model [3] :** In this setting, a graph changes over time and the goal is to keep track of the changes so as to be able to efficiently answer graph queries. The main difference is that here when a change is performed to a graph, the algorithm is notified of the change.

3. **Property Testing Model [5] :** Here the goal is to find whether a graph has some property or is far from satisfying the property using a limited number of queries.

4. **Multi-armed Bandit Model [4] :** In the standard multi- armed bandit setting there are k slot machines (one-armed bandits) and pulling a lever in a slot machine gives a re- ward, which depends on the machine, and reveals informa- tion about the machine. The objective is to select the machines to query so as to maximize the total reward; as in our case, the number of queries in every time step is limited.

5. **Fixed-probe Model [2] :** The time is assumed to proceed in discrete steps, numbered by positive integers. At each time step t, the data is given by a (possibly weighted) graph Gt. The data is changing gradually, i.e., the graph Gt+1 is obtained from Gt by a small random change.2 At each time step t, the algorithm is allowed to probe a small portion of the graph Gt, and then must output a solution for the problem under consideration. We would like this solu- tion to be close to thecorrectsolution for the graph Gt. In this paper we do not impose any constraint on the amount of memory the algorithm maintains or the running time of the algorithm, although all of the algorithms we present are quite efficient with respect to these factors.

6. **Parametric optimization and kinetic problems [1] :**In the parametric optimization model, the edge weights are known continuous functions of a real parameter λ (often referred to as time), and the goal is to identify how the solution changes as λ varies. A kinetic problem combines parametric optimizations and dynamic data structures for insertions and deletions. In such a problem, at the begin- ning a parametric problem of parameter λ is given; as the time λ progresses, the weight functions change and objects (e.g., edges) are inserted or deleted. The goal is to efficiently maintain the optimal solution at each point in time.

2.2 Querying and Mining Dynamic Graphs

In this section we investigate the primitive operations that have been supported on dynamic graphs over the past few years. In doing so, we identify potential open problems which need to be addressed.

2.2.1 Path Related Problems

2.2.2 Sub-graph Problems

2.2.3 Matching Problems

2.3 Data Structures for Approximate and Exact Queries

2.3.1 Index Structures based on Graph Sketches

2.3.2 Spanners, Sparsifiers and Expanders

2.3.3 Approximate operations on Graphs

References

[1] Pankaj K Agarwal, David Eppstein, Leonidas J Guibas, and Monika Rauch Henzinger. Parametric and kinetic minimum spanning trees. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 596–605. IEEE, 1998.

[2] Aris Anagnostopoulos, Ravi Kumar, Mohammad Mahdian, Eli Upfal, and Fabio Vandin. Algorithms on evolving graphs. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 149–160, New York, NY, USA, 2012. ACM.

[3] Camil Demetrescu, David Eppstein, Zvi Galil, and Giuseppe F. Italiano. Algorithms and theory of computation handbook. pages 9–9. Chapman & Hall/CRC, 2010.

[4] Shanmugavelayutham Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

[5] Dana Ron. Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 5(2):73–205, February 2010.

3 Open Problems

4 Seminal Works

5 Application Areas