ALGORITHMS FOR MASSIVE AND DYNAMIC GRAPH DATA PROCESSING

NEHA SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI

SEPTEMBER 2019

ALGORITHMS FOR MASSIVE AND DYNAMIC GRAPH DATA PROCESSING

by

NEHA SENGUPTA

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of **Doctor of Philosophy**

to the

Indian Institute of Technology Delhi
SEPTEMBER 2019

Certificate

This is to certify that the thesis titled **ALGORITHMS FOR MASSIVE AND DYNAMIC GRAPH DATA PROCESSING** being submitted by **Ms. NEHA SENGUPTA** for the award of **Doctor of Philosophy** in **Computer Science and Engineering** is a record of bona fide work carried out by her under my guidance and supervision at the Department of Computer Science and Engineering, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

Maya Ramanath
Associate Professor
Department of Computer Science and Engineering
Indian Institute of Technology Delhi
New Delhi- 110016

Acknowledgements

I thank my advisor, Maya Ramanath, for her invaluable guidance and support. Her clarity and objectivity of thought, unwavering faith in me, and words of encouragement have saved the day during many trying times in the course of my PhD. Her sound judgment and brilliant suggestions have steered my research in the right direction on several occasions. She is a constant source of inspiration to me. I thank my mentors, Srikanta Bedathur and Amitabha Bagchi, for helping to shape my research with their insights, suggestions, and ideas. Other than research, my advisor and mentors have also taught me the value of perseverance and optimism, qualities imperative for conducting research. I owe them thanks for a great many things, but most of all, I thank them for having patience with me.

I thank my thesis committee members, Parag Singla, Saroj Kaushik, and Lipika Dey, for their feedback and suggestions which have been critical to help me refine my ideas and improve my research.

I thank Dorothea Wagner for giving me the opportunity to work with her. The work that I did with her is one of the pivotal components of my PhD. I also thank Michael Hamann for the numerous and highly productive discussions. I also owe thanks to Arkaprava Saha for his hard work.

I thank IBM for supporting me as a graduate student with their IBM PhD fellowship programme. I want to thank the staff of CS & E department and SIT at IIT Delhi for their help in several administrative matters.

The importance of having good friends during graduate school cannot be overstated. I am blessed to have friends like Ashwini Vaidya, Madhulika Mohanty, Prajna Devi Upadhyay, Ankit Anand, Anup Bhattacharya and many others, who have all contributed to making the journey of my PhD enjoyable.

I struggle to find words that can fully express the gratitude I feel for my closest friend and husband, Sourav Guha Roy. I thank him for being the source of strength that has carried me through the most difficult times of my PhD, for his unconditional love and support, and for being the absolutely incredible person that he is. I also thank my parents in law for their support and encouragement.

Last, and the *most*, I thank my parents, Ashok and Mita Sengupta. I could not possibly finish a list of the things I want to thank them for. It suffices to say that all that I know, all that I can do, and by extension the completion of this thesis, is attributed to their efforts. Thank you, Ma and Baba, for inculcating in me even just a fraction of the virtues that you yourselves possess.

Neha Sengupta

Abstract

Graphs have been universally used to represent relationships among real world entities. They can be used to model data in several applications, such as in a social network with nodes as users and edges as friendships, or the World Wide Web with nodes as web pages and edges as hyperlinks. Graphs can be analyzed to obtain information in many application domains, including influence of users on a social network, and the popularity of pages on the World Wide Web. Recent advances have brought about the availability of enormous real-world graph datasets. These graphs contain millions of nodes and billions of edges, making graph analyses computationally challenging. In addition to being massive, many of these graphs also update rapidly, with edges and nodes being added and removed from the graph. In a social network, the formation and discontinuation of friendships cause the insertion and deletion of edges in the graph, while users joining or leaving the platform cause node addition and removals. Several other areas generate large and dynamic graphs, including email networks and academic citations.

A fundamental building block for many graph analysis algorithms is reachability, which asks if there is a path in the graph from a given source node to a destination node. In a

social or telecommunication network, reachability queries can be used to assess the flow of influence or information from one user to another. Most existing methods for determining reachability build sophisticated index structures designed to minimize the time taken to answer the reachability query. For a large and dynamic graph, such methods have a large memory footprint, and a high computation time to update the index as updates are made to the graph. To address this drawback, we develop a technique called ARROW that works by running random walks on the graph, and does not employ any reachability index. We show that being index-free, ARROW has a small memory footprint, and can update the graph extremely fast, outperforming existing index-based techniques on both memory efficiency and update time for massive and highly dynamic graphs.

ARROW is highly flexible, enabling easy adaptation to reachability on temporal graphs, which are graphs that encode their entire history of evolution in the form of intervals associated with each of their nodes and edges. Reachability on a temporal graph has many types due to different temporal constraints. We extend ARROW, called Time's ARROW, for answering various types of reachability queries on temporal graphs. We evaluate Time's ARROW against several baseline algorithms for reachability in temporal graphs and establish its advantages over them.

Even without an index, a massive graph may be too large to fit in memory. To efficiently compute reachability queries on such a graph, we extend an existing compact graph storage technique called Bloom Graphs and enable the use of ARROW on the Bloom graph, called BloomARROW. We propose a variant of Bloom graphs to further reduce memory footprint, called Hybrid Bloom graphs, and evaluate BloomARROW over them.

Bloom Graphs store the neighborhood list of each node in the graph as a Bloom filter, which is a highly popular data structure used to compactly store a set of elements. But, enabling ARROW on the Bloom graph requires running random walks on it, and therefore drawing samples from Bloom filters. Towards this objective, we propose a novel data structure called the BloomSampleTree, which we employ to efficiently obtain samples from a Bloom filter. Using the BloomSampleTree, we also propose an algorithm to reconstruct the set of elements in a Bloom filter.

Another highly important graph problem is that of overlapping community detection, which is the identification of densely linked subgraphs that may be overlapping with each other. Community detection has a number of applications, such as determining groups of users with common interests in a social network. As with reachability, detecting overlapping communities in a dynamic graph is extremely challenging. A big roadblock to the development of such algorithms is the lack of reliably labeled and freely available graph datasets, which are critical to evaluating different algorithms. To address this, we build a benchmark generator that efficiently generates large datasets with an evolving and overlapping community structure, while maintaining statistical properties typical of real-world networks. Using a rigorous empirical analysis, we demonstrate the ability of the benchmark generator to assess the comparative performance of existing algorithms for detection of overlapping communities.

In summary, this thesis provides efficient methods to process and generate large, highly dynamic graphs for two of the most important problems in graph data analysis – that of reachability and community detection.

सार

वास्तविक विश्व संस्थाओं के बीच संबंधों का प्रतिनिधित्व करने के लिए रेखांकन का सार्वभौमिक रूप से उपयोग किया गया है। उन्हें कई अनुप्रयोगों में डेटा मॉडल करने के लिए उपयोग किया जा सकता है, जैसे कि एक सामाजिक नेटवर्क में नोड्स के साथ उपयोगकर्ता और संबंधों के रूप में दोस्ती, या वर्ल्ड वाइड वेब के साथ नोड्स के रूप में वेब पेज और हाइपरलिंक के रूप में संबंधों। सामाजिक नेटवर्क पर उपयोगकर्ताओं के प्रभाव और वर्ल्ड वाइड वेब पर पृष्ठों की लोकप्रियता सहित कई एप्लिकेशन डोमेन में जानकारी प्राप्त करने के लिए ग्राफ़ का विश्लेषण किया जा सकता है। हाल के अग्रिमों ने विशाल वास्तविक-विश्व ग्राफ डेटासेट की उपलब्धता के बारे में बताया है। इन ग्राफ़ में लाखों नोड्स और अरबों संबंधों होते हैं, जिससे ग्राफ विश्लेषण कम्प्यूटेशनल रूप से चुनौतीपूर्ण होता है। बड़े पैमाने पर होने के अलावा, इनमें से कई ग्राफ तेजी से अपडेट भी होते हैं, जिसमें संबंधों और नोड्स को ग्राफ से जोड़ा और हटाया जाता है। एक सामाजिक नेटवर्क में, दोस्ती के गठन और विघटन से ग्राफ़ में संबंधों के सम्मिलन और विलोपन का कारण बनता है, जबकि प्लेटफ़ॉर्म में शामिल होने या छोड़ने वाले उपयोगकर्ता नोड जोड़ और निष्कासन का कारण बनते हैं। कई अन्य क्षेत्रों में बड़े और गतिशील ग्राफ़ उत्पन्न होते हैं, जिनमें ईमेल नेटवर्क और अकादमिक उद्धरण शामिल हैं।

कई ग्राफ विश्लेषण एल्गोरिदम के लिए एक मौलिक बिल्डिंग ब्लॉक रीचैबिलिटी है, जो पूछता है कि किसी दिए गए स्रोत नोड से गंतव्य नोड के लिए ग्राफ़ में कोई रास्ता है या नहीं। एक सामाजिक या दूरसंचार नेटवर्क में, एक उपयोगकर्ता से दूसरे उपयोगकर्ता के प्रभाव या सूचना के प्रवाह का आकलन करने के लिए रीचैबिलिटी प्रश्नों का उपयोग किया जा सकता है। रीचैबिलिटी का निर्धारण करने के लिए अधिकांश मौजूदा विधियाँ परिष्कृत इंडेक्स संरचनाओं का निर्माण करती हैं जो रीचैबिलिटी क्वेरी का उत्तर देने में लगने वाले समय को कम करने के लिए डिज़ाइन की गई हैं। बड़े और गतिशील ग्राफ के लिए, इस तरह के तरीकों में एक बड़ा मेमोरी फ़ुटप्रिंट होता है, और इंडेक्स को अपडेट करने के लिए एक उच्च गणना समय होता है। इस खामी को दूर करने के लिए, हम एरो नामक

एक तकनीक विकसित करते हैं, जो ग्राफ पर यादृच्छिक चलता है, और किसी भी इंडेक्स को नियोजित नहीं करता है। हम दिखाते हैं कि इंडेक्स-फ्री होने के नाते, एरो में एक छोटा मेमोरी फ़ुटप्रिंट होता है, और यह बहुत तेज़ी से ग्राफ़ को अपडेट कर सकता है। एरो मेमोरी दक्षता पर मौजूदा इंडेक्स-आधारित तकनीकों को बेहतर बना सकता है और बड़े और अत्यधिक गतिशील ग्राफ़ के लिए समय अपडेट कर सकता है।

एरो अत्यधिक लचीला है, जो अस्थायी ग्राफ़ पर रीचैबिलिटी के लिए आसान अनुकूलन को सक्षम करता है, जो ऐसे ग्राफ़ हैं जो उनके प्रत्येक नोड्स और संबंधों से जुड़े अंतराल के रूप में उनके विकास के पूरे इतिहास को कूटबद्ध करते हैं। एक लौकिक ग्राफ पर रीचैबिलिटी विभिन्न लौकिक बाधाओं के कारण कई प्रकार की होती है। हम टेम्पोरल ग्राफ़ पर विभिन्न प्रकार के रीचबिलिटी क्वेश्चन का उत्तर देने के लिए, एरो का विस्तार करके टाइम्स एरो का निर्माण करते हैं। हम टाइम्स एरो का मूल्यांकन लौकिक रेखांकन में रीचैबिलिटी के लिए कई आधारभूत एल्गोरिथम के खिलाफ करते हैं और उन पर इसके फायदे स्थापित करते हैं।

इंडेक्स के बिना भी, मेमोरी में फिट होने के लिए एक विशाल ग्राफ बहुत बड़ा हो सकता है। इस तरह के ग्राफ पर रीचैबिलिटी प्रश्नों की कुशलता से गणना करने के लिए, हम ब्लूम ग्राफ नामक एक मौजूदा कॉम्पैक्ट ग्राफ स्टोरेज तकनीक का विस्तार करते हैं और ब्लूम ग्राफ पर एरो के उपयोग को सक्षम करते हैं, जिसे ब्लूम एरो कहा जाता है। हम हाइब्रिड ब्लूम ग्राफ़ नामक मेमोरी फ़ुटप्रिंट को कम करने के लिए ब्लूम ग्राफ़ के एक संस्करण का प्रस्ताव करते हैं, और उनके ऊपर ब्लूम एरो का मूल्यांकन करते हैं।

ब्लूम ग्राफ में प्रत्येक नोड की पड़ोस सूची को ब्लूम फिल्टर के रूप में संग्रहीत किया जाता है, जो तत्वों के एक समूह को कॉम्पैक्ट करने के लिए उपयोग की जाने वाली एक अत्यधिक लोकप्रिय डेटा संरचना है। लेकिन, ब्लूम ग्राफ पर एरो को सक्षम करने के लिए उस पर यादृच्छिक चलता है, और इसलिए ब्लूम फिल्टर से नमूने खींचना आवश्यक है। इस उद्देश्य की ओर, हम एक नय डेटा संरचना का प्रस्ताव करते हैं जिसे ब्लूम सैंपल ट्री कहा जाता है, जिसे हम कुशलतापूर्वक ब्लूम फ़िल्टर से नमूने प्राप्त करने के लिए नियोजित करते हैं। ब्लूम सैंपल ट्री का उपयोग करते हुए, हम ब्लूम फ़िल्टर में तत्वों के समूह को फिर से संगठित करने के लिए एक एल्गोरिथ्म का प्रस्ताव करते हैं।

एक अन्य अत्यधिक महत्वपूर्ण ग्राफ समस्या अतिव्यापी समुदाय का पता लगाने की है, जो घनीभूत रूप से जुड़े उपसमूहों की पहचान है जो एक दूसरे के साथ अतिव्यापी हो सकते हैं। सामुदायिक पहचान में कई एप्लिकेशन होते हैं, जैसे कि सामाजिक नेटवर्क में सामान्य हितों वाले उपयोगकर्ताओं के समूह का निर्धारण करना। रीचैबिलिटी की तरह, गतिशील ग्राफ़ में अतिव्यापी समुदायों का पता लगाना बेहद चुनौतीपूर्ण है। ऐसे एल्गोरिदम के विकास के लिए एक बड़ी चुनौती मज़बूती से लेबल और स्वतंत्र रूप से उपलब्ध ग्राफ़ डेटासेट की कमी है, जो विभिन्न एल्गोरिदम का मूल्यांकन करने के लिए महत्वपूर्ण हैं। हम एक बेंचमार्क जनरेटर का निर्माण करते हैं जो कुशलतापूर्वक एक विकसित और अतिव्यापी सामुदायिक संरचना के साथ बड़े डेटासेट उत्पन्न करता है, जबकि वास्तविक विश्व नेटवर्क के विशिष्ट सांख्यिकीय गुणों को बनाए रखता है। कठोर अनुभवजन्य विश्लेषण का उपयोग करते हुए, हम अतिव्यापी समुदायों का पता लगाने के लिए मौजूदा एल्गोरिदम के तुलनात्मक प्रदर्शन का आकलन करने के लिए बेंचमार्क जनरेटर की क्षमता का प्रदर्शन करते हैं।

सारांश में, यह थीसिस ग्राफ डेटा विश्लेषण में सबसे महत्वपूर्ण समस्याओं में से दो के लिए बड़े पैमाने पर, अत्यधिक गतिशील ग्राफ को संसाधित करने और उत्पन्न करने के लिए कुशल तरीके प्रदान करता है - जो कि रीचैबिलिटी और सामुदायिक पहचान का है।

Contents

List of Figures

List of Tables