ISSN 2186-7437

Generate a standard cover of an NII Shonan Meeting Report.

NII Shonan Meeting Report

No. 2014-12

Algorithms for Large Scale Graphs

Andrew McGregor
Gopal Pandurangan
Sergei Vassilvitskii

October 14–17, 2014

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Algorithms for Large Scale Graphs

Organizers:
Andrew McGregor (University of Massachusetts, Amherst)
Gopal Pandurangan (University of Houston)
Sergei Vassilvitskii (Google)

October 14–17, 2014

Large scale graphs abound in modern data analysis. They are used to model social and communication networks, item-item similarities in recommendation settings, relationships between buyers and sellers in large scale markets and so on. Yet, at the same time, graph algorithms are among the hardest to scale to very large inputs. There are three general directions for scaling such approaches: (i) sparsifying and summarizing the graph, retaining only the crucially relevant information, (ii) streaming it, and thus never storing the full input, or (iii) processing it in parallel. This workshop brought together researchers from all three of the above fields to present results, share ideas, and foster collaboration across different areas.

Schedule

Tuesday

09:00-09:10	Welcome	
09:10-10:30	Sergei Vassilvitskii	Space-{Rounds, Machines} Tradeoffs For Graph Problems
11:00-12:00	Mohammad Mahdian	Composable Core-sets for Diversity and Coverage Maximization
14:00-15:30	Gopal Pandurangan	Distributed Computing and Large-scale Graph Processing: Foundations and Connections
16:00-17:00	Peter Robinson	Distributed Computation of Large-scale Graph Problems
17:00-18:00	Valerie King	Low communication methods for MST in Distributed Dynamic Networks

Wednesday

9:00-10:30	Andrew McGregor	Survey on Graph Streaming and Sketching
11:00-12:00	Rotem Oshman	The Role of Communication Complexity in Distributed Computing
13:30-14:30	Debmalya Panigrahi	Graph Sparsification
14:30-15:30	Qin Zhang	A sketching algorithm for spectral graph sparsification
16:00-17:00	David Woodruff	The Sketching Complexity of Graph Cuts
17:00-18:00	Aleksander Madry	Cuts, Trees, and Electrical Flows
20:00	Open Problem Session	

Thursday

9:00-10:30	Amit Chakrabarti	Maximum Matching (and Generalizations) on Graph Streams
11:00-11:45	Rajmohan Rajaraman	Local discovery and diffusion processes on graphs
11:45 - 12:15	Vladimir Braverman	Approximating Large Frequency Moments with $O(n^{1-2/k})$ Bits
13:30-19:00	Excursion	

Friday

9:00-9:45	Jelani Nelson	The Johnson-Lindenstrauss lemma is optimal for linear dimensionality reduction in l_2
9:45 - 10:30	Maleq Khan	Parallel Graph Algorithms: Challenges and Approaches
11:00-12:00	Nick Harvey	Characterizing Storage Workloads with Counter Stacks

Overview of Talks

Space-{Rounds, Machines} Tradeoffs For Graph Problems

Sergei Vassilvitskii, Google

A tenet of modern parallel algorithms is the fact that the input is too large to be stored on any individual compute node. In this talk we focus on the complexity of standard graph problems as a function of the individual machine size. Specifically, we talk about (i) connectivity and give algorithms with space-round trade-offs for very small and very large machines, and (ii) triangle counting, where we give the optimal partition for machines of any size.

Composable Core-sets for Diversity and Coverage Maximization

Mohammad Mahdian, Google

I will talk about the notion of composable core-sets, and how they can be used to design 1-round map-reduce algorithms (as well as diverse nearest neighbor search and streaming algorithms). We will show efficient constructions of composable core-sets for several diversity maximization problems, and an impossibility result for coverage maximization. This is based on joint work with Piotr Indyk, Sepideh Mahabadi, and Vahab Mirrokni.

Distributed Computing and Large-scale Graph Processing: Foundations and Connections

Gopal Pandurangan, University of Houston

A distributed network consists of a set of processors that communicate by message passing via an underlying communication network. Distributed algorithms been studied extensively for over three decades.

In the first part of the talk, we will focus on two quintessential problems — minimum spanning tree (MST) and random walk sampling — and present our recent results that establish tight time bounds for both problems.

In the second part of the talk. we will focus on distributed processing of large-scale graph data which has become very important with the emergence of "Big Graph Data" in the form of the Web graph, social networks, biological networks etc. Recent graph processing systems such as Pregel and Giraph can do computation on graphs having billions of nodes. These systems use a message-passing "vertex-centric" computation model that mimics the distributed network computation model. We show that algorithms for distributed network computing can be leveraged to obtain efficient large-scale graph algorithms for various fundamental problems including MST and PageRank.

Distributed Computation of Large-scale Graph Problems

Peter Robinson, National University of Singapore

Motivated by the increasing need for fast distributed processing of large-scale graphs such as the Web graph and various social networks, we study a number of fundamental graph problems in the message-passing model, where we have k machines that jointly perform computation on an arbitrary n-node (typically, $n \gg k$) input graph. The graph is assumed to be randomly partitioned among the k machines (a common implementation in many real world systems). We present several lower bounds that quantify the fundamental time limitations of distributively solving graph problems. To complement our lower bounds, we also show how problems such as PageRank, MST, shortest paths, connectivity, and graph covering can be solved in this model.

Low communication methods for MST in Distributed Dynamic Networks

Valerie King, University of Victoria

This talk is about distributed algorithms for communications networks, which have low communication cost and small local memory size. At any point in time, each node is assumed to know its incident edges and their weight. An MST is maintained by the network if the edges of the MST are known by their endpoints to be in the MST.

Nodes communicate by passing O(log n) bit messages across edges. I'll show that a communications network can construct a minimum spanning tree and a spanning forest from scratch in the synchronous model of communication using $O(n \log^2 n / \log \log n)$ messages for MST and $O(n)$ messages for a spanning forest. This is apparently the first algorithm which does not require a message sent across every edge (m messages in total) for either of these problems.

I'll also show how to perform ”impromptu” repairs which require no preprocessing or extra memory between updates. For an MST or ST in the asynchronous model the updates require $O(n \log n / \log \log n)$ messages per edge change in the MST and $O(n)$ messages per edge change for the ST problem. While algorithms for dynamic distributed MST are known which use a similar number of messages per update, they require each node to essentially store the whole tree.

Finally, I'll talk about connections to the greater-than 2 party communication problem.

This is joint work with Shay Kutten, Ben Mountjoy and Mikkel Thorup.

Survey on Graph Streaming and Sketching

Andrew McGregor, University of Massachusetts

Early work on data stream algorithms focused on estimating numerical statistics such as quantiles, frequency moments, and heavy hitters given limited memory and a single pass over the input. However, there's now a growing body of work on analyzing more structured data such as graphs. In this talk, we survey recent research in this area.

The Role of Communication Complexity in Distributed Computing

Rotem Oshman, Tel Aviv University

In distributed systems, the cost of communicating between the participating devices often dwarfs the cost of local computation on the individual devices. Accordingly, when we model a distributed system, we usually ignore local computation and charge only for communication. Traditional cost measures have included the total number of messages sent, or the number of rounds of communication, without restricting the number of bits in a message or a round. However, recently there has been a lot of interest in quantifying the number of bits that need to be sent to solve various tasks, and the interaction between this cost and the round complexity of the task. In this talk I will survey some of the recent work incorporating communication costs into distributed computing models, give some examples of lower bounds, and discuss open problems.

Graph Sparsification

Debmalya Panigrahi, Duke University

Recently, there has been growing interest in graph sparsification — replacing dense/large graphs with sparser/smaller graphs while (approximately) preserving the values of certain graph parameters. In this talk, I will focus on cut sparsification — approximately preserving the weights of all cuts — and describe a general framework for this problem that yields improved combinatorial results and leads to faster and simpler algorithms.

A sketching algorithm for spectral graph sparsification

Qin Zhang, Indiana University at Bloomington

We will discuss a randomized sketching algorithm for spectral graph sparsification: Given an $\epsilon \in (0,1)$, and a n-vertex weighted graph $G = (V, E, w)$, it generates a sketch $sk(G)$ of size $o(n/\epsilon^2)$ such that for any $x \in R^n$, we can compute a $(1+\epsilon)$-approximation of $x^T L_G x$ with high probability. Here L_G is the (unnormalized) graph Laplacian of graph G.

The Sketching Complexity of Graph Cuts

David Woodruff, IBM Almaden

We show an $\Omega(n/\epsilon^2)$ bit lower bound for any data structure which reports $(1+\epsilon)$-approximate values to all cuts of an unweighted graph. The result is based on communication complexity, and strengthens existing lower bounds which only held for spectral sparsifiers and specific estimation functions rather than arbitrary data structures. This shows the ϵ-dependence in the memory required of existing algorithms, both offline and in the streaming model, is optimal.

We also show that if one is interested only in the property that a fixed cut is approximated up to $1+\epsilon$ with high probability, then there is a data structure using $\tilde{O}(n/\epsilon)$ bits of space, and this is optimal up to polylogarithmic factors. This suffices for applications such as min-cut, and shows a separation in terms of ϵ between the "for-each" and "for-all" problems for cut approximation.

Joint work with Alex Andoni and Robi Krauthgamer

Cuts, Trees, and Electrical Flows

Aleksander Madry, EPFL

We discuss some of the recent developments in algorithmic graph theory that are relevant in the context of dealing with massive graphs.

In particular, we present a general framework for obtaining close-to-linear-time approximation algorithms for cut problems in undirected graph. We also introduce the electrical flow paradigm that played key role in some of the recent progress on designing fast algorithms for fundamental flow problems.

Maximum Matching (and Generalizations) on Graph Streams

Amit Chakrabarti, Dartmouth College

The maximum matching problem has proved to be endlessly fascinating: in combinatorics, in the classic theory of algorithms, and now in the data stream model, where the goal is to find an approximately-best matching when the edges of the graph must be processed sequentially, using limited storage.

In this talk I shall survey the rich theory that has developed around this problem, starting in the mid-2000s. We will see how algorithms for this problem got gradually more sophisticated, and then suddenly simpler. We will see two variations of the original maximum-weight matching problem, one that generalizes the objective to a submodular function, and another that extends the streaming model itself. Finally, we shall take a quick look at lower bounds for the problem.

Most of this material is not my own work, the exception being the submodular maximization results which are joint work with Sagar Kale.

Local discovery and diffusion processes on graphs

Rajmohan Rajaraman, Northeastern University

We introduce and analyze two classes of local processes that model the spread of information or epidemics in distributed networks. In the first class of processes – which we term gossip-based discovery – nodes form new neighborhood connections through a local random push or pull process. These processes are motivated by information discovery in large-scale distributed networks such as peer-to-peer and social networks. We present a near-tight analyses of two such processes, starting from arbitrary initial networks.

The second class of processes – which we term cobra walks (for coalescing-branching random walks) – are a generalization of the standard random walk. In each step of a cobra walk, which starts with an arbitrary node being labeled active for step 1, each active node chooses a specified number of random neighbors to become active for the next step ("branching"). A node is active for step

$t+1$ only if it is chosen by an active node in step t ("coalescing"). Cobra walks are useful in modeling and understanding the Susceptible-Infected-Susceptible (SIS)-type of epidemic processes in networks. We present near-tight analyses of cobra walks on expanders, with sufficiently high expansion, grids, and trees.

The talk will focus on the technical challenges we face in our analyses, how we overcome some of them, and will close with several problems that still remain open.

Joint work with Chinmoy Dutta, Bernhard Haeupler, Gopal Pandurangan, David Peleg, Scott Roche, and Zhifeng Sun.

Approximating Large Frequency Moments with $O(n^{1-2/k})$ Bits

Vladimir Braverman, Johns Hopkins University

We consider the problem of approximating frequency moments in the streaming model. Given a stream $D = \{p_1, p_2, \ldots, p_m\}$ of numbers from $\{1, ..., n\}$, a frequency of i is defined as $f_i = |\{j : p_j = i\}|$. The k-th frequency moment of D is defined as $F_k = \sum_{i=1}^{n} f_i^k$.

In their celebrated paper, Alon, Matias, and Szegedy (STOC 1996) introduced the problem of computing a $(1+/-\epsilon)$-approximation of F_k with sublinear memory. We give upper bound of $O(n^{1-2/k})$ bits that matches, up to a constant factor, the lower bound of Woodruff and Zhang (STOC 12) for constant epsilon and $k > 3$.

Joint work with Jonathan Katzman, Charles Seidell and Gregory Vorsanger.

The Johnson-Lindenstrauss lemma is optimal for linear dimensionality reduction in l_2

Jelani Nelson, Harvard University

We show that for any $n > 1$ and $0 < \epsilon < 1/2$, there exists a $poly(n)$-size point subset X of R^n such that any linear map from (X, l_2) to l_2^m with distortion at most $1+\epsilon$ must have $m = \Omega(\min\{n, \epsilon^{-2}\log n\})$. Our lower bound matches the upper bounds provided by the identity matrix and the Johnson-Lindenstrauss lemma, improving the previous lower bound of (Alon '03) by a $\log(1/\epsilon)$ factor.

Joint work with Kasper Green Larsen (Aarhus University).

Parallel Graph Algorithms: Challenges and Approaches

Maleq Khan, Virginia Tech

In general, partitioning the data and balancing load are challenging issues in most parallel algorithms. Dealing with these issues is even more challenging for graph algorithms due to the complex dependencies among the computation paths. Emerging massive graphs pose some additional challenges. In this talk, I will discuss some of these challenges and some possible approaches to deal with them. As examples, I will use some specific problems such as counting triangles, subgraph enumeration, generating random networks, etc.

Characterizing Storage Workloads with Counter Stacks

Nick Harvey, University of British Columbia

Existing techniques for identifying working set sizes based on miss ratio curves (MRCs) have large memory overheads which make them impractical for storage workloads. We present a novel data structure, the counter stack, which can produce approximate MRCs while using sublinear space. We show how counter stacks can be checkpointed to produce workload representations that are many orders of magnitude smaller than full traces, and we describe techniques for estimating MRCs of arbitrary workload combinations over arbitrary windows in time. Finally, we show how online analysis using counter stacks can provide valuable insight into live workloads.