ARC Fellowship Application

Advisor - Richard J. Lipton

1 Efficient Approaches for Random Walks

I am working on approaches for performing several random walks on graphs efficiently. Random walks are fundamental for a wide range of research topics ranging from very theoretical applications like expanders, to understand mixing times and estimating distributions of markov process, finding graph cuts to much more applied areas such as identifying similarities on web-graphs, estimating PageRank vectors, search applications, sampling etc.

Random walks are useful in peer to peer networks and distributed computing for sampling nodes. Node sampling is then used for problems such as calculating the average load on the nodes in the network, malicious behavior detection, search etc. The model in distributed computing is that each node only has local access (to its neighbors) and in every round of communication, two adjacent nodes can communicate along the edge. The number of bits communicated per edge per round is restricted to $O(\log n)$. There are several theoretical and practical papers that use random walks in the distributed computing/P2P settings. Walks of length l have always been done in the naïve $O(l)$ rounds. I am working on improving this bound (with collaborators at GaTech and G. Pandurangan, a faculty at purdue who works in distributed computing theory). We show a rather surprising result that one can in fact perform such a random walk in just $O(l^{2/3}D^{1/3})$ rounds, where D is the diameter of the graph. This is unpublished and work in progress. I plan to work on improving this bound to $\sqrt{lD}$ and probably show a (matching) lower bound. The naïve lower bound is D (assuming D is less than l). I also plan to extend this result to perform a large number of independent random walks, with less than a linear scale-up in rounds. This will lead to improved algorithms for search, monitoring systems, malicious nodes detection on P2P networks. Such efficient techniques are likely to improve most other random walk based algorithms on distributed settings.

Recently, along with S. Gollapudi and R. Panigrahy, we gave a similar surprising result for graphs presented as edge streams, and showed how this leads to a more efficient approach for estimating PageRank vectors than currently used (pods best paper award). This algorithm does not carry over to the distributed setting (nor do several key ideas) as the issues are very different in the distributed setting. We also extended these efficient random walk algorithms to present graph partitioning algorithms on streams (manuscript). I plan to work on improving these random walk algorithms, or coming up with new techniques (such as finding a sparse graph) that can in turn be used for obtaining random walks. Obtaining such small space and pass/round efficient algorithms is an important consideration in real applications such as the web graph, distributed networks etc., due to their scale and rate of change.

To summarize, I am working on efficient approaches to perform several independent random walks, in the graph streaming model (important for database and data mining communities) and the distributed computing model (important for peer to peer networks). I am also working on other algorithmic problems in these models (including a comparative study of certain distributed streaming models. This part is in collaboration with Danupon and Dick, but also discussing related ideas with Jim Xu's post doc Ashwin). Dick Lipton is my advisor.