Overview

What are the primary “characteristics” of large real-world graphs? How can we spot “outlier” nodes and edges in such graphs? Can we generate synthetic but “realistic” graphs? How quickly will a virus spread over such a graph? Will the virus die out or will it become an epidemic? What are the primary “groups” of nodes in such a graph?

These questions show up in one form or the other in many applications: detecting money laundering rings in economic networks or spam rings in the Web, designing computer networks more resistant to viral epidemics, grouping customers based on the products they buy, forecasting the performance of some new graph algorithm by testing it on synthetically-generated “realistic” graphs larger than ones currently available, and the list goes on. In fact, many real-world datasets can be easily expressed in the form of a graph, and hence, problems from a wide range of disciplines can be cast as graph-theoretic questions.

My recent work has been on three major aspects of this area. First, I have worked on finding the basic “laws” that seem to hold for most real-world graphs. Based on this, we have developed the R-MAT graph generator, which can efficiently generate synthetic yet “realistic” graphs matching almost all the aforementioned “laws” better than other published graph generators (including the Barabasi-Albert, the Generalized Linear Preference, and other generators). Second, I have worked on discovering graph properties that affect the spread of information/viruses/rumors in a network, with an emphasis on finding a threshold below which a viral infection dies out quickly, but above which the virus can propagate throughout the population and become an epidemic. Surprisingly, we found that this threshold depends only on one number: the largest eigenvalue of the connectivity matrix of the graph. Third, we have developed completely automated and scalable methods to find “groups” of nodes in a graph, and to detect outlier edges. I will briefly discuss all of these below.

The R-MAT graph generator

What are the key characteristics of real-world graphs? This question has been studied in many fields, and we have compiled a list of graph patterns and “laws”, along with software to run through these patterns given any large graph [3]. These include well-known patterns like power-law degree distributions and small diameters, as well as some interesting patterns found by us. The primary use of these laws is in the development of good graph generators. In many simulation settings, we need large “realistic” graph datasets, but obtaining such graphs from the real world might be expensive or even impossible. Thus, we need graph generators which match as many real-world graph patterns as possible.
Our \textit{R-MAT} generator \cite{4} is able to match almost all of these patterns. In addition, it is also efficient, scalable, requires only 3 parameters, and includes a fast parameter-fitting method too. The basic technique can be easily extended to undirected, directed, bipartite and weighted graphs, making \textit{R-MAT} a very general model for graph generation. This allows \textit{R-MAT} to be applied to situations where most popular/recent generators fail; and when both can be applied, \textit{R-MAT} performs significantly better than other methods.

\section*{Epidemic thresholds in viral propagation}

\textit{How does a virus propagate over a network? What is the value of the epidemic threshold such that a viral infection will die out below it, but might become an “epidemic” above it?} This problem has been studied by epidemiologists, computer security researchers, physicists and probability theorists, but in different variations and with different emphases. Epidemiologists developed different stochastic models of viral propagation: the one we use is called the SIS model, where \textit{susceptible} nodes can get infected on contact with \textit{infected} neighbors; occasionally, \textit{infected} nodes get cured, and immediately return to the \textit{susceptible} state. Simulations on the SIS model had been studied in the security community. Analytic studies had made assumptions about the topology of the network, or assumed infinite network size, or had required other approximations. What is the epidemic threshold on a finite graph of arbitrary topology?

The exact solution requires working on a $2^N$-state Markov chain (where $N$ is the number of graph nodes), and this is impractical for large graphs. We propose \cite{5} a simple yet accurate approximation which reduces the problem size. Most importantly, \textit{no assumptions about the network topology are needed}. Given any topology, we find a closed-form formula for the epidemic threshold, which surprisingly depends on only one number: \textit{the largest eigenvalue of the connectivity matrix of the graph}. This allows us to quickly check the susceptibility of any given network to viral infections, which might help in designing new virus-resistant networks.

\section*{Fully-automatic grouping of nodes}

\textit{How can we scalably and automatically find “natural” groups of nodes in a large graph?} The aim is to provide the user with a tool that takes any large graph as input, and automatically groups the nodes into “natural” clusters based on the linkages between them, while automatically figuring out the proper \textit{number} of groups too. However, how can we define a “good” cluster?

We achieve this by an application of the overall MDL philosophy to our particular problem setting \cite{2, 1}. However, the MDL criterion only lets us \textit{compare} two groupings and choose the better one: in addition, we develop heuristics to \textit{find} a good grouping quickly. We have further extended this work to undirected, directed and bipartite graphs. We have also investigated the use of these groupings in finding outlier nodes and edges, all of which is again done \textit{automatically}.

\section*{Future Research Directions}

My current research has just been a sampling of the possibilities in this field. There are several avenues that I plan to pursue.

One key area is that of \textit{weighted graphs}. Our current algorithm for node grouping works only on unweighted graphs, whereas many real-world datasets have different weights representing varying strengths of linkage. Again, most recent work on finding graph patterns has focused on unweighted
graphs; special features of weighted graphs have not yet been investigated thoroughly. While our R-MAT generator can be easily extended to generate weighted graphs, we need to find weight-specific patterns to verify its accuracy.

Another interesting topic is graph evolution. The Internet has been growing rapidly, company networks change over time, links in the social network change as old acquaintances are “forgotten” and new friendships forged, and so on. How does this process happen? What are its characteristics? If we partially observe this process on a subset of the graph, can we infer its effects on the entire graph? The yet-to-be-discovered “laws” of graph evolution would greatly add to the currently known “laws” for static snapshots of large graphs.

A third area of interest involves extending the work on viral/information propagation to other large and complex settings, such as sensor networks. Sensors can propagate information by transmitting wirelessly over the network. However, this information may be lost due to node or link failures. Under what conditions will the information survive in the network? What happens when each node has a fixed memory, but there are lots of pieces of information? What if the sensors can voluntarily go into “sleep” mode to conserve power? How can we achieve a balance between maintaining information in the network while having low energy consumption? These are fascinating questions which I would like to address.

References


