A Critique on Random Graphs

Grady Ward

February 23, 2016

Existing Models of Random Graphs

The two dominant models/generators for random, undirected graphs were both defined by Erdos and Reyni in TODO. The first is a model which selects an adjacency matrix at random from the set of all binary matrices, and produces a graph from that matrix. The second is a generator which constructs its edges with a fixed probability. Both models have been studied at length, and they have been proven to be asymptotically alike. Though these models both make some intuitive sense, we should carefully examine why they have become dominant in the study of graphs.

For the first model, selecting a random adjacency matrix from that set, it is clear to see that abstracting across all matrices allows us to understand properties of the graph as a whole; i.e. we know the probability of a given number of edges occurring, and we can use existing modes of proof about the probability of an integer in a range having a given property, as this model treats every adjacency matrix as a bit-string, on which common proof techniques can be performed. This random graph generator has continued to be relevant in the field because it links matrix/integer theory to graph theory.

In the second model, the one with a fixed and independent probability of a given edge, has the advantage that it allows us to easily calculate many properties of the graph that are of interest to graph theoreticians. For example, the probability distribution of the number of triangles in a graph is a simple summation of products. Additionally, this model represents many 'real-world' problems that do not require much imagination to dream up. This model of random graphs has been impactful because it allows graph theoreticians to concretely discuss many of the properties that they are interested in.

In short, both models have stuck precisely because they marry a model that can be explained in a sentence with the tools to easily discuss the properties that such 'random' graphs have.

0.1 Definitions

0.2 Computation of the Paths Function