A Multi-objective Optimization Approach for Generating Complex Networks

Viplove Arora
School of Industrial Engineering
Purdue University
West Lafayette, USA
arora34@purdue.edu

Mario Ventresca
School of Industrial Engineering
Purdue University
West Lafayette, USA
mventresca@purdue.edu

ABSTRACT

Complex networks are used to model a wide range of systems in nature and society. One area of importance is the ability to model network formation, which has lead to the development of many different algorithms (network generators) capable of synthesizing networks with very specific structural characteristics (e.g., degree distribution, average path length). In this paper, we propose an approach based on action-based networks, which have the potential to generate more realistic network structures than existing techniques. Tests for robustness of the proposed method to infer other network structural characteristics not included in the set of objectives being optimized show the efficacy of the approach.

1. INTRODUCTION

Complex networks can be used to model complex real world systems using sets of nodes and edges that represent elements and their interactions [4]. A question of fundamental importance is: *can we generate synthetic networks that are statistically representative of real networks?* The answer may come in the form of modeling the underlying processes via an algorithm that creates networks [1]. These algorithms, called network generators, are typically stochastic and through repeated execution, produce a set of networks with specific global structural properties, but are otherwise random [4]. In this paper we introduce a new action-based approach for network generation and examine its ability to reproduce existing network models by observing only a single target network structure. We then use an optimization-based approach to learn the probabilistic model to synthesize networks similar to the target network. Results show that most of them can be interpreted using a simple homogeneous model of the action-based approach.

2. METHODOLOGY

The approach described here allows each node to probabilistically choose from a set of actions to connect to other nodes. Intuitively, action-based network generators (ABNG) model network formation as nodes taking individual actions (decisions) to create a global structure. We define $A = \{a_1, \cdots, a_k\}$ as the set of k node actions and $p_i(A)$ as a probability distribution over actions for node v_i. $p_i(A)$ reflects how a node v_i weighs different actions for forming connections. We can form a $n \times k$ row stochastic matrix $\mathbf{P}$ with the following properties: $p_{ij} \geq 0, \quad \sum_{j=1}^{k} p_{ij} = 1, \forall i = 1, \cdots, n$. If different nodes weigh actions similarly, then $\mathbf{P}$ will have $q \leq n$ distinct rows. We define the Action Matrix, $\mathbf{M} : q \times k$, as a condensed form representation of P such that $\mathbf{M}$ contains all the distinct rows of $\mathbf{P}$ and $\bar{p}$ is a probability vector of length q, which contains the probability of occurrence of the corresponding rows in $\mathbf{P}$.

An action for a node v_i builds undirected edges, all of which have v_i as one endpoint. An action provides us with a well-defined strategy for selecting the other end v_j of the edge (v_i, v_j). Every action for node v_i returns a vector $\hat{p}_i$ of length n, where the j^{th} element $\hat{p}_i^j$ is the probability corresponding to the insertion of an edge between v_i and v_j by node v_i. So, for network $\mathcal{G} = \{V, E\}$:

$$A(V|i) : v_i \rightarrow \hat{p}_i^j \quad 0 \leq \hat{p}_i^j \leq 1 \quad \forall j = 1, \cdots, n \tag{1}$$

where edge (v_i, v_j) is inserted into $\mathcal{G}$ by v_i w.p. $\hat{p}_i^j$.

In the current implementation of ABNG, we use the following eight actions: Preferential attachment based on degree, neighbor degree, PageRank and betweenness; connecting to a second neighbor, node similarity based on Inverse log-weighted and Jaccard similarity and an action adding no new edge. Using these actions and corresponding Action Matrix, networks ($\mathcal{G}$) can be generated by using the following algorithm:

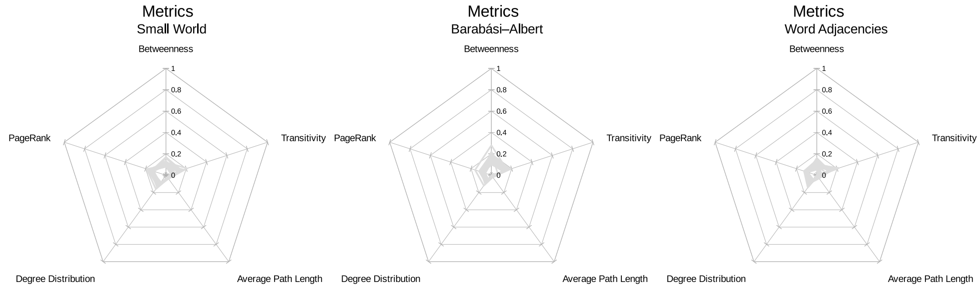
- Visit node v_i, choose $p_i(A)$ from the rows of $\mathbf{M}$ and select an action, a_d with probability p_{id}.
- The action outputs a vector $\hat{p}_i$ of length n, where the j^{th} element $\hat{p}_i^j$ is the probability corresponding to the insertion of an edge between v_i and v_j by node v_i.
- An edge is inserted in $\mathcal{G}$ based on $\hat{p}_i$ until the number of edges in $\mathcal{G}$ is same as the target.

While networks have numerous structural properties (e.g., average path length, transitivity, degree distribution, modularity and so on), there is no universally accepted subset of measures deemed sufficient to accurately compare two networks for dissimilarity [4]. One way of evaluating fitness of the generated networks w.r.t the target as proposed in [3] is to synthesize a set of networks to calculate a summary statistic for the fitness of the generator. Results in

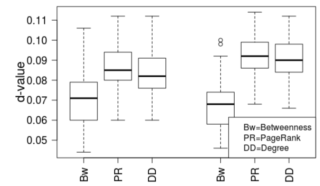
GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2908966



(a)



(b)

Figure 1

Figure 2: (a) Results obtained from the evolved models for the different networks considered here. Network properties other than those used for optimization are also shown. The plots show KS-test d-values with the outer pentagon showing value of 1. The lower the value, better is the generated network. Each plot shows 20 different generations. (b) Network Inference using $\mathbf{M_{200}}$ for $n = 500$, $\mathbf{M_{200}}$ are on the left and $\mathbf{M_{500}}$ on the right. The plots show KS-test d-values which turn out be very similar for the actual and the inferred networks.

[3] indicated that of the examined centrality measures, the degree distribution, betweenness centrality, and PageRank were the most effective for quantifying the (dis)similarity of networks generated by different network models, and will be used here. However, the framework allows for any user-desired measures.

We use Pareto Simulated Annealing [2] to find the action matrix $\mathbf{M}$ that optimizes the fitness of generated networks. We start with a $1 \times k$ action matrix and assume that all nodes are homogeneous w.r.t. to how they form connections. Additional rows are added to $\mathbf{M}$ dynamically.

3. EXPERIMENTS AND RESULTS

Simulation experiments were performed with six synthetic network generators each of which use different criteria for creating networks. The networks used for experimentation were simple (no loops and multi-edges), undirected and unweighted. They were selected for both their historical significance, and because they each exhibit distinct global network properties observed in real networks. The results presented here only include Small World and Barabási–Albert models. A few real world networks were also considered for testing ABNG, results for the network of word adjacencies [5] are shown here. Simulation experiments were run for the six network generation models with $n = 100$ nodes and $m \approx 500$ edges. Networks with this size were chosen because they are large enough to reflect structural properties in the target network while not being too large for testing purposes.

Solutions obtained for the Small World and Barabási–Albert models consisted of a $1 \times k$ action matrix $\mathbf{M}$. It must be kept in mind that due to the stochastic nature of ABNG algorithm, networks generated with an action matrix are not isomorphic as is evident from Figure 1. An experiment was preformed to validate the current model for a growing network using the Barabási–Albert model. A network with $n = 100$ and $m \approx 500$ was allowed to grow to networks with $n = 200, 500$ using the Barabási–Albert model. Action matrices $\mathbf{M_{100}}, \mathbf{M_{200}}, \mathbf{M_{500}}$ were obtained from evolved ABNG solutions. Next, $\mathbf{M_{200}}$ was used to generate the networks with $n = 100, 500$. The networks generated using $\mathbf{M_{200}}$ were statistically as good as the ones generated using $\mathbf{M_{100}}$ and $\mathbf{M_{500}}$ respectively. Figure 1(b) shows equivalence of results between the actual and the inferred network for 100 generated networks. Another observation was that the matrices $\mathbf{M_{200}}$ and $\mathbf{M_{500}}$ gave similar weights to actions.

4. CONCLUSIONS

In this paper we defined an action-based approach for generating complex networks. We have shown the feasibility of this approach to produce complex structure of networks exhibiting different properties by using a variety of target networks. The evolved models also performed well, with respect to the fitness measures, when predicting the growth of networks generated by the target algorithms. Multiple runs of the optimization approach on the same network were observed to converge to similar solutions.

Acknowledgments
This material is based upon work supported by the National Science Foundation under Grant No. 1549608

5. REFERENCES

[1] A. Bailey, M. Ventresca, and B. Ombuki-Berman. Genetic Programming for the Automatic Inference of Graph Models for Complex Networks. *IEEE Transactions on Evolutionary Computation*, 18(3):405–419, 2014.

[2] P. Czyzak and A. Jaszkiewicz. Pareto Simulated Annealing–A Metaheuristic Technique for Multiple-Objective Combinatorial Optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, 1998.

[3] K. R. Harrison, M. Ventresca, and B. M. Ombuki-Berman. Investigating Fitness Measures for the Automatic Construction of Graph Models. In *Applications of Evolutionary Computation*, volume 6025, pages 189–200. 2015.

[4] M. Newman. *Networks: An Introduction.* Oxford University Press, 2010.

[5] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):36104, 2006.