Networks (6.268) Project Report

Recovering the Root in Preferential Attachment: a Mixed-Integer Programming Approach

Samuel Gilmour

May 21, 2021

1 Introduction

A significant thrust of research in complex networks has been concerned with proposing and analyzing generative models that describe the temporal evolution of networks. The proposed models often aim to explain some of the characteristics of real-world networks using random processes that are intuitively reasonable. For example, the Barabasi-Albert model of preferential attachment (PA) has been used to explain the formation of networks according to the *rich get richer* principle – and the resulting networks have scale-free degree distributions that are consistent with those observed in practice [1].

Given these generative models, researchers commonly take on the task of fitting their parameters to the observed growth of networks over time. The quality of the fit can tell us whether the proposed generative model is a reasonable hypothesis for the growth of the network, and if it is, allows us to predict what the network may look like in the future. [2] provides a recent methodology and summary of previous methods that have been used for estimation of parameters in the PA model.

Rather than fitting parameters to an observed sequence of data to check whether a generative model is reasonable, a different stream of research is concerned with reconstructing network histories under the assumption that the generative model drives the growth process – given that we are only able to observe the network at a small fraction of time steps. [3] launched this line of work using a maximum likelihood estimation framework with heuristics designed to address the combinatorial difficulty that is introduced by the need to optimize over graphs. More recently, [4] and [5] have used mixed-integer programming (MIP) techniques to reconstruct histories in biological networks that have been generated according to specific models.

This project aimed to do something similar. Broadly speaking, the aim was to use a maximum-likelihood approach to optimize over graph histories in the PA model and recover the root node. In particular, the two aims were as follows:

1. Explore whether the PA mechanism can be modelled using MIP.
2. If it can, then explore the scalability and accuracy of MIP in recovering the root node.

Section 2 provides a formal definition of the problem and its formulation as a MIP. Section 3 presents some computational experiments that explore how well the technique scales and how well it recovers the root node in the PA model, before Section 4 concludes with some observations.

2 Modelling

This section has three parts. In Section 2.1, we will briefly review the undirected PA mechanism for sequentially generating scale-free graphs. In Section 2.2 we will cast the problem of estimating the root node as a maximum likelihood problem over a sequence of graphs, and in Section 2.3 we will show how this problem can be modelled as a nonlinear convex MIP.

2.1 Undirected Preferential Attachment

The standard undirected preferential attachment model as defined in [1] – which is itself a special case of the more general model in [6] – is a relatively simple model that describes a random process for the evolution of a graph from a single node.

The Mechanism

For notation: at each time step $t \geq 1$, let V_t be the set of nodes that are present, E_t be the set of edges that exist in the network, and $G_t = (V_t, E_t)$.

The process begins at time $t = 1$ with two nodes present, so $V_1 = \{1, 2\}$. The two nodes have an edge between them, so $E_1 = \{\{1, 2\}\}$. For each subsequent time step $t \geq 2$, the following random process takes place to add node $t + 1$ to the network:

(i) Flip a biased coin, $C_t \sim \text{Ber}(p)$.

(ii) If $C_t = 1$, add an edge $\{i, t + 1\}$ (for $i \leq t$) with probability:

$$\frac{D_i(t-1)}{2(t-1)}$$

where $D_i(t-1)$ is the degree of node i at the end of step $t - 1$.

(iii) If $C_t = 0$, add an edge $\{i, t + 1\}$ (for $i \leq t$) uniformly at random.

Note that the mechanism actually has a pair of root nodes. In the rest of this report, we will be concerned with recovering *either* of these nodes and will simply refer to them as a singular root node.

Likelihood Calculations

For notation: we will let $G_{t+1} \in \mathcal{V}(G_t)$ indicate that the graph G_{t+1} can be produced from the graph G_t in a single step of the PA mechanism. Note that given the mechanism description, this simply means that $V_{t+1} = V_t \cup \{t + 1\}$ and $E_{t+1} = E_t \cup \{\{i, t + 1\}\}$ for some $i \leq t$.

We are then able to write the likelihood (conditional on some value of p corresponding to the coin toss) for a transition from G_t to G_{t+1} as:

$$\mathbb{P}(G_{t+1}|G_t, p) = p \cdot \frac{D_i(t)}{2t} + (1 - p) \cdot \frac{1}{t + 1} \tag{1}$$

We will make further use of the likelihood function in Section 2.3. For now, the point is to show that we can write it in a simple form.

2.2 Root Recovery Optimization Problem

Now we will define the optimization problem to be solved, in a general form that is not concerned with a specific MIP formulation.

We are given a graph G_n, which has n nodes and has been generated by the PA mechanism described in Section 2.1. Both the root node and coin probability $p \in [0, 1]$ are unknown; these are the pieces of information we are attempting to recover.

In particular, we approach the problem by finding the most likely valid sequence of graphs and value of p that culminates in G_n (which, to emphasize, is data):

$$\begin{aligned} \max_{p, G_1, \ldots, G_{n-1}} \quad & \sum_{t=1}^{n-1} \log\left(\mathbb{P}\left(G_{t+1} | G_t, p\right)\right) \\ \text{subject to} \quad & G_{t+1} \in \mathcal{V}(G_t) \qquad \forall t \in \{1, \ldots, n-1\} \end{aligned}$$

Then, from G_1^* (in the optimal solution), we obtain an estimate for the root node of the PA process.

It is important to note that this approach for generating a root node estimate does not provide the maximum likelihood estimate of the root node itself. To do so, we would have to marginalize over all graph histories that start with a particular root node and end in G_n – but this is an intractable problem.

Our approach, which simply finds the most likely sequence of graphs, is a simplification of this problem. It hopefully provides a useful estimate of the root, while itself still being a difficult problem that is interesting to try and model with MIP.

2.3 MIP Formulation

Modelling the PA mechanism requires its logic to be broken down into component parts. In this section, we describe the sets of binary variables used in the formulation before showing how they can be constrained to accurately represent the system. Finally, we show how to represent the log likelihood in the objective function.

Establishing some notation is useful at this point. Recall that we are given a graph $G_n = (V_n, E_n)$; for ease we will let $E = E_n$ and $V = V_n$. We also let $\delta(i) \subseteq E$ be the set of edges in the final graph adjacent to node i.

Variables

The two main sets of binary variables and their interpretations are as follows:

$$x_{et} = \begin{cases} 1 & \text{if edge } e \text{ is added at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ikt} = \begin{cases} 1 & \text{if node } i \text{ has degree } k \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

Note that this means the formulation has $O(|V|^3)$ binary variables, since the maximum degree of a node at time t is exactly t, and there are $|V|$ total time steps.

Constraints

The constraints required in the formulation are as follows:

$$\sum_{e \in E} x_{et} = 1 \qquad \forall t \in [n] \tag{2}$$

$$\sum_{t=1}^{n} x_{et} = 1 \qquad \forall e \in E \tag{3}$$

$$\sum_{k=0}^{t} y_{ikt} = 1 \qquad \forall i \in V,\ t \in [n] \tag{4}$$

$$x_{e(t+1)} \leq y_{i0t} + y_{j0t} \qquad \forall e = \{i, j\} \in E,\ t \in [n-1] \tag{5}$$

$$x_{e(t+1)} \leq (1 - y_{i0t}) + (1 - y_{j0t}) \qquad \forall e = \{i, j\} \in E,\ t \in [n-1] \tag{6}$$

$$y_{i11} = \sum_{e \in \delta(i)} x_{e1} \qquad \forall i \in V \tag{7}$$

$$y_{ik(t+1)} \leq 2 - \left(y_{ikt} + \sum_{e \in \delta(i)} x_{et} \right) \qquad \forall i \in V,\ k \in [t],\ t \in [n-1] \tag{8}$$

$$y_{ik(t+1)} \geq y_{ikt} - \sum_{e \in \delta(i)} x_{et} \qquad \forall i \in V,\ k \in [t],\ t \in [n-1] \tag{9}$$

$$y_{ik(t+1)} \leq y_{ikt} + \sum_{e \in \delta(i)} x_{et} \qquad \forall i \in V,\ k \in [t],\ t \in [n-1] \tag{10}$$

$$y_{ik(t+1)} \leq y_{ikt} + y_{i(k-1)t} \qquad \forall i \in V,\ k \in [t],\ t \in [n-1] \tag{11}$$

$$y_{ik(t+1)} \geq y_{i(k-1)t} + \sum_{e \in \delta(i)} x_{et} - 1 \qquad \forall i \in V,\ k \in [t],\ t \in [n-1] \tag{12}$$

Equation (2) ensures that a single edge is selected in each time period and Equation (3) ensures that each edge is selected over the time horizon. Equation (4) is a consistency constraint that requires each node to only be assigned a single degree at each time step.

Equations (5) and (6) together ensure the following requirement: an edge, $e = \{i, j\} \in E$, can only be added at time t when either i or j (but not both) has strictly positive degree at time $t - 1$. This ensures that at each time step, a previously unconnected node is attached to a node that has already been added to the graph.

Note that in general, it would have been possible to define each y_{ikt} variable directly in terms of $\sum_{e \in \delta(i)} x_{et}$ with a standard Big-M constraint. However, a stronger formulation makes use of the relationships between the variables across successive time steps: for instance, y_{ikt} is clearly closely related to $y_{ik(t+1)}$.

With this approach the logic in the constraints needs to be treated more delicately, but the formulation is stronger in the end. Equation (7) sets the initial degrees of the nodes (after only one edge has been selected) according to the relevant $\mathbf{x}$ variables.

Given that the initial y_{ik1} variables have been defined, the logical relationships across successive time periods can be captured by the following observations:

- If node i in time t has degree k, then that same node has degree k at time $t + 1$ if and only if an adjacent edge is not selected.

  If $y_{ikt} = 1$, Equation (8) forces $y_{ik(t+1)} = 0$ whenever an adjacent edge is selected and Equation (9) forces $y_{ik(t+1)} = 1$ when none are selected.

- If node i in time t does not have degree k, then that node has degree k at time $t+1$ if and only if it has degree $k-1$ at time t and an adjacent edge is selected.

  $y_{ikt} = 0$, Equation (10) ensures $y_{ik(t+1)} = 0$ if no adjacent edge is added. Equation (11) forces $y_{ik(t+1)} = 0$ if the degree at the previous time step makes it impossible. Equation (12) ensures $y_{ik(t+1)} = 1$ when $y_{i(k-1)t} = 1$ and an adjacent edge is added.

Since these constraints together characterize the value of $y_{ik(t+1)}$ regardless of the value of y_{ikt}, they completely define the $\mathbf{y}$ variables.

Objective Function

To define the objective function we will first consider the log likelihood of a step between two valid graphs, $\log(\mathbb{P}(G_{t+1}|G_t))$. We have previously defined this probability in Equation (1), but the question remains as to how we can represent it in terms of the variables $\mathbf{x}$ and $\mathbf{y}$.

Equation (13) gives an expression for the likelihood:

$$\log(\mathbb{P}(G_{t+1}|G_t)) = \sum_{e=\{i,j\}\in E} x_{e(t+1)} \left(\sum_{k=1}^{t} (y_{ikt} + y_{jkt}) \log\left(\frac{pk}{2t} + \frac{1-p}{t+1} \right) \right) \tag{13}$$

The outer sum over the $x_{e(t+1)}$ variables picks out the inner term that corresponds to the edge, say $e = \{i, j\}$, that is added at time $t+1$. For this edge, the previous constraints ensure that either i or j is a previously unconnected node, so either $y_{ikt} = 0$ for all $k \geq 1$, or $y_{jkt} = 0$ for all $k \geq 1$.

Similarly, the constraints ensure that either $y_{ikt} = 1$ for some $k \geq 1$, or $y_{jkt} = 1$ for some $k \geq 1$. Therefore there is only a single nonzero term in the sum $\sum_{k=1}^{t}(y_{ikt} + y_{jkt})$ that will pick out the correct value of the log likelihood.

Reformulating the Objective Function

The objective involves the product of binary variables, which can be linearized using standard MIP techniques. We introduce the variable $z_{ek(t+1)} := x_{e(t+1)}(y_{ikt} + y_{jkt})$ to represent these terms. But after doing so, the objective still contains terms that are the product of $z_{ek(t+1)}$ and a nonlinear function of p. Let us define $f_{k(t+1)}(p)$ to be the value of $pk/2t + (1-p)/(t+1)$.

Reformulating this product is possible provided we have bounds on the value of the nonlinear function $f_{k(t+1)}$. It is clear that since $p \in [0,1]$ the value of $f_{k(t+1)}(p)$ lies between $k/2t$ and $1/(t+1)$. It can be verified that $k/2t \leq 1/(t+1)$ when $k = 1$ and $1/(t+1) \leq k/2t$ in all other cases. Let L_{kt} be the lower bound for a fixed k and t.

We introduce a new set of variables $q_{k(t+1)}$ to represent the value of $f_{k(t+1)}$ and define them with a convex constraint:

$$q_{k(t+1)} \leq \log\left(\frac{pk}{2t} + \frac{1-p}{t+1} \right)$$

Finally, we introduce variables $v_{ek(t+1)}$ and force them to be equal to the product $z_{ek(t+1)}q_{k(t+1)}$:

$$v_{ek(t+1)} \geq L_{kt} z_{ek(t+1)} \qquad \forall e = \{i,j\} \in E,\ k \in [t],\ t \in [n-1]$$

$$v_{ek(t+1)} \leq q_{k(t+1)} - L_{kt}(1 - z_{ek(t+1)}) \qquad \forall e = \{i,j\} \in E,\ k \in [t],\ t \in [n-1]$$

This leaves us with a nonlinear convex MIP.

3 Computational Experiments

In line with the second aim of the project that was included in Section 1, the computational experiments were designed to test both the scalability of the MIP approach and its actual ability to recover the root node.

The optimization problem that was formulated in Section 2 is a nonlinear convex MIP. Solvers are available for this class of problem, but we were restricted to an open source solver which could only solve problems on an extremely small scale (graphs with between 5 and 10 nodes).

The nonlinearity in the problem comes from optimizing over p. If we fix p, the problem becomes a standard linear MIP. Because p is a single variable, we can reasonably form a grid of values and solve the linear MIP for each value in the grid – an approach which gives us the ability to solve larger problems with a more robust class of solvers.

The computational experiments in this section correspond to the version of the problem where the value of p is fixed. Experiments were conducted as follows:

- For a fixed value of $p \in [0, 1]$ and $n \leq 50$, generate 5 random instances of a PA graph with n nodes and coin probability p.
- Run the solver on the MIP formulation problem with a time limit of 100 seconds. Though not a large time limit, we found that the majority of time was spent proving optimality by reducing the dual bound rather than obtaining a new solution. Therefore a small time limit still generally allowed a high quality solution to be obtained.

3.1 Solution Time

The solution time of the algorithm (to provable optimality) for problems with up to 20 nodes is shown in Figure 1. Of note is the large jump in solution time when the number of nodes was increased from 10 to 15, which suggests a *strongly* exponential growth trend.

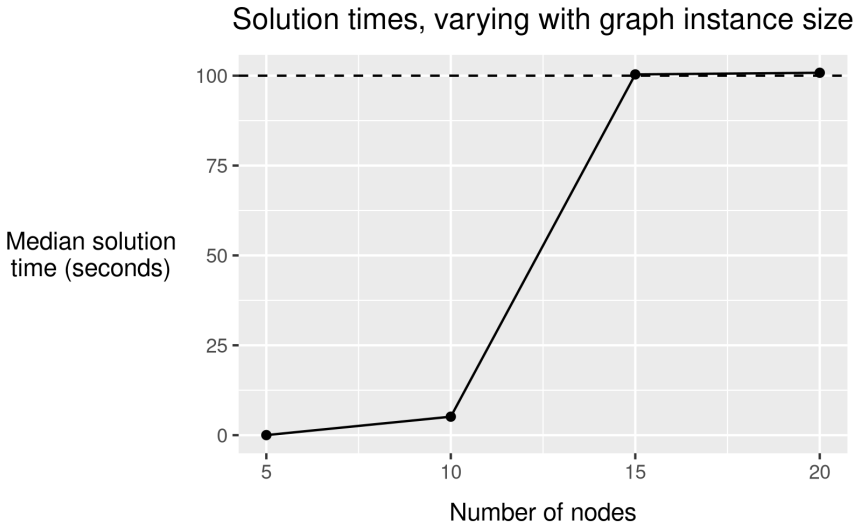

Figure 1: Solution time (to provable optimality) on problems with up to 20 nodes.

3.2 Root Recovery

Once the optimization problem had been solved, we had available to us an estimated order of attachment for the nodes in the PA graph. The performance of the algorithm was measured by the position of the root node in this ordering: if it was in position 1, then the method incurred a *root distance* of 0. If it was in position 2, then the method incurred a root distance of 1.

Figure 2 shows the median root distance over the trials for graphs with up to 20 nodes. The recovery was nearly perfect for graphs of up to 15 nodes, but there was some error once the graph size grew to 20 nodes.

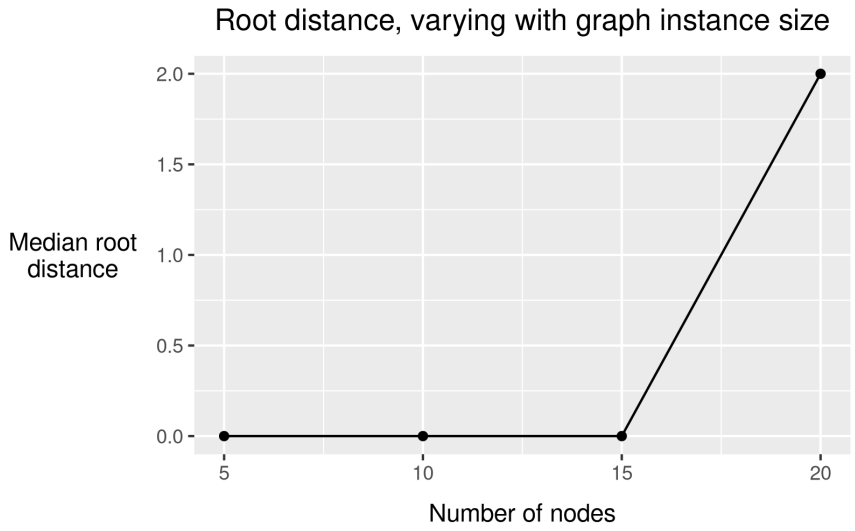

Figure 2: Median root distance for problems with up to 20 nodes.

4 Conclusions and Discussion

Though it was possible to model the PA mechanism using MIP (which itself was interesting), the method was disappointingly unable to scale very well at all. This leads into some questions surrounding whether it would be a useful the combine the MIP model with an aggregation technique that first breaks the given graph down into smaller pieces, before combining information to find the root of the entire graph.

For the problem sizes that it was able to solve, the MIP approach was able to recover the root node fairly accurately. It would have been informative to conduct some further tests that compare its results with some simple heuristics based on measures such as node centrality.

It is likely that the MIP approach suffers from *overfitting*. As mentioned in Section 2, the optimization problem which finds the most likely sequence of graphs provides us with only an estimation of the most likely root node – it does not take into account marginalization over all graph histories. There may be different formulations of the problem that allow us to get closer to finding the most likely root node, and these could be interesting to explore using MIP. However, the issue of scalability limits my excitement for this possibility.

References

[1] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[2] T. Pham, P. Sheridan, and H. Shimodaira, "Pafit: A statistical method for measuring preferential attachment in temporal complex networks," *PloS one*, vol. 10, no. 9, p. e0137796, 2015.

[3] S. Navlakha and C. Kingsford, "Network archaeology: uncovering ancient networks from present-day interactions," *PLoS Comput Biol*, vol. 7, no. 4, p. e1001119, 2011.

[4] V. Rajan, C. Kingsford, and X. Zhang, "Maximum likelihood reconstruction of ancestral networks by integer linear programming," *bioRxiv*, p. 574814, 2019.

[5] J. Cussens, M. Bartlett, E. M. Jones, and N. A. Sheehan, "Maximum likelihood pedigree reconstruction using integer linear programming," *Genetic epidemiology*, vol. 37, no. 1, pp. 69–83, 2013.

[6] D. J. de Solla Price, "Networks of scientific papers," *Science*, vol. 149, no. 3683, pp. 510–515, 1965.