Small Problem 6: Network Analysis Expressiveness Challenge

1. Motivation

Graphs are ubiquitous in many "Big Data" applications. Hence, many machine learning algorithms and methods must incorporate and learn on graphical data. This often requires modeling the generative process of a graph, and applying evidence to the model in the form of properties of the final graph.

2. Model

This task uses a simple generative model of a mixed preferential/uniform attachment graph, as shown in Figure 1.

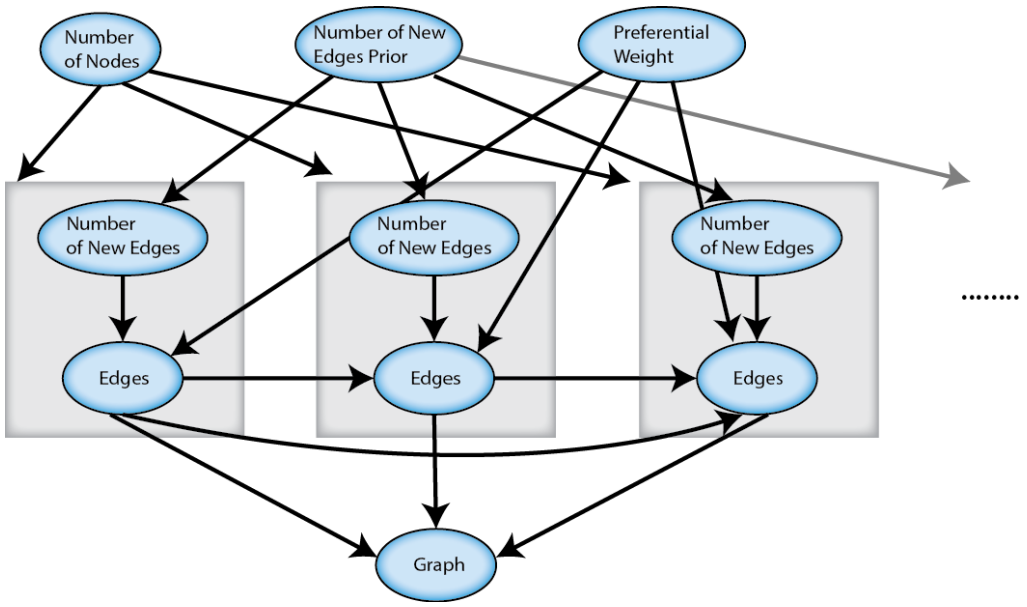

Figure 1: Preferential Attachment Graph Generative Model

The *Number of Nodes* variable determines how many nodes are in the graph. The nodes are generated in succession, and each newly generated node has a distribution over the number of new edges it will create, in the *Number of New Edges* variable. The *Edges* variable for a node represents a set of new edges (whose size depends on *Number of New Edges*). As this is a mixed preferential/uniform attachment model, the *Edges* variable for a node also depends on all of the *Edges* variables for previously created nodes. The variable *Weight* models the weight of the preferential attachment model

for edge creation; values closer to one indicate more weight to a preferential attachment policy of creating new edges, whereas values close zero give more weight to a uniform attachment policy. The probability that a newly created node *i* creates an edge to node *j* is:

$$P(i\ connects\ to\ j) = \ Weight * \frac{|j.Edges| + 1}{|Nodes| + |Edges|} + (1 - Weight) * \frac{1}{|Nodes|}$$

Where $|j.Edges|$ is the number of edges connecting to node *j* and $|Nodes|$ and $|Edges|$ are the current total number of nodes and edges in the graph, respectively.

Finally, the *Graph* variable represents the final graph representation of the generative process. All edges are treated as undirected.

Here is pseudo-code for the graph generation process:

$NumberOfNodes \sim P(NumberOfNodes)$ // Draw the total number of nodes
$Edges := 0$
$Edge := NumberOfNodes \times NumberOfNodes\ array\ of\ zeros$
For $i = 1, \dots, NumberOfNodes$ // Generate edges connecting node i to previously-generated nodes
    $i.NumberNewEdges \sim P(NumberNewEdges)$ // How many edges to attempt to add?
    // Compute the discrete distribution for the edge probabilities
    For $j = 1, \dots, i - 1$
        $P(i,j) := Weight * \frac{|j.Edges|+1}{|i|+|Edges|} + (1 - Weight) * \frac{1}{|i|}$
    // Now add the edges
    For $k = 1, \dots, i.NumberNewEdges$ do
        $j \sim P(i,j)$ // sample a "destination" node
        If $(Edge(i,j) == 0)$
            $Edges := Edges + 1$ // add an edge and count it
            $Edge(i,j) := Edges(j,i) := 1$

3. Challenge Task and Metrics

The task on this problem is to design and implement the graph generative model in such a way that arbitrary hard or soft constraints can be applied to the final, observed graph and used to infer properties of the graph, such as the number of new edges prior or the edge attachment mixing weight.

Performers will be judged on the expressiveness and conciseness of the constraints and efficiency in reasoning with the constraints. Performers may optionally produce performance profile curves (accuracy of answer versus amount of CPU time).

Performers are free to define their own distributions for the model parameters, such as the number of nodes and the number of new edges, as long as they are able to demonstrate reasoning about the posterior distribution given evidence (outlined in the Appendix).

4. Appendix

In this section, $Weight$ is written as $Preferential\ Weight$, since it is the weight placed on the preferential attachment model.

a. Constraint and Query 1

Constraint	Relative Weights
[1]$Clustering\ Coefficient\ (CC)\ of\ Graph$	*0.4 <= CC <= 0.6 -> 3.0* *0.2 <= CC < 0.4 -> 2.0* *0.6 < CC <= 0.8 -> 2.0* *0.0 <= CC < 0.2 -> 1.0* *0.8 < CC <= 1.0 -> 1.0*

Query	Value
$Distribution\ of\ Preferential\ Weight$	?
$Distribution\ of\ Number\ of\ New\ Edges\ Prior$	?

b. Constraint and Query 2

5. Constraint	Relative Weights
[2]$Probability\ of\ Nodes\ with\ at\ least\ k\ Edges\ is\ k^{-2}$	$Normal(P_k - k^{-2}, 0.01)$

Query	Value
$Distribution\ of\ Preferential\ Weight$	?
$Distribution\ of\ Number\ of\ New\ Edges\ Prior$	?

1. The Clustering Coefficient of a graph is defined as the average clustering coefficient of a node. The Clustering Coefficient of a node C_i is defined as:

$$C_i = \frac{2|e_i \in N_i|}{|N_i||N_i - 1|}$$

Where N_i is the set of nodes directly connected to node i and e_i is an existing edge between any two nodes in N_i.

2. This constraint is really on the cumulative degree distribution of the network. That is, the fraction of nodes in the graph with at least one edge should be 1.0, with at least two edges 1/4,

etc. This assumes that the minimum number of edges that at node has is one. Performers may restrict the constraint to a reasonable finite range of node degrees.