Implementation and Analysis of Preferential Attachment Model

Frija Reinert, Sam Haseth

Abstract

This report documents our implementation of a Preferential Attachment Model and its analysis. The model generates graphs with varying numbers of nodes and explores their degree distributions. This report presents the model's implementation details, results, and analysis of log-log plots of degree distributions for simulated graphs with 10,000, 20,000, 50,000, and 100,000 nodes. Additionally, we will discuss the power law behavior of these plots.

Introduction

Preferential attachment is a key concept in network theory, describing the growth of complex networks according to their existing distribution. Where, the more popular an item, or node, is, the more likely it is to continue growing in popularity, or increase its degrees/attachments to other nodes. This preferential attachment model demonstrates this idea that networks related to popularity follow the “rich get richer” growth of power laws. This report explores the implementation of this model and analyzes the degree distribution of generated graphs.

Implementation

Random Selection of Nodes

The *randSelect* function selects a random node based on weighted probabilities from a list of partners. From a list containing all existing partners of nodes, the function first selects a random partner set, and then randomly selects one of those two nodes and returns it.

Cluster Coefficient Calculation

The *calcCCE* function calculates the Cluster Coefficient, which measures the degree of interconnectedness within a network, by dividing the # of closed triplets by the total # of triplets (opened and closed). However, due to the simplicity of this model, the Cluster Coefficient will always be zero.

Data Conversion

The *convertData* function converts the node/partner data to a format which is suitable for creating the log-log graphs. Where originally the data is stored in a format representing how many “partners” (other nodes a single node is attached to) each individual node has, it is converted to a format representing for each amount of partners, how many nodes have that amount of partners. This function then calls the function *writeData*, forwarding the data it just converted to generate a CSV file.

Data Writing

The *writeData* function uses the OpenCSV library to write the converted data to a CSV file which is then used to create log-log graphs. A tutorial provided by GeeksForGeeks was followed when using OpenCSV to create a file and write data to it.

Preferential Attachment Model

The *paModel* function implements the Preferential Attachment Model. It initiates the model with two connected nodes and runs for a specified number of time steps. For each time step, the model creates a new node, randomly selects a partner for that new node using *randSelect*, and updates the data accordingly. After the model has finished running, it calls the *convertData* function to convert the data and export a CSV file.

Results

Graph Generation

The Preferential Attachment Model was used to generate the degree distribution of networks consisting of 10,000, 20,000, 50,000, and 100,000 nodes. After exporting this data in a CSV file, Google Sheets was used to create log-log plots of each set of data. Each graph represents the degree distribution of nodes over the specified number of time steps.

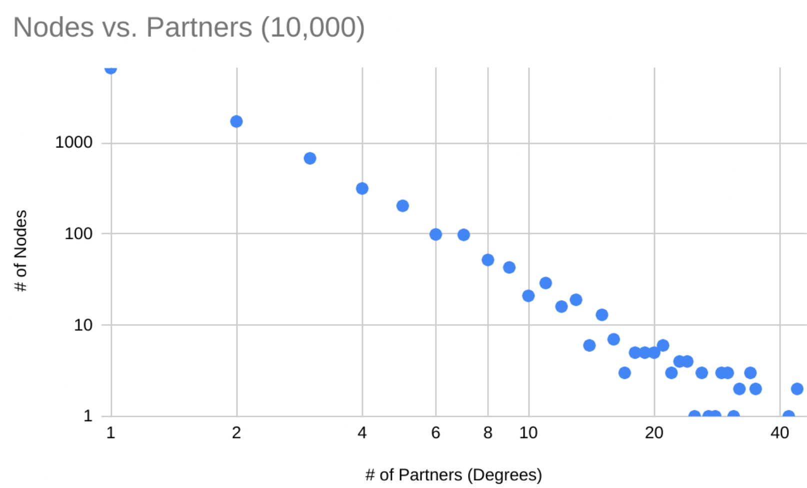

Figure 1a: 10,000 Nodes; Slope = -1.98

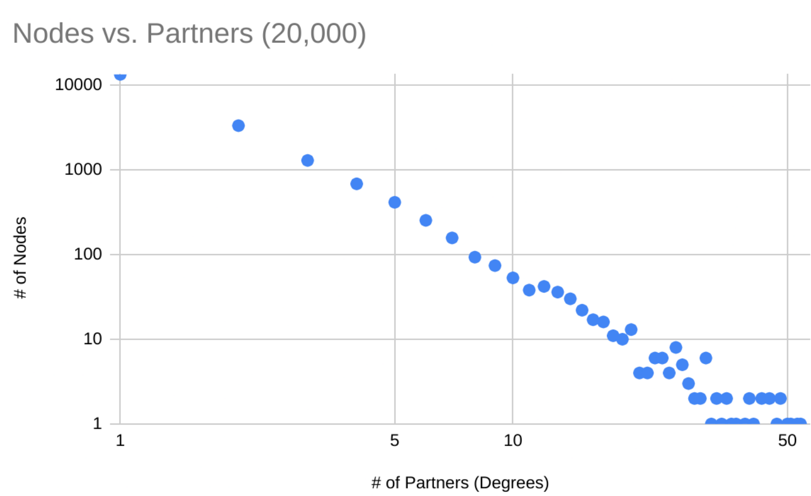

Figure 1b: 20,000 Nodes; Slope = -2.04

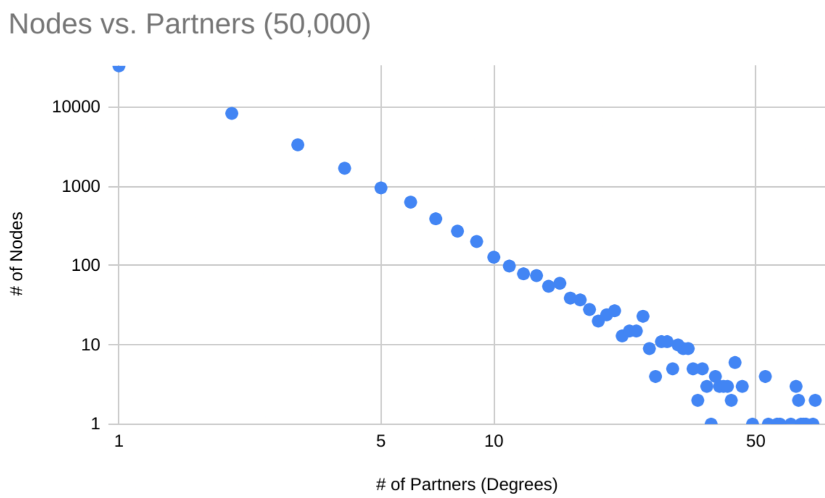

Figure 1c: 50,000 Nodes; Slope = -2.15

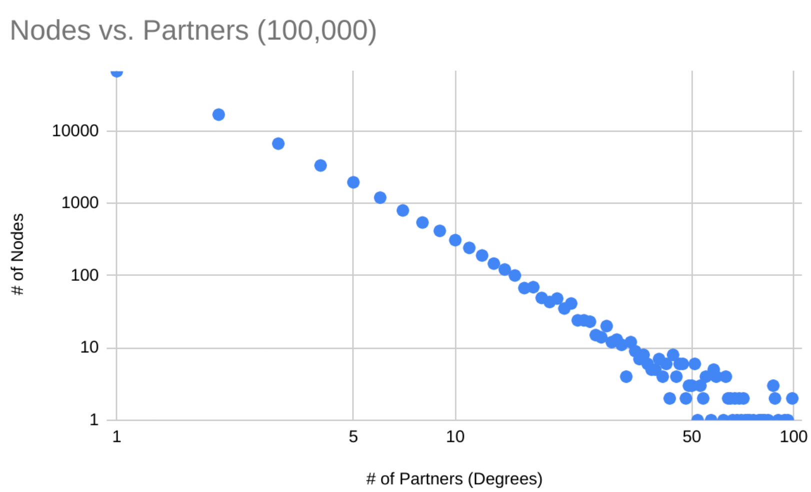

Figure 1d: 100,000 Nodes; Slope = -2.1

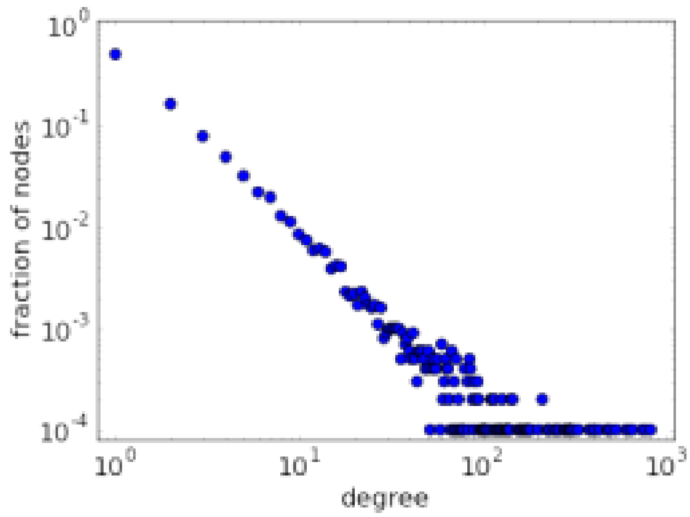

Figure 2: an example of a log-log plot of a power law distribution (Math Insight)

Discussion

Analysis of Log-Log Plots

The purpose of generating log-log plots of degree distributions is to determine whether the generated graphs follow the growth of a power law. When plotted on a log-log scale, a power-law distribution will follow a straight line terminated with a heavy tail. The heavy tail is a result of having very few nodes with exceedingly large amounts of partners/degrees. As the model was run for increasingly large networks, the resulting log-log plot became straighter, denser, and developed a more prominent heavy tail, altogether more closely resembling the distribution of a power law. Looking at the 100,000-node graph (Figure 1d), one can clearly see the line is relatively straight with a heavy tail, but such a distribution is vaguely present in the 10,000 node graph (Figure 1a) as well. Included also, is a log-log scale plot of a power law distribution for reference.

Conclusion

This report presented our implementation of a preferential attachment model and an analysis of its results. The model efficiently generated large networks, and the log-log scale graphs derived from the degree distributions of such networks proved to accurately demonstrate the behavior of a power law. The patterns between the graphs resulting from their scale were also noted.

References

Aman, Singh, and Adnan Irshad. "Writing a CSV file in Java using OpenCSV." *GeeksforGeeks*, 7 Dec 2021, https://www.geeksforgeeks.org/writing-a-csv-file-in-java-using-opencsv/. Accessed 2 Oct 2023.

Jones, Andrew R., and Scott Conway. "OpenCSV-5.8." *SourceForge*, 23 Jul 2023, https://sourceforge.net/projects/opencsv/files/opencsv/. Accessed 2 Oct 2023.

"Plot of power-law degree distribution on log-log scale." *Math Insight*. http://mathinsight.org/image/power_law_degree_distribution_scatter. Accessed 2 Oct 2023.