CS224W Project Milestone Report

Sam Schwager and John solitario

sams95@stanford.edu and johnny18@stanford.edu

November 9, 2018

Abstract

I. Introduction

When analyzing early-stage companies, entrepreneurs and investors alike need to assess a company's exit opportunities with a priority placed on acquisitions and initial public offerings (IPOs). For instance, given information about investments made in a company, an investor may seek to determine whether or not a company will be acquired. Moreover, entrepreneurs may try to predict which types of companies will acquire their companies. In general, acquisition potential can have a substantial effect on a company's outlook, impacting valuations, future rounds of funding, and investor backing.

Currently, in order to assess a company's acquisition potential, interested parties may assess a company's total addressable market, product positioning, and/or financial projections. However, a company's acquisition likelihood depends on a host of additional characteristics, like industry, stage, location, and other recent acquisitions. Nonetheless, copious amounts of past funding and acquisition data exist that an investor or entrepreneur could use to approach the problem more rigorously. However, interested parties may have trouble finding and choosing a tractable representation of the data. In this paper, we seek to determine the acquisition potential of early-stage companies by analyzing networks derived from publicly available acquisition and investment data.

II. Data

In October 2013, Crunchbase, an online platform for finding business information about private and public companies, released investment data for roughly 18,000 startups, nearly 4,700 acquisitions, and over 52,000 investment events. Crunchbase provided the data publicly in four separate datasets: *Companies*, *Rounds*, *Investments*, and *Acquisitions*.

i. Companies

Within the *Companies* dataset, each row corresponds to a company, founded between 1906 and 2013, with the majority of funding rounds occurring between 2010 and 2013. For each company, the dataset provides information about the industry, total funding amount, number of funding rounds, operating status and operating location. The dataset also details several dates related to funding rounds.

ii. Rounds

The *Rounds* dataset provides information about each funding round for the companies listed in the *Companies* dataset. Each row corresponds to a company and its respective funding round (angel, venture, series-a, series-b, series-c+, private-equity, or other). Each row provides basic company information, along with details about funding dates and amounts.

iii. Investments

The *Investments* dataset provides information about investments that companies in the *Companies* dataset have received. Each row corresponds to a specific investment and contains information about the company receiving the investment and the company making the in-

vestment. The dataset also provides details about the size of the investment, the corresponding funding round, and associated dates.

iv. Acquisitions

Within the *Acquisitions* dataset, each row corresponds to an acquisition event for companies in the *Companies* dataset. The dataset also provides information about the acquired company, the acquiring company, the acquisition amount, and any relevant dates.

III. Network Creation

A multitude of different networks naturally arise from the available Crunchbase data. However, in order to assess a company's acquisition potential, we have focused on two networks: one from the *Investments* dataset and another derived from the *Acquisitions* dataset.

i. Investors to Companies

In the first network we built, companies represent one set of nodes, while investors represent the other set of nodes. Since early-stage companies rarely invest in other early-stage companies and investors in early-stage companies rarely invest in other investors (for the most part), the network will have a bipartite structure. Edges within the network represent investment instances, linking investors to companies. The edges are directed, with investors as the source nodes and companies as the destination nodes. Since investors may invest in a company multiple times through subsequent funding rounds, we have allowed for multiple edges between nodes in the network.

ii. Companies to Acquirers

The second network we built links companies to acquirers. Companies represent one set of nodes and acquirers represent another set of nodes. Since early-stage companies rarely acquire other early-stage companies and acquirers rarely acquirer other acquirers (for the most part), the new *Companies-to-Acquirers* network will also have a bipartite structure. Edges within the network represent acquisition instances, linking acquirers to companies. The edges are also directed with companies as destination nodes and acquirers as source nodes.

Metrics	Networks	
	Investors-to-Companies	Companies-to-Acquirers
Company Nodes	11,572	4,563
Investor Nodes	10,465	0
Acquirer Nodes	0	2,641
Total Nodes	21,945	6,926
Total Edges	52,868	4,651
Network Density	1.09×10^{-4}	9.69×10^{-5}
Clustering Coefficient	3.46×10^{-4}	1.25×10^{-4}

Table 1: *Network Statistics for Investors-to-Companies and Companies-to-Acquirers*

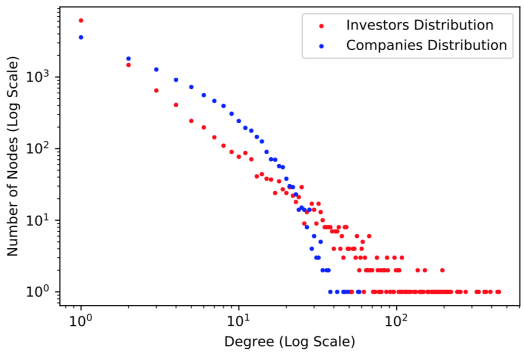

Figure 1: *Degree distribution for the Investors-to-Companies network*

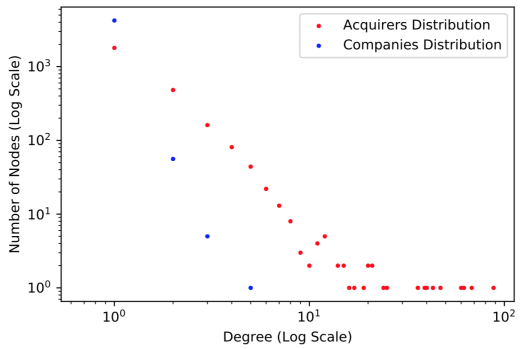

Figure 2: *Degree distribution for the Companies-to-Acquirers network*

IV. Preliminary Network Analysis

After creating the *Investors-to-Companies* network and the *Companies-to-Acquirers* network, we computed a range of statistics for each network (See **Table 1**) and plotted degree distributions (See **Figure 1** and **Figure 2**).

First, we notice that both graphs do not have a true bipartite structure. In the *Investors-to-Companies* network, 92 nodes overlap ($22,037 - 21,945 = 92$), which means 92 companies that received investments have also made investments. In the *Companies-to-Acquirers* network, 278 nodes overlap ($7,204 - 6,926 = 278$), which means 278 companies that made acquisitions were then acquired. Although both networks are not "true" bipartite networks, they both retain a predominantly bipartite structure.

Second, both graphs have very low network densities and clustering coefficients, which arises from their predominantly bipartite structures. Third, the degree distribution plot for the *Investors-to-Companies* network reveals that a wide-range company and investor types exist. A substantial number of companies and investors will only make or receive one investment, while another significant portion will make or receive a multitude of investments. Fourth, the *Companies-to-Acquirers* network similarly shows that a wide-range of acquirers exist, as most acquirers only make one acquisition but some acquirers make dozens of acquisitions. [1]

V. Extended Analysis

In order to further analyze the *Investors-to-Companies* and *Companies-to-Acquirers* networks, we created network projections.

In the *Invested-Companies* network, nodes represent companies, and two companies are connected to each other if there is at least one investor who has invested in both companies. Formally, the *Invested-Companies* network is a graph $G'(V', E')$ with V' = the set of all companies (from *Investors-to-Companies*), and there is an edge (i, j) between companies i and j if there is an investor y, such that $(i, y) \in G$ and $(j, y) \in G$ (where G is the original *Investors-to-Companies* network).

Metrics	Network Projections	
	Invested-Companies	Acquired Companies
Total Nodes	11,572	4,563
Total Edges	398,154	16,184
Network Density	5.95×10^{-3}	1.55×10^{-3}
Clustering Coefficient	0.573	0.381

Table 2: *Network statistics for network projections (folded bipartite graphs)*

In the *Acquired-Companies* network, nodes represent companies, and two companies are connected to each other if they were acquired by the same acquirer. Formally, the *Invested-Companies* network is a graph $G'(V', E')$ with V' = the set of all companies (from *Companies-to-Acquirers*) , and there is an edge (i, j) between companies i and j if there is an acquirer y, such that $(i, y) \in G$ and $(j, y) \in G$ (where G is the original *Companies-to-Acquirers* network).

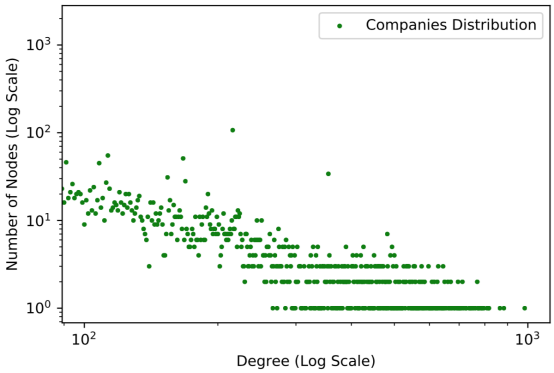

Figure 3: *Degree distribution for the Invested-Companies network*

[1] Note: On the degree distribution plot for the *Companies-to-Acquirers* network, several company nodes have an out-degree greater than one. These nodes represent companies that made acquisitions and were then acquired.

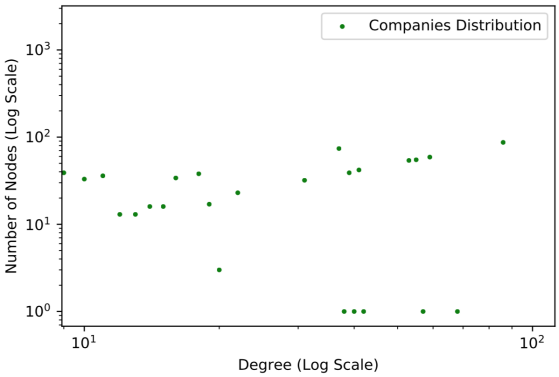

Figure 4: *Degree distribution for the Acquired-Companies network*

After creating the *Invested-Companies* and *Acquired-Companies* network projections, we computed a range of statistics for the new networks (see **Table 2**) and plotted the degree distributions of their nodes (see **Figure 3** and **Figure 4**).

First, we can see that the number of company nodes remains the same for the *Invested-Companies* and *Acquired-Companies* networks, at 11,572 and 4,563 respectively. This is required to ensure proper folding of the original graphs. Second, we see a substantial increase in the number of edges within the *Invested-Companies* network, which means many companies have investors in common. We see similar behavior in the *Acquired-Companies* network but to a lesser degree. The increase in edges corresponds to an equivalent increase in the densities of each network.

Third, we see a significant increase in the clustering coefficients for both networks. In our analysis of the original networks, we noted that a substantial number of prolific investors and acquirers exist. Therefore, in the projected networks, these prolific investors and acquirers lead to the creation of highly clustered groups, which increases the average clustering coefficient. The existence of these prolific investors and acquirers also helps explain the degree distributions for the *Invested-Companies* and *Acquired-Companies* networks, as most nodes have a relatively high degree.

VI. Link Prediction

Only 1,508 out 11,572 companies within the *Investors-to-Companies* network have been acquired with acquisition information coming from the *Companies-to-Acquisitions* network. As a first step, we want to see if we can accurately identify the 1,508 acquired companies within the *Investors-to-Companies* network as being acquired. In order to do so, we will use the Jaccard Similarity Index. For any two nodes i and j, the Jaccard Index is defined as

$$JA(i,j) = \frac{|\ \Gamma_i \cap \Gamma_j\ |}{|\ \Gamma_i \cup \Gamma_j\ |} \tag{1}$$

where Γ_i is the set of neighbors of node i. First, we will choose a subset of the acquired companies in the *Investors-to-Companies* network (i.e. our training set) and then compute the pairwise Jaccard similarity scores for all of the companies in that subset. Next, for every other company in the network not in the training set, we will compute the pairwise Jaccard similarity scores between that company and all of the companies in the training set, and then take the average of all those scores. If the average is above a given threshold (i.e. the average Jaccard index score among all the nodes in the training set), we will predict the given company is an acquired company. Ideally, this method will allow us to predict that a company will be acquired if and only if it is one of the acquired companies that we left out of our training set. Our first implementation using Jaccard Index scores will serve as our baseline.

We plan to further augment our Jaccard prediction mechanism using the weighted characteristics of *Investors-to-Companies* network, which we clarified earlier as funding amount. However, the Jaccard prediction mechanism in general relies on acquired companies sharing common investors. Therefore, as we move forward, we plan to explore a multitude of other prediction mechanisms.

VII. Next Steps

We have laid out the following next steps to augment our basic prediction mechanism:

1. We have approximately 11,500 companies

within our *Investors-to-Companies* network, but we only have acquisition data for 1,500 of them. Therefore, we plan to augment our acquisition data by exploring the outcomes of other companies in *Investors-to-Companies*. We expect to find additional companies that have been acquired.

2. For an early-stage company, an acquisition remains an optimal outcome. However, an IPO is also a very favorable outcome. Therefore, we can further augment our *Investors-to-Companies* network with IPO information.

3. Within the *Companies* dataset, we also have additional information about companies, like location and industry. Thus, we can add this additional information as either new nodes or embeddings.

4. As mentioned early, we also plan to explore additional prediction techniques. Specifically, we will create latent representations of acquired companies. Ideally, we will have similar embeddings for acquired companies and different embeddings for companies that have not yet been acquired. Finally, we will use a variety of machine learning techniques to make predictions based off the embedding.

VIII. Implementation

A considerable amount of work went into creating sufficiently complex graph representations of the investment and acquisition data. The vast majority of our initial development work went into creating a script called CSVToGraph.py (∼350 lines) that would allow us to generate arbitrarily complex graphs from the CSV data. The script allows the user to stipulate (from the command line) what type of graph she wants (TUNGraph, TNGraph, TNEANet) and whether or not the graph is bipartite. If the user selects the more complex TNEANet type, then he can stipulate an arbitrary number of attributes for the edges, source nodes, and destination nodes. These attributes can be of type int, float, or string, as specified in the SNAP documentation for the TNEANet graph type. Additionally, whenever a user generates a graph, a directory is created containing the graph itself and other auxiliary data (e.g. bipartite classes and a mapping from node IDs to original values in the CSV files).

Next, understanding that we would need to fold the bipartite graphs we had created in order to compare companies-to-companies, investors-to-investors, etc., we developed a script called FoldBipartiteGraph.py (∼125 lines) that allowed us to seamlessly fold any of the bipartite graphs we created using the CSVToGraph.py script. This script allows us to fold bipartite graphs in both "directions" (e.g. a bipartite graph with investors as source nodes and companies as destination nodes could be folded such that the resulting projection consists of companies with common investors or investors with common companies). Every time the script is run successfully, a new projection graph of type TUNGraph is stored in the corresponding graph directory, allowing for optimal analysis work flow and organization.

Finally, we developed a script called BasicGraphAnalysis.py along with a Jupyter notebook (∼300 lines total) that we used for analyzing both basic metrics for the graphs we created (e.g. density and clustering coefficient), generating plots (e.g. degree distribution), and performing basic link prediction. At this point, we are using Jaccard similarity in the folded graphs to predict acquisitions, but our link prediction methods will become more sophisticated as we proceed, as discussed in the Next Steps section above.