Venture Capital Investment Networks: Creation and Analysis

Sam Schwager (sams95@stanford.edu) and John Solitario (johnny18@stanford.edu)
https://github.com/SRS95/CS224W

Abstract— **The venture capital landscape and the existence of syndicated investments naturally leads to the formation of intricate networks. However, little attention has been paid to early-stage, start-up companies within these networks. In this paper, we create a variety of networks from publicly available venture capital data. We then perform a variety of analyses on these networks, ranging from basic analysis to sophisticated latent representations and network deconvolution. Our approach can be bucketed into the following categories: basic graph analysis and comparison, analysis of node centrality, community detection, and network deconvolution. Finally, we find promising results leveraging node-level latent representations as features in supervised learning applications.**

I. INTRODUCTION

The inception of the venture capital industry in the United States dates back to the early 1950's, following a few deals made shortly after the end of World War II. The venture capital industry grew slowly through the 1960's and 1970's, but, by the 1980's, the rise of a new institutional foundation allowed for a rapid growth in transactions with respect to volume and value. By the early 2000's, over 103,000 venture capital investments had been recorded, and dozens of companies had grown from early-stage, start-ups to Fortune 500 powerhouses.

Networks feature prominently within the venture capital industry. A given venture capital firm's network might include portfolio companies, investors, and other venture capital firms. In most cases, several venture capital firms will join together to invest in a single start-up, which allows them to distribute investment risk over multiple parties. The combination of investments by multiple venture capital firms in a single start-up is regarded as *syndication* or a *syndicated deal*. Syndicated deals lead to an interconnected network of venture capital firms, related by their co-investments. Although a variety of research explores the emergent properties of venture capital networks, the literature has paid little attention to the most prominent portion of these networks: early-stage, start-up companies. More so, we can derive an extensive amount of information about these start-ups from their positions within venture capital networks.

In this paper, we first create a variety of venture capital networks, consisting of early-stage companies, venture capital firms, investment transactions, and other relevant information. Second, we perform analyses on these networks and various network projections, including a careful evaluation of degree distributions and other network statistics. Third, we explore node centrality for each of the created networks, leveraging degree centrality and eigenvector cenrality. Fourth, we perform community detection, starting with the Louvain Algorithm and then progressing to clustering on latent representations via node2vec. Finally, we perform network deconvolution to extract direct relationships among start-up companies.

II. RELATED WORKS

A. Modeling Venture Capital Networks

In the late 1980's, William Bygrave began exploring the underlying networks of the venture capital community. He started by analyzing joint investments made by venture capital firms in a sample of 1,501 portfolio companies for the period 1966-1982 [3]. He then modeled the venture capital industry as an explicit network, linking venture capital firms together by their joint investments in portfolio companies [4]. With the newly created network, Bygrave performed a set of rudimentary analyses on node centrality with a focus on venture capital firms that invest in "highly innovative technology companies." To measure node centrality, Bygrave leveraged the following metrics:

$$Sum\ of\ Links\ = \sum_j [d(i,j)] \tag{1}$$

$$Sum\ of\ Coinvestments\ = \sum_j [n(i,j)] \tag{2}$$

$$Sum\ of\ Weighted\ Links\ = \sum_j [w(i,j)d(i,j)] \tag{3}$$

where $d(i,j)$ represents distance, $n(i,j)$ represents coinvestment amount, and $w(i,j)$ represents connection strength between two venture capital firms, i and j.

Building off of Bygrave's early work, Podolny showed that venture capital firms with a deal-flow network spanning structural holes invest more often in early product development and more successfully develop their early-stage investments into profitable IPOs [11]. Similar to Podolony's work, Ljungqvist et al. discovered that better networked venture capital firms experience significantly better fund performance, and similarly, portfolio companies of better-networked venture capital firms are significantly more likely to survive to subsequent financing rounds and eventual exit [10].

Stuart and Sorenson extended their research to focus on the geographical distribution of venture capital firms, demonstrating that social networks within the venture capital community diffuse information across boundaries and expand their spatial radii of exchange [12]. In contrast, Kogut et al. showed the rapid emergence of a national network of venture capital syndications by analyzing over 159,561 venture capital investment transactions over nearly 45 years [13]. More so, Kogut et al. posit that a national venture capital investment network subsumes local networks, and new venture capital firms, in general, reject preferential attachment in favor of repeated ties among trusted partners.

B. From basic analysis to latent representations

To perform advanced community detection and link prediction, Hamilton et al. explore *node embeddings*, in which algorithms encode nodes as low-dimensional vectors, summarizing their graph positions and the structure of their local graph neighborhoods [7]. More so, their approach has three key components:

- **Encoder Function:** maps nodes to vector embeddings $z_i \in \mathbb{R}^d$, where z_i corresponds to the embedding for node $v_i \in V$.
$$ENC: \; V \rightarrow \mathbb{R}^d \tag{4}$$
- **Decoder Function:** decodes user-specified graph statistics from node embeddings. The following exemplifies a pairwise decoder,
$$DEC: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+. \tag{5}$$
- **Loss Function:** determines how the quality of the pairwise reconstructions are evaluated in order to train the model.
$$L = \sum_{(v_i, v_j) \in D} l(DEC(z_i, z_j), s_G(v_i, v_j)), \tag{6}$$
where l is a user-defined loss function and D is a set of training node pairs.

Given the above framework, a variety of shallow embedding approaches have been devised to learn node embeddings based on random walk statistics. These approaches learn embeddings to achieve the following:

$$DEC(z_i, z_j) \triangleq \frac{e^{z_i^T z_j}}{\sum_{v_k \in V} e^{z_i^T z_j}} \approx p_{G,T}(v_i | v_j), \tag{7}$$

where $p_{G,T}(v_i|v_j)$ is the probability of visiting v_j on a length-T random walk starting at v_i. More formally, these approaches seek to minimize the following cross-entropy loss:

$$L = \sum_{(v_i, v_j) \in D} -log(DEC(z_i, z_j)), \tag{8}$$

where the training set D is generated by sampling random walks starting from each node.

In particular, node2vec allows for a flexible definition of random walks by introducing two hyper-parameters, p and q, that bias the random walk [6]. The introduction of p and q allow the node2vec algorithm to interpolate between pseudo breadth-first search and depth-first search walks. Therefore, we can leverage node2vec to capture representations of local neighborhoods for a given node, along with more expansive structural roles.

With node embeddings, we can carry out clustering and community detection, which has been shown effective in a variety of applications [5]. In particular, we can apply multiple generic clustering algorithms to our set of learned node embeddings. We can also leverage node embeddings to carry out link prediction (i.e. predict edges that are likely to form in the future) [1].

III. DATA

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

A. Companies

Within the *Companies* data set, 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 data set provides information about the industry, total funding amount, number of funding rounds, operating status, and operating location. The data set also details several dates related to funding rounds.

B. Rounds

The *Rounds* data set provides information about each funding round for the companies listed in the *Companies* data set. 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.

C. Investments

The *Investments* data set provides information about investments that companies in the *Companies* data set have received. Each row corresponds to a specific investment and contains information about the party receiving the investment and the party making the investment. The data set also provides details about the size of the investment, the corresponding funding round, and any associated dates.

D. Acquisitions

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

IV. NETWORK CREATION AND ANALYSIS

A multitude of different networks naturally arise from the available Crunchbase data. However, since we want to analyze early-stage, start-up companies and how they relate to venture capital investors, we focus on four networks derived from the *Investments* data set: *Investors-to-Companies*, *Investors-to-Investors*, *Companies-to-Companies*, and then an augmented version of the *Companies-to-Companies* network. Furthermore, the networks we derive from the *Investments* data set implicitly include relevant information from the *Companies* and *Rounds* data sets.

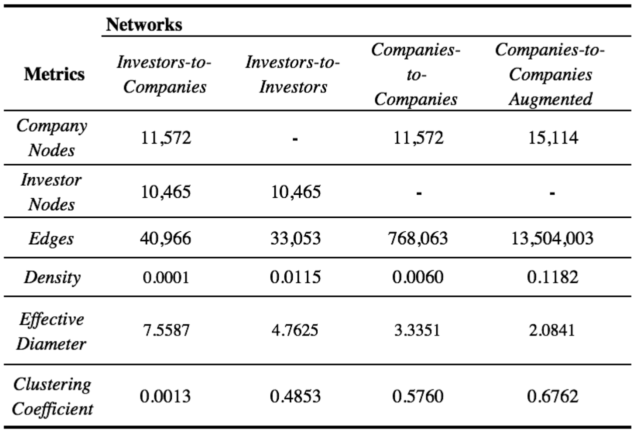
A. Network Creation

In the *Investors-to-Companies* network, 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 venture capital firms rarely invest in other venture capital firms, the network has a bipartite structure. Note the *Investments* data set includes a wide variety of investor types outside of the standard venture capital firms. 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. Although investors can invest in a company multiple times through subsequent funding rounds, we only allow for a single edge between two given nodes for simplicity.

We derive the *Companies-to-Companies* and *Investors-to-Investors* networks by creating network projections of the *Investors-to-Companies* network. In the *Companies-to-Companies* network, nodes represent companies, and two companies are adjacent if there is at least one investor who has invested in both companies. Formally, the *Companies-to-Companies* network is a graph $G'(V', E')$ with V' = the set of all companies from the *Investors-to-Companies* network. 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.

Similarly, in the *Investors-to-Investors* network, nodes represent investors, and two investors are adjacent if they have invested in at least one start-up together. Formally, the *Investors-to-Investors* network is a graph $G'(V', E')$ with V' = the set of all investors from the *Investors-to-Companies* network. There is an edge (i, j) between investors i and j if there is a company y, such that $(i, y) \in G$ and $(j, y) \in G$, where G is the original *Investors-to-Companies* network.

Finally, in order to incorporate more information into the *Companies-to-Companies* network, we define the *Companies-to-Companies-Augmented* network as a network with the nodes being all of the companies we are considering and wherein there exists an edge between two nodes if they share an investor, or if they are in the same region or industry.



	Networks			
Metrics	*Investors-to-Companies*	*Investors-to-Investors*	*Companies-to-Companies*	*Companies-to-Companies Augmented*
Company Nodes	11,572	-	11,572	15,114
Investor Nodes	10,465	10,465	-	-
Edges	40,966	33,053	768,063	13,504,003
Density	0.0001	0.0115	0.0060	0.1182
Effective Diameter	7.5587	4.7625	3.3351	2.0841
Clustering Coefficient	0.0013	0.4853	0.5760	0.6762

Fig. 1: Metrics for the *Investors-to-Companies* network and aforementioned network projections. **Note**: the *Companies-to-Companies Augmented* network has comparatively more companies, as the additional information leads to the inclusion of more company nodes.

B. Preliminary Analysis

After creating the *Investors-to-Companies*, *Investors-to-Investors*, *Companies-to-Companies*, and *Companies-to-Companies-Augmented* networks, we computed a range of statistics (see Fig. 1) and plotted degree distributions for each network (See Fig. 2, Fig. 3, and Fig. 4, Fig. 5).

First, we notice that the *Investors-to-Companies* network does not have a true bipartite structure. Specifically, 92 company and investor nodes overlap within the network, meaning 92 entities that received investments also made investments. Second, the *Investors-to-Companies* network has a very low network density and clustering coefficient, which arises from the predominantly bipartite structure. Third, the degree distribution plot for the *Investors-to-Companies* network reveals that a wide-range company and investor types exist (See Fig. 2). 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.

In the *Companies-to-Companies* network, we see a substantial increase in the number of edges. Therefore, many companies have investors in common, which demonstrates the extensive presence of syndicated investments. The increase in edge count corresponds to a proportional increase in the density of the *Companies-to-Companies* network. Third, we see a significant increase in the clustering coefficients for both the *Companies-to-Companies* and *Investors-to-Investors* networks. In these projected networks, the presence of prolific investors collaborating on syndicated investments leads to the creation of highly clustered groups, which increases the average clustering coefficient. We see similar behavior in the *Companies-to-Companies-Augmented* network.

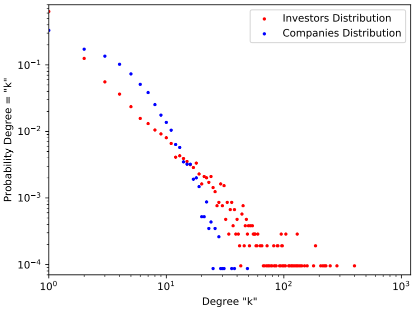


Fig. 2: Degree distributions for *Investors-to-Companies* network.

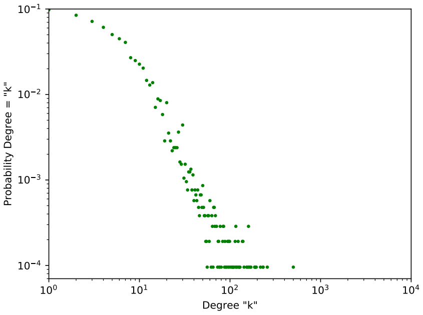


Fig. 3: Degree distributions for *Investors-to-Investors* network.

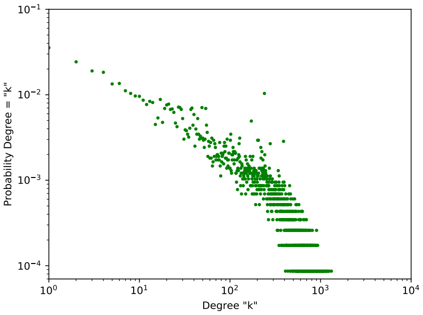


Fig. 4: Degree distributions for *Companies-to-Companies* network.

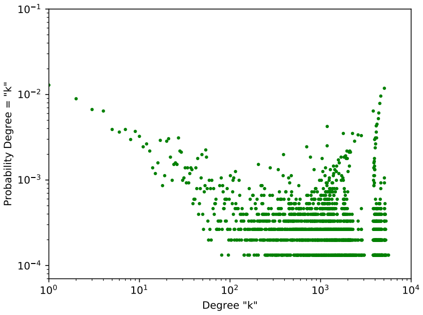


Fig. 5: Degree distribution for *Companies-to-Companies-Augmented* network.

C. Analysis of Degree Distribution

The degree distributions for the *Companies-to-Companies* and *Investors-to-Investors* networks indicate the possibility of a power law relationship. In order to test this hypothesis, we first qualitatively determine values for x_{min} in the power law PDF, which is given by:

$$p(x) = \frac{\alpha - 1}{x_{min}} \left(\frac{x}{x_{min}} \right)^{-\alpha}, \ x \geq x_{min} \tag{9}$$

Looking at the *Investors-to-Investors* distribution, we choose $x_{min} = 20$ as the point at which the power law relationship may begin to come into effect (See Fig. 3). We use the same process to choose $x_{min} = 250$ for the *Companies-to-Companies* network (See Fig. 4).

Now, we find the power law exponent α, using maximum likelihood estimation. Specifically, let n denote our total number of samples. We set $\alpha = \hat{\alpha}_{MLE}$, where $\hat{\alpha}_{MLE}$ is given by:

$$\hat{\alpha}_{MLE} = 1 + n \left[\sum_{i=1}^{n} \left(\frac{d_i}{x_{min}} \right) \right]^{-1} \tag{10}$$

Note that in the equation above, d_i denotes the degree of node i.

After applying the above equation, we find that $\hat{\alpha}_{MLE} = 5.4707$ for the *Companies-to-Companies* network, and $\hat{\alpha}_{MLE} = 2.0285$ for the *Investors-to-Investors* network.

Plotting the resulting exponents against the corresponding complementary cumulative distributions on a log-log scale, we do not find a linear fit, indicating that the degree distributions for the *Companies-to-Companies* and *Investors-to-Investors* networks do not follow power law distributions.

V. Centrality

Centrality measures help indicate the most “important” nodes within a given graph. For instance within the *Companies-to-Companies* network, the most important nodes, depending on the implemented centrality measure, may be companies with the most diverse set of investors or perhaps companies with investments from the “best” investors. Furthermore, in the *Investors-to-Investors* network, the most important nodes may be the most prolific investors or perhaps investors with the highest number of syndicated investments. To measure centrality, we leverage two methods: degree centrality and eigenvector centrality.

A. Degree Centrality

We first employ degree centrality, a simple measure of node centrality that assigns higher centrality scores to nodes with higher degrees. Formally, letting N denote the number of nodes in the graph, the degree centrality of an arbitrary node x is defined as follows:

$$c_{deg}(x) = \frac{deg(x)}{N-1} \tag{11}$$

Applying degree centrality to the *Investors-to-Investors*, we obtain the following top-5 nodes: SV Angel, New Enterprise Associates, Intel Capital, First Round Capital, and Kleiner Perkins Caufield and Byers.

All of these investors are well-known and renowned in the venture capital community. Thus, our results are not surprsing. Interestingly, we note that SV Angel has a substantially higher degree centrality score compared to any other investor (See Fig. 6).

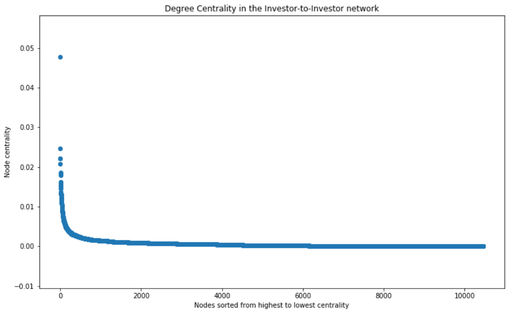


Fig. 6: Degree centrality scores in *Investors-to-Investors*

Applying degree centrality to *Companies-to-Companies*, we obtain the following top-5 nodes: Path, Ark, Dropbox, Twilio, and The Climate Corporation. We note that the the node centrality scores for *Companies-to-Companies* network scores have significantly higher variance than those for the *Investors-to-Investors* network (See Fig. 7).

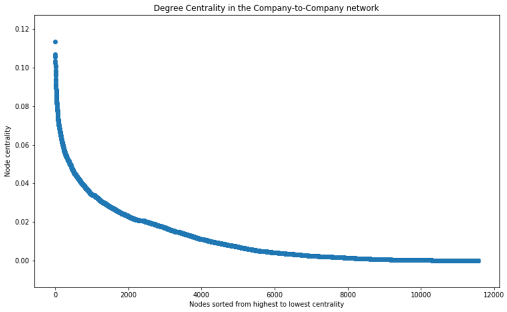


Fig. 7: Degree centrality scores in *Companies-to-Companies*

Finally, applying degree centrality to *Companies-to-Companies-Augmented*, we obtain the following top-5 nodes: Swiftype, Upstart, TrialPay, Kno, and Zaarly. Unlike in the other networks, the degree centrality scores have a demarcated cutoff point at 0.25 (See Fig. 8). Also, it is interesting to note that the incorporation of region and industry in the *Companies-to-Companies-Augmented* network leads to no overlap in the top 5 most central nodes with the *Companies-to-Companies* network.

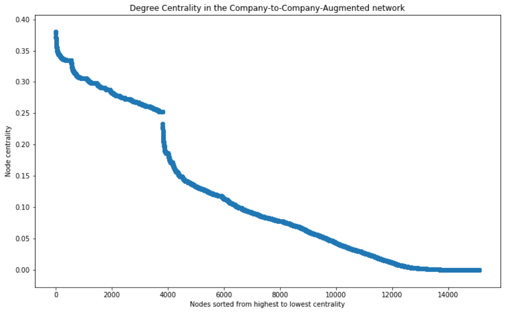


Fig. 8: Degree centrality scores in *Companies-to-Companies-Augmented*

B. Eigenvector Centrality

We next employ eigenvector centrality, a spectral measure of node centrality, where a node's centrality corresponds to the centrality of it's neighbors [8]. More formally, eigenvector centrality measures the influence of a node in a network:

$$c_{eig}(x) = \frac{1}{\lambda} \sum_{y \to x} c_{eig}(y), \tag{12}$$

where c_{eig} converges to the dominant eigenvector of adjacency matrix A, while λ converges to the dominant eigenvalue of A. Eigenvector centrality requires a strongly connected network, but it does not necessitate a directed network, like most other spectral measures.

Applying eigenvector centrality to *Investors-to-Investors*, we obtain the following top-5 nodes: SV Angel, First Round Capital, Andreessen Horowitz, New Enterprise Associates, and Ron Conway. Note that three of the five most central nodes in this case are the same as in the degree centrality case. More so, SV Angel has the highest score in both instances.

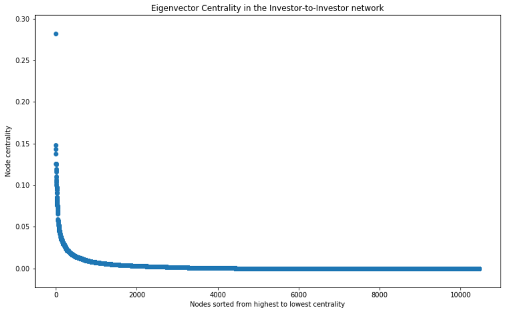


Fig. 9: Eigenvector centralities for nodes in *Investors-to-Investors*

Applying eigenvector centrality to *Companies-to-Companies*, we obtain the following top-5 nodes: Path, IFTTT, The

Climate Corporation, Swiftype, and CrowdMed. Again, the distribution of eigenvector centralities closely match that of the degree centrality scores (See Fig. 10).

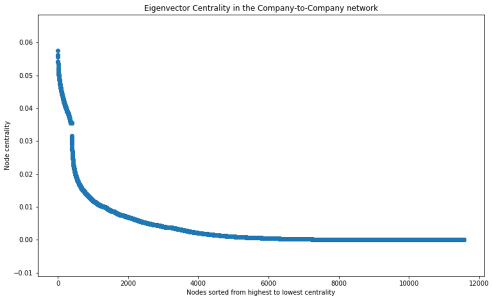


Fig. 10: Eigenvector centralities for nodes in *Companies-to-Companies*

Finally, applying eigenvector centrality to *Companies-to-Companies-Augmented*, we obtain the following top-5 nodes: TrialPay, Swiftype, Kno, Upstart, and Chartbeat. Notably, four of the five most central nodes in this case are the same is in the degree centrality case. It is also interesting to not that the eigenvector centralities create an even more demarcated cutoff when compared to the degree centrality scores (See Fig. 11). As a final note, centrality cutoffs such as this could perhaps be used as a means of detecting communities.

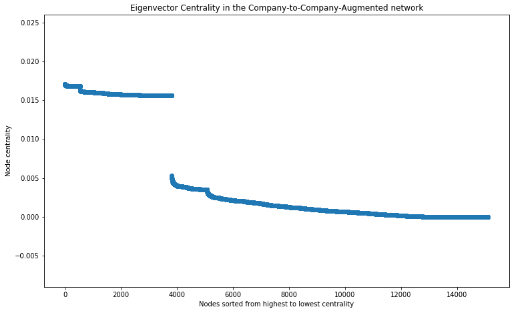


Fig. 11: Eigenvector centralities for nodes in *Companies-to-Companies-Augmented*

VI. Community Detection

Building off of our analysis of node centrality, we explore communities in each of the created networks. Network communities represent sets of nodes with numerous internal connections but few external ones. In order to detect communities, we first leverage the Louvain algorithm. Second, we create latent representations of the nodes in each graph, using node2vec, and then run a variety of clustering algorithms on these embeddings.

A. Louvain Algorithm

The Louvain algorithm iteratively progresses through two pahses: (1) greedily maximize modularity by allowing for changes over local communities and (2) aggregate identified communities to build a new network of communities [2]. We define modularity as the following:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{13}$$

where $2m = \sum_{i,j} A_{ij}$ is the sum of all entries in the adjacency matrix, A_{ij} represents the $(i,j)^{th}$ entry of the adjacency matrix, d_i represents the degree of node i, $\delta(c_i, c_j)$ is 1 when i and j are in the same community ($c_i = c_j$) and 0 otherwise [9]. Note that modularity ranges from [-1, 1].

Running the Louvain algorithm on the *Companies-to-Companies* network results in the formation of **1,405** clusters with an overall modularity of **0.4305**. Upon further evaluation, we recognize that the top-10 clusters, in terms of size, contain approximately 83% of nodes within the network. More so, 97.5% of clusters are of size three or smaller. Therefore, *Companies-to-Companies* has several large, well-defined communities but also many small, non-central communities. This indicates that the algorithm has too limited a picture of the companies landscape to make determinations on many of the companies.

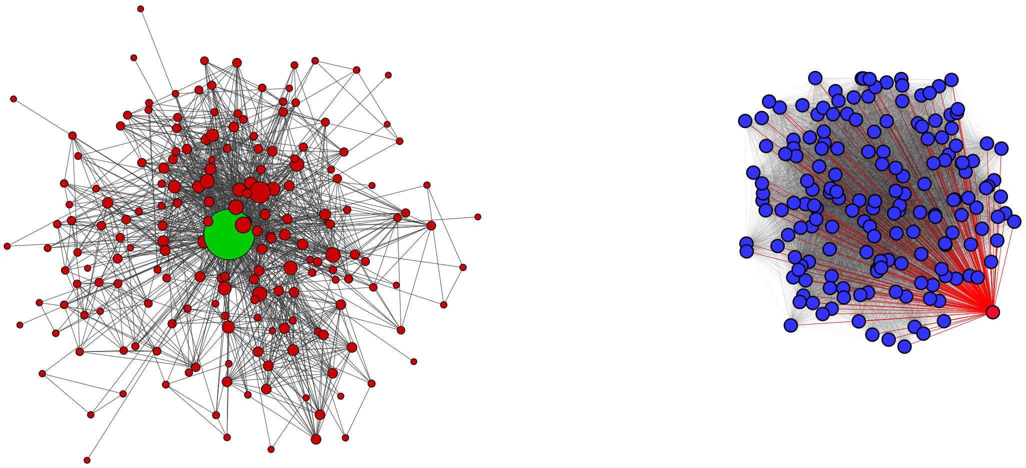
Running the Louvain algorithm on the *Investors-to-investors* network results in the formation of **3,840** clusters with a modularity of **0.64755**. The top-15 clusters, again in terms of size, contain 43.8% of nodes within the network. Similar to the *Companies-to-Companies* network approximately, 96% of clusters are of size three or smaller; however, it is important to note *Investors-to-Investors* has significantly higher modularity.

Realizing that *Companies-to-Companies* and *Investors-to-Investors* omit a substantial amount of information as they simply include an edge between companies (investors) A and B if they share an investor (company) X. In order to incorporate more information from our data sets, we create weighted versions of the *Companies-to-Companies* and *Investors-to-Investors* networks on which we run the Louvain algorithm.

Specifically, we compute the Jaccard Index between all pairs of companies and investors in the original bipartite *Investors-to-Companies* graph. The Jaccard Index is defined as follows:

$$JA(i,j) = \frac{\mid \Gamma_i \cap \Gamma_j \mid}{\mid \Gamma_i \cup \Gamma_j \mid} \tag{14}$$

As such, the weight between two companies C_1 and C_2 is just the number of investors they share, divided by the set of investors that have invested in at least one of C_1 and C_2.



(a) Sequoia Capital Egonet

(b) Reddit Egonet

Fig. 12: (a) and (b) respectively show egonets from the *Investors-to-Investors* and *Companies-to-Companies* networks.

The same applies for investors, as the weight between two investors, I_1 and I_2, is the number of companies they share, divided by the set of companies in which at least one of I_1 or I_2 has invested.

Running the Louvain algorithm with Jaccard weightings produces a much more even distribution of community sizes in both graphs. In the weighted *Companies-to-Companies* graph, the top-10 largest clusters account for approximately 70% of the total nodes, but there are only **152** communities, as opposed to the **1,405** communities found in the unweighted Louvain run. More so, the modularity increases substantially from 0.4305 to 0.5639, equating to a total **increase of 0.1334**.

In the weighted *Investors-to-Investors* graph, the top-10 largest clusters account for approximately 34% of the nodes, but only **700** communities remain, as opposed to the **3,840** communities found in the unweighted Louvain run. Furthermore, the modularity increases significantly from 0.64755 to 0.9533 for a total **increase of 0.3057**.

The increase in modularity for both the *Investors-to-Investors* and *Companies-to-Companies* weighted networks further bolster the validity of our Jaccard weighting system. Intuitively, the edge weights allow the Louvain algorithm to discern between more and less important edges, thereby giving it a more granular picture of the investors and companies landscapes.

B. Node2vec Clustering

Our analysis indicates that Louvain's modularity-optimizing objective performs well separating more "important" companies and investors from less important ones. Nonetheless, Louvain doesn't explicitly capture node-level similarities, which could allow for a more effective means of community detection.

In order to address this, we use node2vec as proposed by Grover et al [6]. Specifically we perform three different runs of node2vec on *Companies-to-Companies*, each time with a different random walk strategy controlled by the node2vec search parameters p and q. For the first run, we perform breadth-first search by setting $p = 1$ and $q = 100$. For the second run we perform depth-first search by setting $p = 1$ and $q = 0.01$. Finally, for the third run, we set $p = q = 1$, thereby using the DeepWalk random walk strategy [15]. Finally, we note that after experiementation, the different random walk strategies correspond to similar results, so our discussion below uses the embeddeings learned from the $p = q = 1$ random walk strategy.

1) Unsupervised Learning: After obtaining the embeddings, namely vectors in $\mathbb{R}^{128}$, for each of the companies, we apply k-means clustering to the embeddings. As a performance metric, we use the Silhouette score, S, defined as:

$$S = \frac{b - a}{max(a, b)} \tag{15}$$

Note that in the equation above, a denotes the mean intra-cluster distance and b the mean nearest-cluster distance. Thus, S ranges from $[-1, 1]$, where 1 denotes the best possible cluster and -1 the worst.

After experimenting with different values of k, we conclude that the the optimal number of cluster k for k-means is **2**, since $k = 2$ clearly achieves the highest Silhouette score (See Fig. 13). This differs drastically from our results from the Louvain algorithm, which, even after adding weights to the network, resulted in optimal numbers of clusters in the hundreds. This make sense because Louvain begins by assigning each node to its own cluster, whereas k-means does not.

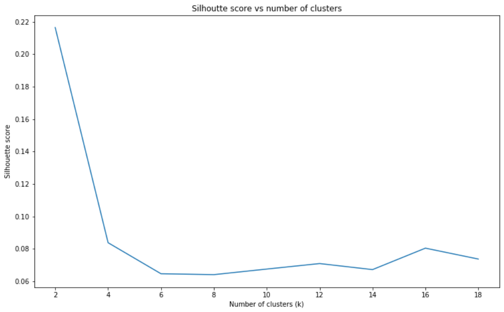
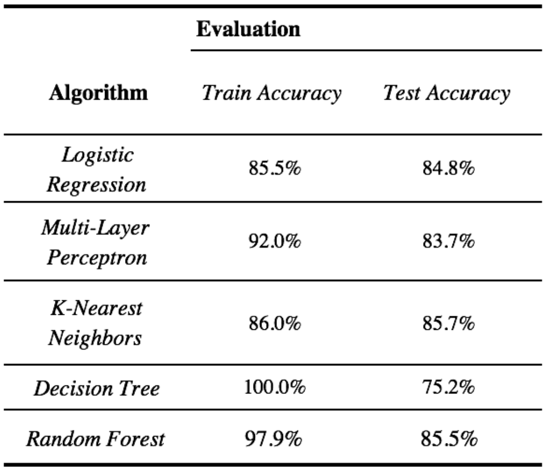


Fig. 13: Silhoutte score vs. k

2) Supervised Learning: As a final experiment, we use the *Acquisitions* data set to perform a supervised experiment designed to assess the quality of the latent representations learned from node2vec. Specifically, for each company in the *Companies-to-Companies*, we assign the company a label of 1 if it was acquired and 0 if it was not. Then, we randomly assign each company to either the train or test set, ending up with approximately 80% of the companies in the train set and 20% in the test set. Now that we have features (i.e. the node2vec embeddings) and labels for all of the companies, along with a train and test set, we apply various supervised learning algorithms, the results of which are detailed below:



	Evaluation	
Algorithm	*Train Accuracy*	*Test Accuracy*
Logistic Regression	85.5%	84.8%
Multi-Layer Perceptron	92.0%	83.7%
K-Nearest Neighbors	86.0%	85.7%
Decision Tree	100.0%	75.2%
Random Forest	97.9%	85.5%

Fig. 14: Discuss Performance

From these results, we conclude that K-Nearest Neighbors, closely followed by Random Forest, performs the best on the given data. As a caveat, we note that many of the companies we labeled as not acquired have likely been acquired since Crunchbase released this data in 2013. Nonetheless, it is fascinating that we are able to obtain such high accuracy using latent representations of the companies. We thus conclude that the acquired companies share some distinguishing characteristics captured by node2vec.

VII. Network Deconvolution

Throughout our analysis, we leverage the *Companies-to-Companies-Augmented* graph, since it incorporates information about each company's investors, industry, and region. However, recall that the *Companies-to-Companies-Augmented* network has $13,504,003$ edges, as opposed to the $768,063$ edges present in the simpler *Companies-to-Companies* network. Since both networks contain the same nodes (i.e. the set of all companies we are considering), it seems highly likely that the *Companies-to-Companies-Augmented* network contains a great deal of spurious edges carrying only indirect information about company relationships.

In order to address this issue and extract direct relationships between companies, we employ network deconvolution. Formally, we let G_{obs} denote the adjacency matrix of the observed *Companies-to-Companies-Augmented* network, and we let G_{dir} denote the adjacency matrix of the "true" network, which we seek to extract from G_{obs}. Next, we model G_{obs} as follows:

$$G_{obs} = \sum_{k=1}^{\infty} G_{dir}^{k} = G_{dir}(I - G_{dir})^{-1} \tag{16}$$

Thus, in order to extract G_{dir}, we consider:

$$G_{dir} = G_{obs}(I + G_{obs})^{-1} \tag{17}$$

In order to implement the above equation, we use Gideon Rosenthal's publicly available implementation of the Network-Deconvolution algorithm originally proposed by Soheil Feizi [14]. We use the unweighted adjacency matrix of the *Companies-to-Companies-Augmented* network as input to the Network-Deconvolution algorithm, which outputs a weighted adjacency matrix with the same number of edges. Crucially, the edge weights in the adjacency matrix output by the deconvolution algorithm are all in the range $[0, 1]$, where higher edge weights correspond to edges of more direct importance, and lower edge weights correspond to indirect edges.

Therefore, we expect for there to be a considerable number of edges with low weights after applying network deconvolution to the *Companies-to-Companies-Augmented* network, since as stated it seems highly likely that there are many indirect relationships in the network. After applying the network-deconvolution algorithm, we obtain the edge weights depicted below:
Figure 15 leads us to conclude that there is a clear cutoff at approximately $w = 0.7$, where w denotes edge weight. The sharp cutoff indicates that there is indeed a considerable amount of redundant information. Thus, in order to reduce the amount of indirect information in the *Companies-to-Companies-Augmented* network, we remove all edges below the $w = 0.7$ weight threshold. Removing these edges ideally

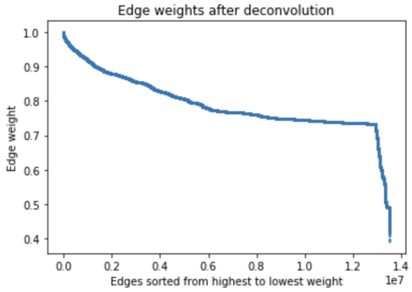


Fig. 15: Edge weights for *Companies-to-Companies-Augmented* after deconvolution

gives us a more "direct" network that compares to the original network as follows:

Compared to the original network, the new *Companies-to-Companies-Augmented* network has 513,694 fewer edges. Furthermore, we see a corresponding increase in network density and a decrease in the average clustering coefficient. We also see that the degree distribution remains relatively unchanged (Compare Fig. 5 to Fig. 17). Thus, the $w = 0.7$ weight threshold may not have been large enough to effectively remove indirect edges.

	Networks	
Metrics	*Companies-to-Companies-Augmented*	*Companies-to-Companies-Augmented* ***with Removed Edges***
Edges	13,504,003	12,990,309
Density	0.1182	0.1239
Effective Diameter	2.0841	2.0772
Clustering Coefficient	0.6762	0.6496

Fig. 16: Metrics for the *Companies-to-Companies-Augmented* network and *Companies-to-Companies-Augmented* with edges removed after deconvolution.

VIII. Conclusions

Recognizing the distinct network structure of the start-up investment ecosystem, we set out to create useful network representations of the Crunchbase 2013 data sets. Similar analysis had been performed on investor networks, but our focus on early-stage start-ups was unique.

We began by folding the *Investors-to-Companies* network in order to analyze relationships among both start-ups and investors. Then we assessed centrality in the folded networks, honing in on early-stage start-ups, which we analyzed via both the simple *Companies-to-Companies* network and the more nuanced *Companies-to-Companies-Augmented* network. Finding similar central nodes with different centrality measures, along with interesting centrality cutoffs, we decided to explore the community structure of the networks. We first employed the Louvain algorithm to detect communities in the *Companies-to-Companies* and *Investors-to-Investors* networks, and ultimately found that the algorithm was significantly more successful when Jaccard weightings were included. Then we applied node2vec to find embeddings for companies in *Companies-to-Companies*, which led to suprisingly meaningful results, especially with the use of the embeddings as inputs to out-of-the-box supervised learning algorithms.

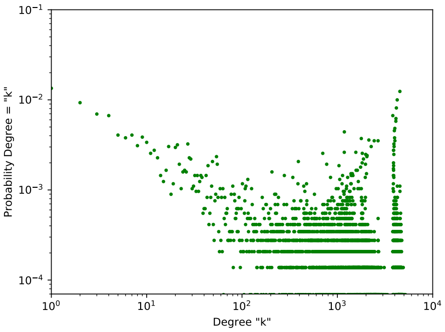


Fig. 17: Degree distribution for the deconvolved *Companies-to-Companies-Augmented* graph with removed edges.

Finally, recognizing that the *Companies-to-Companies-Augmented* network likely contained a great deal of indirect information, we applied network deconvolution to extract direct relationships among companies. We did not find an enormous reduction in the number of edges after applying a cutoff to the weights returned by the deconvolution algorithm, but we were nonetheless able to reduce the size of the network substantially.

Finally, we conclude that further work could be performed in analyzing the results of the deconvolution and how best to apply the resulting weightings to reduce the complexity of the *Companies-to-Companies-Augmented* network. Additionally, we emphasize that the use of embeddings showed incredibly promising results, even when only applied to the simple *Companies-to-Companies* network. As such, we believe that the application of algorithms capable of learning latent representations to larger, more intricate networks is a promising avenue of exploration.

References

[1] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks.In *WSDM*, 2011.

[2] Blondel, Vincent D, et al. Fast Unfolding of Communities in Large Networks. Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 10, Sept. 2008, doi:10.1088/1742-5468/2008/10/p10008.

[3] Bygrave, William D. Syndicated Investments by Venture Capital Firms: A Networking Perspective. *Journal of Business Venturing,* vol. 2, no. 2, 1987, pp. 139154., doi:10.1016/0883-9026(87)90004-8.

[4] Bygrave, William D. The Structure of the Investment Networks of Venture Capital Firms. Journal of Business Venturing, vol. 3, no. 2, 1988, pp. 137157., doi:10.1016/0883-9026(88)90023-7.

[5] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75174, 2010.

[6] Grover, Aditya, and Jure Leskovec. node2vec. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD* 16, 2016, doi:10.1145/2939672.2939754.

[7] Hamilton, W.L., Ying, R., Leskovec, J. (2017). Representation Learning on Graphs: Methods and Applications. IEEE Data Eng. Bull., 40, 52-74.

[8] Leskovec, Jure, and Baharan Mirzasoleiman. Network Centrality. *CS224W: Social and Information Network Analysis.* 15 Nov. 2018, Stanford, Stanford University.

[9] Leskovec, Jure, and Baharan Mirzasoleiman. Community Structure in Networks. *CS224W: Analysis of Networks.* 11 Oct. 2018, Stanford, Stanford University.

[10] Ljungqvist, Alexander, et al. Whom You Know Matters: Venture Capital Networks and Investment Performance. SSRN Electronic Journal, 2005, doi:10.2139/ssrn.631941.

[11] Podolny, Joel M. 1993. A Status-Based Model of Market Competition. *American Journal of Sociology* 98:82972.

[12] Sorenson, Olav, and Toby E. Stuart. Syndication Networks and the Spatial Distribution of Venture Capital Investments. *SSRN Electronic Journal*, 2000, doi:10.2139/ssrn.220451.

[13] Kogut, Bruce, et al. Emergent Properties of a New Financial Market: American Venture Capital Syndication, 19602005. *Management Science,* vol. 53, no. 7, 2007, pp. 11811198., doi:10.1287/mnsc.1060.0620.

[14] Rosenthal, Gideon. MIT Kellis Lab. https://github.com/gidonro/Network-Deconvolution.

[15] Perozzi, Bryan et al. ”DeepWalk: Online Learning of Social Representations.” Stony Brook University, 2014.