CS 224W Final Report
Group 37

Aaron B. Adcock [*] Milinda Lakkam [†] Justin Meyer [‡]

1 Introduction

Much of the current research is being done on social networks, where the cost of an edge is almost nothing; the click of a button in most cases. However, in a graph where each node is an investor and/or investee and each edge is an investment in a company, each edge costs thousands, millions, or potentially billions of dollars. On top of that, each edge is carefully considered, calculated, and subject to a large set of laws, regulation, prior affiliations, and so forth. And most importantly, each edge is intended to ultimately make money for the investor.

Investment networks also have at least two primary node types: investors and investees. Edges would mostly occur between investors and investees (see below for a description of why this is occasionally not true), leading to a nearly bipartite structure which can complicate the network structure for certain tasks. It is possible to separate nodes even further by looking at the kinds of investors that occur in the graph.

In this work, we look at the CrunchBase investment network and compare this network to previously studied social/information networks, investigate the structure induced by the variations in node types, and then attempt the difficult task of investment (link) prediction on the network.

2 Description of Data

The dataset we are using comes from the website CrunchBase [1], a “free database of technology companies, people, and investors that anyone can edit”. We construct a network from the website using companies as nodes and investments as edges. There is a large amount of meta-data available on CrunchBase, some of which we use in the analysis within this paper. The database goes back to 1987 and includes many companies which have not been invested in, companies which we ignore for our analysis.

In this section, we look at some of the basic graph statistics of the CrunchBase network, look at some of the similarities between the CrunchBase network and other common social and information networks, and point out some unique features of this network.

2.1 Statistics

Statistics on the CrunchBase network, along with some statistics of other commonly studied information networks, which are of a similar size to the CrunchBase network, are presented in this section. Note that we often restrict ourselves to the largest connected component of the networks when computing statistics, which gives a few slight irregularities in the data (such as the G_{nm} network having slightly less nodes and edges than the CrunchBase network).

The statistic that immediately stands out is the low clustering coefficient of the CrunchBase network. It is on order with the preferential attachment model and the Gnutella peer-to-peer network available on the SNAP website [3] . Though this may at first seem confusing, a second look at the data shows that this is due to the presence of the node types mentioned in the introduction. Specifically, nodes

[*] Department of Electrical Engineering, Stanford University, Stanford, CA 94305, aadcock@stanford.edu.

[†] Institute for Computational Mathematics and Engineering, Stanford University, Stanford, CA 94305, milindal@stanford.edu.

[‡] Department of Computer Science, Stanford University, Stanford, CA 94305, jlmeyer@stanford.edu.

Table 1: Network Statistics

Network	$\|V\|$	$\|E\|$	$\bar{d}$	$\bar{C}$	D
CrunchBase	20515	39435	3.8445	1.9934×10^{-3}	22
G_{nm}	20034	39423	3.936	2.183×10^{-4}	16
Watts-Strogatz	20515	41030	4	0.37394	27
Preferential Attachment	20515	41026	3.9996	2.7657×10^{-3}	9
CA-AstroPh	17903	196972	22.004	0.6686	14
Cit-HepTh	27400	352021	25.695	0.32864	15
p2p-Gnutella25	22663	54693	4.82663	8.9937×10^{-3}	11
as-caida20071105	26475	53381	4.0326	0.33335	17

can be broadly classified as ‘investors’ or ‘investees’. Since investors are rarely invested in, this makes an almost bipartite network. The few nodes which are classified as both investor and investee are tech companies which are large enough to make investments in startups (two examples would be Cisco and Microsoft). As they were once a startup, they initially received investments, now as a major player in the tech market they make investments and acquisitions.

The bipartite nature of this network leads to a low average clustering coefficient for the network. But, if we flatten the network by restricting ourselves to only investors (with edges indicating a common investment) the clustering coefficient is .67407, similar to previously studied collaboration networks. Similar results are found when we flatten the network to include only investees. This method of ‘flattening’ the network produces networks is similar to the process behind collaboration networks. In a collaboration network, two authors collaborating on a paper produces an edge in the network. If the papers were included in the network, we would have a bipartite paper-author network. Addtionally, the edges (collaborations) do represent a greater cost to create when compared to ‘adding friends’ in a social network making these papers better for comparison with the CrunchBase network. Even so, there is at least one very important difference between the two networks. For one, investees receive additional investors over time, allowing us to predict future investors in companies. With the collaboration networks, even if they were augmented with the specific papers that authors collaborate on, papers have a static author list removing any possibility of predicting future edges to existing papers.

We also looked at some other basic network properties, such as the degree distribution of the network and the network community profile (NCP) plot. Both of these measures showed that the CrunchBase network is very similar to other real networks. For example, Figure 1 demonstrates that the community structure of the CrunchBase network is very similar to existing networks [4]. i.e. there are no well defined communities above a certain size scale, a phenomenon described in great detail by the authors of [4].

2.2 Node types

The analysis of the previous section implies that our analysis will need to take into account the node types present in the network. Broadly, the investors can be classified as financial organizations, tech companies, and individual investors. As each of these entities have different mindsets and different resources available to them, it seems reasonable that they may behave differently within the network. To investigate this, we took the bipartite investment network and flattened it into an investor only network. This network has only investors as nodes and common investments as edges. We can then look at the

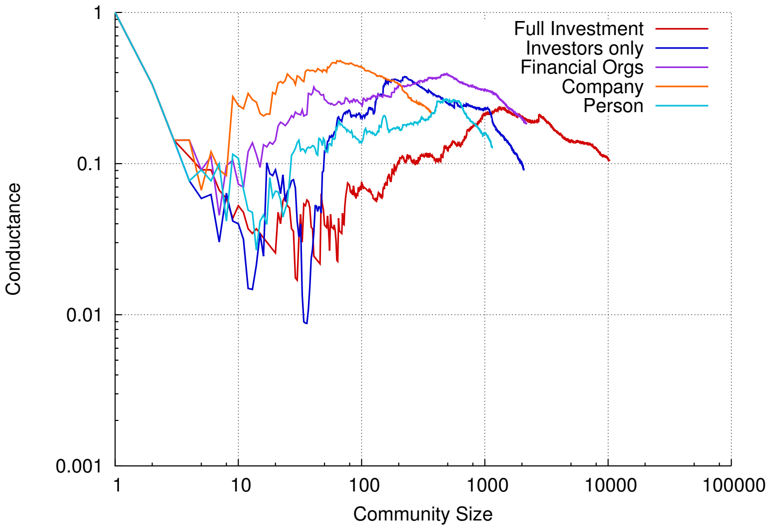


Figure 1: NCP plots of CrunchBase networks

edges among each node type. The results are shown below.

Table 2: Node Type Statistics

Network	$\lvert V\rvert$	$\lvert E\rvert$	$\bar{d}$	$\bar{C}$
All-investor	9999	83145	16.63	.67407
Personal	3363	16510	9.8186	.73531
VC Firm	4792	30770	12.8422	.54872
Tech Company	1005	2926	5.8229	.43342

The clustering coefficients between the different types of investors are especially interesting. The personal investors have the highest average clustering coefficient, indicating that they cluster the most. This suggests that 'nearness' between two personal investors in the investor network is more meaningful than with VC firms or tech companies. There are several possible explanations for this, including that personal investors are more likely to rely on their interactions with other private investors to inform future investments. It is also possible, that private investors could, on average, have less money available to invest leading to a larger number of them being involved in any given investment. We also present a brief table of link prediction results on the flattened network, tabulated by investor type. These results were obtained using the algorithms described below.

The tech companies have the lowest average clustering coefficient, despite being the smallest network, suggesting that they are less likely to cluster together. This could be due to increased competitiveness among companies, which would preclude collaborative investments. It could also be indicative of the fact that many companies choose to acquire small firms they are interested in rather than invest. A second look at the data shows that companies that have venture capital arms are companies likely to make investments (think Google Ventures) whereas companies like Twitter (which do not have separate

Table 3: Link Prediction On Each Investor Type Assuming Investors Are Known

Investor Type	Recall (%)	Precision (%)	Prediction Set Size
All (179)	26.46	0.23	116336
Financial Organizations (118)	18.80	0.21	58641
Companies (45)	12.65	0.23	13321
Persons (16)	6.90	0.44	1805

investment arms listed in CrunchBase) are more likely to make acquisitions later in a company's life.

There are less variations among the investees. Because the CrunchBase website keeps track of only tech companies, with an emphasis on startups, these companies are largely homogeneous. The classifications that we do look at, we use to identify unlikely investments. For example, if a company has been acquired, it is unlikely to receive any more funding. If a company has received all of the typical late stage investment rounds, it is unlikely to need additional investors. These companies are filtered out for our investment prediction algorithms.

3 Relevant Work

There are several earlier papers that use CrunchBase as their test dataset, but most are using machine learning algorithms and semantic features to rank companies and make predictions. Of these papers, only one makes use of the investor network to rank VC investment firms. This work was originally done by Chris Farmer, who did not release his algorithm, only the top ten results. He claims that his idea is analogous to PageRank in the following sense. Each firm is given a ranking. When two firms invest in the same company the firm that invested first then receives ranking from the firm which invest later. Bhat & Sims attempted to replicate this work with their own PageRank based algorithm [2]. We use this ranking to look at our ability to predict the investments of the top ten firms.

For the link prediction algorithms, our work is based on the review paper by Liben-Nowell and Kleinberg [5]. We modified these algorithms to work on a bipartite network.

4 Algorithms

We modified the weighted ranking algorithms presented in [5] and weighted rooted PageRank to apply them to the near-bipartite investment graph. We also use rule-based filtering on top of these algorithms to restrict our prediction set to edges that connect an investor with several recent investments to a company that has recently received early-stage funding and has not yet been acquired, gone public, or gone out of business. We also restrict our predicted edges to be between an investor and an investee of a friend of that investor in the flattened investor collaboration network described in section 2.1. The weight our algorithms use is one of: Common Friends, Jaccard Distance, Adamic-Adar, or Preferential Attachment weighting methods. Definitions are omitted here for brevity but can be found in [5].

4.1 Weighted Ranking

To compute a score for each candidate edge: for each source node s we predict an edge from s to each company c that an investor friend of s has invested in using equation 1, where $weight_s(f)$ is the weight of the edge (s, f) in the investor-only collaboration network and $pathnode_{source}(target)$ is the set of investor friends of 'source' that have invested in the company 'target'.

$$score(\,(s,t)\,) = \sum_{f:pathnode_s(t)} weight_s(f) \tag{1}$$

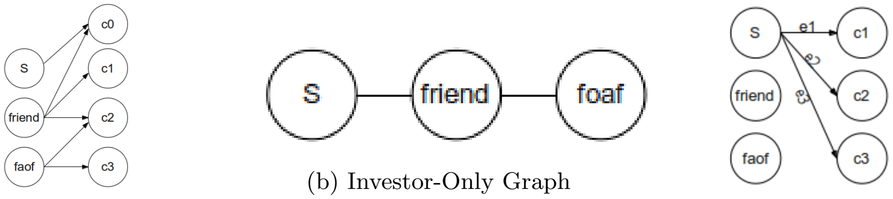
The full candidate set is built by running weighted ranking once for each source node with it as the root. All unlikely edges are filtered out and all likely edges are ranked by score. The prediction set is then all edges with a score greater than some cutoff value. For this algorithm we found the best cutoff value to be the mean normalized score.

4.2 Weighted Rooted PageRank

We ran weighted rooted PageRank ($\beta = 0.85$) once for each source node as the root. The edge weights are given by the weighting methods described above. Then, for each source node, we computed the score of an edge from the source node to each investment (company) of the source node's investor friends and foafs. The score of each of these edges (s, t) is computed using equation 2, where $wrpr_s(node) =$ weighted rooted PageRank score of node, with root s.

$$score(\,(s,t)\,) = \sum_{f:pathnode_s(t)} wrpr_s(f) \tag{2}$$

As an illustration of this computation: Figure 2a shows an investment graph where 3 investors (s, friend, and foaf) invest in 4 companies (c0, c1, c2, c3). Figure 2b shows an investor-only graph where 'friend' is an investor friend of 's' and 'foaf' is a friend of a friend of 's'. One investor node is then selected as the source node and rooted PageRank is run on the investor-only graph. The nodes 'friend' and 'foaf' are then assigned a PageRank score. The candidate edge set is then created by predicting an edge from 's' to each of s's friends' and foafs' investments that 's' has not already invested in (e1, e2, e3) as shown in Figure 2c. Using equation 2: $score(e1) = wrpr_s(friend)$, $score(e2) = wrpr_s(friend) + wrpr_s(foaf)$, and $score(e3) = wrpr_s(foaf)$.



(a) Investment Graph

(b) Investor-Only Graph

(c) Scored Candidate Set

The full candidate set is built after rooted PageRank has been run once for each source node as the root. All unlikely edges are filtered out with our rules and all remaining edges are ranked by score. The prediction set is then all edges with a score greater than some cutoff value. We used the mean normalized edge score as the cutoff value.

5 Result & Analysis

Table 4: Performance of Link Prediction Methods

Algorithm	Precision (%)	Recall (%)	Prediction Set Size
Baselines			
Predict all possible edges	-	100	198528100
Random Baseline	6.50×10^{-4}	1.27	2000000
Node Filtered Random Baseline	1.33×10^{-2}	3.89	300000
Filters			
Rule-Based Node Filtering	7.27×10^{-3}	16.47	10219545
Friend Investment Edge Filtering	6.83×10^{-2}	8.23	543253
Foaf Investment Edge Filtering	1.65×10^{-2}	13.88	3784011
Ranking Algorithms			
Weighted Ranking: Adamic-Adar	3.53×10^{-2}	2.41	161670
Weighted Ranking: Preferential Attachment	3.68×10^{-2}	2.63	168679
Weighted Ranking: Common Friends	5.39×10^{-2}	4.41	192835
Weighted Ranking: Jaccard Distance	9.64×10^{-2}	6.95	170068
Unweighted Rooted PageRank	5.44×10^{-2}	4.45	192907
Rooted PageRank: Adamic-Adar	5.58×10^{-2}	4.15	175646
Rooted PageRank: Preferential Attachment	5.88×10^{-2}	4.49	180275
Rooted PageRank: Common Friends	7.10×10^{-2}	5.42	180159
Rooted PageRank: Jaccard Distance	9.75×10^{-2}	6.10	147726

Our prediction algorithm ranks all candidate edges and then predicts the highest ranked edges. The set of predicted edges can be all edges with a score above a specific cutoff off score or the top N ranked candidate edges for some specific value of N. A good ranking algorithm cuts out a larger percentage of incorrect than correct from the candidate set. To evaluate quality of a ranking method we plot the CCDF of the correct and incorrect predictions, where the x-axis is the normalized edge score and the y-axis is the fraction of edges with score >= x.

The Jaccard ranking beats Adamic-Adar and the other ranking algorithms by a significant amount. This makes intuitive sense, because it measures how much two investors have collaborated in the past, relative to how much they invest overall. Adamic-Adar, in contrast, weights common investors with a low number of investments highly. In contrast to social networks, investors with low numbers of investments are worse predictors of an investments success.

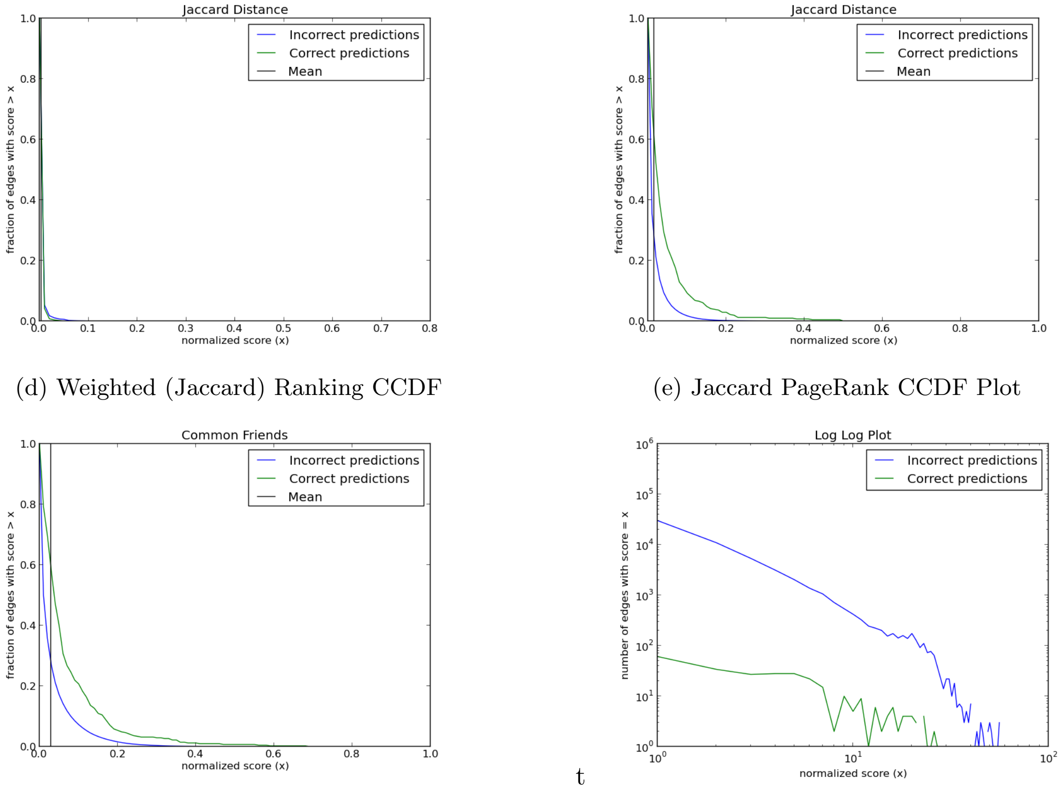
Table 4 shows results for predicting edges without knowing which investors will want to invest and which companies will want funding during the test period. We further evaluate the weighted rooted PageRank algorithm by assuming we know exactly which investors will invest during the test period. This is a reasonable experiment because in the real-world people may know which investors are actively looking for investments and which are not. Results obtained by assuming we know active investors beforehand are shown in Table 5.

Table 5: Comparison Of Link Prediction Algorithms Assuming Active Investors Are Known

Algorithm	Precision (%)	Recall (%)	Prediction Set Size	Cutoff
Weighted Ranking: Jaccard	1.12×10^{-1}	2.67	56071	mean
Unweighted Rooted PageRank	1.78×10^{-1}	1.18	29834	mean + std. dev.
Jaccard Rooted PageRank	3.19×10^{-1}	3.26	24139	mean + std. dev.

5.1 Relative Performance of Algorithms

The prediction set is all edges with a score higher than some cutoff value x and Jaccard Distance weighting achieves the largest $ccdf_x(correct)/ccdf_x(incorrect)$ ratio. Figures 2e and 2f compare the CCDF of two weighting methods. The plots show that the Jaccard Distance weighting is slightly steeper than the common friend weighting, while still separated enough such that a vertical line at the cutoff value x cuts out a larger ratio of incorrect predictions to correct predictions. Figures 2d, 2e show CCDF plots of the weighted Jaccard ranking and weighted rooted Jaccard PageRank algorithms respectively. These were chosen because they gave the best results.



(d) Weighted (Jaccard) Ranking CCDF

(e) Jaccard PageRank CCDF Plot

t

(f) Common Friends PageRank CCDF Plot

(g) Log Log Jaccard CCDF Plot

5.2 Why Precision Is Difficult

The algorithms we tried all achieved relatively low precision compared to some social networks because this network has a much lower edge count. When compared to random baseline, we do approximately 30 times better, comparable to the improvement over baseline in collaboration networks [5]. This can be seen in Figure 2g which shows the number of incorrect candidates dominates the number of correct candidates by several orders of magnitude. Also, the standard link prediction methods were developed for use in unipartite graphs. Our modified algorithms used an intermediate investor-only collaboration network which may have lowered the performance of the algorithms.

Our algorithm can pick out companies highly likely to receive funding, one example being a startup called Path, and predict that several top VCs will invest in it. However, not every VC can participate in a single funding round. So, despite a company and investor being a good match, the investment still may not occur, leading to incorrect predictions. Also, some companies our algorithm predicts to be desirable to investors get acquired rather than funded, leading to incorrect predictions.

Lastly, we note that our algorithm did not consider repeated investments between the same investor and investee. However, it turned out that a significant portion of the test edges were repeated edges. Including repeated edges in our predictions would have greatly improved both precision and recall.

6 Conclusion

We have shown that in several network statistics, the CrunchBase network is similar to previously studied social/information networks. When we flatten the network to investors only, the network statistics are very similar to those of collaboration networks. The network also has interesting structures when it is looked at in terms of personal investors, VC firms, and industrial investors in the flattened network.

However, it is the differences between this network and previously studied networks that make it difficult to predict investments. Previously used link-prediction methods must be modified to work on a nearly bipartite network. The low average degree of the full investment network also makes it difficult to find a prediction set with a reasonable precision. A combination of modified Jaccard weights (from [5]) and rooted PageRank (also from [5]) gives the best results. Future work would involve adding a machine learning component to the network analysis and a more sophisticated handling of the various investor types and other network features, such as repeated investments in the network.

References

[1] Crunchbase. `http://www.crunchbase.com`, Oct 2012.

[2] H.S. Bhat and B. Sims. Investorrank and an inverse problem for pagerank. *Under Review*, 2012.

[3] Leskovec. Stanford network analysis platform. `http://snap.stanford.edu`, Dec 2012.

[4] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[5] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.