Link Prediction in Bipartite Venture Capital Investment Networks

Charles Zhang
Statistics
Stanford University
cyzhang@stanford.edu

Ethan Chan
Computer Science
Stanford University
ethancys@cs.stanford.edu

Adam Abdulhamid
Computer Science
Stanford University
adama94@stanford.edu

Abstract

Our project tests link prediction in a bipartite graph in the context of identifying new investment opportunities. The bipartite graph is a representation of observed investments in the technology start-up world where edges represent specific investments. We attempt to predict future investments using simple preferential attachment scores and an augmented score that makes use of unsupervised learning to encode coarse assortative matching in our model (Segmented Preferential Attachment). We then evaluate the performance of these methods against held-out data and find some evidence for predictive power but low practical significance.

1. Introduction

Identifying investment opportunities is a long-standing research area in many fields, including economics and finance, and has obvious practical implications. Traditional approaches often focus on statistical methodologies that try to understand the relationship between underlying characteristics of investment opportunities and observed investment returns.

There is a dramatically lower amount of observable characteristics and outcomes, however, in the venture capital funding (VCF) context and this relative scarcity of data makes traditional approaches less suitable. On the other hand, network analysis approaches remain competitive in VCF because while there is a relative scarcity of observable characteristics and outcomes, there is still an incredibly rich set of structure in the data that could be exploited for prediction.

In this paper, we suggest a novel network analysis approach to understanding investment decisions. We model investments as edges in a bipartite network of investor and start-up nodes and we re-frame our problem of identifying investment opportunities as the problem of predicting new links in a bipartite graph. These problems are distinct but closely related problems and understanding the network structure of this simplified investment network evolves gives us insight both into venture capital funding networks and the larger problem of identifying investment opportunities.

In Section 2, we review previous literature about network link prediction in bipartite graphs. In Section 3, we discuss our dataset, our modelling choices and our prediction methodologies. In section 4, we present prediction results and finally, in section 5 we discuss our results and potential further research.

2. Review of Previous Work

2.1. Investment networks

Past research on investment networks [5] [9] done on venture capital investment networks have revealed insights about the macro level of the investment networks, but none have come up with a model for predicting future investments.

2.2. Proximity based link prediction

Past link prediction methods on graphs have utilized the assumptions of triadic closure [4], and made use of the observation that nodes tend to form well-connected clusters in the graph[7]. A popular measure by Adamic and Adar[1]:

$$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log \Gamma(z)} \tag{1}$$

refines the simple counting of common features by weighting rarer features more heavily[8]. However, these measures are not applicable in bipartite graphs as there are no common neighbors between the two types of nodes.

2.3. Preferential attachment

Preferential attachment [3] is a good base for link prediction in bipartite graphs that gives a prediction between nodes u and v through the score:

$$\frac{d(u)d(v)}{2|E|} \tag{2}$$

derived from the probability of a link forming between two nodes proportional to the degrees.

2.4. Algebraic link prediction

Algebraic link prediction algorithms [7] are based on the eigenvalue decomposition of its adjacency matrix A:

$$A = U\Lambda U^T \tag{3}$$

A spectral transformation is then usually applied:

$$F(A) = UF(\Lambda)U^T \tag{4}$$

where $F(\Lambda)$ applies a real function to each eigenvalue λ_i. This strategy exploits the fact that the matrix power A^i gives for each pair of nodes u and v the number of paths of length i between u and v. Unfortunately, algebraic link prediction did not predict noticeably better than the preferential attachment model in many real-world bipartite networks. [7]

2.5. Supervised Random Walks Linked Prediction

The supervised random walk approach combines two concepts, viewing link prediction as a classification task and as a task to rank nodes. Link prediction as a classification task where we take pairs of nodes to which s has created edges as positive training examples and all other nodes as negative training examples using node and edge attributes as features. Secondly, viewing link prediction as a task to rank the nodes of the network that will assign higher scores to nodes which s created links to than to those that s did not link to. This approach considers both node and edge features as well as the structure of the network [2].

2.6. Assortative Matching

People are known to take into account cultural and economic factors when people were choosing partners for marriage[6]. This idea could be loosely paralleled in the world of investment networks, where large Venture Capital firms only invest in already large companies, and smaller investors in small companies, where status of companies and investors can be be quantified by the total number and amount of investments made and received so far.

3. Methodology

3.1. Data Collection

We use data from Crunchbase, an aggregator of investment news from venture capital firms, founders and startups. We focused on the Crunchbase Business Graph, which has data on relationships and interactions that occurred between $\sim$ 280k unique persons, $\sim$ 300k unique organizations, $\sim$ 150k investment rounds and $\sim$ 16k acquisitions. This data includes information about the timing and size of investments, founder and employee characteristics and work history and some rough categorization of industries and investments.

3.2. Modeling

As we described earlier, we model this network as a bipartite graph between ”investors” and ”companies”, where investments are directed edges from investors to companies. This model differs from a more commonly seen collaboration network where the nodes (in this context) would be investors and edges would be drawn between any investors who had both invested in the same ”start up”. We chose our representation over the collaboration representation because one of the key behaviors that we are interested in explicitly modelling is investments. This network model also allows for intuitive extensions where investor, company and investment characteristics can be immediately embedded as node and edge attributes.

We subset the Crunchbase data by taking advantage of the timestamps on each edge to create a network that includes all nodes that received or made an investment between 2005-2015. Due to the nature of the data collection, this preserved $> 90\%$ of the Crunchbase data on investments. To make this problem more tractable, we further restricted our analysis to the maximum weakly connected component of the graph (as observed in 2015) which consisted of $\sim$ 105k edges and $\sim$ 55k nodes (the next largest WCC has 20 nodes). In our network, there are $\sim$ 21k investors and $\sim$ 34k companies.

While the majority of our data adhered to the binary division between investor and startups, ~ 400 nodes had both received and given investments over our 10-year observation period. To enforce the bipartite property, we assigned entities with strictly higher out-degree to the investor group (and removed their incoming investment edges) and the other companies to the start-up group (and removed their outgoing investment edges). This simple heuristic preserves the maximum number of edges and, by manual inspection, seems to perform well in distinguishing the primary function of an entity.

Finally, we also created snapshots of the graph to capture the actual observed growth of edges in the graph. We created a set of graphs that capture the state of the graph at three month intervals. This method allows us to create an evaluation procedure that allows us to compare a prediction method (on an earlier snapshot) against the actual observed growth (edges that will appear on a future snapshot).

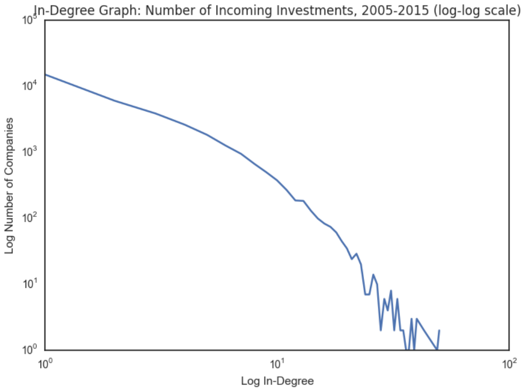

Figure 1. In-Degree Distribution (Log-Log)

3.3. Network Statistics

Investors	Degree Centrality
y-combinator	746
500-startups	698
start-up-chile	690
sv-angel	532
sequoia-capital	524
intel-capital	501

Table 1. Degree Centrality - Investors

Companies	Degree Centrality
ecomom	50
uber	50
flexport	49
opendoor-2	43
careguide	43

Table 2. Degree Centrality - Companies

Investors	Weighted Degree Centrality
goldman-sachs	6351100008
alibaba	5252092061
sequoia-capital	4419774196
warburg-pincus	3853103460
kpcb	3184299724

Table 3. Weighted Degree Centrality - Investors

Companies	Weighted Degree Centrality
uber	6759213252
didi-dache	2852999995
facebook	2165199999
airbnb	2010419995
flipkart	1895428568
clearwire	1706000000

Table 4. Weighted Degree Centrality - Companies

Tables 1-4 present top investors and companies by degree centrality and weighted degree centrality.

In Figure 1, we see that the in-degree distribution for companies (i.e., the degree distribution of nodes in the company partition of the network) shows strong power-law behavior.

In Figure 2, we see similar behavior in the out-degree distribution of investors. This suggests that a preferential attachment model may have some empirical justification.

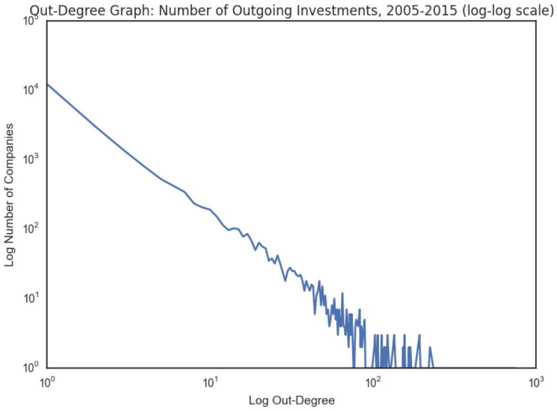

Figure 2. Outdegree Distribution (Log-Log)

In Figure 3, we encoded the size of each investment as an edge weight and used kernel density estimation to estimate the distribution of in-weight for companies (aggregate investments in a company) and out-weight for investor (aggregate investments made). [1] We observe that the power-law behavior also appears in the weighted version of our network model.

3.4. Evaluation

As briefly mentioned in section 3.2, we make use of snapshots to create an earlier version of the final network

[1] We have investment data for specific funding rounds where multiple investors (typically 3-5 investors) simultaneously make an investment in a company. The specific contribution of each investor is not disaggregated, so we have chosen to attribute investment equally to each investor. In addition, we have given no weight to investments with missing data. (This is a first cut of weighting problems and we will consider how to more rigorously impute missing data.)

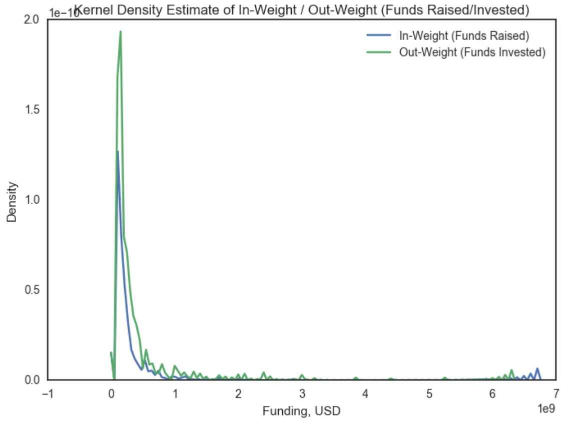

Figure 3. In-Weight and Out-Weight Distribution

that is observed in 2015. This allows us to create a training set (data accessible to our prediction models) and a test set (data used to evaluate our predictions). In this particular case, we have chosen a snapshot containing 70% of the edges in the final graph and we use this as our training data. The remaining edges represent the positive examples that we are trying to predict.

Our evaluation of prediction performance borrows from well-accepted machine learning practices and extend beyond simple accuracy measurements (mis-classification rate). This is a particular concern in our context because we have heavily imbalanced classes – there are far fewer actually observed links ($\sim$ 40k) than possible links ($\sim$ 4M) and we could obtain extremely high accuracy by simply predicting no new links. To address this issue, we will instead examine precision, recall and F1 score as alternative measures of predictor performance.

We note one further complication in our evaluation. Each prediction model will often make predictions using a threshold and the optimal threshold is usually not known a priori. Ideally, we would use something that examines every possible threshold level (e.g. a metric like area under the curve of the ROC graph). However, calculating millions of predictions using features from the graph network structure is computationally intensive and to make this problem tractable, we have simply chosen to enumerate predictor performance at certain points.

3.4.1 Formal Description of Evaluation Procedure

Formally, for an arbitrary prediction model m, we note that the model can be represented as a scoring function $score_m(i, c)$ that satisfies these conditions:

$$0 \leq score_m(i, c) \leq 1, \forall i \in \{\textbf{investors}\}, c \in \{\textbf{companies}\}$$

$$score(i, c) \propto P[(i, c) \in \{\textbf{test set}\}]$$

Then, for some threshold $0 \leq \gamma \leq 1$, we can follow this simple decision rule to make a prediction:

$$score_m(i, c) \geq \gamma \rightarrow \textbf{PREDICT new link}$$

$$score_m(i, c) < \gamma \rightarrow \textbf{PREDICT no link}$$

After making predictions on all of the possible, we can evaluate the performance by calculating the confusion matrix by noting the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives(FN) in our predictions. This allows us to calculate our final performance measures (for some model m and threshold γ):

$$Precision_{m,\gamma} = \frac{TP}{TP + FP}$$

$$Recall_{m,\gamma} = \frac{TP}{TP + FN}$$

$$(F_1)_{m,\gamma} = \frac{2TP}{2TP + FP + FN}$$

3.5. Prediction Methods

3.5.1 Random Link Prediction

$$score_{random}(i, c) \sim Unif(0, 1)$$

We follow the example of previous work and implement random link prediction. This method serves as a baseline to give context to the practical improvement of other methods.

3.5.2 Preferential Attachment Link Prediction

$$score_{PA}(i, c) = \frac{|\Gamma(i)||\Gamma(c)|}{z}$$

where z is a normalization constant and $\Gamma(x)$ indicates the neighborhood of x. Our use of the $\Gamma(x)$ function reflects the fact that, for computational complexity reasons, we implemented PA Link Prediction on the undirected analog of our directed bipartite graph (i.e., edges become undirected) rather than directly on our investment graph.

3.5.3 Weighted Preferential Attachment Link Prediction

$$score_{WPA}(i, c) = \frac{|\rho(i)||\rho(c)|}{z}$$

This is similar to the preferential attachment prediction, except that instead of calculating $|\Gamma(c)||\Gamma(i)|$, we calculate $|\rho(c)||\rho(i)|$ where $\rho(x)$ is sum of incoming weights for companies (equivalently, outgoing weights for investors). Our preliminary experiments found this to perform more poorly than standard PA. Since the behavior of WPA and PA will likely be similar (due to a strong correlation between degree and weight sizes), we decided to spend our limited computational time evaluating a more complex model.

3.5.4 Segmented Preferential Attachment Link Prediction

$$score_{\text{SPA}}(i,c) = \begin{cases} \frac{|\Gamma(i)||\Gamma(c)|}{z} & b_i == b_c \\ 0 & b_i \neq b_c \end{cases}$$

This is similar to preferential attachment but we add the additional constraint that new links can only be formed between investors and companies in the same ”bucket”. This model encodes some beliefs about a positive assortative matching [6] process between investors and companies. For example, this might correspond to a model where high-value and high-degree investors are only interested in investing in high-potential startups (which might be signalled by the number of previous investments).

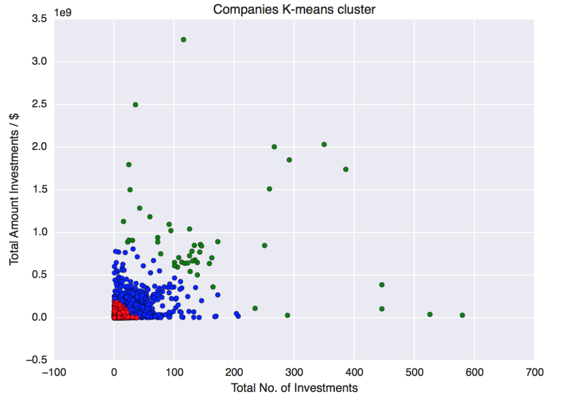

Figure 4. K-Means of Clustering on Companies

Cluster	Companies	Investors
A	11.29	15.64
B	1.85	3.27
C	0.35	0.16

Table 5. Euclidean distance of clusters from origin from origin

We do not have knowledge about these buckets assignments a priori but we use K-Means clustering[2] to create these buckets from observed correlations in the data. Specifically, we clustered on degree and total investments size for investor with a $k = 3$ (and, separately, clustered again for companies). Then, we rearranged the cluster labels so that the investor cluster with the furthest centroid from the origin was matched with the company cluster with the furthest centroid from the origin, etc. Table 4 shows an example of our clustering, where clusters are labelled with different colors. Table 5 reports the Euclidean distance of centroids from the origin of each cluster.

4. Experimental Results and Evaluation

We predicted with 13 different thresholds that were spaced at fairly small thresholds because the average prediction score was fairly low. For each threshold, we examined all three models so we fit 39 models in total.

In Table 6, we report the optimal threshold (as determined by the F_1 score and we also report the precision and recall achieved at that optimal point.

In Figure 5, we plot the precision values and note that the best recall came from segmented PA and that both PA and segmented PA both significantly out preformed random, although the absolute magnitudes were fairly low.

In Figure 6, we plot recall and note that random has high recall (although this will also be driven by the sheer number of positive predictions that the random model will make at low thresholds). Among the PA models, we note that the original PA seems to outperform Segmented PA.

In Figure 7, we plot the F_1 scores, which combine both precision and recall. As you can see, the combined performance of the random model is, as expected, extremely poor at all thresholds. PA outperforms Segmented PA in this metric at all thresholds and seems to follow similar trends. Again, the absolute levels of F_1 scores are extremely low.

Metric/Model	Random	PA	SPA
Optimal Threshold	0.04	0.08	0.05
Precision	2.51×10^{-5}	0.00412	0.00267
Recall	0.954	0.01359	0.00698
F_1	5.02×10^{-5}	0.00632	0.00386

Table 6. Performance Metrics At Optimal F_1 Value

5. Conclusion

Our results showed that making use of the observed preferential attachment behavior that we observed in the global structure gave some insight into local dynamics, especially in the context of link prediction. Our experimental results showed this was significantly better (factor of 100) than random prediction. However, our results also showed that the absolute levels of precision and recall are all extremely low. More specifically, while we have provided some evidence that our models are significantly better than random, our evidence re-confirmed that this has low practical significance

[2] K-Means finds the cluster labellings B through this minimization:

$$\arg\min_B \sum_{i=1}^{k} \sum_{x \in b_i} ||x - \mu_i||^2$$

We use a Lloyd’s Algorithm implementation of K-means to find an locally optimal B.

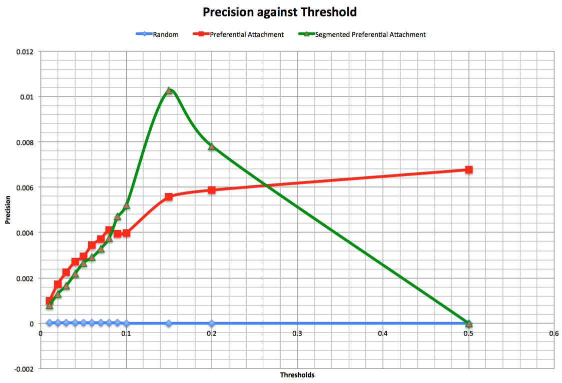

Figure 5. Precision

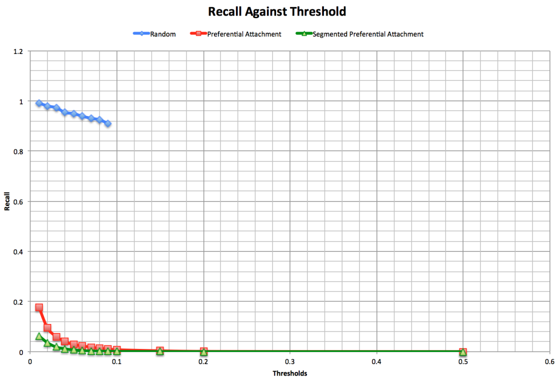

Figure 6. Recall

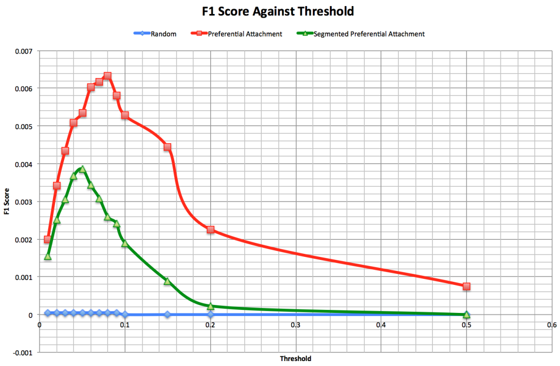

Figure 7. F1 Scores

– using PA scores alone is a poor predictor of new links.

Embedding some domain knowledge into the scoring function did not, on average, improve the performance of our link prediction. We do note, however, that embedding this knowledge significantly improved the maximum precision of the model. Specifically, our formulation of Segmented PA successfully excluded from consideration some extremely unlikely investment-company links.

More generally, our experimental setup holds some promise. Our evaluation procedure could be easily extended to the use of any sort of more sophisticated prediction model (e.g., logistic regression, SVMs, etc.) that could make use of not only the network structure but also any additional node or edge characteristics.

These extensions, however, would be highly dependent on the available computing power. Even with the use of extremely simple prediction models, calculating precision and recall was only marginally feasible on consumer hardware. This suggests that we should make more heavy use of pre-precessing of the graph nodes to reduce the number of edges that must be considered (as we did in Segmented PA). In the situation where we have more significant computing resources, we highlight some potential extensions in the final section.

5.1. Future Work

5.1.1 Use Large Scale Graph Processing Framework

We were limited in the types of scoring functions and features we could compute on our dataset due to limited processing power of our laptops. We intend to use graph processing frameworks such as GraphX on Apache Spark or Giraph on EC2 instances to process our huge network.

5.1.2 All Node Pairs Path Measures

With a large scale processing framework, we will be able to then calculate

1. Shortest path length between every pair of nodes

2. Total number of paths between every pair of nodes due to run time complexity.

We will optimize our current implementation and hopefully get more metrics that we can use to test and build our link prediction model.

5.1.3 Implement a modified Adamic-Adar score

In addition, we intend to modify the Adamic-Adar measure our link prediction model because it outperformed most other similarity measures [7] and fit it to work on our bipartite model.

$$\sum_{z \in \Gamma'(x) \cap \Gamma(y)} \frac{1}{\log \Gamma(z)} \tag{5}$$

where $\Gamma'(u)$ represents the nodes 2 hops away from u, and $\Gamma(u)$ represents the nodes 1 hop away from u. We do know that this is an asymmetric measure, as we have to incorporate this to fit our bipartite graph. We will explore both sides of the asymmetry, and experimentally determine the possible different results that may be caused by this.

5.1.4 Mean Average Precision (MAP) evaluation

As we know the absolute score of precision and recall are quite low, a more insightful metric might be to use the MAP measure to evaluate our models. This presents a better picture as we will input the number of links to predict for each node to the algorithm, hence definitely improving accuracy as we currently use a threshold value to predict links.

$$\text{MAP}(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}\text{Precision}(R_{jk}) \qquad (6)$$

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[3] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[4] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[5] M. Ferrary and M. Granovetter. The role of venture capital firms in silicon valley's complex innovation network. *Economy and Society*, 38(2):326–359, 2009.

[6] M. Kalmijn. Assortative mating by cultural and economic occupational status. *American Journal of Sociology*, pages 422–452, 1994.

[7] J. Kunegis, E. W. De Luca, and S. Albayrak. The link prediction problem in bipartite networks. In *Computational intelligence for knowledge-based systems design*, pages 380–389. Springer, 2010.

[8] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[9] D. Trapido. Mechanisms of venture capital co-investment networks: Evolution and performance implications.