Investor Networks in Venture Capital:

with Silicon Valley Bank

Trevor Gower

M.S. Data Science: GalvanizeU, University of New Have

44 Tehama St. San Francisco, CA

[trevorgowerr@gmail.com](mailto:trevorgowerr@gmail.com)

**I. INTRODUCTION**

Venture capital is a notoriously difficult field for investors. Companies are young, with uncertain financials at best, seeking to break into emerging markets. A successful investment is as much about understanding the people behind the company as it is in understanding the market itself.

Relationships and personal trust play a major role in determining which companies investors will participate in. Similarly, lending institutions develop ongoing relationships with investment firms and with particular partners in those firms. These relationships run between venture partners, firm founders, and the lending officers; in this case we will focus on the investor relationships.

Silicon Valley Bank is the leading institution of venture lending. It has strong ties with the top venture capital firms, particularly in the Bay Area, and is known for providing debt financing to promising companies during early venture stages. Early venture stages range from the seed round to Series A through C, and Silicon Valley Bank often becomes involved at the Series A stage.

Unlike equity lenders, the primary goal of debt lenders is first and foremost to secure repayment of loans. While venture capital funds can afford to have five, seven, even nine losses for every success, a bank requires more reliable rates of return. Write-downs, for a bank, are more costly than several lesser successes.

Silicon Valley Bank, hereafter to be referred to as SVB, was interested in exploring various uses of an investor network graph. Information about the relations of investors to each other and the relationship of investors to successful deals with the bank, could hold valuable insights.

PitchBook is a website devoted to covering venture capital, from seed stage to public offerings. In particular, it is known for keeping records of individual people with firms, including partners in venture capital firms and high-ranking employees and found of startups. Data from PitchBook, combined with SVB’s internal resources, presents a valuable source of potential insights.

**II. Overview**

1. **Hypothesis**

Investors are believed to vary in quality. Obviously some investors bring special expertise to certain industries, but there are other factors to consider. Some may have a remarkable intuition for picking or attracting strong founders, some may have good deal-making skills, and others are likely to attract useful investors at later stages.

Other factors influence the success of a lending investment. These include, but are not limited to, the economic climate, the particular state of the industry, the track record of founders, the strength of competitors, and of course the quality of the company itself.

Three particular approaches to the investor network were considered to be of immediate value.

* + The construction of a query-ready graph database. Through collaborative filtering and the general availability of data, lending officers and relationship experts at SVB may obtain more useful data to contextualize their decisions
  + Clustering of the Investors and Partners. Machine learning techniques can determine, without prior labels, likely clusters of data points within a general set. These methods may be used to discover close networks of investors within the overall social network graph.
  + Classification of deals by investor involvement. Machine learning techniques can classify, with the use of prior labeled data, future data with similar feature sets. By one-hot coding the investors who took part in SVB-related companies, we can obtain a feature set. This may then be trained on labeled data. Methods of classification included in this project are Bayesian, Random Forest, and Neural Nets.

**B. Theory of the Solution**

Graph databases are becoming increasingly common as data-based social networks have become part of our daily lives. The effect of social relationships on investment decisions has been previously documented, and an database of these relationships should prove useful for SVB’s operations.

The general use of SQL queries has become commonplace in financial institutions, only second to spreadsheets. Graph databases can support a similar query language for general use. They are constructed in a way that not only creates simple visualization of the data, but can also allow for the rapid writing of collaborative filtering queries. Such queries, when written for SQL database, tend to be long and expensive to run.

Collaborative filtering consists of making recommendations to an individual based on other preferences they have expressed. In this case, a simple example would be if two investors who have repeatedly invested with another investor, take part in a deal together. It is highly likely that the third investor would also be interested in that deal. That investor may also be interested in doing business with other investors that those two have interacted with. Studying these sorts of recommendations is the purview of collaborative filtering.

Clustering is an unsupervised machine learning model. This means that, as previously stated, the data need not come with prior labels. By using clustering algorithms and exploring hyperparameters, we can find clusters of closely related investors, based on their co-investments. In future, this can also support clustering by other features of similarity.

Classification is a supervised learning model, which means that the data comes with labels. In this case, companies are marked as either positive or negative in accordance with their lending outcomes. The feature set utilized, as previously stated, is a one-hot coded investor matrix.

Random Forest, Naive Bayes, and Neural Nets can all be used to solve classification problems.

Bayesian methods assign a simple probability by which the likelihood to which class a row belongs. This consists of updating the probability of each feature, based on posterior probability and new evidence, then multiplying together the probabilities across the rows. This requires the “naive” assumption that the features are independent.

Random Forest Trees develop individual “decision-based” classifications, using a True/False logic at each stage to split the data. This is done until the data is grouped into largely uniform classes, often by use of cross-entropy and information loss.

Neural nets, at their most basic, accept inputs which are generally run through hidden layers including an activation function, such as sigmoid or tanh. The weights can then be trained using a cost function such as Mean Squared Error. A simple feed-forward neural net without a hidden layer, utilizing a sigmoid function, works in a manner similar to logistic regression.

Singular Value Decomposition (SVD for short) is a method of decomposing a matrix into two orthogonal matrices and the root of their shared eigenvalues. This eigenvalue matrix compresses data into the most important features, and orders the eigenvalues in descending order along the diagonal. SVD’s may be used not only to compress data, but to remove the less significant features by setting the corresponding eigenvalues to 0.

By multiplying this value with the respective orthogonal matrices, we may “rotate” the matrix back into a state that can be trained on. If eigenvalues were zeroed out, the resulting matrix will have a corresponding zeroed out area, which can then be split off as inconsequential to training. This effect is similar to other forms of embedding data in a condensed hidden layer of a neural net. [[1]](#footnote-1)

**C. Prior Literature**

Applications of graph theory to economics are known as an effective area of study. [[2]](#footnote-2)[[3]](#footnote-3)

There have been particular advancements in applying graph theory to venture capital by Crunchbase, Pitchbook, and MatterMark. An exploration of free Crunchbase data is publicly available [[4]](#footnote-4)

More formal studies of the effects of networks on the operations of venture capital have been conducted as well.[[5]](#footnote-5) There exists a lay analysis of this paper, pointing out that the effects of an investor include not only their own intuition as an investor, but the ability to secure funding in later rounds. [[6]](#footnote-6)

**III. Implementation**

**A. Choice of Tools**

Python includes a large number of machine learning tools and packages, including clustering and neural net packages. Python 2.7 was utilized for this project, with primary use of NumPy, SciKit-Learns, and Tensorflow. NumPy assists with numeric computations, particularly matrices. SciKit-Learns contains many machine learning modules and algorithms, including clustering, Bayesian classifiers, and Random Forest Classifiers. Tensorflow supports neural nets.

Python also supports the Neo4J via the Py2Neo package. Neo4J is one of the most commonly used graph databases in existence, with the ability to support basic queries and store information along both nodes and edges.

**B. Choice of Data**

Pitchbook, as previously stated, contains considerable information regarding the involvement of individual people in companies and investments. Most sites that cover venture capital markets focus on the financial information of the deals themselves.

Pitchbook itself can be queried for information. Companies of interest were selected from SVB’s internal data sources. The Series A deals of all negative companies, determined by internal logic, were searched for.

Pitchbook’s Excel Download tool was used to download the information for the A deals for these companies and then to create CSV files. This data included the date of the deal and information identifying the company, the CEO of the company, and the Investors that took part in the deal. In Pitchbook’s case, an Investor refers to a specific ventura capital firm, which may or may not have an individual person (Partner), attached to the deal.

After unpacking the negative deal information, queries for other Series A deals were constructed. These included a query searching for the intersection of other companies that the Partners on the negative deals took part in, and SVB commitments. This returned positive companies that included partners on negative deals.

Further queries on Pitchbook data yielded companies which SVB ended to and also contained known partners of influence. The same data formatting was used each time, producing.

The CSV’s of the deal-level information were unpacked, and the investors in each deal were recorded. Deal-level investor lists were aggregated to the company level. This was then one-hot coded.

There are then two sets of information. A matrix with investors as the feature space over rows of companies, and a positive or negative label for each company. A column vector of the corresponding labels indicating a positive (unstressed) or negative (distressed) company. These labels can be expressed either in a single column, with 1 indicating positive and 0 indicating a distressed deal, or as a two-column vector. In the two-column case, one column indicates a deal is distressed and the other indicates that a deal is positive, with only one column set to one for each row.

Data presently used for training contains a roughly 70-30 class imbalance, with positive labeled companies making up the greater part.

**C. Choice of Models**

DBScan can identify clusters in high-dimensional space. It does so by only grouping together those clusters that are “near” each other, with most features (Investors) being pushed to the edge of the graph by high-dimensionality. In practical terms, this means that only a relatively small number of Investors/Partners will be grouped into clusters. For investment purposes, this is acceptable, as it isolates central Partners and their networks.

Naive Bayes and Random Forest are standard classifiers, and the former is known to perform linearly. These provide a baseline and a useful comparison for neural net performance.

Logistic Regression is also a common choice for work on non-linear models. However the simple neural net was found to train in essentially the same manner, and so it was dropped from comparison in favor of the more robust and customizable neural nets.

Given the broad and sparse nature of the data, feeding the investor matrix through an SVD into a neural net was attempted. This hoped to provide higher metrics, superior speed, and less variance in training.

**D. Choice of Metrics**

As previously stated, banks must reliably produce positive results. Over the course of most years, fewer than 1-2% percent of loans end in default. Even with an expanded query designed to identify deals with poor performance, the percentage of total loans that experience extreme distress is under 4%.

Given limitations on the size of this project’s data downloads, the class imbalance is not so great in the data, yet it is still prominent. Roughly 30% of companies currently included in our data set are negative.

Therefore, although accuracy is a highly readable metric, a “recall” score corresponds most closely to our goal. Although this is part of an F1 function, it is not the sole factor.

Furthermore, in terms of business operations, the highest value may be yielded by determining those deals most certain to have negative performance, rather than simple binary classification. The models therefore predicted the probability that a company would fall into distress and a ROC curve was fit to these.

Grid-searching the ROC curve discovers those thresholds that, when used to round the probabilities, return the highest number of distressed deals identified compared to the number of good deals that are incorrectly identified.

In other terms, a grid-search that discovers the set of simple local minima for recall provides a business-ready way to evaluate the value of each model.

**IV. Exploratory Data Analysis**

As stated, searches were performed at the company level and the Series A deal for those companies were downloaded. It may not be the case that SVB became involved at the time of the Series A deal, but the date and investor involvement at that time provide our feature space.

Negative deal companies were dated by the time at which they became distressed, not on their Series A. The query for distressed deals collected all those deals which became distressed during or after 2011. The Series A deals for these companies traced further back than that, including some as early as 2004.

Analysis of SVB’s lending performance is based off the entirety of deals returned by Pitchbook. Although PitchBook did not contain all the deals found in SVB’s commitments, it may be assumed that the ratios in the PitchBook explorations are relatively consistent.

Each year there are roughly 340 companies which complete their Series A deals and go on to do business with PitchBook. Of these, roughly 3% fell into distress.

The study indicates that SVB already does a fairly thorough job of filtering investors. PitchBook could only identify a handful of investors who were on multiple distressed deals. All other specific partners that PitchBook could identify were involved with just one distressed company each.

There was more information at the firm level; 37 investment firms invested in two or more distressed companies. These companies took part in a substantial percentage of the total number of distressed companies. There was also notable overlap of these companies among the distressed deals.

The data suggests that SVB already is effective at constraining poor investor performance. Lending officers are aware of the importance of past performance and the presence of investor networks, and they include this criteria in choosing companies for lending. This machine learning project will particularly seek to assist them in identifying those companies which are likely to fall into distress.

**V. Results**

**A. Clustering**

The investor matrix, formed as previously indicated, was clustered using DBScan. In large, sparse datasets, there is a tendency for points on the data to be pushed either very close together or very far apart. DBScan handles this well because it is less influenced by points at the edges. DBScan therefore performs well by selecting clusters within the data, the actual dense networks, and dropping the further points.

The input parameter for DBScan is the minimum number of points within a cluster. This means that a cluster of 4 investors would appear with a minimum of 3, but not at a minimum of 5. A search through several values at this parameter better helps us understand the density of sub-clusters.

At a minimum of 3, nearly 150 clusters were found. At a minimum of 10, 15 clusters were found. At a minimum of 20, 3 clusters were found. In the final case, fewer than 100 investors were found to be in clusters. These are investors of particular interest for SVB’s operations.

**B. Classification**

Using the investor matrix and the company label array, Bayesian, Random Forest, and a Neural Net classier were used to train predictive models. The goal of classification is to assist SVB in identifying likely negative companies and, by dropping these, to improve portfolio performance.

**i. Naive Bayes, Random Forest, Single Layer Neural Net**

SKLearn contains packages which were used to generate random sets of data for training and testing. Given the limited data for this project, the train-test split was set at 0.1. Training accuracy was seen to decrease significantly at .15, but the train-test split is expected to increase as more data becomes available. When testing models either against each other or for finding hyper parameters, each model was run over a number of random seeds, and the selection was determined by overall performance.

SKLearn also contains packages with which to run Naive Bayes and Random Forest Decision trees. Given that Random Forest is designed to accept a multi-column label vector, this project utilized the two-column label array for that classifier. All other classifiers utilize a single column vector for labels.

Both Naive Bayes and Random Forest were set to fit a model to the train set and tested for accuracy and portfolio performance on the test set. Random Forest also requires a hyper parameter for the number of trees which will be created and used in “voting.” A grid search was performed on this hyper-parameter, and it was found that at least 150 trees were necessary to optimize accuracy. This is a somewhat high number, but reasonable give the feature set size and the relatively small number of companies.

A simple feed-forward neural net was trained on the data. Both the Avant-de Glorot and Apres Glorot methods of assigning initial weights were tested. Significant differences were not observed between the two methods, but Apres Glorot was selected for reasons of theoretical stability.

The data was run through a simple activation function: sigmoid, tanh, and relu were all tested. Sigmoid was found to produce the most consistent strong results in testing. Tan also produced some highly accurate results, but displayed a greater variance in error rates and an increased tendency to overfit.

Naive Bayes severely underperforms all other models. This is unsurprising, given the non-linear relationships that often underlie graph data.

The simple feed-forward neural net typically performs at least on par with Random Forest in terms of accuracy, but with considerably less variance. The feed forward net averages around 68% accuracy, the value most often arrived at by the Random Forest. However, the Random Forest may also significantly underperform in terms of accuracy and produces a less “flexible” set of optimal threshold.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neural Net Without Embedding | |  | Random Forest | |
| Accuracy: 68% | |  | Accuracy: 68% | |
| 7.7% | 16.7% |  |  |  |
| 15.4% | 50% |  | 7.7% | 50% |
| 46.2% | 66.7% |  | 58.3% | 78.6% |
| 65.4% | 91.7% |  |  |  |
|  |  |  |  |  |
| Percentage of Good Deals Miscategorized | | Percentage of Distressed Deals recognized | |  |

When the predicted labels of the models are then run through business logic, we may determine a new portfolio of recommended investments. By dropping all deals marked by the model as distressed, and recalculating on their actual labels, this project can observe the resulting portfolio size and performance. Performance, in this case, is measured as the percentage of actual positive deals in the portfolio.

It was observed that the portfolio performance resulting from Naive Bayes was found to produce slightly higher portfolio performance, but at the cost of portfolio size. The preference between these two may be determined based on business needs.

**ii. SVD Embedding**

Neural Nets may also be trained on compressed data, using an embedding layer or SVD. By compressing the feature space into a fraction of its former size, sparsity will prove less of an issue.

In order to test the value of compressing the data, an SVD was used to compress the feature space down to sizes of 10, 30, 50, 100, and 300. This embedded data was then run through the feedforward neural net.

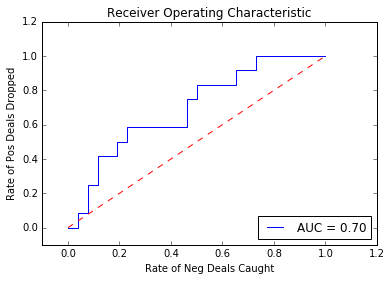
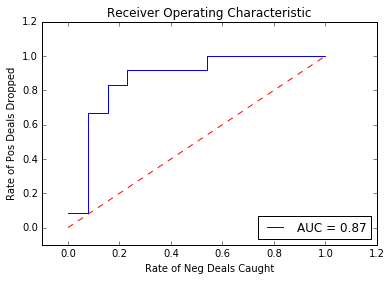
Initially the accuracy declined significantly, due to a weak selection of labels. At first the labels were submitted to the net as a two-column array, rather than a single binary value. The SVD model performance improved tremendously after switching to a single binary column.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Neural Net With Embedding | | | | |
| Accuracy: 86% | |  | Accuracy: 78-81% | |
| 3.8% | 25% |  | 3.8% | 25% |
| 7.7% | 50% |  | 7.7% | 33.3% |
| 19.2% | 83.3% |  | 26.9% | 66.7% |
| 23.1% | 91.7% |  | 30.8% | 83.3% |
|  |  |  | 34.6% | 91.7% |
|  |  |  |  |  |
| Percentage of Good Deals Miscategorized | |  |  |  |
| Percentage of Distressed Deals recognized | |  |  |  |

Training with SVD produced superior accuracy and generated superior portfolio results to the other models. This appeared to be true over all the compressions, but the increase in performance and resilience against overfitting appeared to be strongest at a new feature size of 100 or 300.

The SVD training time was also far faster than the simple neural net training, taking an order of magnitude fewer steps. It did, however, appear particularly sensitive to changes in the initial learning rate.

Shown below are the portfolio recommendations based on SVD Neural Net runs.



SVD Net

Single Layer Net

**VI. Conclusions**

From the data we may see that investors present at the Series A round significantly influence the expectancy of that company’s success. SVB practices already significantly filter the quality of investors and deals, and machine learning can further identify potentially distressed deals. While overlap in the investor group prevents generally prevents a perfect recall, results are produced that, if acted on, generate superior portfolio performance at relatively little cost.

This project is not at liberty to indicate the total portfolio performance that may arise from application of these models, as it treats with potentially confidential data. However, some of the performance increases were notable.

Expansion of the feature set, including industry, clusters, founders, and venture partner information are likely to continue improving the results.

This supports the concept that investor participation and condensed networks are strong predictors of investment success. Furthermore, it demonstrates the applicability of DB Scan and SVD to non-linear graph data.

**VII. Steps Forward**

Although alternative neural nets, including one with two hidden layers and another with a particle swarm optimizer, were attempted, they did not initially produce better results. However, modification and comparison of the SVD Neural Net with neural net embedding techniques may yield further improvements.

Future steps will include the use of cluster and industry information as features apart from direct investment. The development of a recall optimization for gradient descent will likely generate stronger and more consistent portfolio performance

Furthermore, combining the neural net predictions of likely successful deals and recommendation based off collaborative filtering from the Neo4J graph, it will be possible to automatically search and recommend likely positive companies as they are added to PitchBook.

1. <https://arxiv.org/abs/1509.08360> [↑](#footnote-ref-1)
2. <https://www.sg.ethz.ch/media/publication_files/KonigBattiston2009.pdf> [↑](#footnote-ref-2)
3. <http://www.isid.ac.in/~dmishra/mpdoc/lecgraph.pdf> [↑](#footnote-ref-3)
4. <https://linkurio.us/crunchbase-graph-analysing-graph/> [↑](#footnote-ref-4)
5. <http://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/VC_networks.pdf> [↑](#footnote-ref-5)
6. <http://insight.kellogg.northwestern.edu/article/whom_you_know_matters> [↑](#footnote-ref-6)