Applying Software Development and Artificial Intelligence to Venture Capital

MEng Thesis Proposal

Yasyf Mohamedali {yasyf@mit.edu}
Thesis Supervisor: John Guttag {guttag@csail.mit.edu}

Abstract

We enumerate several opportunities for software development and artificial intelligence to be applied to the day-to-day operations of seed-stage venture funds, and identify two such opportunities to build automated tools around. The first tool is a graph-based ranking system that ingests information and structure from the Internet, and outputs interesting (given a venture context) nodes. The second tool is a recommender system for early founders to find investors, alongside a public-facing tool to collect data and surface recommendations.

I. Introduction

Our goal is to successfully apply techniques from software development and artificial intelligence to the day-to-day operations of a venture capital firm, in order to increase the efficiency with which the firm deploys its capital to the optimal set of startup companies. In order to describe our path to this goal, it is first necessary to define what a venture capital firm is, what its goals are, how efficiency is defined, and how success is measured. Following this, we will explore several opportunities for modern computer science to enable venture firms to operate more efficiently, proposing a product or tool for each. Finally, we hope to build a system which implements two of these tools, and evaluate the efficacy of these tools in the real world.

A. *Definitions*

For the purposes of this thesis, we will consider the standard structure of a seed-stage venture firm, as follows. A venture firm, or VC, is composed of a central pool of capital, contributed by individuals or organizations known as Limited Partners (LPs). This pool is managed by individuals known as General Partners (GPs), who are compensated for their work both with a fraction of the pool (the management fee) as well as a fraction of the returns on their investments (the carry).

In our simplified model, the sole goal of a VC is to trade capital from the pool for equity in companies that will later either enter public markets (via an Initial Public Offering, or IPO) or get acquired by another company. These liquidation events allow the VC to sell their equity for more than the original purchase price. Thus, the success of a VC is measured by the realized capital gains that are accrued when they sell equity in a now-public investment, and the objective function they optimize for is the expected value of this gain over all their investments. We will define efficiency as the fraction of hours in a given period of time the GPs must spend working in order to achieve some level of expected returns.

For more background on venture capital and the ongoing economic research in the field, we refer the reader to [1].

B. *Opportunities*

The time GPs spend working is split between the activities of sourcing, analyzing, and supporting startups. One can imagine these forming a funnel-like pipeline: sourcing looks to fill the top of the funnel with as many high-quality companies as possible, analyzing seeks to filter these companies down to only the investment-worthy ones, and supporting aims to lengthen the lifespan of the existing companies by as much as possible.

While it is clear how sourcing additional companies and doing a better job of analyzing potential investments is beneficial to the bottom line of a firm, it is not self-evident that investing time into supporting portfolio companies leads to greater expected returns. To mitigate these concerns, we refer the reader to [2], which shows that supporting portfolio companies results in "an increase in innovation and the likelihood of a successful exit.".

While the proportion of time spent on each activity varies greatly between firms, we will assume a roughly equal split, such that increasing efficiency in any one is equally impactful. Importantly, these three activities correspond to the three opportunities we will consider for increased efficiency in venture capital.

1) Sourcing: Sourcing entails GPs leveraging their networks and any available information (free or proprietary) to discover the optimal set of companies to consider investment in. The stream of companies that are being considered are known as "deal flow". This is commonly split into outbound and inbound flow. Outbound flow is generated by the partners attending events and scouring their digital and analog networks for new companies being started. Inbound flow, on the other hand, is generated by startup founders reaching out to the firm and requesting consideration for investment.

Much of sourcing requires humans integrating large swaths of linked information, resulting in a few highlights in the form of "interesting" companies. The more data that can be ingested, the more interesting companies are surfaced. We model this as an unsupervised graph problem, where nodes represent information accessible to a firm, and explore how we can learn to identify interesting nodes at a scale no human could manage.

2) Analyzing: The process of analyzing and doing due diligence on startups is how the GPs of a firm decide whether or not to invest. This can include reviewing the product, financials, and traction of the startup, in addition to doing research on the founders and broader industry at hand.

The lowest-hanging fruit in this process is the notion of automatically filtering, categorizing, and ranking pipeline items. Investors currently limit the number of companies they are considering at any given time to the few they can learn absolutely everything about. Furthermore, they pass on many companies on the basis of cheap filters and pattern-matching historic successes. The problem of clustering and raking companies can be modeled as a supervised, structured problem, leveraging both historic successes as well as past misses.

3) Supporting: Providing what is known as "portfolio support" is how venture firms attempt to ensure the companies they invest in survive long enough to IPO (thereby allowing the VCs to cash out). This encompasses everything from advising the founders, to making key introductions, to helping the company raise further funds, to helping publicize big product announcements.

The opportunities in this area seem to be around how we can use commoditized machine learning techniques to solve problems more optimally. These include matching founders to investors (effectively the Netflix Prize [3] problem) and predicting success of social media blasts (with something like SEISMIC [4]).

II. Opportunities

We have spent time exploring possible tools that could be built to aid in various stages of the venture pipeline. Each tool is identified below, along with the motivation and a brief summary of the technical challenges involved.

A. Sourcing

We have identified two opportunities to do with sourcing, both on the outbound flow side.

1) A system to aggregate signals from founders and predict the intent to start a company: Founders often emit signals that indicate they are starting a new company, often long before they officially announce their new endeavor. These signals can be explicit (changing a job title on LinkedIn, or biography line on Twitter) or implicit (leaving a job, moving cities, or attending entrepreneurial events). In isolation, these signals are not strong, but in aggregate they can be strongly correlated with the intent to start a company.

We see an opportunity for a system that monitors the social networks of a GP, identifying and aggregating potential signals. The technical challenges include linking seemingly-unrelated signals across networks and schemas, and inventing a ranking algorithm which can present the most likely potential founders given a set of signals. We would likely use these signals as machine learning features.

2) A system to discover and monitor promising out-of-network individuals and organizations: While there are a plethora of announcements and releases online which would indicate an investment-worth company has formed, humans are not capable of monitoring and filtering the wealth of information generated on the internet on an ongoing basis. Thus, a GP's sourcing abilities are largely limited to the founders they can discover in their network.

We propose a system which treats the relevant information on the Internet as a connected, directed graph, which can be monitored and have its nodes ranked (as PageRank does for search engines). Every interesting community (such as educational institutions) could have its own independent graph, and the top-ranked nodes of each graph could be surfaced for easy human review. The technical challenges around this system include a lack of labeled training data (what constitutes an "interesting" node?) and the noisiness of the web (there are many sites linked from a community that contain irrelevant or even misleading information).

B. Analyzing

When is comes to analyzing, there are two major project proposal we considered.

1) A system to filter, categorize, and rank the companies in a venture pipeline: Many seed-stage funds suffer today from an overwhelming pipeline of startup companies to consider. There is considerable data available on these companies which seems to be correlated to how investment-worthy the company is at first glance. At the very least, the cheap filters applied by investors are mimicable through existing data (alma maters of founders, size of initial market, sentiment of partners after first meeting).

We propose a system which uses the information associated with pipeline companies to categorize each company into buckets that predict how far in the pipeline the company will move, using these buckets to filter and prioritize the pipeline. This will be an online, semi-supervised clustering problem which receives constant feedback from partners. The technical challenges include identifying and extracting the relevant features (which may include leveraging NLP techniques on descriptions, pitch decks, and meeting notes), and finding a way to incorporate user feedback in a meaningful way. Evaluation methods are also difficult to formulate a priori.

2) A system to surface and summarize key trends and news in a given industry: Many hours of time is wasted at venture firms serially researching and identifying key facts and risks about both a company and its broader industry. This act of information extraction and summarization is well-suited for classic Natural Language Processing.

We propose a system which ingests both internal data on the company at hand, as well as recent news and evergreen data sources (such as Wikipedia) and delivers a digest of key risks identified in the company (based on pitch decks and partner notes), as well as a one-pager on the given industry.

C. Supporting

Finally, with regards to portfolio support, there are two tools we considered building.

1) A system for the discovery of and supporting outreach to the optimal set of seed-stage investors: It is widely accepted in the venture industry that there is significant merit to a founder finding the "right" set of investors when raising money. Not only does the strategic focus of a firm and its network impact said firm's ability to help a company, but the particular focus of a partner within a firm can also influence whether or not a company even gets funded. There is strong empirical evidence that partners at venture firms do indeed specialize and focus on a very specific subset of companies [5].

Matching a founder to the most relevant partner at each firm, and the most realistic and appropriate firms at each funding stage, is a challenging problem for humans to tackle alone.

We propose a hybrid recommender system which suggests relevant and strategic investors to founders, based on their company and ideal investor profile. This would follow the models laid out in recent literature on recommender systems [6].

Our tool would also provide an interface for planning and tracking the process of reaching out to these investors, as a way to collect structured training data for future iterations. Technical challenges here include building a sufficiently strong user experience so as to inspire trust in the tool, determining how to identify a user as features, and building a labeled database of investors and the founders they have backed.

2) A system to predict and propagate viral company news: As the number of companies in a seed-stage venture firm's portfolio grows, it becomes increasingly difficult for partners to keep track of the movements of each company. This makes it difficult to identify when a company is in the process of making a big press release (which the VC could support). Furthermore, there is no easy way for a VC to know the latest public change in each of their companies.

We propose a tool to monitor the social media accounts of portfolio companies, summarizing news and sharing the posts that are estimated to be the most popular or viral. Text summarization is an open research problem that has several standardized solutions [7], each of which can be tuned for the domain with manual feature engineering and additional rule-based systems. Estimating social media popularity and virality can be done with linear point-process models such as SEISMIC [4], or more complex Bayesian models like the one presented in [8], which uses more features from the graph generated by the post and its shares. The biggest technical challenges here are around coaxing and tuning these algorithms to give sufficiently good results for our domain.

III. Preliminary Work

The past six weeks have been spent exploring the opportunities described above, envisioning what each solution would look like, and which would provide the greatest combination of technical novelty, feasibility, and impact for a venture fund. After completing roughly 20 user interviews with venture partners in Boston and San Francisco, and a further 10 with startup founders across the country, two of the proposals stand out as the best to dive deeper into. We have decided to build *a system to discover and monitor promising out-of-network individuals and organizations* ("FounderRank") and *a system for the discovery of and supporting outreach to the optimal set of seed-stage investors* ("VCWiz").

A. FounderRank

For FounderRank, we have started the process of building a graph of relevant institutions and entities, as well as preliminary work on ranking the nodes of this graph. For now, we have limited the graph to a given educational institution.

To build the graph, we first start with the `.edu` domain of the institution. From there, we use Google to find the websites of the entrepreneurship organizations and clubs on campus. We then crawl these to find startup websites, and crawl those startup sites to find news and press sites. Finally, we add backlinks from the press sites to any other existing startup node in the graph.

We end up with a graph of authority hubs, which link to startups and their founders, as well as news sites that provide relative rankings for the other linked entities.

We initially ran the standard PageRank [9] algorithm (with `NetworkX` [10]) on the graph crawled as described above, with fairly abysmal results. There is far too much noise on the web, particularly in the variety of (potentially irrelevant) sites that an entrepreneurship center can link to. We attempted to filter out some of this noisiness by pruning "company" nodes which only have one inbound edge. Adding only backlinks from news sites was also an attempt to make the final graph much less noisy. When tested on MIT and evaluated manually, these two strategies resulted in a set of nodes that were valuable to a venture firm, though the accuracy of the collection process could be much improved.

To further increase the efficacy of the system, we need a way to filter out appropriate nodes at each level of the graph. For example, we first sought to isolate only the startups linked from each entrepreneurship center, ignoring all other outgoing links. We attempted to train a binary classifier on nothing but the domain name itself, a humorous attempt that proved somewhat successful.

For our training data, we used the top 20000 sites from Alexa [11] as negative data points, and the domains of the 20000 most recent startups in Boston and San Francisco, as reported by Crunchbase [12], as the positive. Note the equal numbers of data points to avoid class bias. We started by mapping each domain to a padded vector of ASCII character codes. These character code vectors were then fed into a neural architecture (using Tensorflow [13]) containing an embedding layer, followed by an LSTM layer, and finally a sigmoid output layer. Biasing the decision threshold to avoid false negatives (thresholding around 40%) gave around 70% accuracy.

Examining the results of this classifier did not seem to indicate that the high-information top level domain (TLD) was being taken into account, so we developed a second model which takes the output of the LSTM from the previous, as well as two engineered features (the length of the domain and a one-hot representation of the TLD), and feeds into a fully-connected net. Optimizing for a weighted combination of the loss of the original model and the loss of the new model renders an accuracy of 89%.

A linear model with only the two engineered features as input performs fairly well, giving about 76% accuracy.

Future work will involve more complicated categorization and filtering of websites based on bodies, not just domain names.

B. VCWiz

Our work on VCWiz to date has involved exploring what data we need collect and how we can collect it, on the discovery side, and identifying the ideal user interface and experience on the outreach side. We realized that in order to provide truly useful investor recommendations to founders, it was necessary to collect and leverage a large corpus of user-sourced data on founder-investor pairings.

We first spent time talking to startup founders going through YCombinator and other well-known accelerators, right as they were about to begin raising their seed funding rounds. They all identified a need for personalized suggestions of investors. Our initial idea was to collect information from each founder on their ideal investor (characterizing investors and firms with features such as industry, check size, and location), and generate suggestions from a cluster of similar investors (using a k-nearest neighbors algorithm). With this in hand, we build the first iteration of the VCWiz application.

The first version of VCWiz collected a founder's ideal investor profile, as well as basic company information, before taking them to a screen of recommendations. The founder had the option to add any of the recommended investors to their list to begin tracking them, and were then taken to the main card-based view of the app. This view presented a series of stages, from "Waiting for Into" to "Waiting for Response", to "Need to Respond", to "Interested" or "Not Interested". Investor cards, which showed summaries of a partner at a firm, alongside community notes on both the partner and firm, could be moved between stages through dynamic buttons, which captured the transitions between stages. At this time, the collected data was not leveraged, save for sorting the recommendations for new users by popularity in the existing user base.

We learned a few crucial insights through the launch and test of this first iteration of the application.

The first was that it was very difficult to convince users to trust us with their investor data, and that anything we could do to build credibility (auto-filling form fields or leveraging existing brands, to name a couple examples) vastly increased willingness to share data.

The next big learning was that our users were very familiar with a spreadsheet-based experience (which is what they most commonly used before our tool existed), and that trying to replace it was difficult and unnecessary.

Finally, the most important takeaway from this iteration was the realization that founders do not find investors by looking at similar investors; they instead find investors by examining the previous investors of similar companies. This presented us with a nice property: similar users (founders/companies) were positive about similar products (investors). Our recommendation problem had now been reduced to the Netflix Prize [3] problem, opening up a large body of existing research.

We have begin work on the second version of this tool, which includes an augmented spreadsheet-like interface, and a recommendation set that is seeded by asking the founder to identify similar companies that are more established. We have also build infrastructure for crawling and ingesting the large databases of information available on websites like Crunchbase [12] and Angelist [14].

Future work will involve building out a new signup flow for the tool that better reflects the data needed to seed recommendations, additional progress on the recommendation algorithms and techniques, and efforts to de-duplicate and standardize the information crawled from various sources.

IV. Related Work

On the surface, it appears that there has not been much academic interest in combining computer science and venture capital. This is because most of the work done in leveraging data and machine learning to identify new opportunities and make superior investment decisions happens at venture firms where proprietary knowledge is a business advantage, and there is no incentive to share information. We therefore see very few papers published in the space, despite organizations like SignalFire [15], building what amounts to “a mini proprietary Google” [16] to aid in investment decisions and strategic portfolio support, and Correlation Ventures, which has built “one of the worlds most complete databases of venture capital financings” [17] for use in their predictive models.

There has also been significant work done in financial and economic academic explorations of venture capital that we can leverage. For example, we plan on borrowing several learnings from [18], including the list of sector names to use as binary features for a company and the calculated features for both investors and founders.

Another example are the various economic models summarized in [1], which include the problems of picking startups, matching founders to investors, and the interactions between venture firms and companies. While none of these consider the practical ways we can improve these processes through computer science, they provide a background for the challenges at hand, and present mathematical abstractions which may be useful in our models.

Finally, there have been several publications describing the difficulties accompanying the sparse, noisy data found in venture. Thomas Stone’s thesis on Computational Analytics for Venture Finance [19] delves into many of the problems with the publicly-available datasets. He proposes solutions to issues such as poor class labels, with supervised learning models that define a new, more granular schema, using existing schemas as input.

For the work remaining on the proposed tools, there has also been recent and relevant literature published.

FounderRank

We will be building on two bodies of work for the further exploration and implementation of FounderRank: document classification techniques for filtering and categorizing nodes in our graph, and node ranking algorithms, for surfacing the most relevant nodes in our final graphs.

Much of the work to be done on FounderRank requires shaping our web page crawl graphs to match the structure we expect. Much literature has been produced in the domain of text classification.

The first falls into the camp of extracting features from documents, and then feeding these feature vectors into standard models, such as Support Vector Machines (SVM) or a Naive Bayes classifier. The common feature vector is a bag-of-words model, and adding complex linguistic features to this model does not provide much benefit [20]. The time-tested term frequency-inverse document frequency (tf-idf) statistic is also often used in place of word counts [21]. Extrapolating on the word2vec methodology [22] of learning an embedding of word vectors, a team at Google has also proposed the “Paragraph Vector” [23], a distributed fixed-length feature representation of documents that automatically captures word ordering and semantics with recurrent neural networks (RNNs).

In the second camp are solutions that attempt to categorize documents in one shot, learning a classification directly from the text. These often employ convolutional neural nets (CNNs). One solution models documents in low-dimensional representations from hierarchical filters at the sentence and document level [24] (similar to Paragraph Vectors, but using CNNs to capture local context instead of the temporal context provided by RNNs). Another uses character-level convolution, treating the document as a series of raw signals and using convolutional filters to generate representative features [25]. In their case, these representations are fed into fully-connected predictive layers of a neural net which can output distributions over categories.

When it comes to ranking nodes in a graph, there is a large body of literature to reference. The canonical starting point is of course PageRank [9], which recursively estimates node importance by analyzing the importance of nodes which link to the node in question. HITS [26], a simple algorithm which calculates authority and hub scores, is also commonly used. Several variations exist, included a Weighted Page Rank [27]. While these algorithms do a good job ranking nodes in a graph for search relevance, they don’t necessarily capture desirable characteristics for interesting nodes in a venture context. It is unclear if there even exists a canonical ranking for nodes in a graph for venture, since desirable properties depend on the company.

Thus, we look at learned graph node ranking algorithms. Strategies such as the Graph Neural Network (GNN) [28] help neural nets directly process graphs, which has lead to the development of systems which use GNNs to rank web pages [29]. Crucially, this algorithm does not require explicitly determining which factors are important for ranking, and can be learned from a small number of training examples in the form of inequalities. Further work has then been done extending the idea of GNNs, with gating and other modern enhancements [30].

VCWiz

The work of Stone at UCL is the best starting point for related previous work in the area of recommendation systems for venture capital. In [5], the authors explore the difficult task of building a top-N recommendation system for venture firms considering investments - the inverse of the problem we are trying to solve. While not the same problem, Stone et al. discovered the difficulty in building a recommendation system with hyper-sparse data sets such as the set of venture fundings in the US, which is roughly the same data set we will be using (albeit from different sources). Their insight of leveraging both content-based and collaborative filtering, combined via a linear ensemble method, will be the inspiration for our hybrid classifier.

Current literature in recommendation systems defines (at least) two broad categories of systems: content-based and collaborative filtering [6]. Content-based systems characterize users with features extracted from the items they have preferences for, then use these features to find other items with similar features. Collaborative filtering, on the other hand, characterizes users by the set of preferences they have for a canonical set of items (without knowledge of the actual items), and suggests new items by finding users with similar preferences, and returning the items that they prefer. These two systems take different information into account, and can be combined into hybrid models that capture both perspectives.

Combining the models helps us mitigate some of the adverse effects of using one or the other. For example, content-based systems struggle to recommend items which are not associated with existing user items, since there are no similar features. Collaborative systems fix this by pulling items from similar users, agnostic of the item itself, but suffer from other issues, such as degraded results when users each only review a few items.

While there are other recommendation models to explore, it's accepted that "learning-based technologies work best for dedicated users who are willing to invest some time making their preferences known to the system" [6], which reflects our situation.

V. Future Work

A. FounderRank

Our proposal for the FounderRank project is to continue building a system which maps out a subset of the Internet as a graph, then filters and ranks the nodes to surface the most interesting nodes for a seed-stage venture fund.

The plan is to first finish building the infrastructure for crawling and scraping the nodes in question. Currently, we start at the entrepreneurship sites of academic institutions, and create one graph per institution, but we would like to explore alternative root nodes, and combined graphs.

Once we have crawled the subset of nodes we'd like to consider, the next challenge will be to adequately filter the graph, to remove noise and irrelevant nodes. We will continue our work website classification, leveraging the two methods described in the previous section. We will create baseline classifiers with a simple bag-of-words and Naive Bayes classifier, then evaluate the improvement rendered by distributed representations such as Paragraph Vectors, as well as one-shot classification attempts with convolutions.

The classifiers we anticipate needing are as follows, although there may be more required as the work progresses.

- a 4-way classifier indicating if a website represents a person, a company, a news site, or none of the above
- a binary classifier indicating if a person is a founder, given their website
- a binary classifier indicating if a company is a startup, given its website
- a binary classifier indicating if a company is an investor, given its website

Leveraging these classifiers, we can shape our graph into the following levels.

1) root nodes
2) founders and their startups
3) news and press sites and blogs

The second level will be the nodes we care about surfacing, while the third level will be (in part) where authority is derived from.

Once we have this graph, the goal will be to learn a ranking (per-graph) that successfully identifies interesting nodes to consider, where "interesting" is defined by the given examples. We will use inequalities between nodes as the training examples, which will be sourced from a combination of real world data from the venture firms we are working with, CrunchBase historic funding data, and some manual identification.

Our strategy for learning the ranking will be to learn one ranking function per graph, using the GNN-based method identified earlier. The benefit of learning this function is that, though the neural net structure changes when we add nodes to the graph, the shared weights stay the same, so our technique will work on previously-unseen nodes.

We will have to explore which features to use as the "label" of each node. This will require experimentation with features extracted from the page, and may also include some features derived from the graph (for example, we could include metrics of connectedness or centrality). Another path to explore for features are metrics derived from the news and press sites that reference the website in question, perhaps adapted from investor neighborhoods [18].

Finally, we will build an interface to easily surface these nodes and collect human feedback on the rankings.

B. VCWiz

Our proposal for VCWiz is first to finish building the second version of the public-facing tool, which we will launch to help us collect data. This tool will be a research, discovery, and outreach organization tool, where seed-stage founders can learn about the investors they are considering, get recommendations for new investors, and track their outreach (with email integrations for easy updating of the system).

Once the new tool is up and running, we will begin work on the sophisticated recommender system. Based on our research identified above, we will try out two models for recommendations, and then combine them into a hybrid model. In all cases, we will bootstrap the system by asking the founder to identify three companies similar to their own, which we will pull investor suggestions from.

The first model is based on collaborative filtering, leveraging founder's insights and the discovery that similar founders often prefer similar investors. We will determine the best way to extract a "rating" from the data we have collected, including the outreach tracking data users will enter into the tool. Bootstrapping will happen by simply suggesting the investors of the identified similar companies, using those choices alongside imported investor targets to build a user profile. The challenge with this model will be the extreme sparsity in the data, as identified earlier. However, this is where most of the insights from our novel data set will be incorporated.

The second model is a content-based filtering approach, leveraging our knowledge about investors and how to characterize them. Bootstrapping will happen by aggregating features of the identified similar companies (perhaps by averaging), building a target profile which is then updated as more data is collected. We will identify suggested investors via a k-Nearest Neighbors approach, as was detailed in [19].

In order to build a content-based model, we need to identify features to characterize both founders and investors. Many of the features we are considering can be automatically calculated, while a few will require user authorization or input.

The features we are considering for startups and founders are as follows.

- binary presence of industry tags (from CrunchBase or the founder)
- features extracted from the time-series data of investor outreach interactions (such as mean response time)
- binary *Job IPO*, *Job Acquired*, *Executive IPO*, *Executive Acquired*, *Advisory IPO*, and *Advisory Acquired* from [18] (all sourced from CrunchBase)
- *investor neighborhood*, *maximum IPO fraction*, and *maximum acquisition fraction* from [18]
- fraction of the leadership that had previously founded a company before the given company was founded [18]
- *number of companies affiliated*, *work overlap*, *education overlap*, *major overlap*, *from top school* [18]
- *major company similarity* [18], but taking into account the textual description of the startup

as well as the sector(s)
- mean *leadership age* [18]
- company age

The features we are considering for investors are as follows.

- binary presence of startup industry tags (based on known existing investments)
- binary industry preferences (explicitly published or from the Pitchbook data set)
- aggregations (e.g. means) of the startup features above, from the portfolio of the investor

We will consider using imputation methods such as Soft-Impute [31] to fill our feature matrices in the event of insufficient data.

Finally, we will combine the two models into a hybrid system. Methods to explore include the *weighted*, *mixed*, and *cascade* models from [6], as well as more complicated Bayesian methods [32] of model combination, if time permits.

VI. Evaluation

A. FounderRank

Evaluation for the filtering subsystems of FounderRank is simple; we will use common model accuracy metrics to evaluate the ability to categorize webpages verses some naive baseline (for example, looking for keywords), as well as a random model. Evaluation for the system as a whole is more difficult, and is something we are actively researching.

B. VCWiz

Evaluating the VCWiz system will be difficult, but we can leverage historic funding event data from sources like CrunchBase to evaluate the relevance of our system's recommendations. We will use the evaluation metrics described in [19], using a random model as a baseline, and a simple knowledge-based filtering system (given the industry of the startup in question) as a competitive alternative.

VII. Timeline

July 2017	**General**: Identify and enumerate opportunities for CS in VC **FounderRank**: Initial proof-of-concept system (web scraping and graph construction) **VCWiz**: N/A
August 2017	**General**: Write proposal **FounderRank**: N/A **VCWiz**: Import investors from all data sources, V1 of public-facing tool
September 2017	**General**: N/A **FounderRank**: N/A **VCWiz**: V2 of public-facing tool, public launch, V1 of recommendation engine
October 2017	**General**: N/A **FounderRank**: N/A **VCWiz**: Iterate on recommendation engine
November 2017	**General**: N/A **FounderRank**: Filtering models and V1 of ranking algorithm **VCWiz**: N/A
December 2017	**General**: N/A **FounderRank**: Iterate on ranking algorithm **VCWiz**: N/A
January 2018	Break
February 2018	**General**: N/A **FounderRank**: Build interface for surfacing results **VCWiz**: Evaluate and collect metrics
March 2018	**General**: Write Thesis **FounderRank**: N/A **VCWiz**: N/A
April 2018	Buffer

References

[1] M. D. Rin, T. F. Hellmann, and M. Puri, "A survey of venture capital research," National Bureau of Economic Research, Working Paper 17523, October 2011. [Online]. Available: http://www.nber.org/papers/w17523

[2] S. BERNSTEIN, X. GIROUD, and R. R. TOWNSEND, "The impact of venture capital monitoring," *The Journal of Finance*, vol. 71, no. 4, pp. 1591–1622, 2016. [Online]. Available: http://dx.doi.org/10.1111/jofi.12370

[3] J. Bennett, S. Lanning, and N. Netflix, "The netflix prize," in *In KDD Cup and Workshop in conjunction with KDD*, 2007.

[4] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, "SEISMIC: A self-exciting point process model for predicting tweet popularity," *CoRR*, vol. abs/1506.02594, 2015. [Online]. Available: http://arxiv.org/abs/1506.02594

[5] T. Stone, W. Zhang, and X. Zhao, "An empirical study of top-n recommendation for venture finance," in *Proceedings of the 22nd ACM international conference on Conference on information &#38; knowledge management*, ser. CIKM '13. New York, NY, USA: ACM, 2013, pp. 1865–1868. [Online]. Available: http://doi.acm.org/10.1145/2505515.2507882

[6] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, Nov 2002. [Online]. Available: https://doi.org/10.1023/A:1021240730564

[7] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, “Text Summarization Techniques: A Brief Survey,” *ArXiv e-prints*, Jul. 2017.

[8] T. Zaman, E. B. Fox, and E. T. Bradlow, “A bayesian approach for predicting the popularity of tweets,” *Ann. Appl. Stat.*, vol. 8, no. 3, pp. 1583–1611, 09 2014. [Online]. Available: http://dx.doi.org/10.1214/14-AOAS741

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Tech. Rep., 1999.

[10] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug. 2008, pp. 11–15.

[11] “Alexa internet,” 2017. [Online]. Available: http://www.alexa.com/

[12] “Crunchbase,” 2017. [Online]. Available: http://crunchbase.com/

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[14] “Angellist,” 2017. [Online]. Available: https://angel.co/

[15] “Signalfire,” 2017. [Online]. Available: http://www.signalfire.com/

[16] C. Loizos. (2015, October) Watch out, vcs: Chris farmer plans to massively disrupt the industry. [Online]. Available: https://techcrunch.com/2015/10/22/watch-out-vcs-chris-farmer-says-hes-about-to-massively-disrupt-the-industry/

[17] “Correlation ventures - our approach - about us,” 2017. [Online]. Available: http://correlationvc.com/approach/about

[18] D. S. Hunter and T. Zaman, “Picking Winners: A Framework For Venture Capital Investment,” *ArXiv e-prints*, Jun. 2017.

[19] T. R. Stone, “Computational analytics for venture finance,” Ph.D. dissertation, UCL (University College London), 2014.

[20] A. Moschitti and R. Basili, *Complex Linguistic Features for Text Classification: A Comprehensive Study*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 181–196. [Online]. Available: https://doi.org/10.1007/978-3-540-24752-4_14

[21] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[23] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[24] M. Denil, A. Demiraj, N. Kalchbrenner, P. Blunsom, and N. de Freitas, “Modelling, visualising and summarising documents with a single convolutional neural network,” *CoRR*, vol. abs/1406.3830, 2014. [Online]. Available: http://arxiv.org/abs/1406.3830

[25] X. Zhang, J. J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” *CoRR*, vol. abs/1509.01626, 2015. [Online]. Available: http://arxiv.org/abs/1509.01626

[26] J. M. Kleinberg, “Authoritative sources in a hyperlinked environment,” *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[27] W. Xing and A. Ghorbani, “Weighted pagerank algorithm,” in *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*. IEEE, 2004, pp. 305–314.

[28] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[29] F. Scarselli, S. L. Yong, M. Gori, M. Hagenbuchner, A. C. Tsoi, and M. Maggini, “Graph neural networks for ranking web pages,” in *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*. IEEE, 2005, pp. 666–672.

[30] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, “Gated graph sequence neural networks,” *CoRR*, vol. abs/1511.05493, 2015. [Online]. Available: http://arxiv.org/abs/1511.05493

[31] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral regularization algorithms for learning large incomplete matrices,” *Journal of machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010.

[32] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, “Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks,” *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 785 – 799, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888613X10000460