Capital Crunch: Predicting Investments in Tech Companies

Zifei Shan, Haowen Cao and Qianying Lin
Department of Computer Science, Stanford University
{zifei,caohw,qlin1}@stanford.edu

Abstract

For many start-ups, lack of investment and capital has become the bottleneck for development. This phenomenon inspires us to use machine learning algorithms to find patterns in investment behavior from major investors. We propose to use various domain-specific features to predict which investors would potentially invest in a particular company. This would not only reveal important information about investment strategies and behaviors of investors, but also give startups ideas on where to seek potential investment and how to adjust their strategies so as to attract potential investors.

Our work is grounded in CrunchBase, an accessible knowledge base that maintains full records of company and people information.

There are two primary goals of our work:

(1) To predict whether an investor would invest in a particular start-up based on textual, topological and domain-specific signals from both the investor and start-up.

(2) To analyze and reveal the factors that would prompt an investor to invest in startups so as to shed light on the adjustments the start-ups could make to attract more investments.

1 Dataset

We downloaded all the data from *CrunchBase.com*, one of the biggest databases with company information.

1.1 Accessing data

CrunchBase provides indexing data and an API for full access of their data. We requested the CrunchBase education plan API, which gives us a much higher throughput when acquiring data, and we were able to get the full dataset.

The full CrunchBase dataset includes 214,290 companies and 286,659 people. We also parsed all known investments in CrunchBase, and got 31,942 investments.

2 Approach

2.1 Data Model

The CrunchBase dataset has a variety of entities: organization, person, product, etc. There are also different relations including investment, acquisition, degree, founder, etc.

Our currently focus is on investment relationships. For simplification, we categorize organizations into **startups** and **investors**, and we care about predicting **investment** relationship between them.

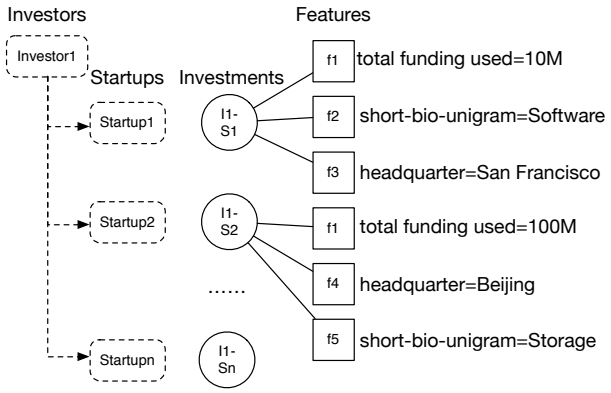

Figure 1: Logistic Regression model (for a single investor)

The data model is defined below:

- $Startup(startupId, [attributes...])$
- $Investor(investorId, [attributes...])$
- $Investment(investorId, startupId, isTrue)$

Where we use attributes (features) in *Startup* and *Investor* entities to predict *Investment* relations.

2.2 Problem definition

Our formal problem is:

Definition 1 *Problem: given the full **Startup** relation and **Investor** relation, predict **isTrue** value in **Investment** table, which determines if any given investor invests a startup.*

The desired output is a predicted probability between each investor and a startup.

2.3 Algorithm

In our initial model, we train an independent logistic regressor for each individual investor, which takes a feature vector of a start- up and predicts a label.

The logistic regression model is a subcase of a factor graph model: each *Investment* relation is a boolean variable that we are predicting, and the features are unary factors applied to variables. The factor graph for logistic regression is demonstrated in Figure 1. Each circle is a variable and each square is a unary factor. Note that the figure only represents the factor graph for a certain investor. For each investor we train a model like this, and their factor graphs are disconnected.

The drawback of this model is that it cannot utilize investor-based attributes and higher-level knowledge such as network topology.

As an improvement, we propose a factor graph / CRF model that correlates features across this graph. In a factor graph, and features in both sides of investors and startups can be correlated.

Figure 2 presents a possible design of a factor graph model. In this design, we add a binary factor $f3$ to connect factor graphs:

A factor $Equal(I_1S_1, I_2S_2)$ is applied if I_1 and I_2 has a common attribute a_i, and S_1 and S_2 has a common attribute a_s, and the weight (coefficient) is determined by (a_i, a_s). Intuitively, **investors that have similar interest would prefer to invest in similar startups,** and the degree is determined by the specific attributes.

We use DeepDive [4], a highly scalable inference engine to tackle the problem. L-1 regularization is applied during weight learning, and Gibbs Sampling is used for inference.

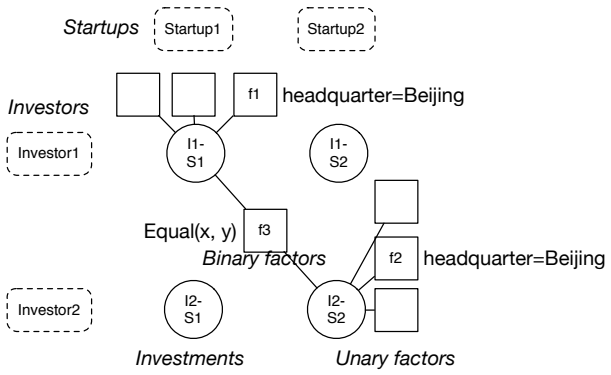

Figure 2: Factor graph model that captures similarity

2.4 Features

A rich set of features can be applied to predict investments. A list of features are described as below:

Start-up attributes:

- Unigrams (words) of short description of the start-up, with stopwords removed.
- Total funding used: we think the more funding the startup has used, the more promising it is and the more likely it will get another funding raise.
- Founded year: it is used combining with the total funding rounds. We use this to generate the negative example.
- Current team size: the size of founders in the start-up. We think the larger it is, the more likely it will be invested.
- Number of Competitors: we think the more competitors the start-up has, the less likely it will get the investment.
- Headquarters: the location of headquarters of the startup. We find that some specific VC likes to invest the startups in some specific areas.
- Category: the category of the service of the startup. We know that some VC are fond of the startups running on some certain kinds of service.
- Total items of websites: we think that the more official websites the startup, the startup are more likely to get the investment since websites are indicative of the marketing potential.

Note that in a logistic regression model, each investor will have different weights (coefficients) for these different attributes. For example, investor *A* may prefer start-ups with headquarter in Beijing, *B* may prefer start-ups with larger team size, *C* will prefer start-ups in the category of artificial intelligence, etc.

Investor attributes:

- Total number of investments: if this investor has already invested more items, the more likely it gives a start-up investment under the same situation
- Total number of acquisitions: it is similar to the total investment items.
- Headquarters: the location of headquarters of the investor. It is similar to the feature of the startup's headquarter.
- Category: the category of the service of the investor. It is similar to the feature of the startup's category.

Note these features are not used in training independent logistic regressors, but they are useful in the full factor graph model.

2.5 Getting training data

To train the predictor, we take ground truth investments in CrunchBase as positive training examples, that is, if an investor I has invested in a startup S, we obtain a training example $(I, S, true)$ in *Investment* relation.

For the negative training examples, it might not be desirable to simply label all pairs of $(investor, startup)$ that do not have a known investment as negative, because (1) this makes positive examples extremely sparse and introduce a data skew, (2) this yields too many training examples and makes training harder, and (3) even if an investor have not invested a startup right now, it is still possible that the investment will happen in the future.

For now we sub-sample random pairs of investors and startups without known investment happening as negative examples. In future work, we will apply some heuristics to label negative examples. The proposed method is described below.

2.5.1 Proposed heuristics for negative example generation

If a certain investor does not invest a startup, we cannot simply label it as a negative example since there is a chance that this investor will invest this startup in the future. Besides, there are competitions between investors as the amount of investment for a start-up is limited. If investor A has made the investment for startup X but investor B has not, we cannot say (B, X) is a negative example because B may have the intention to invest X but A's investment prevents it happening. Furthermore, some investors may not be early enough to know the company but they actually have huge interest in the start-up.

So we propose a model to help us generate the negative examples. For all the startups, we have already known their founded year and funding rounds. We believe that the longer this start-up exists and the less funding rounds it has, the less likely it has the investment from other investors and vice versa. Therefore we use this equation to give startup X a probability to have negative examples:

$$P(X = N) = e^{-K}, K = \frac{\textbf{funding rounds} + 1}{\textbf{running time} + 1}$$

Intuitively, if the start-up has already run for a long time, it should have more funding rounds. Otherwise, it will have a high probability to get negative answers from investors. In contrast, if the start-up only runs for a short time but has already got many rounds of funding, it is very likely to get positive answers from other investors. We add one to the numerator and denominator to smooth it and also prevent illegal equation.

3 Evaluation

3.1 Evaluation metrics

We evaluate our models based on ground truth investments (positive examples) as well as randomly / heuristically labeled negative examples. Specifically, we hold out a fraction of training examples, and use the predictions on these relations for evaluation.

We examine the calibration plots, where we layout the predicted probabilities and the accuracy in test set in buckets. A discussion of interpreting calibration plots is in the paper [4].

We look at the accuracy of most confident predictions: we hope to get a trained system where the most confident predictions are very likely to be true. Specifically, we evaluate the precision where the system predicts a probability above 0.95, as well as the recall.

We also evaluate the recall of predictions above probability 0.5, as well as the recall.

We compare our results with a naive baseline model: a random predictor that predict random labels based on class priors. An oracle model would be one that knows all existing investments and give correct predictions, which have precision and recall of 1.0.

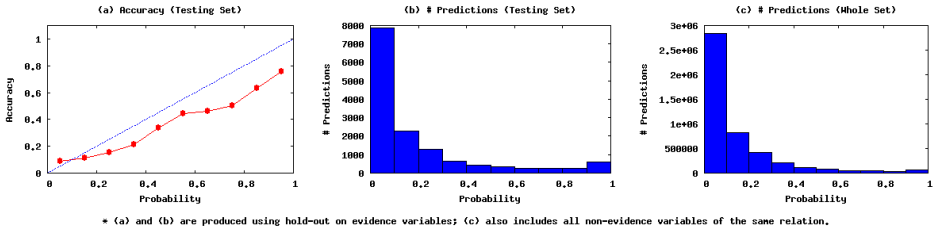

Figure 3: Calibration plot for investment predictions

3.2 Initial Results

We run experiments on a filtered dataset where each investor have at least 5 investments. We randomly label investor-startup pairs that do not have known investments as negative examples with probability of 0.01. This yields a dataset of 9,521 positive examples, and 47,490 negative examples.

Factor graph size:

- 4.7 million variables
- 115 million factors
- 3.1 million different weights

Calibration plot: See Figure 3.

High confidence (probability > 0.95 predictions):

- Precision 81.22% (over labeled data)
- Precision 0.91% (if count all predicted unlabeled data as false positives)
- Recall 3.13% (over all 2460 held-out positive examples)

Half confidence (probability > 0.5 predictions):

- Precision 59.34% (over labeled data)
- Precision 0.37% (if count all predicted unlabeled data as false positives)
- Recall 40.7% (over all 2460 held-out positive examples)

3.3 Initial error analysis

The results are not optimal. Specifically, we see severe underfitting from the calibration plot. Possible reasons include inexpressive features or models, or bad labeling strategy. We will discuss improvements in future work.

Table 1 shows some high-confidence examples with features. A view like this is useful for future error analysis.

3.4 Result analysis

Here is a list of top indicative features for the investor *Sequoia Capital*:

- headquarter=Shanghai
- headquarter=Beijing
- num_websites==5
- num_competitors==2
- category=Web Hosting
- founded_on_year=2005
- short_bio_1gram=services
- num_websites==4

investor_id	startup_id	is_true	expectation	features
sequoia-capital	influitive	f	0.982	category=Communities, category=Marketing Automation,...
new-enterprise-associates	liquidia-technologies	t	0.988	category=Nanotechnology, founded_on_year=2004,...
intel-capital	mobixell	t	0.998	category=Mobile, founded_on_year=2000,...
index-ventures	abes-market	t	0.98	category=E-Commerce, founded_on_year=2009,...
y-combinator	homejoy	t	0.982	category=Hospitality, founded_on_year=2012, headquarter=San Francisco,...
high-tech-gruenderfonds	fraud-sciences	f	0.99	category=Security, headquarter=Palo Alto,...
insight-venture-partners	twitter	t	0.996	category=Messaging, category=MicroBlogging,...
amicus-capital	getable	t	0.992	category=Curated Web, founded_on_year=2010,...

Table 1: Examples of predicted relations

- headquarter=Houston
- num_competitors==10
- founded_on_year=2004
- category=Databases
- headquarter=Pune
- category=Technology

We see some interesting results in these features: this investor prefers Asia startups, in field of web hosting, services, databases and technology.

Analysis like this might help startups find ideal investors, making marketing strategy and technical decisions.

4 Future Work

We clearly see several directions of improvements towards a better predictor:

(1) Improve features. We will add textual features in company descriptions, rather than using simple unigrams of short-bio. We have already parsed all the sentences in company descriptions using Stanford CoreNLP. We have also downloaded information about startup founders, which might help improving features.

(2) Improve models. We will add the similarity CRF rule discussed above, and propose more expressive rules such as people-related rules, or propose HITS/PageRank-like models (good investors invest in good startups, and vise versa), whose idea is similar with a previous work discussed in [3].

(3) improve supervision methods. We should integrate the heuristics of generating negative examples discussed above. Further questions include: how many negative examples to generate? How to do proper train/test split? Specifically, we might hold out start-ups, rather than single investment edges.

5 Related Work

In the paper [1], the author discussed a methodology to match proposals from start-ups to the potential investors on Kickstarter with linear regression, SVM-linear, SVM-poly and SVM-RBF, with an accuracy rate of 82% for static data features and 73% for dynamic data features. Their features are mostly updates made to the tweets, number of comments and so on, and we could widely expand the feature set.

Another paper [2] used a discriminant analysis to classify the potentially successful and unsuccessful companies. Their feature sets are worth noting, including individual characteristics of the entrepreneurs, the efforts by entrepreneurs (i.e. whether they actively look for resources and help), degree of innovation and so on. Though this paper is more on the social science side, we would like to scrutinize the feature sets so as to explore more meaningful and insightful features. For example, we could extend individual characteristics to how many start-ups the CEO has founded and their histories.

References

[1] J. An, D. Quercia, and J. Crowcroft. Recommending investors for crowdfunding projects. In *Proceedings of the 23rd international conference on World wide web*, pages 261–270. International World Wide Web Conferences Steering Committee, 2014.

[2] W. Gartner, J. Starr, and S. Bhat. Predicting new venture survival: an analysis of anatomy of a start-up. cases from inc. magazine. *Journal of Business Venturing*, 14(2):215–232, 1999.

[3] Z. Shan, S. Li, and Y. Dai. Gamerank: Ranking and analyzing baseball network. In *Social Informatics (SocialInformatics), 2012 International Conference on*, pages 244–251. IEEE, 2012.

[4] C. Zhang, C. Ré, A. A. Sadeghian, Z. Shan, J. Shin, F. Wang, and S. Wu. Feature engineering for knowledge base construction. *arXiv preprint arXiv:1407.6439*, 2014.