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 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

Our dataset is from *CrunchBase.com*, an accessible knowledge base that maintains up-to-date information of companies.

CrunchBase provides indexing data and an API for full access of their data to get the full dataset of companies, people and investments.

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

The data model is defined below:

- $Startup(startupId, [attributes...])$

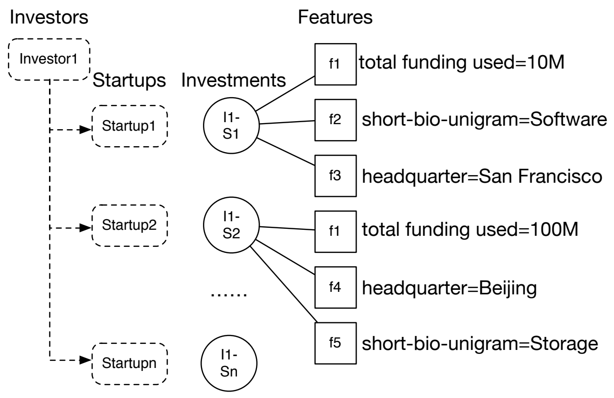

Figure 1: Logistic Regression model (for a single investor)

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

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 first 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.

One drawback of this model is that it cannot utilize investor-based attributes and higher-level knowledge such as network topology. As an improvement, we implement a factor graph / CRF model that correlates features across this graph. In a factor graph, features in both sides of investors and startups can be correlated.

Figure 2 presents our implemented factor graph model. In this design, we add a binary factor $f3$ to connect investment variables:

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-2 regularization** is applied during weight learning, and Gibbs Sampling is used for inference. In our evaluation, we also tried L-1 regularization, and made a comparison between these different regularization methods.

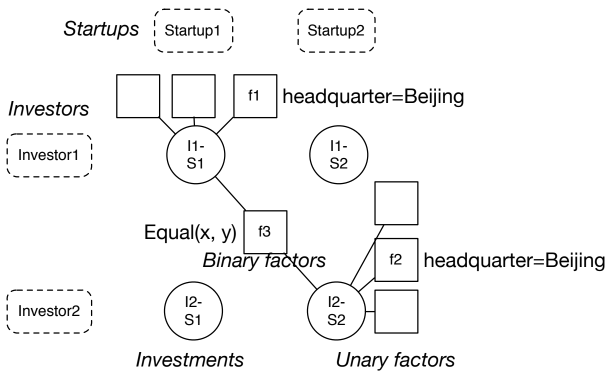

Figure 2: Factor graph model that captures similarity

2.4 Features

A rich set of features can be applied to predict investments. We group our features into basic startup attributes, people attributes, and linguistic attributes. They are described in the lists below.

Basic attributes:

- 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.
- Founded year: a VC may make more investments in certain years than others.
- Number of competitors: in a competitive area, it might be harder to get investments, where in a newer area without many competitors, it might be easier to get investments.
- Number of websites: we think that the more official websites the startup has, the more likely it will get investments since websites are indicative of the marketing potential.
- Team size: Some investors may prefer to invest in a larger startup while some may prefer smaller startups, which is characterized by the startup team size.

People attributes

Investors may evaluate the founders and CEOs before making an investment, therefore we think information of these people might be important features for predicting investments. The following features are extracted for each founder or CEO of startups:

- Names.
- University of graduation: some investors may have certain preference of people from some specific universities.
- Has obtained MBA: investors may have preference to people who has MBA degree.
- Number of degrees obtained.
- Company worked in: this helps the investors to evaluate the working experience of founders.

Linguistic attributes

We try to utilize the linguistic attributes of startups to help improve the prediction accuracy. These linguistic features are extracted from the description of the startups. These descriptions may reveal important aspects for startups, such as what problems they are trying to solve, what technologies they are using, etc.

We use Stanford CoreNLP to extract the following features from the description:

- Unigrams of lemmatized nouns. Nouns carry important concepts that reveals specifications of the startup, such as used technologies and marketing strategies.

Category	Positive Example	Negative Example
Total	7749	43207
Training	5831	32462
Test	1918	10745

Table 1: Dataset Statistics

- Location phrases. We extracted all phrases of location, such as countries or cities. This feature may indicate important location information of startups, such as offices or targeted areas, and it contains more information than "headquarters" in basic attributes.

2.5 Getting Labeled Data

2.5.1 Positive Examples

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.

2.5.2 Negative Examples

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.

So we take startups that satisfies both following conditions

1. Have been founded more than 6 months.
2. Have not been invested or acquired.

For each startup S among these, we randomly generate edges with known investors in I, to obtain negative examples $(I, S, false)$. The intuition is that we think any investor is not likely to be interested in startups that have not got any investment after founded for a long time.

2.5.3 Train / Test Split

We randomly hold out investment edges for 25% startups from all labeled data as test set. Table 1 shows statistics for the training set and test set.

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 mentioned above. Specifically, we hold out a fraction of training examples, and use the predictions on these relations for evaluation.

We compute precision, recall and F1 score on the test set, for different models and feature combinations.

Since our models give predicted probabilities to each possible investment pair, we are able to choose a value as the *decision boundary*, where we predict true for all investments that has predicted probability above this value. For example, if we choose 0.9 as decision boundary, then predicted investments will be only ones that the system is very confident of. Since we want to balance precision and recall, we choose decision boundaries that maximizes F1 score.

Feature combinations: we cluster all features into basic / people / linguistic, and try different combination of features.

Model	Features	Decision bounary	Precision	Recall	F1 score	F1 on training set
Baseline	N/A	N/A	0.151	1.000	0.263	0.263
LR	basic-only	0.6	0.856	0.480	0.615	0.810
	people-only	0.6	0.753	0.218	0.338	0.482
	nlp-only	0.8	0.896	0.409	0.562	0.867
	no-basic	0.6	0.788	0.525	0.630	0.941
	no-people	0.7	0.880	0.591	0.707	0.994
	no-nlp	0.6	0.852	0.504	0.633	0.833
	all features	0.7	0.889	0.585	0.706	0.999
CRF	all features	0.9	0.495	0.527	0.510	0.999
Oracle	all + funding	0.3	0.864	0.894	0.879	0.999

Table 2: Evaluation Results

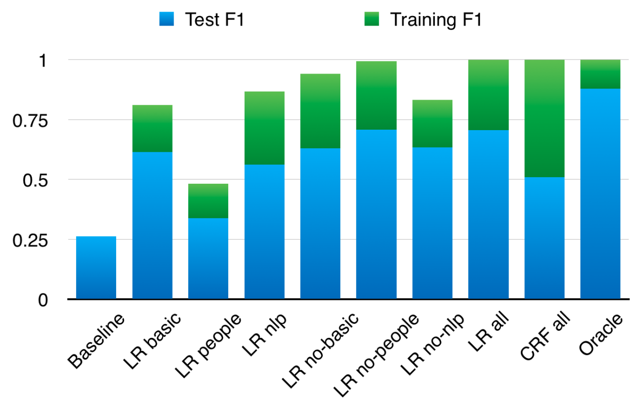

Figure 3: Gaps between training set and test set F1 scores

3.2 Baseline and Oracle

To understand the quality of predictions, we compare our results to a baseline and an oracle model.

Baseline: simply predicting all true for every test example.

Oracle: The oracle is Logistic Regression with all features plus additional features: *the number of funding rounds, and the total amount of funding*. This oracle is used to understand how well the model can perform when it "cheats" with much more informative information. The additional features are directly indicative of whether a startup has been invested, and is not available for investors before investments happen.

We also 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].

3.3 Results

Table 3.3 shows the results for different models and feature combinations. LR with best features has F1 score of 0.707. The baseline has F1 score of 0.263, and all our models has significant improvement than the baseline. The oracle has F1 score of 0.879.

We further visualize the gap of training test set, in Figure 3. We see that there are quite a large gap between training set and test set F1 scores, for most models and features. Even on our best feature combination (no-people), the F1 score on training set is 0.287 higher than that on test set.

This indicates that our models has notable overfitting, therefore we may want to do feature selection, or use better features. It also indicates that investments happen pretty randomly, and it is generally hard to use shallow linguistic indicators inside a small dataset to predict them.

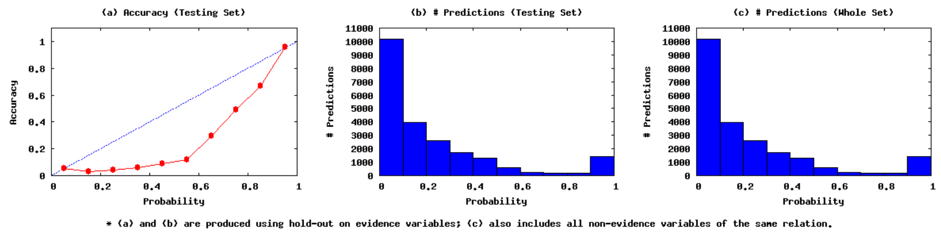

Figure 4: Calibration plot for investment predictions

Calibration plot is shown in Figure 4, which also indicates overfitting since the predicted probabilities are not well calibrated.

3.4 Overall analysis

Among results of different feature combinations, we see that the features that works best are basic attributes and linguistic attributes.

In basic attributes, *headquarter* and *category* get the highest weights. In linguistic features, *lemmatized nouns* in description get the highest weights.

People attributes turn out to be bad features, which are mostly because most startups do not have people features: among 46971 startups that we predict for, only 5776 startups have people features.

CRF does not work well: CRF model does not generalize to test. Severe overfitting happens in CRF model, which indicates that the underlying assumptions may not be valid.

These results can be indicative on what factors are important in investments.

3.5 Feature analysis by investor

We conduct feature analysis for a specific investor, Sequoia Capital. Table 3 shows some top positive features for Sequoia Capital.Table 4 shows top negative features for Sequoia Capital. By this analysis we can understand preferences for this investor, as well as understand how overfitting happens in the current model.

For example, headquarter and location phases are quite useful for predictions, however some linguistic (NLP) features are leading to overfitting, such as “friend”, “s”. We can see that lots of negative features do not make sense, which indicates that we should improve negative example generation to get better results — the current negative examples are too “far away from the decision boundary”.

We also 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.

3.6 Experiment with L-1 Regularization

Since there are overfitting, we try to use L-1 regularization withto do feature selection. However L-1 regularization did not improve the results. See Table 5, experiments results for Logistic Regression with “no-people” feature combination. This indicates that feature selection might be generally hard in this problem, since the investment behaviors are pretty random and there are hardly any expressive features that always work. To improve this, we may need to dig into deeper features, which may even not exist on CrunchBase.

4 Future Work

There are some future directions of improvements:

Type	Feature	Weight
basic	headquarter=San Francisco	2.18276
nlp	noun-1gram=Jive	1.49204
nlp	location=China	1.4918
basic	num_competitors==10	1.44375
nlp	noun-1gram=management	1.25531
basic	headquarter=Fremont	1.09784
basic	headquarter=Beijing	1.07851
basic	founded_on_year=2005	1.05513
basic	headquarter=Mountain View	1.03978
basic	founded_on_year=2006	1.01425
basic	category=Hardware + Software	0.937676
nlp	noun-1gram=weather	0.932523
nlp	noun-1gram=friend	0.91767
nlp	noun-1gram=s	0.907115
basic	num_competitors==1	0.894676
nlp	noun-1gram=Sequoia	0.862655
basic	category=Software	0.862347
nlp	noun-1gram=enterprise	0.855152
nlp	noun-1gram=advertisement	0.836432
nlp	noun-1gram=Capital	0.801734

Table 3: Top positive features for Sequoia Capital

Type	Feature	Weight
nlp	noun-1gram=bitcoin	-1.569
basic	founded_on_year=2013	-1.23817
nlp	noun-1gram=Bitoomba	-1.17675
basic	headquarter=Fountain Valley	-1.06557
basic	headquarter=Zug	-0.928788
nlp	noun-1gram=billing	-0.904699
basic	category=Real Estate	-0.893038
nlp	noun-1gram=need	-0.877832
basic	category=Travel	-0.851232
nlp	noun-1gram=design	-0.822234
nlp	noun-1gram=country	-0.816831
basic	headquarter=Berlin	-0.774059
nlp	noun-1gram=subscription	-0.748185
nlp	noun-1gram=world	-0.73763
nlp	noun-1gram=Daum	-0.735467
nlp	noun-1gram=LLC	-0.725189
basic	num_websites==3	-0.724794
nlp	noun-1gram=sport	-0.715075
nlp	noun-1gram=link	-0.708986
nlp	noun-1gram=place	-0.705011

Table 4: Top negative features for Sequoia Capital

(1) Improve features. As we mentioned above, we need to dig into more indicative features to reduce overfitting.

(2) Improve models. Although the similarity CRF rule discussed above did not work well, there are other possible CRF rules such as people-related rules, or HITS/PageRank-like models.

(3) Improve supervision methods. The current negative exmaple generation are too weak and we may need better examples closer to the decision boundary.

We also see other interesting topics, such as **ranking investors and startups** based on investment relationships. Intuitively, good investors invest in good startups, and good startups are invested by

Paramater	Decision boundary	Precision	Recall	F1 score
0.1	0.6	0.592	0.219	0.320
1	0.6	0.591	0.222	0.323
10	0.6	0.587	0.217	0.317

Table 5: Results with L-1 regularization

good investors. There is a similar work in baseball, [3], and it would be interesting to see how the algorithm generalize here.

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.