Capital Crunch: Predicting Investments in Tech Companies

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

Introduction

Motivation: find patterns in investment behavior from major investors and successful startup strategies.

Problem: predict whether an investor would potentially invest in a startup.

Contribution:
- Understanding investment strategies and behaviors of investors
- Give startups ideas on where to seek potential investment and how to attract potential investors.

Data Model

Data source: CrunchBase
- Entities: Organization, person, product…
- Relations: investment, acquisition, founder…

Data Processing
- Categorize organizations to start-ups and investors

Data Model
- *Startup(startupId, [attributes…])*
- *Investor(investorId, [attributes…])*
- *Investment(investorId, startupId, isTrue)*

Getting Labeled Data

Positive Examples
Use ground truth investments in CrunchBase:
- if an investor *I* has invested in a startup *S*, we obtain a training example *(I, S, true)* in *Investment* relation.

Negative Examples
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, randomly generate edges with known investors in *I*, to obtain negative examples *(I, S, false)*.

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

Example	Positive example	Negative example
Total	7749	43207
Training	5831	32462
Testing	1918	10745

Table 1: Dataset statistics

Features

Basic attributes
- Headquarter (e.g. SF)
- Category (e.g. software)
- Founded year
- Number of Competitors
- Number of websites

People attributes
For founders and CEOs:
- Names
- University of graduation
- Company worked in
- Has obtained MBA
- #degrees obtained

Linguistic attributes
NLP features from description of start-ups:
- Location phrases
- Unigram of lemmatized nouns

Feature Analysis: Table 2 shows the most indicative features of investor *Sequoia Capital,* according to learned weights from our system.

Top Positive Features	Top Negative Features
location=China	noun-1gram=VitaCig
headquarter=San Francisco	num_websites=2
headquarter=Beijing	founded_on_year=2014

Table 2: Top features for Sequoia Capital

Algorithms

Logistic Regression (LR) model: we train an independent logistic regressor for each investor, which takes a feature vector of a start-up and predicts a label.
- This model cannot utilize investor-based attributes.

Factor graph (CRF) model: we introduce a binary factor to utilize attributes of investors.
- A factor *Equal*$(I_1 S_1, I_2 S_2)$ 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.

Annotations in Figure 1 / 2:
- Circle: variables. Square: factors
- Each *Investment* relation is a boolean variable, and the features are unary factors.
- Note that Figure 1 only represents the factor graph for *one* investor.

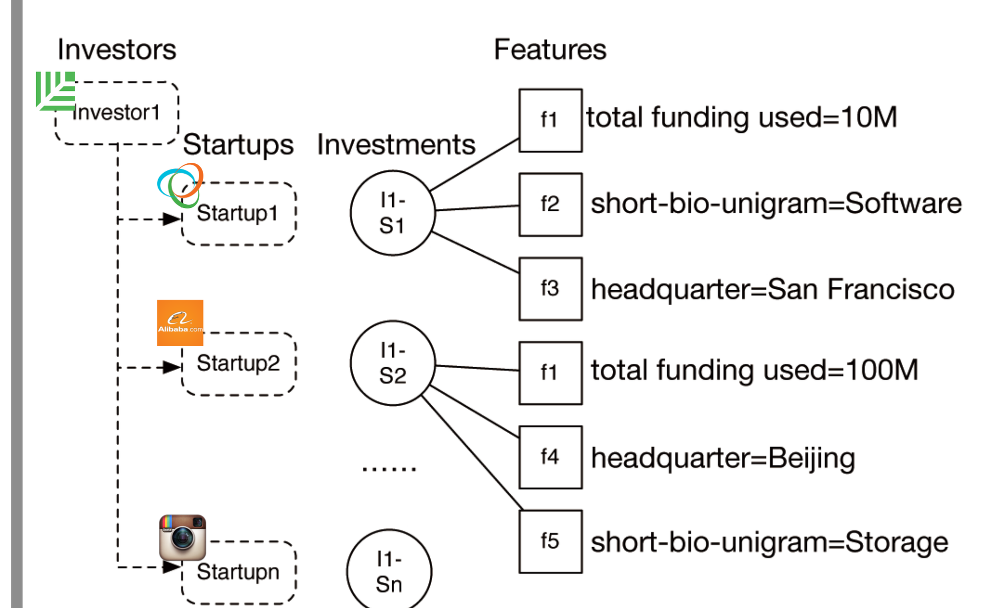

Figure 1: Logistic regression model (for one investor)

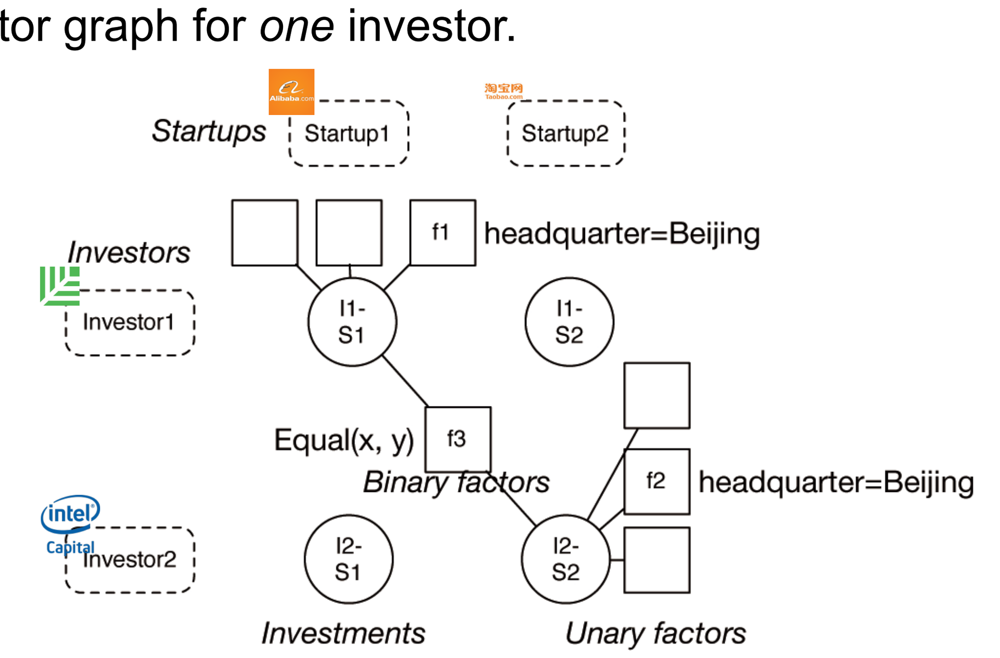

Figure 2: Factor graph model that captures similarity

Evaluation

For evaluation, we compute precision, recall and F1 score on the test set, for different models and feature combinations. We choose decision boundaries to optimize F1.

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

Baseline: simply predicting all true for every test example. Has F1 of 0.263.

Oracle: Logistic Regression with all features plus information about *number of funding rounds* and *total funding raised*. Has F1 of 0.879.
- These features in oracle are directly indicative of whether a startup has been invested, and will not be usable in real cases.

Results: Table 3 shows the results for different models and feature combinations. LR with best features has F1 0.707, much better than baseline 0.263, close to oracle 0.879.
- **Good features:** basic attributes and linguistic attributes.
  Especially: Headquarter, category, lemmatized nouns in description.
- **Bad features:** people attributes
- **CRF does not work well:** CRF model does not generalize to test set, possibly because of overfitting, or the underlying assumptions is not valid.

Model	Features	Decision bounary	Precision	Recall	F1
Baseline	N/A	N/A	0.151	1.000	**0.263**
LR	basic-only	0.6	0.856	0.480	0.615
	people-only	0.6	0.753	0.218	0.338
	nlp-only	0.8	0.896	0.409	0.562
	no-basic	0.6	0.788	0.525	0.630
	no-people	0.7	0.880	0.591	**0.707**
	no-nlp	0.6	0.852	0.504	0.633
	all features	0.7	0.889	0.585	0.706
CRF	all features	0.9	0.495	0.527	0.510
Oracle	all + funding_rounds +total_funding	0.3	0.864	0.894	**0.879**

Table 3: Results for different features applied to the model

Figure 3 shows the calibration plot of the best feature selection (no-people).
- Most predictions has very low / very high probabilities.
- Confident predictions are reliable: error rate is 5.5% when predicted probability < 0.1, 4.0% when predicted probability > 0.9

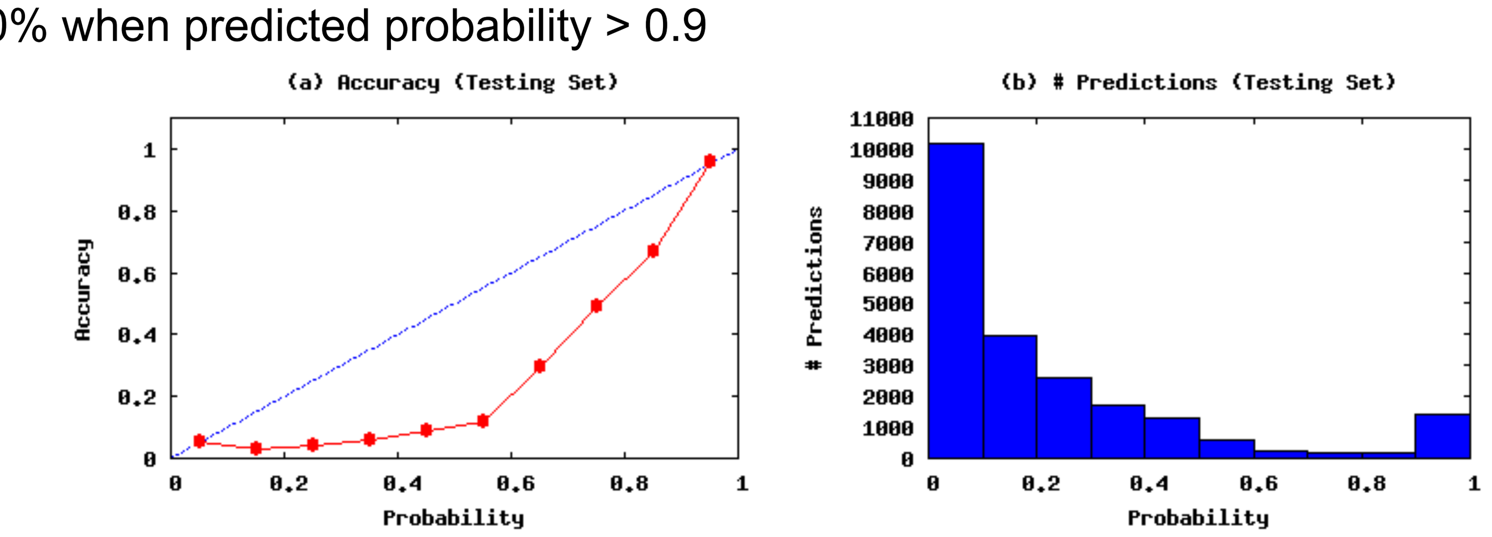

Figure 3: Calibration plot for the best feature selection

Stanford University