# Exploiting Social Relations and Sentiment for Stock Prediction


*Jianfeng Si[*], Arjun Mukherjee[†], Bing Liu[†], Sinno Jialin Pan[*], Qing Li[‡], Huayi Li[†]*

** Institute for Infocomm Research, Singapore*

*†Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA*

*‡Department of Computer Science, City University of Hong Kong*



## Abstract

- The Web has seen a tremendous rise in social media.
- Information in social media text (e.g., Twitter, Facebook ) not only contains Opinion, but also Relations.
- The goal of this paper is to exploit social relations and social sentiment for stock market prediction.
- We build a Semantic Stock Network (SSN) from the co-occurrence statistics of cash-tags in Twitter messages. This SSN summarized discussion topics about stocks and stock relations.
- Experimental results demonstrate that topic sentiments from close neighbours are able to help improve the prediction of a stock markedly.


## Key Tasks

- Data collection.
- Build the stock network.
- Derive the topics over nodes and edges.
- Regress stock price with sentiment time-series derived from the network in an autoregressive framework for market prediction.

## Data Collection

- Collected streaming tweets using Twitter's REST API.
- Query keywords: ticker symbols from S&P100 stocks.
  - $APPL, $GOOG, $AMZN, $MSFT…
  - ""$AAPL is loosing customers. everybody is buying android phones! $GOOG."

## Tweets in Relation to the Stock Market

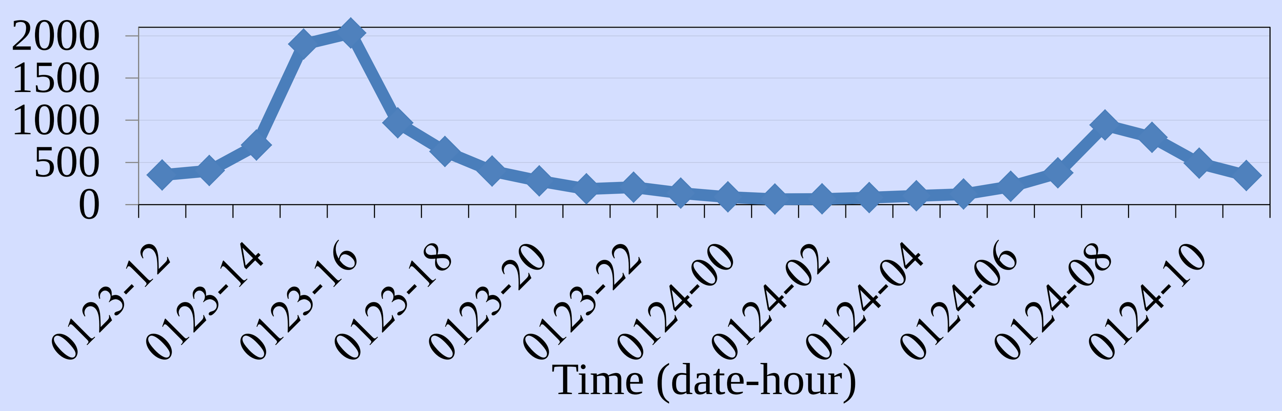


Figure 1. Tweet Activity around $aapl's earnings report date on Jan.23 2013.

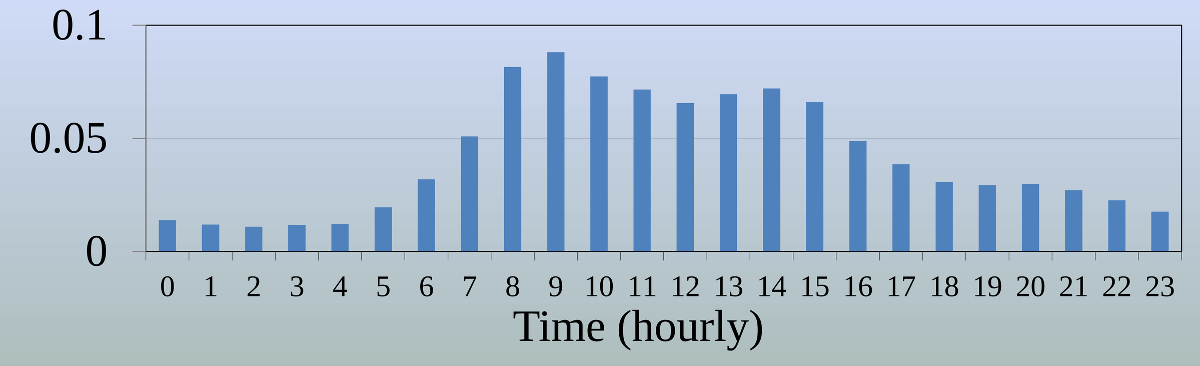


Figure 2. Tweet volume distribution in our data over hours averaged across each day.

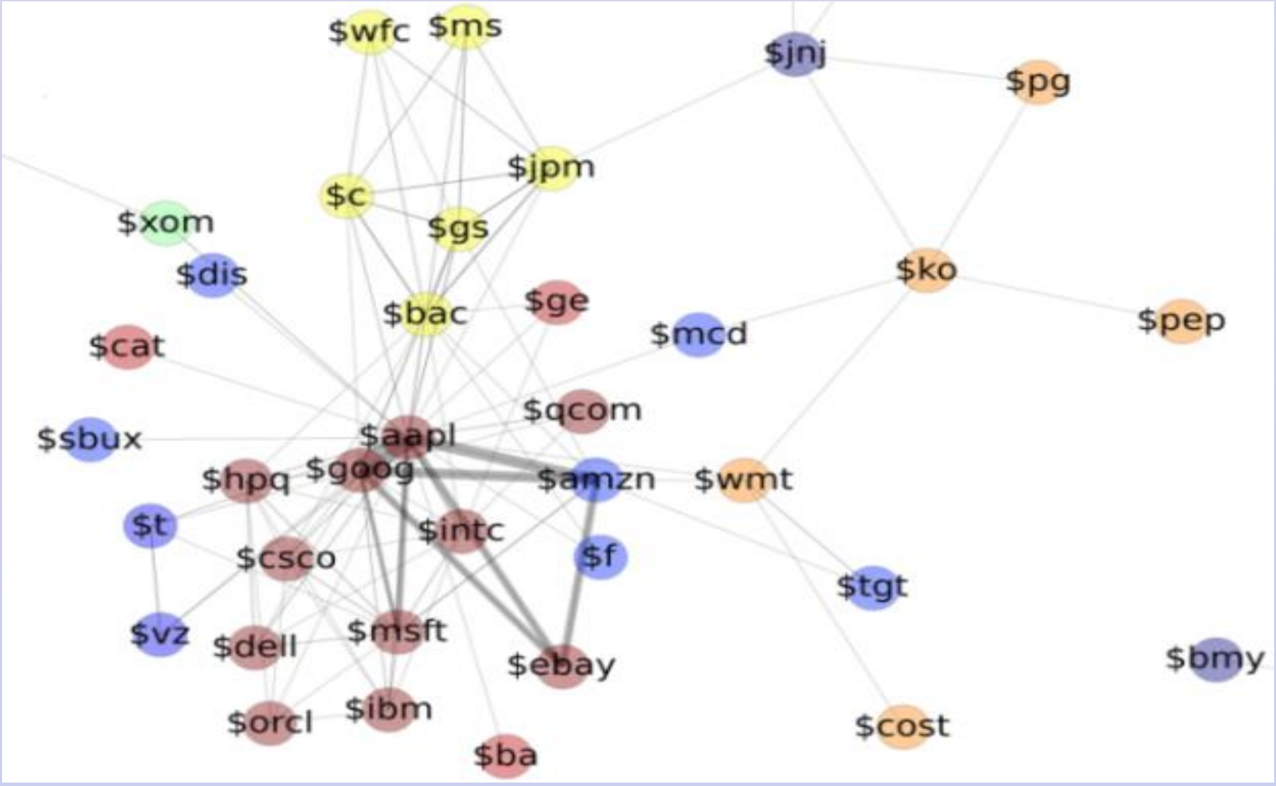


Figure 3. An example stock network.

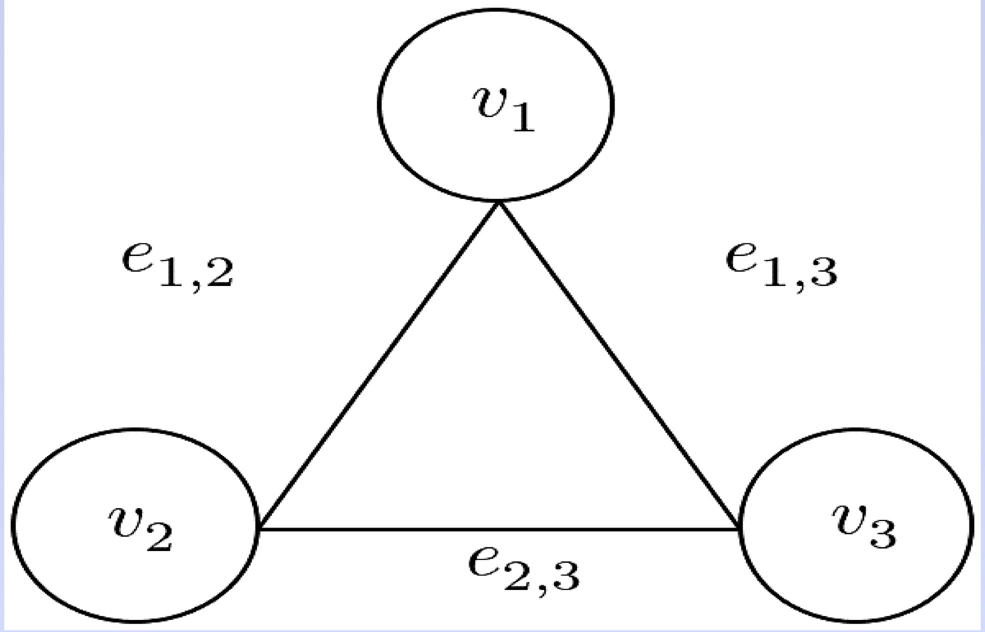


Figure 4. Tweet label design.

## The Semantic Stock Network (SSN)

- We define the stock network as an undirected graph: $G = \{V, E\}$.
  - $V$ comprises stocks.
  - $e_{u,v} \in E$ stands for the edge between stock nodes $u$ and $v$.
- For a tweet, $d$ with three cash-tags: $\{v_1, v_2, v_3\}$, we annotate $d$ with the label set as $L_d = \{v_1, v_2, v_3, e_{1,2}, e_{1,3}, e_{2,3}\}$
  - E.g., $e_{1,2}$ is "aapl_goog" if $v_1$ is "aapl", and $v_2$ is "goog"
- Further apply the Labeled-LDA on this labeled data set.
  - $p(z_i = k|z_{-i}) \sim \frac{N(d_i,k)_{-1} + \alpha}{N(d_i,*)_{-1} + |L_{d_i}| * \alpha} * \frac{N(k,w_i)_{-1} + \eta}{N(k,*)_{-1} + |V| * \eta}$
  - $\beta_{k,w} = \frac{N(k,w_i) + \eta}{N(k,*) + |V| * \eta}$
  - $S(k) = \sum_{w=1}^{|V|} \beta_{k,w}\, l(w)$, $S(k) \in [-1,1]$, $l(w)$ is the opinion polarity of word $w$.

## Stock Market Prediction

- Two-dimensional ($\{x_t\}$, $\{y_t\}$) vector autoregression model (VAR)
  - Regress $y$ on $x$ using least square regression in R.
  - $y_t = \sum_{i=1}^{lag} (\vartheta_i^x x_{t-i} + \vartheta_i^y y_{t-i}) + \varepsilon_t$
- Experiment with different window sizes and lags.
- Evaluate prediction accuracy of Price (↑/↓) movement.

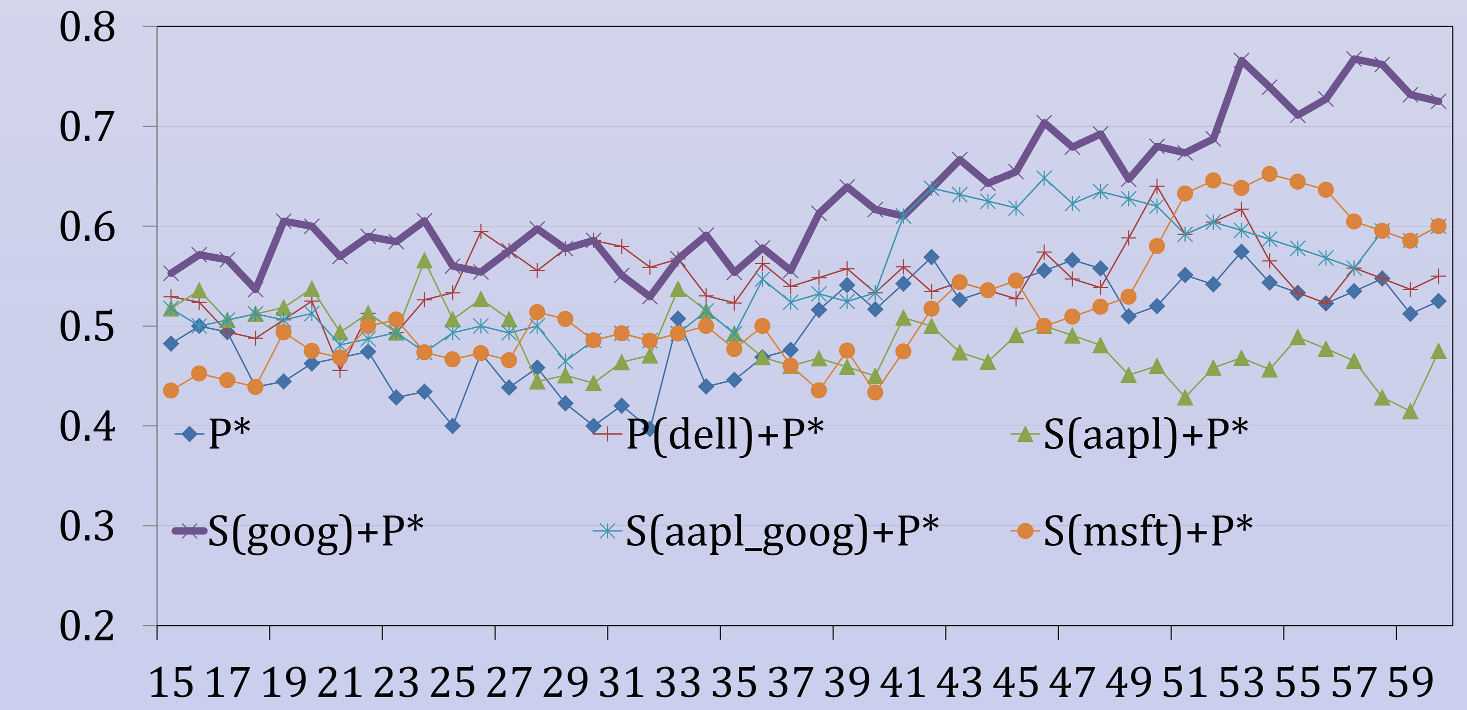


Figure 5. Prediction on $apple on lag 2. (x- axis is training window size, y- axis is the accuracy.)

| Target | $lag$ | $P^*$ only | CSN: P(.)+$P^*$ | SSN: S(.)+$P^*$ | | |
|---|---|---|---|---|---|---|
| goog | | | *dis(0.96)* | *goog* | *aapl* | *amzn* |
| | 2 | 0.48(0.59) | 0.53(0.60) | **0.59(0.65)** | 0.44(0.53) | 0.42(0.49) |
| | 3 | 0.46(0.54) | 0.53(0.62) | **0.56(0.67)** | 0.50(0.59) | 0.43(0.49) |
| amzn | | | *csco(0.90)* | *amzn* | *goog* | *msft* |
| | 2 | 0.48(0.54) | 0.48(0.55) | 0.47(0.54) | 0.57(0.66) | **0.60(0.68)** |
| | 3 | 0.46(0.53) | 0.49(0.53) | 0.43(0.50) | 0.55(0.63) | **0.57(0.66)** |
| ebay | | | *amzn(0.81)* | *ebay* | *amzn* | *goog* |
| | 2 | 0.49(0.55) | 0.51(0.57) | 0.44(0.53) | **0.57(0.64)** | 0.56(0.62) |
| | 3 | 0.48(0.58) | 0.49(0.54) | 0.45(0.58) | **0.54(0.64)** | 0.54(0.61) |
| tgt | | | *vz(0.88)* | *tgt* | *wmt* | *amzn* |
| | 2 | 0.43(0.53) | 0.43(0.54) | 0.46(0.55) | 0.49(0.56) | **0.49(0.59)** |
| | 3 | 0.44(0.50) | 0.40(0.53) | 0.44(0.48) | 0.41(0.48) | **0.48(0.54)** |
| wmt | | | *tgt(0.86)* | *wmt* | *tgt* | *amzn* |
| | 2 | 0.53(0.59) | 0.53(0.63) | 0.52(0.61) | 0.52(0.60) | **0.60(0.65)** |
| | 3 | 0.53(0.64) | 0.48(0.57) | 0.55(0.66) | 0.48(0.58) | **0.58(0.66)** |
| qcom | | | *pfe(0.88)* | *qcom* | *aapl* | *intc* |
| | 2 | 0.53(0.6) | 0.55(0.63) | 0.57(0.61) | 0.46(0.54) | **0.63(0.70)** |
| | 3 | 0.54(0.61) | 0.48(0.55) | 0.56(0.65) | 0.51(0.61) | **0.61(0.67)** |

Table 1. Average and best (in parentheses) prediction accuracies (over window sizes of [15, 60]) of some other cases with different covariates, cell of *dis(0.96)* means "$*dis*" takes the maximum price correlation strength of 0.96 with "$*goog*" (similar for others in column CSN). The best performances are highlighted in bold.

## Conclusion & Future Work

- SSN is robust to find stock pairs with real-world relationship.
- Sentiment based approaches perform better than all price based ones. Furthermore, sentiment of the neighbors in SSN performs best in general.
- The business of offline companies like Target Corp. ($*tgt*) and Wal-Mart Stores Inc. ($*wmt*) are highly affected by online business like $*amzn*.
- Future Work:
  - Fully exploit the network power.
  - Connect social media text to financial reports.