PREDICTING MARKET VOLATILITY FROM FEDERAL RESERVE BOARD MEETING MINUTES

Reza Bosagh Zadeh and Andreas Zollmann
Lab Advisers: Noah Smith and Bryan Routledge
GOALS

- Make Money!
  - Not really.

- Find interesting patterns in meeting minutes
  - Meetings happen roughly 10 times a year
  - Interest rate changes are decided, along with other qualitative assessments of US Economy
  - Minutes freely available on the web for meetings from 1967 to 2008

- “Idea”: Use established tools from NLP and ML
Efficient market hypothesis:
- excess returns per unit risk cannot be consistently generated using public information
- Stock prices react on news in split-seconds
- Automated analysis can outperform humans because of processing speed

Strongest form of EMH: all market correction due to insiders
- Gidofalvi and Elkan (2003): News-based prediction model has highest predictive accuracy over the 20 minutes trading window before the publication time of the respective article
## Previous Attempts

<table>
<thead>
<tr>
<th>Prototype idea</th>
<th>Prototype 3.1</th>
<th>Prototype 3.2</th>
<th>Prototype 3.3</th>
<th>Prototype 3.4</th>
<th>Prototype 3.5</th>
<th>Prototype 3.6</th>
<th>Prototype 3.7</th>
<th>Prototype 3.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aims to forecast…</td>
<td>price trends</td>
<td>price trends</td>
<td>volatilities</td>
<td>price trends</td>
<td>price trends</td>
<td>price trends</td>
<td>price trends</td>
<td>volatilities</td>
</tr>
<tr>
<td>Underlying</td>
<td>equity index</td>
<td>single stock</td>
<td>single stock</td>
<td>single stock</td>
<td>exchange rate</td>
<td>single stock</td>
<td>single stock</td>
<td>single stock</td>
</tr>
<tr>
<td>Forecasting horizon</td>
<td>24 hours</td>
<td>1 hour</td>
<td>N/A</td>
<td>1 hour</td>
<td>3 hours</td>
<td>1 hour</td>
<td>N/A</td>
<td>15 minutes</td>
</tr>
</tbody>
</table>

| Text mining parameter | | | | | | | | |
| Feature definition | manually | automated | manually | automated | manually | automated | automated | semi-automated |
| Number of features | 423 | N/A | 145 | 1000 | 400 | N/A | 200 | 85 |
| Feature granularity | tuple (words) | terms | tuple (terms) | single words | tuple (words) | single words | single words | tuple (terms) |
| Primary classifier | Naive Bayes | Naive Bayes | decision rules | Naive Bayes | decision rules | linear SVM | regression | polynomial SVM |
| Number of categories | 3 | 5 | 39 | 3 | 3 | 5 (training: 3) | 2 | 4 (training: 3) |

| Input data | | | | | | | | |
| Information age | 2 - 15 hours | 0 hours | 0 - 24 hours | 0 hours | 0 - 2 hours | 0 hours | 0 hours | 0 hours |
| Text analyzed | headline, body | headline, body | headline | headline, body | headline | headline, body | headline, body | headline, body |
| Labeling | automated | automated | manually | automated | automated | automated | automated | automated |
| Price frequency | daily close | 10 min. | daily close | 10 min. | 60 min. | intraday | daily close | 15 sec. |

| Test | | | | | | | | |
| Training/Test split | 3 months rolling | 3 / 15 months | 8 / 5 months | 5.5 / 2 months | 1 month rolling | 6 / 1 month(s) | cross validation (90% / 10%) | cross validation (90% / 10%) |
| Prototype vs. random | 44% vs. 33% | N/A | N/A | 40% vs. 33% | 50% vs. 33% | N/A | 61% vs. 30% | 45% vs. 33% |
| Rolling per year | - 600 | > 100’000 | (200) | < 600 | N/A | N/A | N/A | < 500 |
| Profit per roundtrip as reported | 13 bps | 23 bps | (first phase: 10 bps) | 10 bps | N/A | N/A | N/A | 29 bps |
| Market | DJIA, Nikkei, FTSE, HS, ST | 127 stocks (USA) | constraints: Russell 3000 | constraints: DJIA | USD/DEM and USD/JPY | 614 stocks (Hong Kong) | constraints: DAX100 | constraints: S&P500 |

Past work: features used in text-based prediction

- BOW, BObigrams, NPs, NNPs, NEs (frequency, TF-IDF score, or information gain)

- Lerman et al. (COLING’08):
  - News-focus features:
    - change in occurrence frequency of a word in the current day's news coverage compared to the average news coverage of the past N days
  - Dependency features
CLASSIFICATION VS. REGRESSION

- Most past work: predicts increase or decrease in prices/volatility
- Kogan et al. (NAACL’09): predict indicator (stock volatility) directly using Support Vector Regression
TAPPING THE FED

- Our aim: use FOMC meeting minutes to predict financial indicators
- No previous attempts to our knowledge
- Boukus & Rosenberg: market participants do extract complex signals from these minutes
  - found correlations of e.g. Treasury yields with specific themes of the meeting minutes using Latent Semantic Analysis
OUR ATTEMPT

Predicting Prices is too hard. Focus on Volatility:

\[ \text{vol} = \frac{1}{n-1} \sum_{i=1}^{n} [\ln \text{return}(t + i) - 1/n \sum_{j=1}^{n} \ln \text{return}(t + j)]^2, \]
where \( t \) is the time of the meeting and
\[ \text{return}(t) = \frac{\text{price}(t)}{\text{price}(t - 1)} - 1. \]

Predict volatility of
- S&P 500
- 13-week Treasury Bills
- 10-year Treasury Notes
Machine Learning Setting

- Take meeting minutes from minutes held on day $t$, and predict volatility $n$ (look-ahead) days ahead.

- Not I.I.D. training data at all. But let’s hide that under the carpet.

- Features:
  - Bag of Words: unigrams, bigrams
  - Dependency fragments
  - Volatility from $n$ days ago
Dependency features

- (S
  (NP Business inventories)
  (VP climbed)
  (ADVP significantly further)
  (PP in)
  (NP September))) .)

- Using seedword ‘inventories’, extract fragments:
  Inventories ➔ climbed, business ➔ inventories ➔ climbed, inventories ➔ climbed ← further
FEATURES: BAG OF WORDS

- **TF-IDF:**
  \[
  (tf-idf)_{i,j} = tf_{i,j} \times idf_i
  \]
  
  \[
  tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}
  \]
  \[
  idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}
  \]
  
  - IDF dampens effect of common words

- **Log1P:**
  \[
  \log |1 + \frac{n_{i,j}}{\sum_k n_{k,j}}|
  \]
  
  - Don’t really need IDF since we already removed stop-words
DATA MINING

- Obtained meeting text from PDF and HTML files available at
  http://www.federalreserve.gov/monetarypolicy/fomc.htm

- Our corpus available at
  http://rezab.ca/useful/fomc_minutes.html

- Stemmed using Porter 2 stemmer. Removed stop-words using available online list.
PREDICT WHAT?

- Regression:
  - Actual value of the volatility

- Classification:
  - Two classes, volatility goes UP or DOWN

First set of experiments:
Classification for different indices.
S&P 500 – Classification – Short Periods

Higher is better.
10 Year Treasury Note Classification – Short Periods

[Graph showing data points for base and 10yr-tnote over a period of 11 years.]
13 Week Treasury Bills – Classification – Short Periods
13 Week Treasury Bills – Classification – Long Periods
Example decision tree

This had 64% Accuracy
**SVM Prominent Terms**

**negative:**
-0.5677 * (normalized) w_action
-0.554 * (normalized) w_mnufactur
-0.4998 * (normalized) w_sowli
-0.4965 * (normalized) w_craven
-0.4947 * (normalized) w_recent
-0.4771 * (normalized) w_outcom
-0.3755 * (normalized) w_crude
-0.3732 * (normalized) w_institut
-0.3718 * (normalized) w_affect
-0.3694 * (normalized) w_climb
-0.3551 * (normalized) w_canadian
-0.3533 * (normalized) w_cumul

**positive:**
0.3664 * (normalized) w_surg
0.3679 * (normalized) w_polici
0.3735 * (normalized) w_warehous
0.375 * (normalized) w_resum
0.3864 * (normalized) w_job
0.4039 * (normalized) w_impliment
0.4054 * (normalized) w_outlook
0.4059 * (normalized) w_struckmey
0.4068 * (normalized) w_cutback
0.6298 * (normalized) w_downward
0.6536 * (normalized) w_curtail
Conclusions from Classification

- Shorter Periods are easier to predict than longer periods

- 13 Week Treasury bills are easier to predict than S&P 500 and 10 Year Treasury notes

- Bigrams don’t help in our case

- Dependency fragments don’t help either

Now onto Regression...
S&P 500 – Regression – Short Periods

Now lower is better.
S&P 500 – Regression – Long Periods
CONCLUSIONS FROM REGRESSION

- Regression for S&P 500 is hard – can’t beat simple straw man baseline using only words

- Oddly enough, training on the previous volatility does worse than just predicting the previous volatility.
  - Over-fitting happening with just two dimensions – very surprising, a testament to the difficulty of the problem.