Boosting Financial Trend Prediction with Twitter Mood Based on Hidden Markov Models

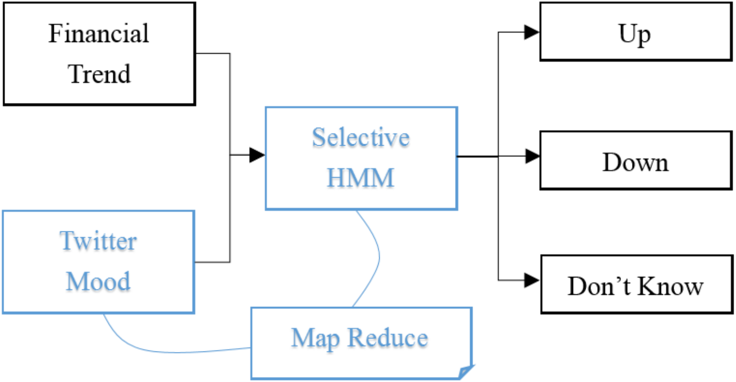
Yifu Huang

School of Computer Science, Fudan University
huangyifu@fudan.edu.cn

COMP630030 Data Intensive Computing Project, 2013

Outline

Introduction



Financial Trend

- Focus on market index
  - DJIA, S&P500, ...
- Raw data
  - (Date, Open, High, Low, Close, Volume, Adj Close)
- Compute trend
  - $Trend_{t+1} = sign(Close_{t+1} - Close_t)$

Financial Trend

- Focus on market index
  - DJIA, S&P500, ...
- Raw data
  - (Date, Open, High, Low, Close, Volume, Adj Close)
- Compute trend
  - $Trend_{t+1} = sign(Close_{t+1} - Close_t)$

Financial Trend

- Focus on market index
  - DJIA, S&P500, ...
- Raw data
  - (Date, Open, High, Low, Close, Volume, Adj Close)
- Compute trend
  - $Trend_{t+1} = sign(Close_{t+1} - Close_t)$

Twitter Mood

- Behavioral finance
  - Individual mood -> individual decision
  - Societies mood -> societies decision
- Societies mood measurement
  - Twitter, sense the world
- POMS Bipolar word list
  - Composed/anxious, agreeable/hostile, elated/depressed, confident/unsure, energetic/tired, clearheaded/confused
  - WordNet extend
  - Granger causality analysis

Twitter Mood

- Behavioral finance
  - Individual mood -> individual decision
  - Societies mood -> societies decision
- Societies mood measurement
  - Twitter, sense the world
- POMS Bipolar word list
  - Composed/anxious, agreeable/hostile, elated/depressed, confident/unsure, energetic/tired, clearheaded/confused
  - WordNet extend
  - Granger causality analysis

Twitter Mood

- Behavioral finance
  - Individual mood -> individual decision
  - Societies mood -> societies decision
- Societies mood measurement
  - Twitter, sense the world
- POMS Bipolar word list
  - Composed/anxious, agreeable/hostile, elated/depressed, confident/unsure, energetic/tired, clearheaded/confused
  - WordNet extend
  - Granger causality analysis

Twitter Mood (cont.)

- Key idea
  - Twitter data is large, so map it to different nodes, extract poms vector from each tweet, and reduce them to overall poms index
- Map (offset, line, date, poms_individual)
  - Filter
    - Ignore {http:, www.}, hold {i feel, makes me, ...}
  - Stem
    - Agreed -> agree, disabled -> disable, ...
  - Analyze
    - Seren -> composed, shaki -> anxious, ...
- Reduce (date, poms_individual, date, poms_societies)
  - Average

Twitter Mood (cont.)

- Key idea
  - Twitter data is large, so map it to different nodes, extract poms vector from each tweet, and reduce them to overall poms index
- Map (offset, line, date, poms_individual)
  - Filter
    - Ignore {http:, www.}, hold {i feel, makes me, ...}
  - Stem
    - Agreed -> agree, disabled -> disable, ...
  - Analyze
    - Seren -> composed, shaki -> anxious, ...
- Reduce (date, poms_individual, date, poms_societies)
  - Average

Twitter Mood (cont.)

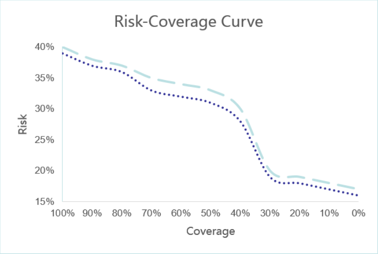
- Key idea
  - Twitter data is large, so map it to different nodes, extract poms vector from each tweet, and reduce them to overall poms index
- Map (offset, line, date, poms_individual)
  - Filter
    - Ignore {http:, www.}, hold {i feel, makes me, ...}
  - Stem
    - Agreed -> agree, disabled -> disable, ...
  - Analyze
    - Seren -> composed, shaki -> anxious, ...
- Reduce (date, poms_individual, date, poms_societies)
  - Average

Selective Prediction

- Not ignorance, but ignorance of ignorance is the death of knowledge
- Definition

$$Y_{t+1} = \begin{cases} F(X_t), & if\ G(X_t) = 1 \\ reject, & if\ G(X_t) = 0 \end{cases}$$

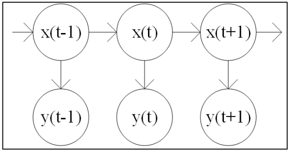
- Evaluation



Hidden Markov Models

- Definition
  - $\lambda = \{N, M, \pi, A, B\}$
  - N, state number
  - M, observation number
  - $\pi = \{\pi_1, \pi_2, ..., \pi_N\}$, initial probability
  - $A = \{a_{ij} | i, j = 1, 2, ..., N\}$, state transfer probability
  - $B = \{b_{ij} | i = 1, 2, ..., N, j = 1, 2, ..., M\}$, observation probability distribution

Hidden Markov Models (cont.)



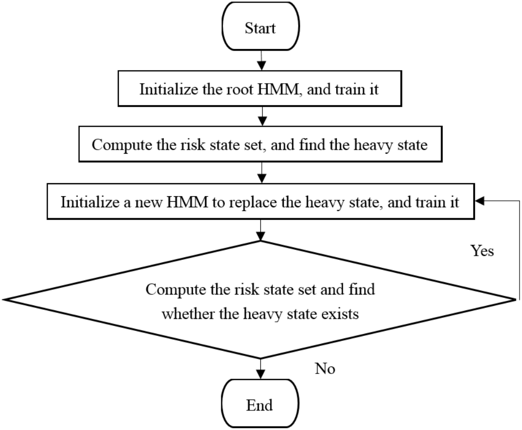
- Problem 1: probability computation
  - Forward backward algorithm
- Problem 2: state annotation
  - Viterbi algorithm
- Problem 3: model train
  - Baum-Welch algorithm

Selective HMM

- Model
  - Multiple Stream
  - Selectivity
    - Visit rate $v_i = \frac{1}{T}\sum_{t=1}^{T}\gamma_{ti}$
    - Risk rate $r_i = \frac{\frac{1}{T}\sum_{t=1,l_t\neq l_i}^{T}\gamma_{ti}}{v_i}$
    - Risk state set $RS = \{i_1, ..., i+k|\sum_{j=1}^{K} v_{i_j} \leq 1 - C_B, \sum_{j=1}^{K+1} v_{i_j} > 1 - C_B\}$
  - Scale

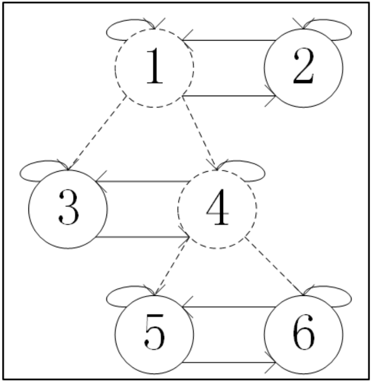
Selective HMM (cont.)

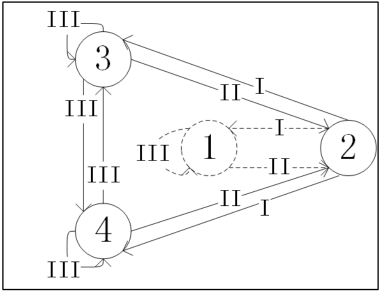
- Train



Selective HMM (cont.)

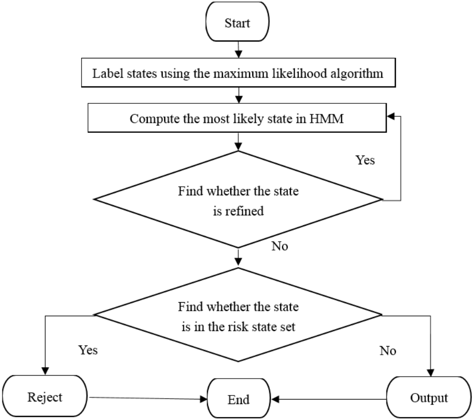
- Train
    - Recursive refine





Selective HMM (cont.)

- Predict



Selective HMM (cont.)

- Key idea
  - Random initialization number is large, so map selective HMM to different nodes, get error rate from each selective HMM after train and predict, and reduce them to overall error rate
- Map (offset, line, reject_bound, error_rate)
  - Train
  - Predict
  - Get error rate
- Reduce (reject_bound, error_rate, reject_bound, avg_error_rate)
  - Average

Selective HMM (cont.)

- Key idea
  - Random initialization number is large, so map selective HMM to different nodes, get error rate from each selective HMM after train and predict, and reduce them to overall error rate
- Map (offset, line, reject_bound, error_rate)
  - Train
  - Predict
  - Get error rate
- Reduce (reject_bound, error_rate, reject_bound, avg_error_rate)
  - Average

Selective HMM (cont.)

- Key idea
  - Random initialization number is large, so map selective HMM to different nodes, get error rate from each selective HMM after train and predict, and reduce them to overall error rate
- Map (offset, line, reject_bound, error_rate)
  - Train
  - Predict
  - Get error rate
- Reduce (reject_bound, error_rate, reject_bound, avg_error_rate)
  - Average

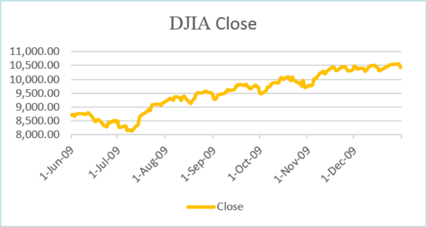
Setup

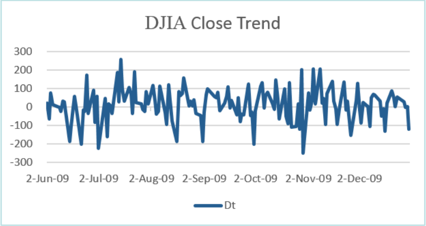
- Cluster
  - Hadoop 1.2.1
  - Masters:1, slaves:24
  - Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz
  - 4GB RAM
- Data
  - DJIA, 200906~200912, ~130day
  - Tweet, 200906~200912, ~50GB

Setup

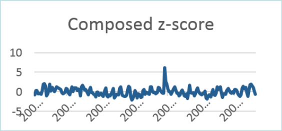
- Cluster
  - Hadoop 1.2.1
  - Masters:1, slaves:24
  - Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz
  - 4GB RAM
- Data
  - DJIA, 200906~200912, ~130day
  - Tweet, 200906~200912, ~50GB

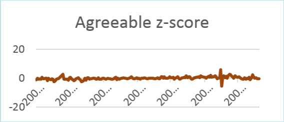
DJIA

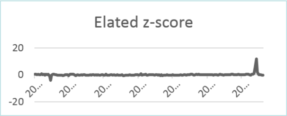

DJIA Close
11,000.00
10,500.00
10,000.00
9,500.00
9,000.00
8,500.00
8,000.00
1-Jun-09
1-Jul-09
1-Aug-09
1-Sep-09
1-Oct-09
1-Nov-09
1-Dec-09
Close

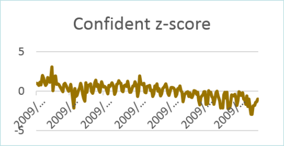

DJIA Close Trend
300
200
100
0
-100
-200
-300
2-Jun-09
2-Jul-09
2-Aug-09
2-Sep-09
2-Oct-09
2-Nov-09
2-Dec-09
Dt

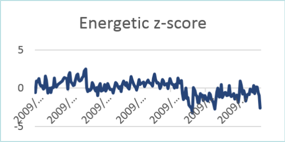
POMS

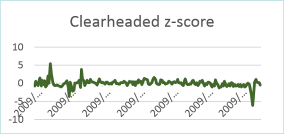

Composed z-score
10
5
0
-5


Agreeable z-score
20
0
-20


Elated z-score
20
0
-20

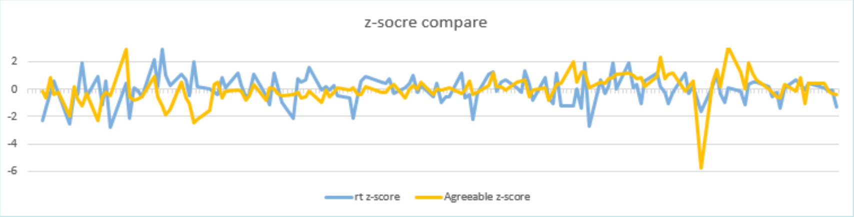

Confident z-score
5
0
-5


Energetic z-score
5
0
-5

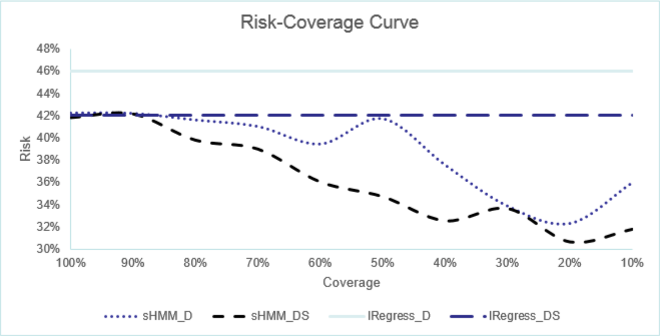

Clearheaded z-score
10
5
0
-5
-10

Granger Causality Analysis

Lagged Days	Composed /Anxious	**Agreeable /Hostile**	Elated /Depressed	Confident /Unsure	Energetic /Tired	Clearheaded /Confused
1	0.723009776	0.512862214	0.9399375	0.880644906	0.857355253	0.342346356
2	0.86129301	0.166551184	0.8289756	0.576292251	0.933422157	0.310755746
3	0.434470424	**0.062817907**	0.9608715	0.455076866	0.993825935	0.377955186
4	0.435631775	0.127495831	0.9903607	0.637129619	0.803028135	0.514455259
5	0.593896982	0.212591485	0.9854185	0.534574688	0.755306207	0.708745583
6	0.630440149	0.206866576	0.9689204	0.656838808	0.557477213	0.738674666
7	0.694607494	0.107745913	0.9858471	0.688712317	0.577784406	0.851840215



Risk-Coverage Curve



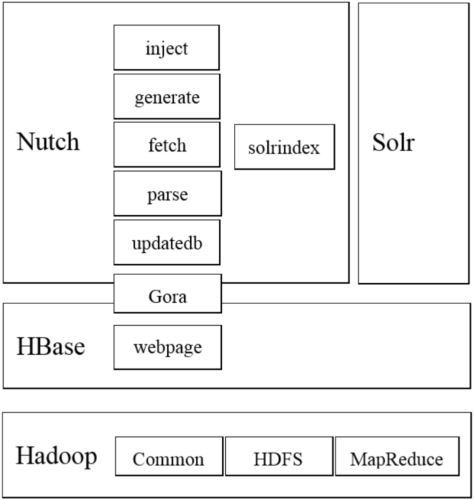
Discussion

- Conclusion
  - Twitter mood can indeed improve prediction accuracy
  - Selective HMM outperforms linear regression
  - Selectivity can indeed control prediction accuracy
- Future Work
  - Risk-coverage curve of selective HMM should decrease monotonically

Discussion

- Conclusion
  - Twitter mood can indeed improve prediction accuracy
  - Selective HMM outperforms linear regression
  - Selectivity can indeed control prediction accuracy
- Future Work
  - Risk-coverage curve of selective HMM should decrease monotonically

BBS Search Engine



BBS Search Engine (cont.)

10.171.5.222:3000

References I

- [1] Selective Prediction of Financial Trends with Hidden Markov Models. NIPS2011.
- [2] Twitter Mood Predicts the Stock Market. JCS2011.
- [3] Data-Intensive Text Processing with MapReduce. 2010.
- [4] Hadoop: The Definitive Guide. 2012.