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

Yifu Huang[1], Shuigeng Zhou[1](✉), Kai Huang[1], and Jihong Guan[2]

[1] Shanghai Key Lab of Intelligent Information Processing,
School of Computer Science, Fudan University, Shanghai 200433, China
{huangyifu,sgzhou,kaihuang14}@fudan.edu.cn
[2] Department of Computer Science and Technology,
Tongji University, Shanghai 201804, China
jhguan@tongji.edu.cn

Abstract. Financial trend prediction has been a hot topic in both academia and industry. This paper proposes to exploit Twitter mood to boost financial trend prediction based on *selective hidden Markov models* (sHMM). First, we expand the *profile of mood states* (POMS) Bipolar lexicon to extract rich society moods from massive tweets. Then, we determine which mood has the most predictive power on the financial index based on *Granger causality analysis* (GCA). Finally, we extend sHMM to combine financial index and the selected Twitter mood to predict next-day trend. Extensive experiments show that our method not only outperforms the state-of-the-art methods, but also provides controllability to financial trend prediction.

1 Introduction

Financial trend prediction has been a hot research and practice topic in both academia and industry. The purpose of financial trend prediction is to predict the ups and downs of financial trends by building models based on historical financial data [11,15,23,36]. Besides historical financial data, more and more additional indictors such as news reports [14,26], Twitter mood [2,20,27,28] and trading relationship [30] have been used to improve financial trend prediction.

According to behavior finance [21], society mood is correlated with and even has predictive power on public financial index. Si *et al.* [27] modeled society mood of Twitter to support financial trend prediction. They utilized topic-based model to extract sentiments from Twitter posts, and then regressed the stock index and the sentiment series in an autoregressive framework. They achieved improved prediction performance when taking advantage of Twitter mood. However, in addition to prediction accuracy, *controllability* is also an important issue in financial trend prediction. Some works [4,7] modeled controllability on *selective prediction*—a prediction framework that can qualify its own prediction results and reject the outputs when they are not confident enough. Selective prediction can provide a trade-off between *coverage* (indicating how many predictions

M. Renz et al. (Eds.): DASFAA 2015, Part II, LNCS 9050, pp. 435–451, 2015.
DOI: 10.1007/978-3-319-18123-3_26

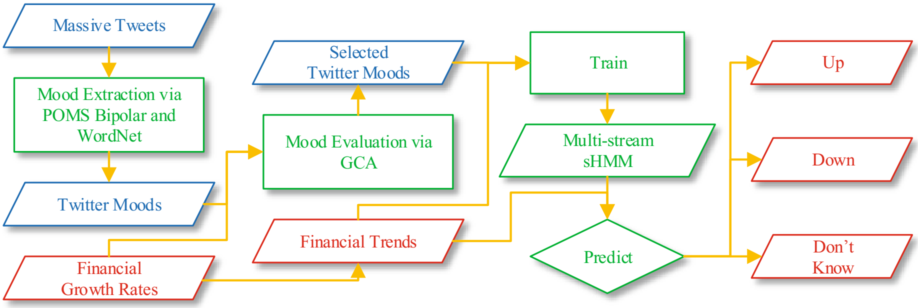


Fig. 1. The workflow of our approach

are made) and *accuracy* (indicating how many predictions are correct). This mechanism allows the users to interfere the prediction process, which is a desirable feature from the practice point of view. Three years ago, *hidden Markov model* (HMM) was introduced to the selective prediction framework, which leads to the *selective hidden Markov models* (sHMM) [23].

In this paper, to further boost financial trend prediction, we propose to combine Twitter mood and sHMM so that we can not only achieve high prediction performance but also obtain good controllability to financial trend prediction. To the best of our knowledge, this is the first attempt to exploit Twitter mood to predict financial trend with controllability. Concretely, we first use the *profile of mood states* (POMS) [18] Bipolar lexicon expanded by WordNet [19] to efficiently extract six-dimensional society moods from massive tweets, including *composed-anxious*, *agreeable-hostile*, *elated-depressed*, *confident-unsure*, *energetic-tired* and *clearheaded-confused*. Second, we perform *Granger causality analysis* (GCA) [9] between financial index and each Twitter mood to effectively determine which Twitter mood has the most predictive power on the index.

Then, we extend sHMM to multiple data streams so that historical financial data and selected Twitter mood can be combined to train the sHMM. Finally, we identify low-quality states of the trained sHMM according to given coverage and prevent predictions that are made from the low-quality states. Extensive experiments over real datasets show that our method not only outperforms the state-of-the-art methods, but also provides controllability to financial trend prediction. Fig. 1 illustrates the workflow of our approach.

The rest of this paper is organized as follows: Section 2 reviews the related work and highlights the differences between our work and major existing methods. Section 3 introduces the details of Twitter mood extraction and evaluation. Section 4 extends sHMM to combine index data and Twitter mood to predict financial trend. Section 5 presents performance evaluation and comparison. Finally, Section 6 concludes the paper.

2 Related Work

The purpose of financial trend prediction is to build models based on historical financial data and then employ the built models to predict the trend of future financial data. Up to now, various models have been proposed and different data were exploited. Existing works focus mainly on two aspects: data selection and model selection. For data selection, Idvall and Jonsson [11], Lin *et al.* [15], Pidan and El-Yaniv [23] and Zhang [36] used historical financial data only to predict financial trend, while Bollen *et al.* [2], Mittal *et al.* [20], Si *et al.* [27] and Sprenger *et al.* [28] combined historical financial data with some additional indicators such as Twitter mood to predict. As for model selection, Bollen *et al.* [2] exploited non-linear Self-Organizing Fusion Neural Networks (SOFNN) model, Si *et al.* [27] employed linear Vector Autoregression (VAR) model, while Idvall and Jonsson [11], Pidan and El-Yaniv [23], and Zhang [36] proposed HMM based methods. Since we do prediction by considering historical financial trends and Twitter moods together, some works [3,33,34] about how to model multiple time series that may correlate with each other and how to use multiple time series together to make prediction, are also highly related. The closest works to ours are [2], [23] and [27]. In what follows, we give a detailed description of the three methods and highlight the differences between them and our method.

Bollen *et al.* [2] investigated whether society moods have predictive on stock index. The authors extracted society moods from tweets through a lexicon called Google-POMS. Then, they performed a non-linear model called SOFNN to predict future trend. Major technical differences between their work and ours are: They used a lexicon called Google-POMS for Twitter mood extraction, which is not publicly available. Instead, we build a new lexicon by expanding POMS Bipolar with WordNet. Considering the importance of controllability, we utilize the sHMM rather than their SOFNN model to provide controllability to the prediction process.

Pidan and El-Yaniv [23] introduced the selective prediction framework into HMM, and addressed the importance of compromising coverage to gain better accuracy in a controlled manner. This method can identify low-quality HMM states and prevents predictions at those states. However, the training sequence of this model is only historical financial data. In this paper, we extend the sHMM to adopt both historical financial data and Twitter mood as input to boost the prediction performance.

Si *et al.* [27] proposed to leverage topic based sentiments from Twitter to help to predict the stock market. The authors utilized continuous Dirichlet process mixture (DPM) model to extract sentiments from Twitter posts, and then regressed the stock index and the sentiment time series in an autoregressive framework. They focused mainly on topic based sentiment analysis of tweets and used a simple prediction model, while we combine lexicon-based Twitter mood extraction and sHMM — an advanced model with controllability.

3 Mood Extraction and Evaluation

In this section, we extract and evaluate Twitter moods. Our aim is to predict public financial index, so we focus on analyzing sentiments of global tweets. First, we build a sentiment lexicon based on POMS Bipolar and WordNet. Then, we leverage the MapReduce framework to retrieve Twitter moods from massive tweets. Finally, we evaluate the predictive power of different Twitter moods via GCA, and determine the most predictive Twitter mood.

3.1 Basics of Sentiment Analysis

Behavior finance [21] shows that society mood has powerful influence on society decision. Furthermore, society mood is correlated with and even has predictive power on public financial index. To acquire society moods, we can analyze the sentiments of social media such as tweets that are fresh opinions shared by citizens. Sentiment analysis, *a.k.a.* opinion mining [17,22], is an application of *Natural Language Processing* (NLP) that aims at extracting subjective information such as author attitudes from texts [16,25]. Author attitudes may reflect the judgements or opinions of the authors, mood states or sentiments that the authors want to disseminate to the public. A major task of sentiment analysis is to extract multi-dimensional polarities from texts. Generally, there should be a pre-defined sentiment lexicon [10,32] for each polarity. A polarity is a time series obtained by first counting the sentiment word frequencies and then aggregating the frequencies in terms of a certain time granularity.

3.2 Expanding POMS Bipolar Lexicon by WordNet

According to the research of multi-dimensional sentiment analysis, human mood is very rich in social media, and a piece of text may contain multiple sentiments such as calm and agreement. POMS [18] is a questionnaire designed by psychologists to assess human mood states, and it already has three versions, namely, POMS Standard, POMS Brief and POMS Bipolar. POMS Bipolar consists of 6 polarities called *composed-anxious* (**Com.**), *agreeable-hostile* (**Agr.**), *elated-depressed* (**Ela.**), *confident-unsure* (**Con.**), *energetic-tired* (**Ene.**) and *clearheaded-confused* (**Cle.**), respectively. Each polarity contains 12 adjectives, and each of the 12 adjectives can increase either positive or negative polarity.

Due to the small size of the POMS Bipolar lexicon, it cannot capture all sentiments from texts in practice. So there should be some method to expand the POMS Bipolar lexicon. This paper employs WordNet synsets to expand the POMS Bipolr lexicon. WordNet [19] is an English language lexicon that subsumes English words into groups of synonyms called synsets. By mapping 72 words in the POMS Bipolar lexicon to their WordNet synsets, we get an expanded lexicon consisting of 638 words.

Algorithm 1. Twitter mood extraction

```
def MAP(date $d$, tweet $t$)
    $\boldsymbol{v} \leftarrow$ ANALYZE(STEM(FILTER($t$)))
    EMIT(date $d$, vector $\boldsymbol{v}$)
end def
```

```
def REDUCE(date $d$, vectors $[\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots]$)
    $\boldsymbol{v_{avg}} \leftarrow$ AVERAGE($[\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots]$)
    EMIT(date $d$, vector $\boldsymbol{v_{avg}}$)
end def
```

3.3 Mood Extraction from Massive Tweets

Because of the massive amount of tweets, we leverage the MapReduce [6] framework to efficiently extract Twitter moods. The algorithm is outlined in Algorithm 1.

In the Map stage, the FILTER method discards tweets containing spam keywords such as "http:" and "www.", and keeps tweets containing subjective phrases such as "i feel" and "makes me". The STEM method normalizes terms in a tweet by eliminating prefixes and suffixes. The ANALYZE method computes the six-dimensional sentiment vector of a tweet using the expanded POMS Bipolar lexicon. In the Reduce stage, we compute the average sentiment vector of each trading day.

3.4 Mood Evaluation via Granger Causality Analysis

After sentiment analysis of tweets, we get six-dimensional sentiment series. Following that, we need to effectively evaluate them to determine which mood can help mostly predict market trend. GCA was first proposed by Clive Granger [9], it is a statistical hypothesis test for determining whether one time series is useful in forecasting another. Clive Granger argued that causality in economics could be reflected by measuring the ability of predicting the future values of a time series using past values of another time series. Formally, a time series X is said to Granger-cause another time series Y if it can be shown, usually through a series of t-test and F-test on some lagged values of X and Y, that those X values have statistically significant influence on the future values of Y. Formally, the following two equations hold:

$$Y_t = y_0 + \sum_{i=1}^{lag} y_i Y_{t-i} + \varepsilon_t, \tag{1}$$

$$Y_t = y_0 + \sum_{i=1}^{lag} y_i Y_{t-i} + \sum_{i=1}^{lag} x_i X_{t-i} + \varepsilon_t. \tag{2}$$

Above, t and i are time variables (in days), and lag is the upper bound of lagged days. We perform GCA in the same way as [8] between the growth rate of financial index (Y) and each Twitter mood (X). We determine the Twitter mood that has the most predictive power and its corresponding lagged value according to the following two rules: 1) find which p_{value} is at statistically significant level ($p_{value} \leq 0.1$); 2) find which p_{value} decreases significantly comparing to its precursor ($difference$ <-0.25). The first rule is for selecting the Twitter mood with the corresponding lagged days that has significant predictive power to the growth rate of financial index. The second rule is for guaranteeing that the selected Twitter mood of a certain day can provide significant improvement on predictive power comparing to that of the preceding day. The parameter *lag* of the selected Twitter mood is further used as the encoding length of the observation in our prediction model.

4 The Multi-stream sHMM

Here we extend sHMM to handle multi-streams. We call the extended model *multi-stream sHMM*, or *msHMM* in short. First, we introduce the basic concepts of HMM. Then, we briefly introduce sHMM. And finally, we present multi-stream sHMM, including the training and prediction processes as well as model evaluation with a large number of random starts based on the MapReduce framework.

4.1 HMM

HMM is a generative probabilistic model with latent states, where hidden state transitions and visible observation emissions are assumed to be Markov processes. Given an observation sequence O=$\{o_1, o_2, ..., o_T\}$ that is generated by a HMM λ, we associate O with a latent state sequence S=$\{s_1, s_2, ..., s_T\}$ that most likely produces O. λ can be formally defined as a quintuple $\{N, M, \boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$. Here, N is the number of states in the state set Q=$\{q_1, q_2, ..., q_N\}$; M is the number of observations in the observation set U=$\{u_1, u_2, ..., u_M\}$; $\boldsymbol{\pi}$ is the initial probability vector of states and π_i=$P(s_1$=$q_i)$ is the initial probability of state q_i; $\boldsymbol{A}$ is the transition probability matrix of states and a_{ij}=$P(s_{t+1}$=$q_j|s_t$=$q_i)$ is the transition probability from state q_i to state q_j; $\boldsymbol{B}$ is the observation emission probability matrix of states and b_{ij}=$P(o_t$=$u_j|s_t$=$q_i)$ is the emission probability of observation u_j at state q_i.

4.2 sHMM

Selective prediction [4,7] is a prediction framework that can qualify its own prediction results and reject outputs if they are not confident enough. Pidan and El-Yaniv [23] introduced the selective prediction framework [7] to HMM, and thus developed sHMM. As in [23], we add state label p_i, empirical visit rate v_i and empirical state risk r_i to each state q_i, and add reject subset RS and heavy state q_h to HMM λ. For better understanding sHMM and the following multi-stream sHMM, we recall the major definitions of sHMM as follows in the context of financial trend prediction.

Definition 1. *Given an observation sequence* O=$\{o_1,\ o_2,\ ...,\ o_T\}$ *(indicating historical financial trend), a relative label sequence* L=$\{l_1,\ l_2,\ ...,\ l_T\}$ *(indicating* *next-day financial trend) and a HMM λ, the state label p_i denotes the most probable label that state q_i should have. Formally,*

$$p_i = \arg \max_{l=up,down} \sum_{t=1,l_t=l}^{T} \gamma_{ti}. \tag{3}$$

Above, γ_{ti}=$P(s_t$=$q_i|O,\ \lambda)$ denotes the probability that the HMM λ stays at state q_i at time t, which can be computed by the forward-backward procedure [24].

Definition 2. *Given an observation sequence O=$\{o_1,\ o_2,\ ...,\ o_T\}$ and a HMM λ, the empirical visit rate v_i denotes the fraction of time that the HMM λ spends at state q_i,* i.e.,

$$v_i = \frac{1}{T}\sum_{t=1}^{T} \gamma_{ti}. \tag{4}$$

Definition 3. *Given an observation sequence O=$\{o_1,\ o_2,\ ...,\ o_T\}$, a relative label sequence L=$\{l_1,\ l_2,\ ...,\ l_T\}$ and a HMM λ, the empirical state risk r_i denotes the rate of erroneous visits to state q_i. Formally,*

$$r_i = \frac{\frac{1}{T}\sum_{t=1,l_t\neq p_i}^{T} \gamma_{ti}}{v_i}. \tag{5}$$

Furthermore, we sort all HMM states by their empirical state risks in descending order and record them as Q_d=$\{q_{d_1},\ q_{d_2},\ ...,\ q_{d_N}\}$ (for each $j < k$, $r_{d_j} \geq r_{d_k}$). The low-quality HMM states, also called *reject states*, constitute the *reject subset RS*. Predictions at those states are prevented.

Definition 4. *Given a coverage bound C_B, we label the reject states sequentially until their cumulative empirical visit rate $\sum_{j=1}^{K} v_{d_j}$ exceeds 1-C_B. Formally, the reject subset RS is defined as*

$$RS = \{q_{d_1}, q_{d_2}, ..., q_{d_K} | \sum_{j=1}^{K} v_{d_j} \leq 1 - C_B, \sum_{j=1}^{K+1} v_{d_j} > 1 - C_B\}. \tag{6}$$

Definition 5. *Given a visit bound V_B, state $q_{d_{K+1}}$ is identified as a heavy state q_h if its visit rate $v_{d_{K+1}} > V_B$.*

The heavy state q_h is the cause of coarseness problem as described in [23], and it should be recursively refined in the training stage. Another issue should be taken into consideration in practice is scaling, because floating point underflow can easily happen in the forward-backward procedure that is the fundamental of HMM. We handle it with the solution provided by [24].

4.3 Multi-stream sHMM

To combine financial index and Twitter moods to sHMM, we extend sHMM to process multiple data streams. We treat historical financial trend and Twitter mood trends as multiple observation sequences generated by sHMM, and formulate multiple observation sequences as $O_K{=}\{O^{(1)}, O^{(2)}, ..., O^{(K)}\}$ where $O^{(k)}{=}\{o_1^{(k)}, o_2^{(k)}, ..., o_{T_k}^{(k)}\}$. The observation is gained by encoding the trend with *lag* length, which is determined in "Mood Evaluation via Granger Causality Analysis" subsection. In the training stage, likelihood function $P(O_K|\lambda)$ (indicating the probability that multiple observation sequences are produced by the model) is maximized via a variation of the Baum-Welch algorithm [1]:

$$P(O_K|\lambda) = \prod_{k=1}^{K} P(O^{(k)}|\lambda) = \prod_{k=1}^{K} P_k. \tag{7}$$

For example, denote 1 as *up trend* and 0 as *down trend*. Say we have the financial trend sequence $A{=}\{0, 0, 0, 1, 1, 0, 1\}$, the selected Twitter mood trend sequence $B{=}\{1, 0, 1, 0, 1, 0, 1\}$ and the encoding length $lag = 3$. Based on A, B and lag, we can get the encoding financial trend sequence $A_e{=}\{0, 1, 3, 6, 5\}$ and the encoding selected Twitter mood trend sequence $B_e{=}\{5, 2, 5, 2, 5\}$. Our prediction model multi-stream sHMM use A_e and B_e as training sequences to get model parameters.

Training. Given a coverage bound C_B, multiple observation sequences O_K and a label sequence L, we train a multi-stream sHMM and recursively refine the heavy state q_h until there is no heavy state remaining. The training process consists of the following steps:

1. Initialize the root HMM λ_0 with random parameters, and train it with the set O_K of historical financial trend and Twitter mood trends. The train algorithm is Baum-Welch [1] variation adjusted to multiple observation sequences.
2. Compute state label p_i, empirical visit rate v_i and empirical state risk r_i for each state q_i in the root HMM λ_0. Under the coverage bound C_B, compute the reject subset RS of the root HMM λ_0 to identify which state is the heavy state q_h. If there is no heavy state, then the training is done.
3. Initialize a random HMM λ_{random} to replace the heavy state q_h, so a refined HMM λ_{refine} is obtained. Train the refined HMM λ_{refine} with the previous set O_K of multiple observation sequences until it converges. Details of this step are described in Algorithm 2.

$$\pi_j = \frac{\sum_{i=1}^{K} \frac{1}{P_i}(\gamma_{1j}^{(i)} + \sum_{t=1}^{T_i-1} \sum_{k=1,k\neq h}^{N} \xi_{t,k,j}^{(i)})}{Z}, \tag{8}$$

$$a_{jk} = \frac{\sum_{i=1}^{K} \frac{1}{P_i} \sum_{t=1}^{T_i-1} \xi_{t,j,k}^{(i)}}{\sum_{l=N+1}^{N+n} \sum_{i=1}^{K} \frac{1}{P_i} \sum_{t=1}^{T_i-1} \xi_{t,j,l}^{(i)}}, \tag{9}$$

$$b_{jm} = \frac{\sum_{i=1}^{K} \frac{1}{P_i} \sum_{t=1, o_t^{(i)}=u_m}^{T_i} \gamma_{tj}^{(i)}}{\sum_{i=1}^{K} \frac{1}{P_i} \sum_{t=1}^{T_i} \gamma_{tj}^{(i)}}. \tag{10}$$

Algorithm 2. Training the refined model

```
Input: HMM $\lambda$ with $N$ states, heavy state $q_h$, multiple sequences $O_K$={$O^{(1)}$, $O^{(2)}$, ..., $O^{(K)}$}
1: Initialize a random HMM $\lambda_{random}$ with $n$ states
2: For each $j$=1, 2, ..., $N$, $j \neq h$, replace transition $q_j q_h$ to $q_j q_{N+1}$, $q_j q_{N+2}$, ..., $q_j q_{N+n}$ and
   transition $q_h q_j$ to $q_{N+1} q_j$, $q_{N+2} q_j$, ..., $q_{N+n} q_j$
3: Record the heavy state $q_h$ as a refined state $q_{refine}$ and remove the observation emission prob-
   ability vector from it. For each $j$=$N$+1, $N$+2, ..., $N$+$n$, set state label $p_j$=$p_h$
4: while not converged do
5:     For each $j$=1, 2, ..., $N$, $j \neq h$, $k$=$N$+1, $N$+2, ..., $N$+$n$, update $a_{jk}$=$a_{jh}\pi_k$, $a_{kj}$=$a_{hj}$
6:     For each $j$=$N$+1, $N$+2, ..., $N$+$n$, update $\pi_j$=$\pi_h\pi_j$
7:     For each $j$, $k$=$N$+1, $N$+2, ..., $N$+$n$, update $a_{jk}$ = $a_{hh}a_{jk}$
8:     For each $j$=$N$+1, $N$+2, ..., $N$+$n$, re-estimate $\pi_j$ by Eq. (8)
9:     For each $j$, $k$=$N$+1, $N$+2, ..., $N$+$n$, re-estimate $a_{jk}$ by Eq. (9)
10:    For each $j$=$N$+1, $N$+2, ..., $N$+$n$, $m$=1, 2, ... $M$, re-estimate $b_{jm}$ by Eq. (10)
11: end while
12: Perform the operations of Lines 5-7 once again
Output: HMM $\lambda$ with $N$-1+$n$ states
```

Above, Z is the normalization factor of π_j, and $\xi_{t,j,k}$=$P(s_t$=q_j, s_{t+1}=$q_k|O$, $\lambda)$ is the transition probability of HMM λ from state q_j to state q_k at time t, which can be efficiently computed by the forward-backward procedure [24].

4. Compute empirical visit rate v_i and empirical state risk r_i for each state q_i in the refined HMM λ_{refine}. Under the coverage bound C_B, compute the reject subset RS of the refined HMM λ_{refine} to identify which state is the heavy state q_h. If there is a heavy state, go to Step 3.

Prediction. Given a trained multi-stream sHMM and a new observation sequence O=$\{o_1, o_2, ..., o_T\}$, we predict the last label l_T in the relative label sequence L according to O through recursively finding the most probable state q_{most}. The prediction process consists of the following steps:

1. Find the most probable state q_{most} at the last time T by computing γ_{Ti} of all states in the root HMM λ_0.
2. If q_{most} is a refined state $q_{refined}$, reset the most probable state q_{most} by computing γ_{Ti} of new states added to the next level refined HMM λ_{refine}, and go to Step 2.
3. If q_{most} is in the reject subset RS, no prediction is made; otherwise, the label p_{most} of state q_{most} is output as the prediction result of the last label l_T in the relative L according to O.

Large-Scale Evaluation. As the parameters of HMM are randomly initialized, and the train algorithm such as Baum-Welch [1] is sensitive to the initial parameters, it may converge to different local maxima for different initializations. So we may get different predictions with the same training and test sequences. To reduce the random effect caused by parameter initialization, we run the algorithm a number N_{rs} of times, and evaluate the averaged empirical error rate as the performance measure. To make the evaluation efficient, we adopt the MapReduce [6] framework. The procedure is outlined in Algorithm 3. In the Map stage, given a coverage bound C_B, we train a multi-stream sHMM (by the

Algorithm 3. Large-scale performance evaluation

```
def MAP(id $i$, coverage $C_B$)
    $e \leftarrow$ PREDICT(TRAIN($C_B$))
    EMIT(coverage $C_B$, error $e$)
end def

def REDUCE(coverage $C_B$, errors $[e_1, e_2, \ldots]$)
    $e_{avg} \leftarrow$ AVERAGE($[e_1, e_2, \ldots]$)
    EMIT(coverage $C_B$, error $e_{avg}$)
end def
```

TRAIN method) using different parameter values, then do prediction (by the PREDICT method) based on the trained model, and get the error rate for different parameter values. In the Reduce stage, we compute the averaged error rate for the given C_B.

5 Experimental Evaluation

In this section, we present experimental evaluation results. First, we introduce experimental datasets and computing environment. Then, we present and analyze the results of GCA.

Finally, we compare our method with seven existing approaches to demonstrate the advantage of our method.

5.1 Experimental Setup

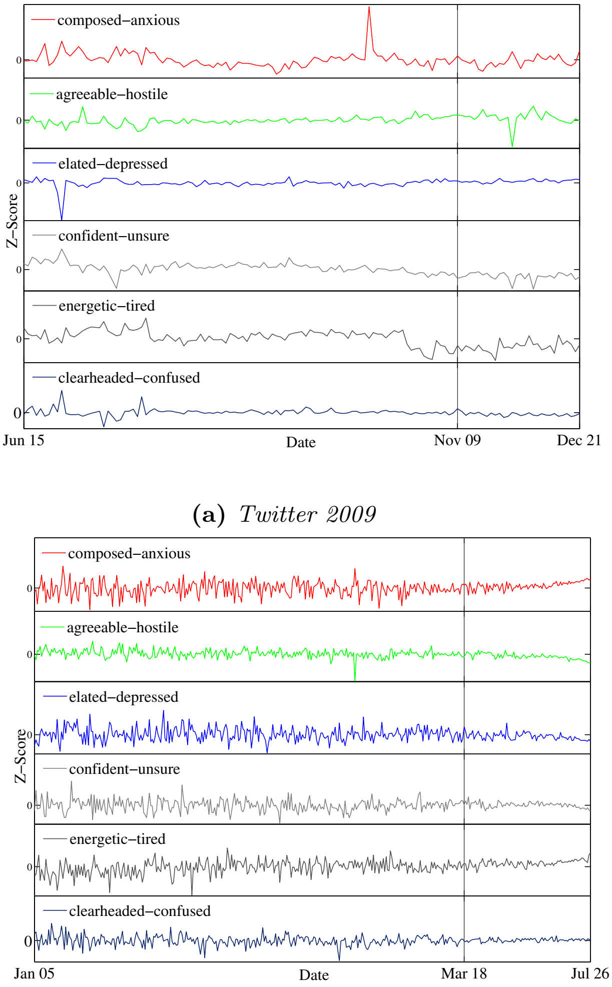
There are two Twitter datasets used in our experiments. The first one is from [35]. It contains 467 million Twitter posts that were published on Twitter in a seven-month period from Jun. 2009 to Dec. 2009. We call it *Twitter2009*. The second is from [13]. It contains 50 million tweets that cover a 20 month period from Jan. 2010 to Aug. 2011. We call it *Twitter2011*.

The financial data used are the S&P500 Index and NYSE Composite Index from Yahoo! Finance. For *Twitter2009* that covers the time period from 06/12/2009 to 12/21/2009, we train the model with data of the first 100 days (before 11/09/2009), and test the model using data of the next 30 days. As for *Twitter2011* that covers the time period from 01/04/2010 to 07/26/2011, we train the model with data of the first 300 days (before 03/13/2011), and test the model using data of the next 90 days. Here, we focus on predicting the trend of daily close price data.

We implemented our method in the MapReduce framework at a Hadoop platform, which was built on a Hadoop cluster that consists of 1 namenode/jobtracker and 24 datanodes/tasktrackers. Each node is equipped with an Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz and 4GB RAM.

5.2 Statistics of Extracted Twitter Moods

We utilize Algorithm 1 to extract Twitter moods from massive tweets, and keep the extracted daily Twitter moods during the period of trading days. For these

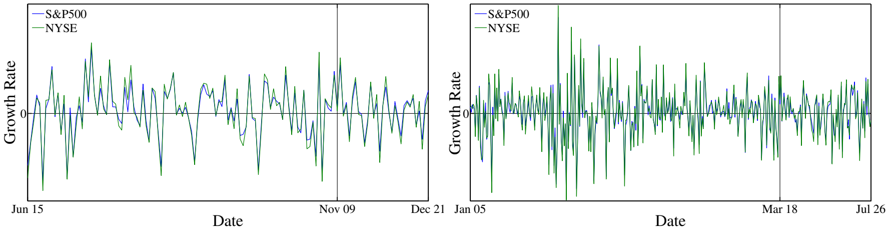


(a) *Twitter 2009*

(b) *Twitter 2011*

Fig. 2. Extracted Twitter Moods

days that have not enough tweets, we get the Twitter moods by linear interpolation. We compute the z-scores of all Twitter moods, including *composed-anxious* (**Com.**), *agreeable-hostile* (**Agr.**), *elated-depressed* (**Ela.**), *confident-unsure* (**Con.**), *energetic-tired* (**Ene.**) and *clearheaded-confused* (**Cle.**). The results of *Twitter 2009* and *Twitter 2011* are plotted in Fig. 2a and Fig. 2b respectively. We also discretize Tweet moods in daily trends, which will be fed to our multi-stream sHMM later.



(a) *Twitter 2009* **(b)** *Twitter 2011*

Fig. 3. S&P500 and NYSE growth rates

5.3 Growth Rates of Financial Indexes

Given the daily close price $close_t$ at day t, daily growth rate of day t is evaluated as follows:

$$growth_t = \frac{close_t - close_{t-1}}{close_{t-1}}. \tag{11}$$

The days with positive growth rates are labeled as up trends, while the days with negative growth rates are labeled as down trends. The computation results of S&P500 Index and NYSE Composite Index for *Twitter 2009* and *Twitter 2011* are shown in Fig. 3a and Fig. 3b respectively.

5.4 Results of Granger Causality Analysis

We perform GCA on the extracted Twitter moods via Eq. (1) and Eq. (2). We treat the S&P500 growth rate and NYSE Composite growth rate as the time series Y respectively, while taking each Twitter mood as the time series X. The *lag* range is set from 1 to 7. We present the p_{value} results of S&P500 and NYSE for *twitter2009* in Table 1 and for *twitter2011* in Table 2 respectively. By checking the results in the two tables, we select 3-day (*i.e.*, *lag*=3) lagged *agreeable-hostile* (**Agr**.) Twitter mood as the predictive indicator of public financial index due to following two reasons: 1) Most *p*-values under 3-day lagged **Agr**. Twitter mood achieve significant level (≤ 0.1), and their values are 0.159, **0.096**, **0.071** and **0.054**, respectively. 2) All *p*-values under 3-day lagged **Agr**. Twitter mood decrease significantly ($difference$ <-0.25) comparing to *p*-values under 2-day lagged **Agr**. Twitter mood. The differences are -0.278, -0.276, -0.284 and -0.409, respectively.

5.5 Prediction Performance Comparison

We compare our method with seven existing methods, in which six methods exploit Twitter mood. These six methods and our method all incorporate 3-day lagged **Agr**. Twitter mood to predict public financial trend. The six methods are:

Table 1. p_{value} results of S&P500 and NYSE for *Twitter2009* (all $p_{value}{}^* \le 0.1$)

Lag	S&P500						NYSE					
	Com.	**Agr.**	**Ela.**	**Con.**	**Ene.**	**Cle.**	**Com.**	**Agr.**	**Ela.**	**Con.**	**Ene.**	**Cle.**
1	0.704	0.226	0.681	0.696	0.535	0.270	0.739	0.179	0.756	0.625	0.529	0.385
2	0.764	0.437	0.648	0.588	0.722	0.305	0.851	0.372	0.664	0.444	0.746	0.417
3	0.228	0.159	0.856	0.276	0.741	0.338	0.231	$\mathbf{0.096}^*$	0.876	0.238	0.772	0.489
4	0.234	0.233	0.516	0.386	0.886	0.127	0.214	0.134	0.615	0.349	0.900	0.232
5	0.379	0.389	0.515	0.315	0.966	0.159	0.348	0.258	0.569	0.275	0.974	0.241
6	0.301	0.145	0.186	0.439	0.949	0.180	0.277	$\mathbf{0.061}^*$	0.228	0.405	0.948	0.277
7	0.428	0.148	0.331	0.262	0.955	0.218	0.364	$\mathbf{0.094}^*$	0.418	0.231	0.941	0.296

Table 2. p_{value} results of S&P500 and NYSE for *Twitter2011* (all $p_{value}{}^* \le 0.1$)

Lag	S&P500						NYSE					
	Com.	**Agr.**	**Ela.**	**Con.**	**Ene.**	**Cle.**	**Com.**	**Agr.**	**Ela.**	**Con.**	**Ene.**	**Cle.**
1	0.352	0.153	0.991	0.565	0.223	0.596	0.401	0.209	0.811	0.584	0.137	0.542
2	0.690	0.355	0.924	0.450	$\mathbf{0.082}^*$	0.747	0.707	0.463	0.885	0.463	$\mathbf{0.060}^*$	0.772
3	0.876	$\mathbf{0.071}^*$	0.897	0.415	0.172	0.842	0.855	$\mathbf{0.054}^*$	0.950	0.409	0.132	0.821
4	0.886	0.131	0.963	0.524	0.241	0.525	0.864	$\mathbf{0.099}^*$	0.986	0.490	0.216	0.490
5	0.929	0.215	0.981	0.647	0.328	0.498	0.913	0.174	0.993	0.629	0.270	0.475
6	0.872	0.309	0.994	0.705	0.266	0.559	0.837	0.261	0.999	0.646	0.156	0.523
7	0.885	0.476	0.999	0.524	0.109	0.621	0.840	0.413	1.000	0.472	$\mathbf{0.071}^*$	0.587

1. **VAR**. The Vector Autoregressive (**VAR**) framework [29] treats historical financial data and Twitter moods as an integrated vector to make prediction based linear regression.
2. **HMM**. Hidden Markov model (**HMM**) [24] considers two states of "up" and "down", and treats both historical financial data and Twitter moods as observation sequences to make prediction.
3. **CRF**. Conditional Random Field (**CRF**) [12] model also considers "up" and "down" states, and uses historical financial data and Twitter moods as observation sequences to make prediction.
4. **SVM**. Support Vector Machine (**SVM**) [5] combines both historical financial data and Twitter moods as a feature vector to make prediction.
5. **NN**. Neural Network (**NN**) [31] uses two nodes of "up" and "down" as output layer, and puts both historical financial data and Twitter moods to input layer to make prediction.
6. **cDPM**. It was proposed in [27], which uses a topic-model based approach to extract Twitter sentiments and then combines historical financial data and Twitter sentiments into an autoregressive framework.

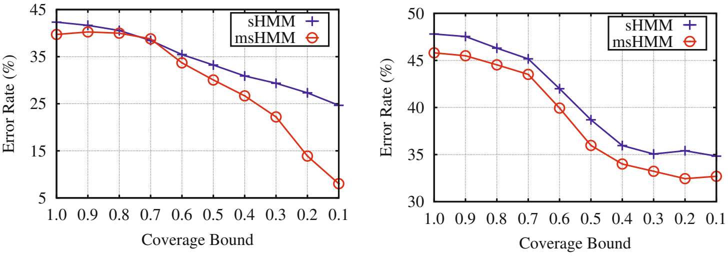
The only compared method that does not use Twitter mood is sHMM. It was developed in [23], which uses sHMM for financial trend prediction without using any social media information. This is the best existing model for financial trend prediction with controllability. For discrimination, we call our method **msHMM** as it uses multiple sequences for training.

Comparison with existing methods using Twitter mood. As **msHMM**'s performance is adjustable by the coverage bound C_B, we set four values between 1.0 and 0.1 for C_B to evaluate **msHMM**. A smaller C_B value means that **msHMM**

Table 3. Comparison with six methods using Twitter mood

Model	**Error Rate (%)**			
	Twitter2009		*Twitter2011*	
	S&P500	**NYSE**	**S&P500**	**NYSE**
VAR	*26.667*	*33.333*	46.667	44.444
HMM	36.667	46.667	44.444	54.444
CRF	40.000	40.000	45.556	44.444
SVM	40.000	46.667	50.000	44.444
NN	36.667	46.667	*37.778*	*32.222*
cDPM	40.000	43.333	48.889	38.889

msHMM(C_B)	39.715(1.0)
	30.055(0.5)
	22.209(0.3)*
	8.033(0.1)
		45.802(1.0)
		35.972(0.5)
		33.227(0.3)*
		32.694(0.1)
			45.796(1.0)
			42.408(0.5)
			36.622(0.2)*
			35.380(0.1)
				45.802(1.0)
				35.972(0.5)
				31.869(0.1)*
				31.869(0.1)

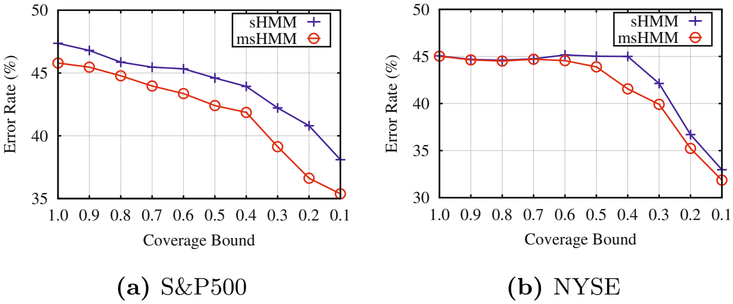


(a) S&P500 **(b)** NYSE

Fig. 4. Risk Coverage Curves for *Twitter2009*

puts more restriction on prediction output, which leads to a smaller error rate. For the six existing methods, we tune their parameters to get the best results. All experimental results are presented in Table 3. We can see that among the six existing methods, **VAR** has the smallest error rates (26.667% for S&P500, 33.333% for NYSE) for *Twitter2009* and **NN** obtains the smallest error rates (37.778% for S&P500, 32.222% for NYSE) for *Twitter2011*. For the four cases, by reducing C_B's value to 0.3, 0.3, 0.2 and 0.1 respectively, **msHMM** can get smaller error rates (22.209%, 33.227%, 36.622% and 31.869% respectively) than the six existing methods. And with C_B=0.1 **msHMM** achieves the lowest error rate on all two datasets. More importantly, the error rate of **msHMM** is controllable, while the six existing methods do not have such a feature.

*Comparison with **sHMM**.* For different coverage bound C_B values from 1.0 to 0.1, we first run **sHMM** on the S&P500 and NYSE index data, and then combine historical financial data and 3-days lagged agreeable-hostile Twitter mood to run our method **msHMM**. The results are plotted in Fig. 4 and Fig. 5, which show the *Risk Coverage* (RC) curves for both **sHMM** and **msHMM**. When taking a smaller coverage, **msHMM** rejects to make predictions if not confident enough, so a smaller error rate is obtained. For example, as shown in Fig. 4a, when C_B=0.1, **sHMM** gets a 24.67% error rate, while **msHMM** achieves a 8.03% error rate. From Fig. 4 and 5, we can see that our method



(a) S&P500 **(b)** NYSE

Fig. 5. Risk Coverage Curves for *Twitter2011*

msHMM obviously outperforms **sHMM**. On the one hand, given a certain error rate, **msHMM** can achieve a larger coverage than **sHMM**; On the other hand, given a certain coverage bound, **msHMM** can obtain a smaller error rate than **sHMM**. Another advantage of **msHMM** over **sHMM** is that **msHMM** lays down a way to utilize more additional indicators to boost financial trend prediction performance.

6 Conclusion

We proposed to utilize Twitter moods to boost financial trend prediction based on sHMM. First, we used the POMS Bipolar lexicon expanded by WordNet to extract six-dimensional society moods from large scale tweets, then we performed GCA between financial index and each Twitter mood to determine which Twitter mood has the most predictive power on financial index. Finally, we extended sHMM to combine financial index and Twitter moods to predict next-day trend. Experiments on the S&P500 and NYSE Composite index show that our method with 3-days lagged agreeable-hostile Twitter mood not only performs better than the state-of-the-art methods, but also provides a controllability mechanism to financial trend prediction.

Note that the major contribution of our work is combining financial data and Twitter moods into sHMM. We treat financial trend and Twitter mood trends as multiple observation sequences generated by sHMM. In this work, we use bivariate GCA to determine which Twitter mood has the most predictive power. For future work, we will explore multivariate GCA to select the optimal combination of multiple Twitter moods to improve prediction performance. Furthermore, we will investigate more advanced data combination methods for sHMM, and try other prediction models with controllability.

Acknowledgments. This work was partially supported by the Key Projects of Fundamental Research Program of Shanghai Municipal Commission of Science and Technology under grant No. 14JC1400300, and the Innovation Research Program of Shanghai

Municipal Education Commission under grant No. 13ZZ003. Jihong Guan was supported by National Natural Science Foundation of China (NSFC) under grant No. 61373036.

References

1. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The annals of mathematical statistics **41**(1), 164–171 (1970)
2. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of Computational Science **2**(1), 1–8 (2011)
3. Brand, M.: Coupled hidden markov models for modeling interacting processes. Tech. rep., MIT (1997)
4. Chow, C.K.: On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory **16**(1), 41–46 (1970)
5. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995). http://dx.doi.org/10.1007/BF00994018
6. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008)
7. El-Yaniv, R., Wiener, Y.: On the foundations of noise-free selective classification. The Journal of Machine Learning Research **11**, 1605–1641 (2010)
8. Gilbert, E., Karahalios, K.: Widespread worry and the stock market. In: Proceedings of the Fourth International Conference on Weblogs and Social Media, pp. 59–65 (2010)
9. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society **37**(3), 424–438 (1969)
10. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)
11. Idvall, P., Jonsson, C.: Algorithmic trading: hidden markov models on foreign exchange data. Master's thesis, Södertörn University (2008)
12. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289 (2001). http://dl.acm.org/citation.cfm?id=645530.655813
13. Li, R., Wang, S., Deng, H., Wang, R., Chang, K.C.C.: Towards social user profiling: unified and discriminative influence model for inferring home locations. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1023–1031 (2012). http://doi.acm.org/10.1145/2339530.2339692
14. Li, X., Wang, C., Dong, J., Wang, F., Deng, X., Zhu, S.: Improving stock market prediction by integrating both market news and stock prices. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part II. LNCS, vol. 6861, pp. 279–293. Springer, Heidelberg (2011)
15. Lin, Y., Guo, H., Hu, J.: An svm-based approach for stock market trend prediction. In: The 2013 International Joint Conference on Neural Networks, pp. 1–7 (2013)
16. Liu, B.: Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing, 2nd edn (2010)

17. Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers (2012)
18. McNair, D.M., Lorr, M., Droppleman, L.F.: Profile of mood states. Educational and Industrial Testing Service (1971)
19. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
20. Mittal, A., Goel, A.: Stock prediction using twitter sentiment analysis. Tech. rep., Stanford University
21. Nofsinger, J.R.: Social mood and financial economics. The Journal of Behavioral Finance **6**(3), 144–160 (2005)
22. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**(1–2), 1–135 (2008). http://dx.doi.org/10.1561/1500000011
23. Pidan, D., El-Yaniv, R.: Selective prediction of financial trends with hidden markov models. In: Advances in Neural Information Processing Systems, pp. 855–863 (2011)
24. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
25. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 105–112 (2003)
26. Schumaker, R.P., Chen, H.: A discrete stock price prediction engine based on financial news. Computer **43**(1), 51–56 (2010)
27. Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., Deng, X.: Exploiting topic based twitter sentiment for stock prediction. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (vol. 2: Short Papers), pp. 24–29 (2013)
28. Sprenger, T.O., Tumasjan, A., Sandner, P.G., Welpe, I.M.: Tweets and trades: the information content of stock microblogs. European Financial Management (2013). http://onlinelibrary.wiley.com/doi/10.1111/j.1468-036X.2013.12007.x/abstract
29. Stock, J.H., Watson, M.W.: Vector autoregressions. The Journal of Economic Perspectives **15**(4), 101–115 (2001). http://www.jstor.org/stable/2696519
30. Sun, X.Q., Shen, H.W., Cheng, X.Q.: Trading network predicts stock price. Scientific Reports **4**(3711), 1–6 (2014)
31. Trippi, R.R., Turban, E.: Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance. McGraw-Hill, Inc (1992)
32. Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Opinionfinder: a system for subjectivity analysis. In: Proceedings of HLT/EMNLP on Interactive Demonstrations, pp. 34–35 (2005)
33. Wu, D., Ke, Y., Yu, J.X., Yu, P.S., Chen, L.: Detecting leaders from correlated time series. In: Kitagawa, H., Ishikawa, Y., Li, Q., Watanabe, C. (eds.) DASFAA 2010. LNCS, vol. 5981, pp. 352–367. Springer, Heidelberg (2010)
34. Yang, B., Guo, C., Jensen, C.S.: Travel cost inference from sparse, spatio temporally correlated time series using markov models. Proc. VLDB Endow. **6**(9), 769–780 (2013). http://dx.doi.org/10.14778/2536360.2536375
35. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proceedings of the fourth ACM international conference on Web search and data mining, pp. 177–186 (2011)
36. Zhang, Y.: Prediction of financial time series with Hidden Markov Models. Master's thesis, Simon Fraser University (2004)