Predicting Stock Price Trends using News Headlines
(10-701 Final Report)

Xiaote Zhu
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
xiaotez@andrew.cmu.edu

Sunny Nahar
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
anahar@andrew.cmu.edu

Abstract

We consider the task of predicting S&P 500 stock price movements, as a binary response variable, from news headline data. We experimented with two models: a logistic regression using bag-of-words headline features, and a Markov Random Field to incorporate pairwise stock correlations. We observe an accuracy of 49.91% using just headlines features and an improvement of 53.42% when including pairwise correlations. The results using pairwise correlations and both headlines and correlation were statistically significant.

1 Introduction

News media has a tremendous effect on the stock market prices [1]. Financial news articles are read everyday by millions, interpreted by companies, and analyzed by automated stock algorithms at quantitative trading firms. Key news and data about company trades, revenues, market climates, and geopolitical conflicts strongly influence trading and market transactions. In our project, we address the problem of predicting stock price trends using news headlines. We want to determine the extent to which we can predict the movement of stock prices using features extracted from news headlines. As shown in Figure 1, we want to correlate words with increase or decrease.

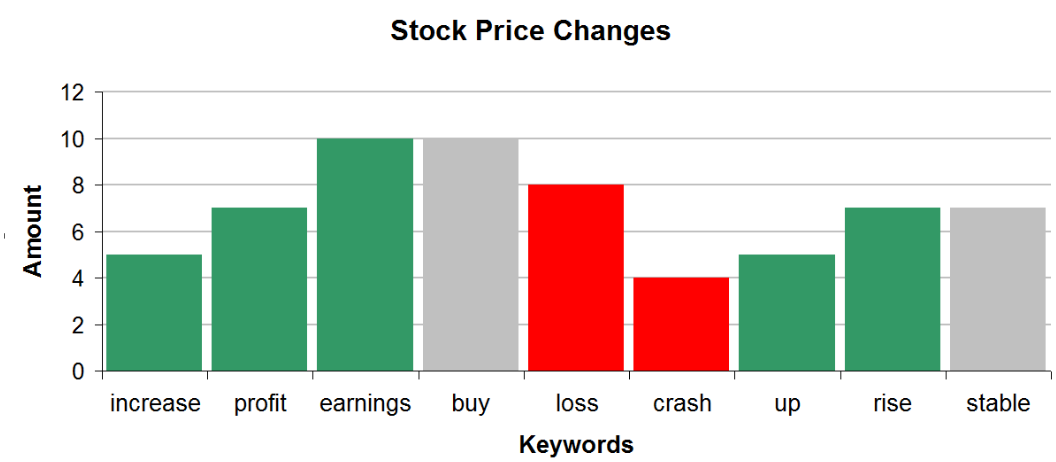

Figure 1: Increase and decrease of stock price when certain keywords are observed in headlines.

Typically, features are extracted from news articles using a bag-of-words representation of the article's text; that is, each article is represented as a vector of counts, where each dimension corresponds to the number of times a certain word appears in the article. However, strictly using the words in an article makes the feature space susceptible to noise and large variance. We want to identify keywords which will have the largest impact in influencing stock price trends. We argue that news headlines, are a proxy to news events –typically 8 to 10 words, headlines capture the essence of a news article, and will contain important keywords relating to the event. Hence, in our project, we will focus on using headline keywords as our main features.

On the other hand, we recognize that stock price predictions is an extremely difficult task. The "efficient market hypothesis" tells us that in financial markets, profit opportunities are exploited as soon as they arise, hence stock prices follow a random walk and are extremely difficult to predict [2]. However, Lavrenko et al. [3] found that in the stock market setting, generating profitable action signals (buy and sell) is more important than accurately predicting future values of a time series. Furthermore, stock prices are heavily influenced by "technical" indicators such as company revenues and profits, corporate strategy and market demands [4, 5]. Hence, we simplify our task to predict just price movements – day-to-day up and down changes – as opposed to predicting prices, and try to determine if news headlines have sufficient predictive ability in this setting.

2 Related Work

Stock market prediction is an area of very active research. Attempts have been made to predict stock prices using technical information such as company technicals [4–6] and price history [7–11]. Recent work has begun to incorporate natural language text data into prediction – for example, using full-text of news articles [12, 13], hidden Markov model incorporating news headlines [14] and sentiment in social media [15–18]. Most of these works has worked with short term price movements (on the order of minutes), while we will focus on end of day stock prices; we find that a longer time horizon is less susceptible to noise, and more beneficial to the common investor.

3 Predicting stock movements

Problem definition Suppose we are given news headlines, $\boldsymbol{h}_{di}$, represented as a bag-of-words vector, for each stock $i = 1, \ldots, N$ on day $d = 1, \ldots, D$. Our prediction task is to predict the movement of the stock end-of-day price. We treat this as a binary classification task, where we predict the label $s_{di} = -1$ if the stock moved lower on day d (i.e price on day d is lower than price on day $d-1$), and $s_{di} = +1$ otherwise.

Prediction with headlines We used the ℓ_1-regularized logistic regression to predict stock movements using our headlines. For a given stock i on a given day d, the logistic model assumes that

$$p(s_{di} = -1 \mid \boldsymbol{w}_i) = \frac{1}{1 + \exp(\boldsymbol{w}_i^\top \boldsymbol{h}_{di})}$$

where $\boldsymbol{w}_i$ is a weights vector corresponding to each stock. The weights in $\boldsymbol{w}_i$ signifies the relative contribution of each word in the day d's headlines to the log-odds of the stock movement. To learn $\boldsymbol{w}_i$, we maximize the regularized log-loss training objective

$$\mathcal{L}(\boldsymbol{s}_i, \boldsymbol{w}_i) = \sum_d \left[\frac{1}{D} \sum_{d=1}^{D} \frac{(s_{di}+1)}{2} \boldsymbol{w}_i^\top \boldsymbol{h}_{di} \right] - \eta \|\boldsymbol{w}_i\|_1$$

where η is a regularization constant to control the sparsity of the weight vector. We used the ℓ_1 regularizer, which is akin to placing a Laplacian prior with variance η^{-1} [19, 20]. The Laplacian prior encourages sparsity and is important because of the high dimensionality of our headlines [21].

The objective function is convex but requires special treatment due to non-differentiability when any elements are zero; we use the Orthant-Wise Limited-memory Quasi-Newton algorithm (OWL-QN) [22] to solve $\boldsymbol{w}_i$ for each stock independently. OWL-QN is a variant of L-BFGS [23] which is scalable to high dimensions, and uses a projection method to constrain the search space within the correct orthant. The OWL-QN algorithm requires computation of a gradient w.r.t $\boldsymbol{w}_i$, which is

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{s}_i, \boldsymbol{w}_i)}{\partial \boldsymbol{w}_i} &= \frac{1}{D}\sum_{d=1}^{D} \boldsymbol{h}_{di} - \frac{\exp(\boldsymbol{w}_i)}{\sum_v \exp(w_{iv})} - \eta \frac{\partial \|\boldsymbol{w}_i\|_1}{\partial \boldsymbol{w}_i} \\ &= \langle \boldsymbol{h}_i \rangle - \frac{\exp(\boldsymbol{w}_i)}{\sum_v \exp(w_{iv})} - \eta \frac{\partial \|\boldsymbol{w}_i\|_1}{\partial \boldsymbol{w}_i}\end{aligned} \tag{1}$$

We see that the optimization is similar to *moment matching* between the empirical and expected means of the headlines vector (if we did not have regularization).

Modeling pairwise stock correlations It is widely known that certain subsets of stocks are highly correlated [24]. Hence, we decide to model the correlations between stock movements using a Markov Random Fields (MRF) [25]. A pairwise MRF allows us to capture such pairwise correlation in stock movements, while letting us incorporate headlines information in a systematic manner. Suppose $S_i \in \{-1, +1\}$ is a random variable that denotes the movement of stock i on a single day. Our MRF can be represented by the undirected graph $G = (V, E)$, where V indexes the set of random variables $\mathcal{S} = \{S_1, S_2, \dots\}$ and an edge $(i, j) \in E$ denotes the pairwise correlation between S_i and S_j. We define our conditional distribution over $\mathcal{S}$ as

$$p(\mathcal{S} = \{s_1, s_2, \dots\} \mid \boldsymbol{\theta}, \boldsymbol{w}) = \frac{1}{Z_{\boldsymbol{\theta}}} \prod_{(i,j)\in E} \phi_{ij}(s_i, s_j; \boldsymbol{\theta}_{ij})$$

Here, s_i is the observed value of S_i, $\boldsymbol{\theta}_{ij}$ corresponds to the parameters of the pairwise potential function $\phi_{ij}(\cdot, \cdot; \boldsymbol{\theta}_{ij})$, and $Z_{\boldsymbol{\theta},\boldsymbol{w}}$ is the *partition function*

$$Z_{\boldsymbol{\theta}} = \sum_{\boldsymbol{s}\in\{-1,+1\}^{|\mathcal{S}|}} \prod_{(i,j)\in E} \phi_{ij}(s_i, s_j; \boldsymbol{\theta}_{ij})$$

which is defined as the sum of potentials of all possible assignments of stock movements. Since the potential functions are positive, we can parametrize the potentials as log-linear models, where

$$\phi_{ij}(s_i, s_j; \boldsymbol{\theta}_{ij}) = \exp\left(\boldsymbol{\theta}_{ij}^{\top} \boldsymbol{f}(s_i, s_j)\right)$$

and $\boldsymbol{\theta}_{ij} \in \mathbb{R}^4$ are the parameters, and $\boldsymbol{f} : \{-1, +1\} \times \{-1, +1\} \to \{0, 1\}^4$ is a feature function that maps each of the 4 possible configurations to a binary vector on the 4-dimensional $\{0, 1\}$-simplex. Thus, we can learn $\boldsymbol{\theta}$ by optimizing the ℓ_2-regularized training objective

$$\mathcal{L}(\boldsymbol{s}, \boldsymbol{\theta}) = \frac{1}{N}\sum_{i,j}\sum_{d=1}^{N} \boldsymbol{\theta}_{ij}^{\top}\boldsymbol{f}(s_{di}, s_{dj}) - \log Z_{\boldsymbol{\theta}} - \frac{\lambda}{2}\sum_{i,j} \|\boldsymbol{\theta}_{ij}\|_2^2$$

Like before, we place an ℓ_2-regularizer, λ, which is like a Gaussian prior with variance λ^{-1} on the parameters to avoid overfitting. We use the limited-memory BFGS algorithm [23] to find the maximum *a posteriori* estimate of $\boldsymbol{\theta}$. L-BFGS is a quasi-Newton optimization algorithm which uses an approximation of the inverse Hessian matrix, while storing only a few vectors to approximate the Hessian implicitly (thus limited-memory). It requires a gradient for the maximization, which is

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{s}, \boldsymbol{\theta}_{ij})}{\partial \boldsymbol{\theta}_{ij}} =& \frac{1}{N}\sum_{d=1}^{N} \boldsymbol{f}(s_{di}, s_{dj}) - \frac{\exp(\boldsymbol{\theta}_{ij})}{\sum_{k=1}^{4} \exp(\theta_{ijk})} - \lambda\boldsymbol{\theta}_{ij} \\ =& \langle \boldsymbol{f}(s_i, s_j) \rangle - \frac{\exp(\boldsymbol{\theta}_{ij})}{\sum_{k=1}^{4} \exp(\theta_{ijk})} - \lambda\boldsymbol{\theta}_{ij}\end{aligned} \tag{2}$$

where $\langle \boldsymbol{f}(s_i, s_j) \rangle$ is the expected feature vector of pairwise movements between s_i and s_j. In other words, $\langle \boldsymbol{f}(s_i, s_j) \rangle$ can be seen as the empirical distribution of the features. Without the regularization term, the optimization will be akin to *moment matching*, where the first order conditions is satisfied by setting the gradient to zero.

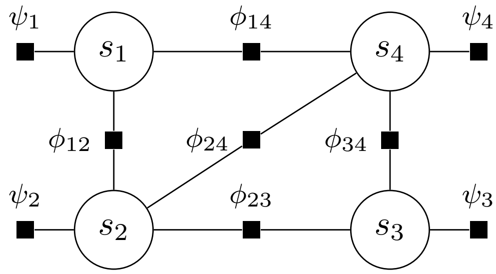

Figure 2: An example factor graph for our Markov random field with 4 stocks; ϕ_{ij} are pairwise factors (stock correlations) and ψ_i are unary factors (informed by headlines).

Combining headlines with stock correlation The choice of using an MRF provides us with a principled way to incorporate our background knowledge about the movement of each individual stocks (i.e headlines) into a joint model that takes into account the correlated movements between stocks. We incorporate the headlines as a unary factor in the MRF, with log-linear potentials

$$\psi_i(s_i; \boldsymbol{w}_i) = p(s_i \mid \boldsymbol{w}_i) \propto \exp(\boldsymbol{w}_i^\top \boldsymbol{h}_i)$$

The joint log-likelihood of our data is thus

$$\begin{aligned}\mathcal{L}(\boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{w}) = \sum_{(i,j)} &\left[\frac{1}{N} \sum_{d=1}^{N} \boldsymbol{\theta}_{ij}^\top \boldsymbol{f}(s_{di}, s_{dj}) - \frac{\lambda}{2} \|\boldsymbol{\theta}_{ij}\|_2^2 \right] \\ &+ \sum_i \left[\frac{1}{N} \sum_{d=1}^{N} \frac{(s_{di}+1)}{2} \boldsymbol{w}_i^\top \boldsymbol{h}_{di} - \eta \|\boldsymbol{w}_i\|_1 \right] \\ &- \log Z_{\boldsymbol{\theta},\boldsymbol{w}} \end{aligned} \tag{3}$$

where $Z_{\boldsymbol{\theta},\boldsymbol{w}}$ is now the partition function taking into account $\boldsymbol{w}$. The gradients for (3) remain the same (see (1) and (2)).

We used `liblbfgs`, a C implementation of L-BFGS and OWL-QN available at `https://github.com/chokkan/liblbfgs` for our joint optimizations. Figure 2 presents an example factor graph representation of our MRF.

Inference With parameters estimated from data, our inference task is to find the best assignment of stock movements to every stock based on the day's headlines, and pairwise correlations. Exact inference in our MRF model thus requires evaluating over all $2^{|\mathcal{S}|}$ stock movement assignments to find the maximum likelihood assignment. This is intractable, hence we resort to using an approximate inference algorithm – Metropolis-Hastings [26]. Metropolis-Hastings (MH) algorithm is a Markov chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution.

Given some sample of stock movements $\boldsymbol{s}^{(t)}$, we uniformly selected a stock, s_u and reversed its movement (i.e $\boldsymbol{s}'_u = -s_u$).[1] Thus, $\boldsymbol{s}' = \{s_1^{(t)}, \ldots, s'_u, \ldots\}$ is our proposed state, which we choose to accept or reject depending on the ratio r of its likelihood to $\boldsymbol{s}^{(t)}$, i.e $r = \frac{p(\boldsymbol{s}')}{p(\boldsymbol{s}^{(t)})}$. Therefore if $r \geq 1$, we accept the state, and update the model. Otherwise if $r < 1$, then we accept the state with probability r, and update the model. We note that the likelihood ratio between two states whose

[1] We note that this proposal distribution is symmetric.

difference is stock s_u depends only on its Markov blanket. Hence,

$$\frac{p(\boldsymbol{s}')}{p(\boldsymbol{s}^{(t)})} = \exp\left(\log \psi_u(s'_u) + \sum_{j<u} \log \phi_{uj}(s'_u, s'_j) + \sum_{j>u} \log \phi_{ju}(s'_j, s'_u) \right.$$
$$\left. - \log \psi_u(-s'_u) - \sum_{j<u} \log \phi_{uj}(-s'_u, s_j^{(t-1)}) - \sum_{j>u} \log \phi_{ju}(s_j^{(t-1)}, -s'_u)\right) \quad (4)$$

Algorithm 1 describes our Metropolis-Hastings procedure in detail.

Algorithm 1 Metropolis-Hastings algorithm for inference.

Initialization: Pick an initial state $\boldsymbol{s}^0$ at random.
for iteration $t = 1, \ldots, T$ **do**
  $u \leftarrow$ UnifInt(1, N) ▷ Select a stock index at random from uniform integers.
  $\boldsymbol{s}' = [s_1^{(t-1)}, \ldots, -s_u^{(t-1)}, \ldots, s_N^{(t-1)}]$ ▷ Flip the state of stock u.
  $r \leftarrow \frac{p(\boldsymbol{s}')}{p(\boldsymbol{s}^{(t)})}$ ▷ Acceptance probability accoding to Eq. (4).
  if Unif$(0, 1) < \min(1, r)$ **then** ▷ Flip a coin to decide if we accept the proposed state.
    $s^{(t)} \leftarrow \boldsymbol{s}'$
  else
    $s^{(t)} \leftarrow \boldsymbol{s}^{(t-1)}$
  end if
end for

For our experiments, we generated 1000 samples from the posterior, discarded the first 500 samples for burn-in, and discarded every 5th sample to reduce auto-correlation. The remaining samples are used to compute the posterior mean assignment.

4 Experiments

Data Set Our data set consists of news headlines related to 466 S&P 500 stocks and their end of day prices from Feb. 25, 2008 to June 21, 2013.[2] We have selected these stocks such that they all have at least 1,000 data points for stock prices. The average percentage of +1 label for each stock is 52% with a standard deviation of around 1.5. Figure 3 shows the distribution of our headline data.

For headlines, we removed stop-words, punctuations and lowercased them. Furthermore, we removed tokens that appeared less than 5 times in our corpus resulting in a vocabulary size of 30,011.

We extract features from headlines using a bag-of-words representation. To compensate for data sparsity (many stocks have no headlines on a given day), we concatenate all the headlines on a given day to form our feature vector. For example, a feature for stock i on day d (i.e feature vector $\boldsymbol{h}_{di}$) would be "`APPL_sell=1`", which represents the word "sell" appeared 1 time in a headline related to stock ticker AAPL.

Stock movement prediction We split our data randomly into 70% for training and 30% testing. We experimented with predicting stock movements using the different models described in §3. Table 1 illustrates the prediction accuracies of our model. Using headlines only (HEADLINES), the accuracy is just below 50%. This shows that the model consisting of simply the headline factors was insufficient. When considering just correlations between pairs of stocks (CORRELATION), we obtained an accuracy of 0.5328, which is better than the headline data. When both features are considered (BOTH), the accuracy improved slightly (but significantly) to 0.5342. When comparing the top 6 most accurately predicted stocks for BOTH, we see that the list is similar to that of CORRELATIONS (but quite different from HEADLINES). This means that headlines provide useful signals in stock price movements, although the bulk of the accuracy improvements is attributed to the stock correlations. Table 2 presents the top stocks (by accuracy) that our models predict.

[2] We obtained news headlines using Dow Jones API available at `http://betaweb.dowjones.com/api/`. Stock data was downloaded from `http://pages.swcp.com/stocks/`.

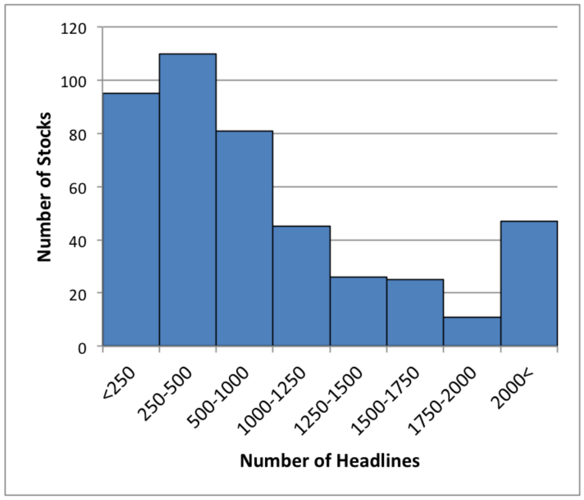

Figure 3: Distribution of headlines data. There are an average of 944 headlines for 466 stocks.

Model	Accuracy
MAJORITY	0.5207
HEADLINES	0.4991 $\pm$ 0.0245
CORRELATION	0.5328 $\pm$ 0.0214
BOTH	**0.5342** $\pm$ 0.0210

Table 1: Accuracies of our different models. All results are statistically different from random and each other at $p < 10^{-10}$ using the paired t-test.

Unigram features extracted from headlines did not perform better than random guessing. There are a number of possible explanations why this may be the case. When using a bag-of-words model, the context is lost; words are reduced to independent identities, and certainly, words in a headline are not independent. Additionally, our unigram model does not take into account metadata associated with stock. For instance, words relating to stocks in a particular industry may have stronger influence than the same words in another industry. We were able to get statistically significant results when using pairwise correlation. With more work, we can improve upon the existing model. We will discuss some of our ideas for future work in §5.

Stock correlations From our MRF, we can analyze the edges that exhibits the strongest correlations between stock pairs. We measure the "related"-ness of stock pairs by calculating for each stock pair i and j:

$$\textsc{related}(i, j) = \frac{\phi_{ij}(-1,-1) + \phi_{ij}(+1,+1)}{\phi_{ij}(-1,+1) + \phi_{ij}(+1,-1)}$$

Table 3 lists such top 10 stock pairs. We see that these highly related stock pairs are in the same industries, which matches widely held beliefs that stocks in the same industry move together [24].

5 Future Work

There are numerous exciting opportunities for future improvements. Since our headline data was sparse, we can look into getting more headline data by using general business related articles, instead of just financial news related to stocks. General news articles will contain signals from non-financial events that may have an influence on the stock market (i.e political events, riots, etc). Moreover, we miss out contextual information by just using unigrams. This can be remedied by exploring higher order $n-$grams, such as bi-grams and tri-grams. Although this would make the feature space sparser, we can use phrase extraction techniques such as that proposed by Justeson and Katz [27].

HEADLINES		CORRELATION		BOTH	
Stock	Acc	Stock	Acc	Stock	Acc
INTC	0.5671	PLD	0.5896	PLD	0.5920
AET	0.5572	AMAT	0.5846	AMAT	0.5895
AES	0.5547	PLL	0.5821	GT	0.5845
TYC	0.5547	IR	0.5797	LLY	0.5820
HIG	0.5522	GT	0.5796	IR	0.5820
BTU	0.5522	FITB	0.5796	PLL	0.5796

Table 2: Top 6 stocks by accuracy for each of our models.

Stock pair	Industry	RELATED
AVB:EQR	Construction	5.872
DHI:LEN	Construction	5.634
HCP:VTR	Real estate	5.541
BXP:VNO	Real estate	5.537
CSX:NSC	Transportation	5.036
HOT:MAR	Hotel	4.956
ALTR:XLNX	Electronics	4.925
ADI:LLTC	Electronics	4.899
BTU:CNX	Energy	4.826
BHI:SLB	Energy	4.801

Table 3: Top 10 "related" stock pairs

Our project focused on data from a large period of time spanning six and a half years. The stock market is highly unpredictable, and its behavior changes over time. We can look into modelling shorter periods of time, such as a month or a week. This may be a promising route since local fits such as these may be much better at modelling the current situation. Also, data is released continuously every day, so getting the previous week's or month's data should not be an issue.

Another point of interest is grouping stocks by industry. Our model included stocks from a variety of industries. It is possible that certain words were conflated in our model which would not be conflated in industry specific models. These would essentially be MRF's consisting of only the stocks associated to one industry. It is likely that similar words would be used within the industry, so the feature space of headlines would not be as large. Moreover, coupling this with a bi-gram or tri-gram feature space could improve accuracy. More difficult endeavors include trying to predict percentage or absolute increase.

6 Conclusion

While headlines by itself are insufficient in predicting stock price movements, we find that it can be used as a feature in joint models to improve prediction. In this paper, we experimented with using headlines as side information in a pairwise Markov random field, which captures stock correlations. We find that in such a setting, headlines were able to contribute to better predictive performance. Our results reflect the difficult task of forecasting the stock market. Like previous work [12, 13, 16], we achieved comparable performance using limited noisy information. Given the proliferation of digital information related to companies today, the key challenge would be synthesizing different sources of data in a principled and efficient manner in a computational model. In future work, we hope to consider including more data – historical data and stock metadata – building upon a more sophisticated model that can capture the influences of varied sources of information.

References

[1] Paul C. Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168, 2007. ISSN 1540-6261. doi: 10.1111/j.1540-6261.2007.01232.x. URL http://dx.doi.org/10.1111/j.1540-6261.2007.01232.x.

[2] Halbert White. Economic prediction using neural networks: The case of ibm daily stock returns. In *Neural Networks, 1988., IEEE International Conference on*, pages 451–458. IEEE, 1988.

[3] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Mining of concurrent text and time series. In *IN PROCEEDINGS OF THE 6 TH ACM SIGKDD INT'L CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING WORKSHOP ON TEXT MINING*, pages 37–44, 2000.

[4] John Y Campbell and Robert J Shiller. Stock prices, earnings, and expected dividends. *The Journal of Finance*, 43(3):661–676, 1988.

[5] Mark L Mitchell and Erik Stafford. Managerial decisions and long-term stock price performance*. *The Journal of Business*, 73(3):287–329, 2000.

[6] Pei-Chann Chang and Chen-Hao Liu. A tsk type fuzzy rule based system for stock price prediction. *Expert Systems with Applications*, 34(1):135–144, 2008.

[7] Takashi Kimoto, Kazuo Asakawa, Morio Yoda, and Masakazu Takeoka. Stock market prediction system with modular neural networks. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 1–6. IEEE, 1990.

[8] Ken-ichi Kamijo and Tetsuji Tanigawa. Stock price pattern recognition-a recurrent neural network approach. In *IJCNN International Joint Conference on Neural Networks*, pages 215–221. IEEE, 1990.

[9] Youngohc Yoon and George Swales. Predicting stock price performance: A neural network approach. In *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, volume 4, pages 156–162. IEEE, 1991.

[10] Steven H. Kim and Se H. Chun. Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index. *International Journal of Forecasting*, 14(3):323–337, 1998.

[11] Ping-Feng Pai and Chih-Sheng Lin. A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505, 2005.

[12] Gyozo Gidofalvi and Charles Elkan. Using news articles to predict stock price movements. Technical report, Department of Computer Science and Engineering, University of California, San Diego, 2001.

[13] Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.

[14] Rowland O. Flaherty and Woody Folsom. Predicting stock trends using natural language processing of headlines. Technical report, 2012.

[15] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[16] Malcolm Baker and Jeffrey Wurgler. Investor sentiment in the stock market. Working Paper 13189, National Bureau of Economic Research, June 2007. URL `http://www.nber.org/papers/w13189`.

[17] Werner Antweiler and Murray Z Frank. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294, 2004.

[18] Sitaram Asur and Bernardo A Huberman. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 492–499. IEEE, 2010.

[19] Mário AT Figueiredo. Adaptive sparseness for supervised learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1150–1159, 2003.

[20] Kenneth Lange and Janet S. Sinsheimer. Normal/independent distributions and their applications in robust regression. *Journal of Computational and Graphical Statistics*, 2(2):pp. 175–198, 1993. URL `http://www.jstor.org/stable/1390698`.

[21] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[22] Galen Andrew and Jianfeng Gao. Scalable training of ℓ_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40. ACM, 2007.

[23] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989. doi: 10.1007/BF01589116. URL http://dx.doi.org/10.1007/BF01589116.

[24] Benjamin F King. Market and industry factors in stock price behavior. *Journal of Business*, pages 139–190, 1966.

[25] Ross Kindermann, James Laurie Snell, et al. *Markov random fields and their applications*, volume 1. American Mathematical Society Providence, RI, 1980.

[26] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[27] John S. Justeson and Slava M. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27, March 1995.