Expert Systems With Applications 114 (2018) 388–401

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An investor sentiment reward-based trading system using Gaussian inverse reinforcement learning algorithm

Steve Y. Yang*, Yangyang Yu, Saud Almahdi

Financial Engineering Program, School of Business, Stevens Institute of Technology, 1 Castle Point on Hudson, Hoboken, NJ 07030, USA

ARTICLE INFO

Article history:
Received 12 February 2018
Revised 29 May 2018
Accepted 27 July 2018
Available online 29 July 2018

Keywords:
Investor sentiment
Inverse reinforcement learning
Support vector machine learning
Sentiment reward

ABSTRACT

Investor sentiment has been shown as an important factor that influences market returns, and a number of profitable trading systems have been proposed by taking advantage of investor sentiment signals. In this paper, we aim to design an investor sentiment reward-based trading system using Gaussian inverse reinforcement learning method. Our hypothesis is that while markets interact with investor's sentiment, there exists an intrinsic mapping between investor's sentiment and market conditions revealing future market directions. We propose an investor sentiment reward based trading system aimed at extracting only signals that generate either negative or positive market responses. Such a reward extraction mechanism is based not only on market returns but also market volatility representing a succinct and robust feature space. The back-test results show that the proposed sentiment reward-based trading system is superior to various benchmark strategies on S&P 500 index and market-based ETFs as well as few other existing news sentiment-based trading signals. Moreover, we find that sentiment reward trading system is much more effective in a volatile market, but it is sensitive to transaction costs.

1. Introduction

Many studies have emerged aiming to explain the phenomenon that irrational decisions tend to be biased on the same direction rather than holding rational expectations (De Long, Shleifer, Summers, & Waldmann, 1990; Kahneman & Tversky, 1979; Shiller, 2003; Shiller, Fischer, & Friedman, 1984). Investor sentiment has become a focus of behavioral finance in recent years (Antoniou, Doukas, & Subrahmanyam, 2013; Barber & Odean, 2012; Sun, Najand, & Shen, 2016; Tetlock, 2007; Yang, Mo, & Zhu, 2014). On the one hand it directly challenges the assumption that participants of the financial markets are rational but on the other hand, it has inspired researchers to design novel trading strategies to exploit premiums caused by investor's irrational behaviors driven by the market sentiment.

The aim of this research is to reveal the intrinsic relationship between investor sentiment and market returns using the inverse reinforcement learning (IRL) approach by designing an effective trading system. In this study we answer three questions: (i) What is the interaction mechanism between investor sentiment and financial market? (ii) Is there an inherent mapping between this interaction mechanism and future financial market directions? (iii) Can we take advantage of this interaction mechanism to consistently beat the market? Most of the effort in this field has been focused on building direct relationships between investor sentiment proxies and future financial market movements (Bollen, Mao, & Zeng, 2011; Kurov, 2010; Yang, Song, Mo, Datta, & Deane, 2015), where these relationship indicators are then used to design trading strategies (Yang, Mo, Liu, & Kirilenko, 2017). However, in this paper we model the financial market dynamics as a Markov decision process and regard investor sentiment as a series of actions taken at different market states. Assuming there exists an intrinsic market reward function governing this process, we then extract the reward function using Gaussian process based inverse reinforcement learning algorithm (Qiao & Beling, 2011), and use the rewards to forecast directions of the future marketx returns.

The intuition of this approach rests on the evidence that investor sentiment has significant influence on financial market movement. When investors are optimistic about the market, they tend to long assets and contribute to the boom of the market. When investors are pessimistic about the future of the market, they tend to take short positions on assets and contribute to the downturn of the market. Investor sentiment would also adjust itself to these market movements when the asset values deviate too far from the fundamental values. During this process, we posit that the reward function from the inverse reinforcement learning framework is "the most succinct, robust, and transferable defini-

* Corresponding author.
E-mail addresses: steve.yang@stevens.edu (S.Y. Yang), yyu8@stevens.edu (Y. Yu), salmahdi@stevens.edu (S. Almahdi).

https://doi.org/10.1016/j.eswa.2018.07.056
0957-4174/

tion of the task" (Abbeel & Ng, 2004), assuming the market as a whole follows a Markov process. In other words, the reward function is a feature that can be extracted from the observations of the past interactions between investor sentiment and financial market movement. This process naturally filters out the sentiment signals that do not generate market reactions, and hence the reward-based signals should improve the quality of the forecast where the reward function contains cleaner information of such interactions than the raw sentiment measures at least. Based on this intuition, we hypothesize that the market sentiment reward function should be a good feature space for predicting future financial market directions. Then we can design a profitable strategy based on the sentiment rewards extracted from this inverse learning process.

In this paper, we use news sentiment from Thomson Reuters news analytics database as the proxy of investor sentiment toward the general U.S. market. We apply Gaussian process based inverse reinforcement learning (GPIRL) method (Qiao & Beling, 2011) on past observations of market states and investor sentiment shocks. In this process, a preference graph is constructed to capture the distinctive state transition choices of the Markov decision process. One unique feature of this GPIRL method is that it requires less data observations than other linear IRL methods, moreover it is less susceptible to observation noise which is very prevalent when dealing with financial market data. We then compare the performance of the sentiment-reward signals with other sentiment signals such as the raw sentiment score (Feuerriegel & Prendinger, 2016; Sun et al., 2016; Yang et al., 2014), sentiment shock (Song, Liu, & Yang, 2017; Yang, Song et al., 2015), and sentiment trend (Song, Liu et al., 2017) using support vector machines (SVM) method to generate trading signals. The out-of-sample tests show significant improvement over the existing sentiment signals. Moreover, we examine the sensitivity of the sentiment reward-based signal using different machine learning methods such as Random Forest, Boosting, k-NN (k-nearest neighbors), Decision Tree and Bagging along with SVM method. We show that the SVM method combined with the sentiment reward signals outperforms all the other methods, suggesting an optimal sentiment reward-based trading system.

The major contribution of the paper is to propose a news sentiment based trading system where the trading signals are generated based on market's rewards to investor sentiment. We argue investor sentiment signals are mostly very noisy, and often markets do not react to these signals. There are two sources of noise that prevent better model forecasting: a) news sentiment is only a proxy of the "true" investor sentiment given the investor sentiment is unobservable, and proxies are mostly deployed to indirectly measure the underlying investor sentiment (Antweiler & Frank, 2004; Barber & Odean, 2012); b) news sentiment measure itself is noisy where many expressions may be used in news articles, but not all accurately describe market conditions (Feuerriegel, Heitzmann, & Neumann, 2015; Feuerriegel & Neumann, 2013). Although efforts to filtering the noise existing in sentiment measures show improvement (Song, Almahdi, & Yang, 2017), their effect of applying direct filtering is rather limited (Song, Liu et al., 2017). In this study we propose an investor sentiment reward-based trading system aimed at extracting only signals that generate either negative or positive market responses. This filtering mechanism is realized through training a model based on how market gives rewards to sentiment signals using a supervised learning method. Because if the market does not reward certain news sentiment signals, the learned model will automatically filter out such sentiment signals. As a result, the model will only capture the effect of "true" investor sentiment on the market. Moreover, such a reward extraction mechanism is based not only on market returns but also market volatility representing a succinct and robust feature space. To the best of our knowledge, this approach is the first in representing investor sentiment influence to market returns in the reward space. It can be broadly applied to other market sentiment proxy based trading systems.

The rest of the paper is organized as follows. In Section 2, we review the literature about investor sentiment and news sentiment, as well as machine learning and reinforcement learning based trading strategies. Then we introduce the Gaussian process based inverse reinforcement learning technique and construct state space and action space and present our proposed trading strategy in Section 3. In Section 4, we assess the performance of this sentiment reward-based trading system with other existing sentiment filtering methods along with some popular passive trading strategies. In the last section we make a conclusion of our research and highlight the contributions of the paper.

2. Literature review

In this section we review two strands of related literature to set the background of our work. First, we review the reinforcement learning and inverse reinforcement learning approaches applied in the financial market forecasting and trading. In the second part we examine the recent work in combining investor sentiment and machine learning methods in building trading systems.

2.1. Reinforcement learning and related trading strategies

Reinforcement learning is a method based on finding the optimal policy with the knowledge of state space, action space, transition mechanism and reward function. In applying artificial intelligence to stock market investment, scholars have investigated how to combine reinforcement learning and other machine learning methods to design optimal trading rules. Lee (2001) developed a stock prediction approach where the author applied the temporal difference TD(0) method which is a prediction based reinforcement learning algorithm to predict the stock market returns . The author combined the TD(0) method with a neural network in order to minimize the mean squared error (MSE), and showed that reinforcement learning methods can be used with other neural networks for time series forecasting. Kuremoto, Obayashi, and Kobayashi (2007) proposed a self-organized fuzzy neural network (SOFNN) with reinforcement learning applying the Stochastic Gradient Ascent (SGA) for time series forecasting where the input layer in the neural network is the historical data and the hidden layer consists of a Gaussian membership functions. Forecasting is based on a probabilistic policy which determines actions and the forecasting error is identified as the reward function. Many scholars also applied reinforcement learning in market actions. Instead of only predicting prices, scholars also tried to determine the next optimal action to take in the market. Neuneier (1996) applied a Q-learning method to optimize a portfolio. The author developed a neural network to predict price movement and then applied Q-learning to find an optimal policy. In another paper (Neuneier, 1998), the author enhanced Q-learning in portfolio optimization by introducing the neural network with 8 hidden neurons and a linear output. A trading algorithm by Sharpe ratio maximization was investigated by Gao and Chan (2000) where the author utilized Q-learning in FX market trading by back-propagating the TD error from the Q-value using neural network in order to find an optimal action. The action selection is based on the Boltzman distribution using two parameters only where the action can be either 1 or 0. The transaction cost of 0.5% was applied only when there are changes between currencies. Scholars have also tested mixing reinforcement learning with other machine learning methods or technical

analysis strategies. In a paper by Chen, Mabu, Hirasawa, and Hu (2007), the authors applied genetic network programming method with SARSA model which is a reinforcement learning algorithm (GNP-RL). The authors added a technical index that would deliver information to the GNP-RL where SARSA will learn the Q-values and determine buying and selling timings. A different approach to reinforcement learning in trading is the deep Q-learning algorithm introduced by Zhai et al. (2016) where the authors developed a deep Q-network trading system that would use Q-learning to develop a loss function. Then they trained a deep network using a loss function obtained by the Q-values. The authors concluded that deep Q-learning can extract more features and learn online.

Inverse reinforcement learning (IRL) problem is aimed at extracting a reward function given observed actions, which are assumed to be the optimal actions from experts in a Markov decision process Ng and Russell (2000). The IRL problem was first introduced by Ng and Russell (2000) in a machine learning setting with a Markov decision process. The authors aimed to identify a specific reward function using linear programming and remove the degenerative solutions by maximally differentiating between the observed policy and the sub-optimal policies. Abbeel and Ng (2004) studied the apprenticeship learning through inverse reinforcement learning. They converted the maximizing reward function problem into a linear combination of known features with specific relative weightings. The authors used this mechanism to reflect the apprenticeship learning in practice showing that learners trade off different factors in order to learn from experts. Ramachandran and Amir (2007) incorporated a Bayesian formula framework into the inverse reinforcement learning problem instead of the traditional linear approximation of the reward function. They considered the background information about the specific IRL problem into the prior probability of the reward function. Then the authors updated the prior probability using observations of the expert action. A probabilistic approach based on the principle of maximum entropy to solve the IRL problem was proposed by Ziebart, Maas, Bagnell, and Dey (2008). This algorithm is very good in dealing with the noise and imperfect behavior demonstrated by the expert and it is well suited in modeling real world driving behaviors. Another very important branch in the literature of IRL research is the Gaussian process-based IRL (GPIRL) model. This branch can be dated back to the work of Ramachandran and Amir (2007). Qiao and Beling (2011) proposed the IRL with Gaussian process based on the Bayesian framework. By assigning a Gaussian prior on the reward function. They changed this IRL problem into a convex quadratic programming where it can be efficiently solved. At the same time the Gaussian process is also very good in dealing with noisy observations generated by incomplete policies in practical situations. Another very important contribution of their research is that they incorporated a preference graph in the action space to represent multiple observations in the same state. This technique has a great meaning for practical interest as it successfully describes the non-deterministic policies for a certain state. Yang, Qiao, Beling, Scherer, and Kirilenko (2015) successfully applied the Gaussian process-based IRL technique into finance research. Identifying algorithmic trading strategies in the financial market, they constructed the feature space that can best capture the nature of traders decision making process. The authors incorporated a preference graph in describing the non-deterministic nature of the observed multiple trading behaviors in the same market states where the reward function is achieved by solving the GPIRL problem. They applied this method to the E-Mini S&P 500 Futures market data. The result showed that the accuracy of identifying high frequency trading strategy according to the reward function is higher than the traditional statistic-based trader classification approaches.

2.2. Investor sentiment based trading systems

The literature on investor sentiment has been growing very fast during the last decade partly due to the advancement in news and social media information processing technologies. Baker and Wurgler (2006) examined the impact of investor sentiment to different categories of firm's future returns. They found a pattern implying that when the financial market is optimistic, stocks attracting optimists and speculators; like small stocks, young stocks, high volatility stocks, unprofitable stocks, non-dividend-paying stocks, extreme growth stocks and distressed stocks usually have low returns in the following periods. Yu and Yuan (2011) found that investor sentiment has an obvious influence on the mean-variance trade-off for portfolio management. There is a strong positive trade-off when investor sentiment is low, but there is no obvious relationship when investor sentiment is high. In analyzing the interaction mechanism between financial market conditions and investor sentiment, Yang, Liu, Chen, and Hawkes (2017) employed a multivariate Hawkes process model to reveal the complex self-excitation and cross excitation relationships between investor sentiment and market returns. Their empirical results are based on the 15 minutes time scale of *S&P*500 return data and Thomson Reuters News sentiment data from 2008 to 2014. In measuring investor sentiment quantitatively, Muhammad, Wiratunga, and Lothian (2016) designed a lexicon-based sentiment measure system that captures the polarity of content both from words interaction with their textual neighborhood and from text genre. Day and Lee (2016) applied deep learning technique into investor sentiment analysis with financial news from different sources. They empirically showed that the forecast ability of different proxies varies with different financial news sources.

A number of artificial intelligence trading support systems based on sentiment prediction have emerged in the current literature. Nasseri, Tucker, and de Cesare (2015) proposed a way to combine text-mining techniques, feature selection and decision tree algorithms to develop a superior trading system. They first extracted semantic terms expressing sentiments from micro-blogging messages. Then they generated trading decisions using the decision tree algorithms and a filtering approach for feature selection. An experiment using the Dow Jones Industrial average was performed to demonstrate the success of this trading system. In a paper by Pröllochs, Feuerriegel, and Neumann (2016), the main contribution was exploring the impact of negation scope detection on sentiment analysis of financial news. A reinforcement learning approach was proposed to predict negation scopes. Reinforcement learning was compared with Hidden Markov Model and conditional random fields approaches. The goal of the comparison was to explore how the sentiment analysis of financial news can be improved in terms of classification accuracy. In conclusion, reinforcement learning leads to a better classification accuracy compared to other machine learning methods as well as in terms of sentiment analysis where an improvement in correlation between sentiment value and stock returns was achieved. Feuerriegel and Prendinger (2016) developed an automated decision-making system based on supervised and reinforcement learning methods using news-based sentiment data with the addition of price momentum. Using random forest as the supervised learning strategy and Q-learning as the reinforcement learning method the authors were able to find quantitative evidence that a news trading system can successfully make profitable investment decisions. Ho, Damien, Gu, and Konana (2017) tested the time varying nature of social media sentiment effect on stock prediction. The main focus is to examine possible dynamic relationship between social media sentiments and future stock returns. The authors studied the relevance between sentiment and stock prediction where sentiments are treated as time-varying. A combination of Bayesian Dynamic Linear

Models and Seemingly Unrelated Regression was used to tackle the problem. In conclusion, they found that the impact of sentiment over future stock returns varies over time. Their results show that the time-varying sentiments coefficients are more stable in 2011 compared to 2009 due to the high volatility in the time of the recession when applied to the empirical analysis of the Dow Jones Industrial Average. Oliveira, Cortez, and Areal (2017) tested the effectiveness of the microblogging data in stock trading where the sentiment and survey indicators were extracted from microblogging data to predict stock returns, volatility and trading volume. The authors used Kalman-Filter to merge microblog and survey data. Regression was applied using five different machine learning methods on a rolling window of data, and Diebold–Mariano method was used to test the usefulness of sentiment and attention based predictions. Checkley, Higón, and Alles (2017) modeled the forecast-ability of micro-blogging sentiment metrics to stock price return, volatility and volume. The authors found that a predictive time-horizon of minutes yields better results than hours or days and that sentiment metrics are better at predicting the volatility and volume. The test was applied on five major stocks and a limited evidence of Granger-causality was found from social media sentiment metrics to the stock market behavior in returns, volatility and volume.

3. Methodology and data

In this section, we propose a sentiment reward-based inverse reinforcement learning approach to estimate the reward function based on past observations. We use support vector machine (SVM) to classify these rewards into up-trend or down-trend indicators. We also discuss how to estimate the reward function using Gaussian process based inverse reinforcement learning technique, followed by how to construct the state space and action space in our case as well as the design of a trading strategy. At last, we design a trading system according to these indicators and the market conditions.

3.1. Inverse reinforcement learning

Here we consider a finite countable MDP defined as a tuple $\mathbf{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r)$, where

- $\mathcal{S} = \{s_n\}_{n=1}^{N}$ is a set of N states. Denote $\mathcal{N} = \{1, 2, \cdots, N\}$.
- $\mathcal{A} = \{a_m\}_{m=1}^{M}$ is a set of M actions. Denote $\mathcal{M} = \{1, 2, \cdots, M\}$.
- $\mathcal{P} = \{\mathbf{P}_{a_m}\}_{m=1}^{M}$ is a set of state transition probabilities.
- $\gamma \in [0, 1]$ is a discount factor.
- r is the reward function defined as

$$r(s_n, a_m) \triangleq \sum_{n' \in \mathcal{N}} \mathbf{P}_{a_m}(s_n, s_{n'}) r(s_n, a_m, s_{n'}). \tag{1}$$

The reward function depends on both state and action. We denote the $\mathbf{r}$ function as

$$\begin{aligned} \mathbf{r} =& (\mathbf{r}_1(s_1), \cdots, \mathbf{r}_1(s_N), \cdots, \mathbf{r}_M(s_1), \cdots, \mathbf{r}_M(s_N)) \\ =& (\mathbf{r}_1, \cdots, \mathbf{r}_M) \end{aligned} \tag{2}$$

In the reinforcement learning framework, a rational agent takes actions at each state aiming at maximizing the value function. We use the following theorems:

Theorem 1. *(Bellman Equations): Given a stationary policy π, $\forall n \in \mathcal{N}$, $m \in \mathcal{M}$, $V^{\pi}(s_n)$ and $Q^{\pi}(s_n, a_m)$ satisfy*

$$V^{\pi}(s_n) = r(s_n, \pi(s_n)) + \gamma \sum_{n' \in \mathcal{N}} \mathbf{P}_{\pi(s_n)}(s_n, s_{n'}) V^{\pi}(s_{n'}), \tag{3}$$

$$Q^{\pi}(s_n, a_m) = r(s_n, a_m) + \gamma \sum_{n' \in \mathcal{N}} \mathbf{P}_{a_m}(s_n, s_{n'}) V^{\pi}(s_{n'}). \tag{4}$$

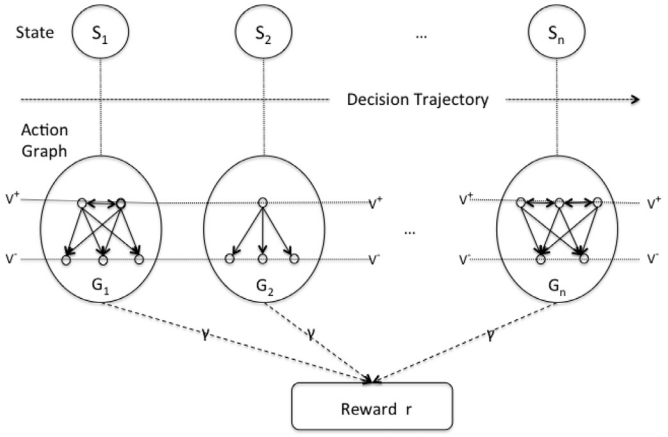

Fig. 1. Pairs of state and action preference graph (Yang, Qiao et al., 2015).

Theorem 2. *(Bellman Optimality): π is optimal if and only if $\forall n \in \mathcal{N}$, $\pi(s_n) \in arg\max_{a \in \mathcal{A}} Q^{\pi}(s_n, a)$.*

Based on the above theorems, we write the Q-function as:

$$Q^{\pi}(s_n, a_m) = \mathbf{r}_m(s_n) + \gamma \mathbf{P}_{a_m}(s_n, :)(\mathbf{I} - \gamma \mathbf{P}_{\pi})^{-1} \hat{\mathbf{I}} \mathbf{r} \tag{5}$$

where $\mathbf{P}_{\pi}$ denotes the state transition probability matrix by following an optimal policy π at every state, $\hat{\mathbf{I}}$ is an identity matrix with N rows and $N \times M$ columns.

We define action preference relation as:

1. Action $\hat{a}$ is weakly preferred to $\check{a}$, denoted as $\hat{a} \succeq_{s_n} \check{a}$, if $Q(s_n, \hat{a}) \geq Q(s_n, \check{a})$;
2. Action $\hat{a}$ is strictly preferred to $\check{a}$, denoted as $\hat{a} \succ_{s_n} \check{a}$, if $Q(s_n, \hat{a}) > Q(s_n, \check{a})$;
3. Action $\hat{a}$ is equivalent to $\check{a}$, denoted as $\hat{a} \sim_{s_n} \check{a}$, if and only if $\hat{a} \succeq_{s_n} \check{a}$ and $\check{a} \succeq_{s_n} \hat{a}$.

Here we define an action preference graph as a direct graph showing preference relationships among all actions at a certain state. $G_n = (\mathcal{V}_n, \mathcal{E}_n)$ represents an action preference graph at state s_n with nodes set $\mathcal{V}_n$ and edges set $\mathcal{E}_n$. In the above expression each node represents an action in $\mathcal{A}$, and each edge represents a preference relationship. In this study, if we observe action $\hat{a}$ at state s_n, we have $\hat{a} \succeq_{s_n} \check{a}, \forall \check{a} \in \mathcal{A} \backslash \{\hat{a}\}$. In Bellman optimality view, the action $\hat{a}$ is observed if and only if $\hat{a} \in arg\max_{a \in \mathcal{A}} Q(s_n, a)$. In other words,

$$Q(s_n, \hat{a}) > Q(s_n, \check{a}), \forall \check{a} \in \mathcal{A} \backslash \{\hat{a}\} \tag{6}$$

According to the above consideration, we use a two-layer directed graph to represent the action preference graph G_i at a certain state s_i (see Fig. 1). The top layer $\mathcal{V}_n^+$ contains the nodes that represent the observed actions. While the bottom layer $\mathcal{V}_n^-$ contains the nodes that represent the other actions. In our case the nodes in top layer denote the highest frequently observed actions in this certain state, and the nodes in bottom layer denote other less frequently observed actions in this state. Sometimes we may observe two equally frequent actions as shown in G_1 in Fig. 1. This can be easily extended to more nodes on the top layer. G_2 in Fig. 1 is a typical situation that only one action is most frequently observed.

Given the observed data $\mathcal{O}$ that contain the real financial market conditions and the corresponding investor sentiment, we can collect the action preference information at each state using the pairs of state and action preference graph (Yang, Qiao et al., 2015) (see Fig. 1).

Gaussian process based IRL is developed on the foundation of Bayesian IRL. The posterior probability of the reward given obser-

vation data is defined as:

$$p(r|\mathcal{O}) \propto p(\mathcal{O}|r)p(r) \propto \prod_{(s,a)\in\mathcal{O}} p(a|s,r). \tag{7}$$

The IRL problem is to find the reward which maximizes the posterior probability. In Gaussian process based IRL, we assume that the reward is contaminated by Gaussian noise. So we model the reward function by $r + \mathcal{N}(0, \sigma^2)$. Here we denote r_m as the reward of action a_m, which follows $\mathbf{r}_m \sim \mathcal{N}(0, \mathbf{K}_m)$. The $\mathbf{K}_m$ is a covariance matrix generated by function $k_m(s_i, s_j)$. The joint prior probability of reward is the product of multivariate Gaussian, $p(\mathbf{r}|\mathcal{S}) = \prod_{m=1}^{M} p(\mathbf{r}_m|\mathcal{S})$. Then $\mathbf{r} \sim \mathcal{N}(0, \mathbf{K})$, where $\mathbf{K}$ is a positive definite covariance matrix and it is block diagonal of $\{\mathbf{K}_1, \mathbf{K}_2, \cdots, \mathbf{K}_M\}$, which is based on the assumption that rewards of different actions are uncorrelated. More specific, we define the $k_m(s_i, s_j)$ as a squared exponential kernel function:

$$k_m(s_i, s_j) = e^{\frac{1}{2}(s_i - s_j)\mathbf{T}_m(s_i - s_j)} + \sigma_m^2 \delta(s_i, s_j), \tag{8}$$

where $\mathbf{T}_m = \kappa_m \mathbf{I}$. Here $\delta(s_i, s_j) = 1$ if $s_i = s_j$; otherwise $\delta(s_i, s_j) = 0$. The property of this covariance matrix is that the elements of this matrix are almost the same when the two inputs are very close in Euclidean space, and the value decreases as their distance increases.

We then have the likelihood function as:

$$p(\hat{a} \succ_{s_n} \check{a}|\mathbf{r}) = \Phi\left(\frac{Q(s_n, \hat{a}) - Q(s_n, \check{a})}{\sqrt{2}\sigma}\right), \tag{9}$$

$$p(\hat{a} \sim_{s_n} \hat{a}'|\mathbf{r}) \propto e^{-\frac{1}{2}(Q(s_n,\hat{a}) - Q(s_n,\hat{a}'))^2}. \tag{10}$$

In Eq. (9), $\Phi(x)$ is the standard Gaussian accumulative probability function. The property of Eq. (9) is that the more $Q(s_n, \hat{a})$ is larger than $Q(s_n, \check{a})$, the more it approaches 1. For Eq. (10), it equals 1 when $Q(s_n, \hat{a}) = Q(s_n, \hat{a}')$; otherwise it decreases.

In this study, we obtain statistics of the observed frequencies of the M sentiment actions at each of the N states. Then we rank the frequencies from the highest to the lowest at each state. If there are several actions having the same highest frequency at certain state, we use Eq. (10) to describe this probability. The other strict preference relationships at certain state are described in Eq. (9).

Then the likelihood function given the observation data $\mathcal{O}$ becomes

$$\begin{aligned} p(\mathcal{O}|\mathbf{r}) \propto & p(\mathcal{G}|\mathcal{S}, \mathbf{r}) = \prod_{n=1}^{N} p(G_n|s_n, \mathbf{r}) \\ & = \prod_{n=1}^{N} \prod_{k=1}^{n_n} p((\hat{a} \succ_{s_n} \check{a})_k|\mathbf{r}) \prod_{l=1}^{m_n} p((\hat{a} \sim_{s_n} \hat{a}')_l|\mathbf{r}). \end{aligned} \tag{11}$$

where n_n denotes the number of strictly preference at a certain state, and m_n denotes the number of equivalent preference.

Combining the above prior probability and likelihood probability along with the following Bayes rule:

$$p(\mathbf{r}|\mathcal{S}, \mathcal{G}, \theta) \propto p(\mathcal{G}|\mathcal{S}, \theta, \mathbf{r}) p(\mathbf{r}|\mathcal{S}, \theta), \tag{12}$$

we can reach at the following negative log posterior formula

$$\begin{aligned} U(\mathbf{r}) \triangleq & \frac{1}{2} \sum_{m=1}^{M} \mathbf{r}_m^T \mathbf{K}_m^{-1} \mathbf{r}_m - \sum_{n=1}^{N} \sum_{k=1}^{n_n} \ln \Phi\left(\frac{Q(s_n, \hat{a}) - Q(s_n, \check{a})}{\sqrt{2}\sigma}\right) \\ & + \sum_{n=1}^{N} \sum_{l=1}^{m_n} \frac{1}{2} (Q(s_n, \hat{a}) - Q(s_n, \hat{a}'))^2. \end{aligned} \tag{13}$$

So we only need to minimize the Eq. (13) to obtain the reward. Qiao and Beling (2011) have already proven Eq. (13) is a convex function.

In the above formula, we have a hyper-parameter vector $\theta = (\kappa_m, \sigma_m, \sigma)$. The κ_m and σ_m regulate the shape of covariance matrix and the σ controls the shape of Gaussian noise of the reward. Here we need to optimize the hyper-parameters by maximizing $\log p(\mathcal{O}|\theta, \mathbf{r}_{MAP})$, which is the Laplace approximation of $p(\theta|\mathcal{O})$.

Concluding the above description, we estimate the reward function by iteratively applying the following procedures:

1. Estimate $\mathbf{r}_{MAP}$ by minimizing Eq. (13), while fixing the values of hyper-parameter vector $\theta = (\kappa_m, \sigma_m, \sigma)$.
2. Optimize the hyper-parameter vector $\theta = (\kappa_m, \sigma_m, \sigma)$ by maximizing $\log p(\mathcal{O}|\theta, \mathbf{r}_{MAP})$, while fixing the values of $\mathbf{r}$.

3.2. Sentiment-reward based GPIRL

Using Gaussian process based inverse reinforcement learning method described above, we define a sentiment reward-based GPIRL approach to extract market return signals. In this system, we assume the market as a dynamic adaptive process. We assume a learning agent is observing how the market is responding to sentiment signals (actions) and aims to recover the market's reward function under different market states. As such, the reward is a functional mapping of the market states (market conditions) and investor sentiment signals (actions) per definition in Eq. (1). If there is a mapping between an investor sentiment signal and a market state, such a mapping can then be used in training a supervised model to capture future market directions. And we further assume the market responds to sentiment shocks following a Markov decision process, and its reward at each state is contaminated with a Gaussian noise process. Moreover, we also introduce an action preference graph to further constrain the action and state pairing for our particular problem. Using Eq. (13), we then extract the rewards at different market states. In the following sections, we describe the state space, action space, and the preference graph we choose for this study.

3.2.1. State space

As it is well-established, a market can be characterized by its return and volatility in general. Though higher moments of the return variable, such as Skewness and Excess Kurtosis, etc., also convey market information, their changes and implications are more subtle. Given the computational constraint, we have to make a trade-off between the complexity and computational tractability. We deffer the consideration of Skewness and Excess Kurtosis into future studies. In this study, we choose log return and volatility as our state variables to describe the financial market conditions. We calculate the volatility using the log return from $t_0 - 21$ to t_0, where t_0 is the current period. Then we discretize the values of each of the two variables into three levels:

- High level:

$$Value > \mu + \sigma, \tag{14}$$

- Neutral level:

$$\mu - \sigma \leq Value \leq \mu + \sigma, \tag{15}$$

- Low level:

$$Value < \mu - \sigma. \tag{16}$$

where μ is the mean value of the observation period and σ is the standard deviation of the observation period. Upon discretizing the two variables into three levels each, we obtain a total of $3^2 = 9$ states for the state space.

3.2.2. Action space

We define investor sentiment shocks to the market as actions. We then can naturally categorize them into three categories: positive, neutral, and negative. We discretize the sentiment polarity score based on a threshold T into the following three action types in the action space:

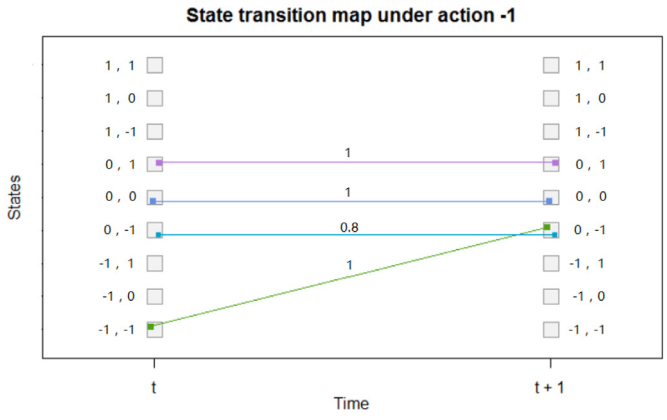

Fig. 2. State transition map under the action -1 at each state. Note: the numbers in the first column indicates return levels, and the numbers in the second column indicates volatility levels at both t and $t+1$.

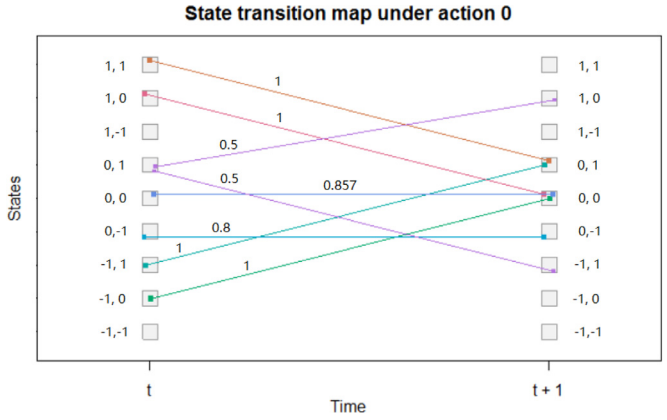

Fig. 3. State transition map under the action 0 at each state. Note: the numbers in the first column indicates return levels, and the numbers in the second column indicates volatility levels at both t and $t+1$.

- Action (positive) $+1$:

$$\sum_{i=t_0-N}^{t_0} \Delta S_i > T, \tag{17}$$

- Action (neutral) 0:

$$-T < \sum_{i=t_0-N}^{t_0} \Delta S_i < T, \tag{18}$$

- Action (negative) -1:

$$\sum_{i=t_0-N}^{t_0} \Delta S_i < -T. \tag{19}$$

Where t_0 denotes the current period, ΔS_i is the change of sentiment on period i, N is the moving window to sum the change of sentiment and T is the threshold value. We define the action as $+1$ if the cumulative value is larger than T. We define the action as -1 if the cumulative value is smaller than $-T$. Otherwise the action is 0. N and T are two variables controlling the percentage of the different action categories in the whole observation period. Here we adopt the values $N = 15$ and $T = 0.12$ as this can generate balanced number of positive, neutral and negative sentiment signals.

3.2.3. State transition graph

We obtain the state transition probability matrix by taking statistics of the state transition observations under each action type. In our study, we have four state transition probabilities under four different actions defined as a_1, a_2 and a_3. We define a^* as the action following the optimal policy at each state. The action value should be a ternary such as $\{1, 0, -1\}$.

Fig. 2 is the state transition map under action -1, which we only show the transitions whose probability is larger than or equal to 0.5. This map is drawn from the statistical values of the trajectory starting from 9:30 a.m. Jan 2nd, 2008. Fig. 3 is the state transition map under action 0, and Fig. 4 is the state transition map under action 1. Both of these two maps are drawn from the same observed trajectory as Fig. 2. From the three state transition maps we can see that the sentiment mainly has an influence on the change of return. When the return is low, the sentiment will lift the return and when the return is high the sentiment will depress the return. This phenomenon is consistent with the mean reversion theory.

3.3. Adaptive trading system with sentiment reward signals

The financial market is a dynamic system. Since we model the market and sentiment interaction as a Markov decision process,

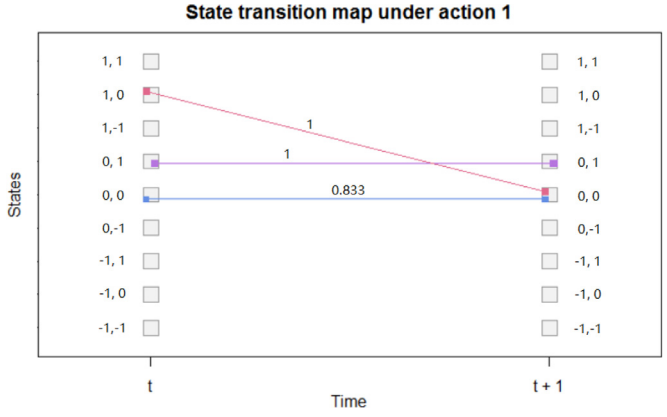

Fig. 4. State transition map under the action 1 at each state. Note: the numbers in the first column indicates return levels, and the numbers in the second column indicates volatility levels at both t and $t+1$.

it is critical that the Markov process captured in our training set is representative of the near future market mechanism. Assuming a stationary Markov decision process for a given period of time, we design a trading system based on a supervised learning model with reward function learned from the GPIRL model. In another word, for a given period of time, we assume the market and investor sentiment interaction mechanism as a stationary system. Otherwise the predictive power of classification models may be impaired. Therefore, we apply a moving window training scheme in our system assuming within the moving window the system is stationary. If the market gives the same reward to certain market condition, we then assume this reward can be used to predict future market directions under a stationarity assumption. Obviously, the sentiment signals that do not generate market rewards will be treated as noise and be filtered out through the modeling process. Moreover, we use the stop-loss market conditions to accommodate the non-stationarity nature of the financial market. Based on the findings from Lo and Remorov (2017), stop-loss mechanism will generate additional profit when market exhibits regime switch patterns. We therefore design an adaptive trading system with the GPIRL sentiment rewards as market direction signals and consider market regime change for retraining (see Fig. 5). This system consists of four major components as described in the following subsections.

As the first step of the trading system, we need to select an optimal sample size for the GPIRL learning (see step ① in Fig. 5). Given a fixed historical training dataset, we argue that when the

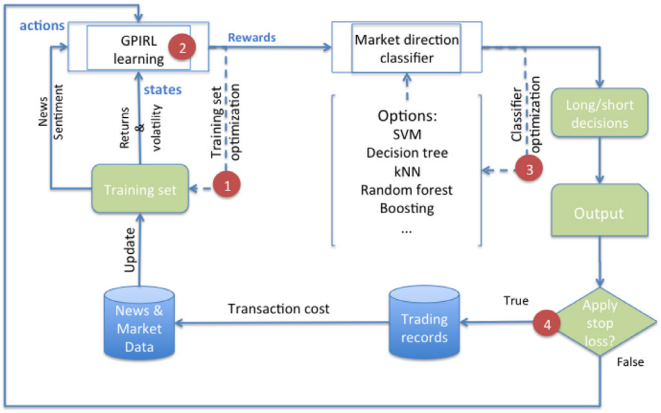

Fig. 5. Trading system flow diagram.

historical data length is too short, the sentiment reward estimates contain too little state transition information to reveal the underlying interaction mechanism between financial market and investor sentiment. However, when the historical data length is too long, it leads to too much overlap between two adjacent trajectories, and the same information will be used to estimate two adjacent sentiment reward trajectories. Then it is too difficult to classify two different directions based on these similar sentiment rewards. There needs to be a balance in finding such an optimal sample size for reward learning. During this phase of the system design, we analyze the effect of the training set size on sentiment reward trading strategy performance. The increase of the training set length would cause computational time to increase without improvement of the system performance. We use classification confusion matrix to select an optimal training set length and overlap length between consecutive training sets.

Once we obtain a set of training data, we follow the GPIRL method to extract the reward function using the method we described in the previous sections (see step ② in Fig. 5). Following the method described in Section 3.2, we obtain states, actions, and transition graphs from the training set. We then apply GPIRL to extract the rewards at different time points t. In this process, we examine the convergence of the learning process to make sure the rewards extracted are stable before stopping the learning process.

Now, we obtain features in the reward space, and then we can train a classifier to predict the future market directions (see step ③ in Fig. 5). In other words, the input feature is the rewards extracted from the GPIRL algorithm. The label corresponding to the feature is whether the market will go up or go down Δt hours later. For the samples of the training set, if the close price of Δt hours later is higher than the close price of current time point, we mark the label as 1; if the close price of Δt hours later is lower than the close price of current time point, we mark the label as 0. We train the classification model using the feature and label of the training set. Then we predict the label of the samples in the testing set using the trained model and input features. If the predicted label is 1, we believe the market will go up Δt hours later. We long the asset. If the predicted label is 0, we believe the market will go down Δt hours later, and we then short the asset.

Support vector machine (SVM) is one of the most important supervised learning techniques in pattern recognition. It has elegant theoretical foundation and has been successfully applied to diverse problems. SVM is to look for the optimal hyper-plane that separates the examples of different categories as wide as possible. Then the new example is predicted to belong to one category based on this hyper-plane. In addition to linear classification, SVM is also flexible in performing non-linear classification using different kernel functions. The most popularly used kernel functions are linear, polynomial, radial basis and sigmoid.

In this study, we adopt a simple market regime change indicator to apply a dynamic updating mechanism to update the training set to improve the prediction power of SVM. Almahdi and Yang (2017) applied a dynamic stop-loss to retrain the parameters of their recurrent reinforcement learning system. We apply this concept in our trading system to update the training set according to the following rule:

$$\frac{r_{t-n,t-1}}{\sigma_{t-n,t-1}} \leqslant -m. \tag{20}$$

Here the $r_{t-n,t-1}$ is the sum of simple returns from $t-n$ to $t-1$, $\sigma_{t-n,t-1}$ is the moving volatility window from $t-n$ to $t-1$, and $-m$ is the threshold to evoke the retraining mechanism (see step ④ in Fig. 5). Initially we regard the first 150 samples as our training set. If the retraining condition (Eq. (20)) is met, we update the training set using the latest samples from $t-150$ to $t-1$. This time point t is the "retraining position". If the condition does not hold, we keep on using the existing model parameters for trading.

Finally, the system will keep records of past profit/loss and transaction costs. The stop/loss mechanism can be used to safeguard the overall system performance.

3.4. Data collection

In this study, we use S&P 500 Index return as the market return and Thompson Reuters' News Sentiment as the proxy of the investors' sentiment. In addition, we also use Thomson Reuters ETF return data as our performance benchmarks. We specifically use intraday market and news data for our model construction. Since GPIRL model training requires a lot of observations, we have to choose an adequate time scale for our study. We use 15 min time interval to measure the return and news sentiment based on a recent study where Yang et al. showed that 15-min time scale is the highest frequency that financial market and investor sentiment have a significant statistical interaction (Yang, Liu et al., 2017). All the data used in this study are listed here:

- SPX: S&P 500 Index
- News sentiment score
- IWD: iShares Russell 1000 Value
- IWC: iShares Micro-Cap
- SPY: SPDR S&P 500 ETF
- DEM: WisdomTree Emerging Markets High Dividend
- VTI: Vanguard Total Stock Market ETF

We obtain the SPX and five ETF data from Thomson Reuters Tick History database. SPX is a stock market index calculated based on the market capitalization of 500 large companies listed on NYSE or NASDAQ. It is a leading indicator of the American financial market performance. IWD is an ETF that seeks to track the investment performance of the Russell 1000 index, which is composed of large and median capitalization of the U.S. equity market. IWC is an ETF that seeks to track the investment performance of Russell micro-cap index. It aims at long-term growth. SPY is an ETF that seeks to track the performance of S&P 500 Index. DEM is an ETF that seeks to track the Wisdom Tree Emerging Markets High Dividend Index, which is based on the companies in the emerging markets region. VTI is an ETF that seeks to track the CRSP US Total Market Index, which measures the investment performance of the overall stock market. We extract the close prices of the above Index and ETFs every 15 min. So each trading day has 26 data points. The data range is from January 2, 2008 to December 31, 2015.

We get the news sentiment data from Thomson Reuters News Analytics (TRNA) database, which is a commercial database and provides quantitative measures of news sentiment. The TRNA

transforms unstructured real-time news from Reuters News and third-party services into a machine readable feed. Companies or assets mentioned in each sentence of the story are identified and the related content is analyzed. The sentiment extraction algorithm considers a number of factors in quantifying a news article such as the order of words, adjectives, common finance phrases, etc. The algorithm is trained on a database of several thousand randomly selected stories that are manually tagged by former market participants.[1] Each news article is described by more than 40 pieces of metadata including Identifier (company/asset), Timestamp, Sentiment (*POS*, *NEUT*, and *NEG*), Relevance, etc. The formula for calculating 15 min average sentiment score is defined as follows:

$$Avg\_Senti = \frac{1}{N}\sum((POS \times 1 + NEUT \times 0 + NEG \times (-1)) \times Relevance) \tag{21}$$

where *POS* is the probability that sentiment of the news article is positive for a certain stock; *NEUT* is the probability that sentiment of the news article is neutral for a certain stock; *NEG* is the probability that sentiment of the news article is negative for a certain stock. The sum of *POS*, *NEUT* and *NEG* is equal to 1. *Relevance* is a number between 0 and 1, and it measures the degree of relevance of a certain news article to a stock. After obtaining the sentiment score of one news article about a certain stock, we get the sentiment score of the whole financial market by averaging all the news' sentiment score to every stock included in the S&P 500 index in the 15 min interval. The data span of news sentiment is the same as the SPX and ETFs data. We use the news sentiment scores for the SPX to estimate the reward signals and we apply these signals combined with the supervised machine learning techniques such as SVM to evaluate the trading strategy on the SPY ETF. We also use other market related ETFs as performance benchmarks for the proposed trading system.

4. Experiments and discussions

In this section, we perform a number of experiments to evaluate the system parameter choices and assess the subsequent performance of the proposed sentiment reward-based trading system.

We first discuss the training data used in the experiments. As discussed in Section 3.3, we use a moving window of certain number of hours to learn the rewards, and then we use a number of learned rewards to predict the market direction of the next period under a stationarity assumption. This moving window sample will be retrained based on the stop-loss conditions assuming the stop-loss provides a measure of market regime change (Lo & Remorov, 2017).

More specifically we use a 13.5 h period and then re-adjust when the stop-loss is triggered. As shown in Fig. 6, the observed historical data between t_{0-h} to t_0 are used to compute sentiment reward, and we want to predict the direction of t_0 to t_{0+p}. Then the next trajectory is from t_{1-h} to t_{1+p}, where t_1 is equal to t_{0+p}. In this study, we use 13.5 h of historical data to predict the SPX market direction. As we use the 15-min data, the 13.5 h historical data contain 54 data points and 4.5 h prediction period contains 18 data points. We obtain this optimal training set duration by examining the training duration ranging from 9 h to 2 months. We find that 13.5 h historical data is an optimal trajectory length for reward learning that can best capture the dynamic interactions between the market and news sentiment measured by prediction confusion matrix. We obtain this number based on the initial training dataset and update it according to a moving window of 150 samples, but

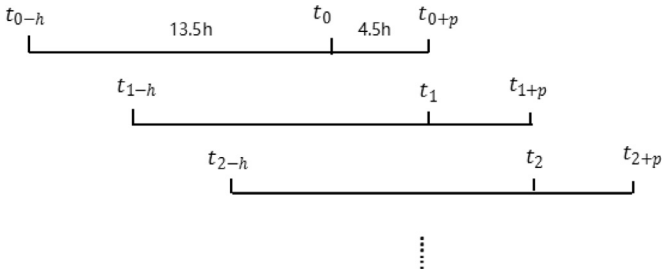

Fig. 6. Data scheme of this research.

for the entire sample data we do not see significant difference in results. Therefore, to reduce the computational complexity, we use the same trajectory size through out the entire study. However, as we pointed out in step ① of the trading system in Section 3.3, the optimal training dataset length should be reevaluated when the stop-loss trigger is activated. Otherwise, there is a danger of data snooping, and the system performance may suffer as a result. After predicting the SPX market direction, we will trade the SPY ETF according to it.

4.1. Strategy comparison

In this experiment, we compare the performance of the proposed sentiment reward based trading strategy with sentiment score, sentiment trend and sentiment shock based trading strategies (Song, Almahdi et al., 2017; Song, Liu et al., 2017; Yang, Song et al., 2015) to assess whether this sentiment reward trading strategy is superior to other existing sentiment based trading strategies in the current literature.

To construct the sentiment score based trading strategy, we use the 13.5 h historical sentiment score data, which contains 54 data points as the input feature, to predict the SPX direction of the next 4.5 h. Then we buy long or sell short according to the direction labels predicted by SVM. The mechanism for sentiment trend and sentiment shock strategies is the same as the sentiment score based trading strategy. The only difference is that we replace the 13.5 h historical sentiment score data by sentiment trend data or sentiment shock signals accordingly. As Yang, Song et al. (2015) use daily sentiment trend and sentiment shock to design their trading strategies, we assign daily sentiment trend and daily sentiment shock signals to the last 15-min of this certain day in order to make the strategies comparison fairly. The sentiment shock is constructed according to Yang, Song et al. (2015) and Song, Liu et al. (2017) in the following way:

- Positive shock:
$$S_{t_0} > \mu + M \cdot \sigma, \tag{22}$$
- Neutral shock:
$$\mu - M \cdot \sigma < S_{t_0} < \mu + M \cdot \sigma, \tag{23}$$
- Negative shock:
$$S_{t_0} < \mu - M \cdot \sigma. \tag{24}$$

Here S_{t_0} is the sentiment score of day t_0, μ is the mean of sentiment score calculated from $t_0 - N$ to $t_0 - 1$ and σ is the standard deviation of sentiment score from $t_0 - N$ to $t_0 - 1$. N is the number of look-back days and M is the multiplier. We set $N = 5$ and $M = 1.5$ as chosen by Yang, Song et al. (2015).

We define sentiment trend due to Yang, Song et al. (2015) and Song, Liu et al. (2017) as:

$$\text{Sentiment trend: } \sum_{i=t_0-N}^{t_0} |\Delta S_i| > T \tag{25}$$

[1] For more information about Thomson Reuters News Analytics (TRNA) database please visit its official website at https://financial.thomsonreuters.com/en/products/data-analytics/financial-news-feed/world-news-analysis.html.

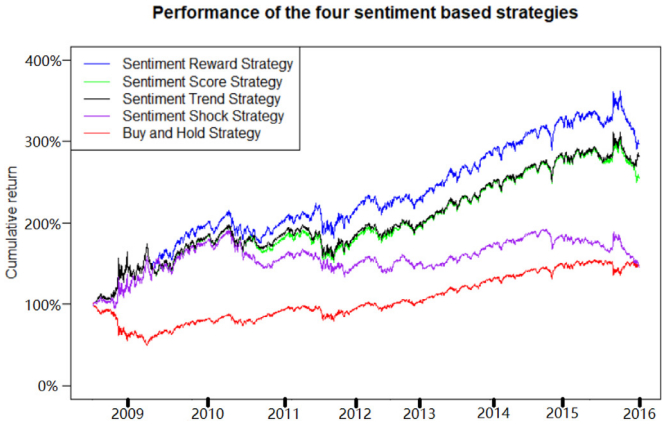

Fig. 7. Comparison of performance for sentiment reward strategy, sentiment score strategy, sentiment trend strategy, sentiment shock strategy and buy-and-hold strategy from 2008 to 2015. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index.

Table 1
Sentiment based strategies performance comparison in 2008–2015 period.

Strategy	Max. drawdown	Annualized performance			
		Mean return	Volatility	Sharpe ratio	Sterling ratio
Total backtest period					
Senti. reward	13.02%	15.48%	20.43%	0.76	0.67
Senti. trend	13.02%	14.70%	20.43%	0.72	0.64
Senti. score	13.02%	13.23%	20.43%	0.65	0.57
Buy and hold	8.18%	5.31%	20.44%	0.26	0.29
Senti. shock	13.02%	5.13%	20.44%	0.25	0.22
High volatility regime					
Senti. reward	13.02%	67.87%	41.33%	1.64	2.95
Senti. trend	13.02%	54.77%	41.36%	1.32	2.38
Senti. score	13.02%	54.18%	41.36%	1.31	2.35
Buy and hold	8.18%	−31.70%	41.42%	−0.77	−1.74
Senti. shock	13.02%	49.55%	41.37%	1.2	2.15
Low volatility regime					
Senti. reward	2.84%	9.33%	12.00%	0.78	0.73
Senti. trend	2.84%	12.85%	11.99%	1.07	1.00
Senti. score	2.84%	9.59%	12.00%	0.80	0.75
Buy and hold	3.40%	13.09%	11.99%	1.09	0.98
Senti. shock	2.84%	−0.31%	12.01%	−0.03	−0.02

Notes: In this research the high volatility period is from the beginning of 2008 to the June 1, 2009, whose mean annualized 6-month realized volatility is 31.08% and mean VIX close index is 35.13; the low volatility period is from beginning of 2013 to the end of 2015, whose mean annualized 6-month realized volatility is 11.16% and mean VIX close index is 15.08. We trade on the SPY ETF using the signals generated from news sentiment and SPX Index.

where ΔS_i is the change of sentiment on day i, t_0 represents the current day. N is the moving window size to sum the change of sentiments within it, and T is the threshold value.

All of the four trading strategies would update the training set to the latest 150 samples at every time step. Additionally, we also use the performance of buy-and-hold of SPY ETF as a benchmark. In order to test the performance of sentiment reward strategy in different market conditions, we back-test these strategies in high volatility period and low volatility period.

From Fig. 7 and Table 1, we can see that the sentiment reward trading strategy significantly outperforms all other benchmark sentiment based trading strategies during 2008–2015 period. Sentiment trend trading strategy achieves the second highest performance. Sentiment score trading strategy performs better than sentiment shock trading strategy. Only the sentiment shock trading strategy achieves a little worse performance than the naive buy and hold trading strategy. During the high volatility period, the performances of the four sentiment based trading strategies

Table 2
Performance statistics of four sentiment based trading strategies.

	Sentiment reward	Sentiment trend	Sentiment score	Sentiment shock
Accuracy	54.20%	53.77%	53.73%	52.46%
Positive predictive value	50.61%	49.54%	49.44%	46.67%
Negative predictive value	55.16%	54.80%	54.76%	54.12%
Sensitivity	23.06%	21.09%	20.77%	22.67%
Specificity	80.78%	81.66%	81.86%	77.88%
Transactions	99	39	57	289

are in the same order as in whole period, and they all beat the benchmark by a large amount. In the low volatility period, the sentiment trend trading strategy outperforms other sentiment based trading strategies by a small amount, but it is still a little lower than the benchmark's performance. We think in the high volatility period, the investor sentiment switches from optimistic to pessimistic or from pessimistic to optimistic frequently. So the sentiment reward strategy, which reveals the interaction mechanism between investor sentiment and market condition along with the other sentiment based trading strategies, is able to take advantage of investor sentiment signal very well. In general, we observe that all the sentiment signal based trading strategy perform at the same level of profitability as the naive buy and hold strategy in the low volatility period.

From Table 2, we can observe that the accuracy, precision/positive predictive value and negative predictive value have the same trend as the performance of the four sentiment based trading strategies. However, we find the accuracy, precision/positive predictive value and negative predictive value from the sentiment reward trading strategy have the highest value. Then it is followed by the sentiment trend and sentiment score trading strategies respectively. The sentiment shock trading strategy has the lowest performance. The precision/positive predictive value is a measure of the SPX downward prediction correctness in the 4.5 h. The negative predictive value a measure of the SPX upward prediction correctness in the 4.5 h. Thus the results show that the sentiment reward trading strategy has the highest rate in predicting both the SPX up direction and down direction resulting in a better performance than the other three sentiment based trading strategies. The sensitivity in this study measures the proportion of SPX's downward direction in the next 4.5 h that are correctly predicted as such by the trading system. The specificity measures the proportion of SPX's upward direction in the next 4.5 h that are correctly predicted. In an ideal situation, we would expect both high sensitivity and high specificity, but given that the specificity for all other sentiment signals are around 80%, 80.78% should be considered good enough in our case. The overall performance of the sentiment reward-based strategy is clear better than that of the other sentiment based strategies.

Fig. 8 shows the sentiment reward trading with every step update of trading signals from 2008 to 2015. The blue background areas indicate the long periods; the orange background areas indicate the short periods. The sentiment score is based on 4.5 h interval. Fig. 9 show the number of news articles along with sentiment score from 2008 to 2015. The number of articles is calculated based on 4.5 h interval. Fig. 10 is the sentiment reward with every step update of trading strategy signals of high volatility period (from the beginning of 2008 to the June 1, 2009). From Fig. 9, we can see that more news articles are published during the high volatility period than the low volatility period. We observe the news sentiment score, which is measured from the published news articles, is more closely correlated with the market price during high volatility period than the low volatility period. However, we see less news articles, less correlation, and conse-

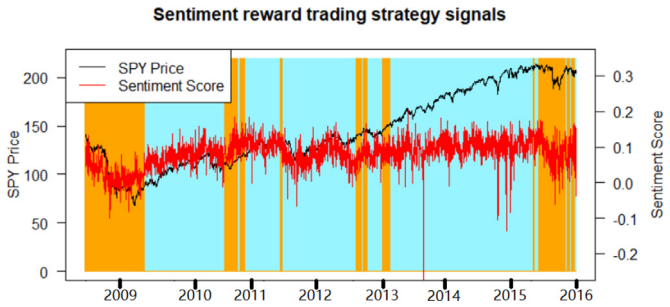

Fig. 8. Sentiment reward trading with every step update of trading signals from 2008 to 2015. Note: The blue background areas indicate the long periods; The orange background areas indicate the short periods. The sentiment score is based on 4.5 h interval. We trade on the SPY ETF using the signals generated from news sentiment and SPX Index. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

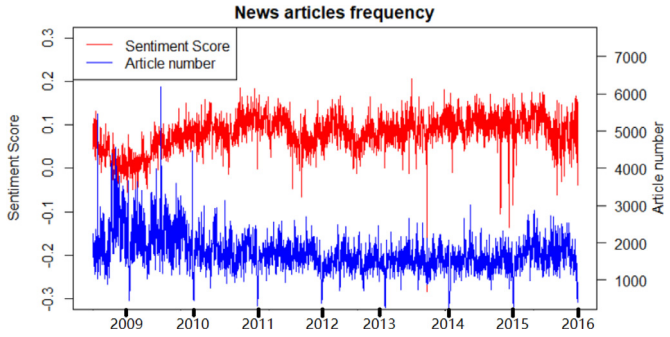

Fig. 9. News articles frequency from 2008 to 2015. The number of articles is based on 4.5 h interval.

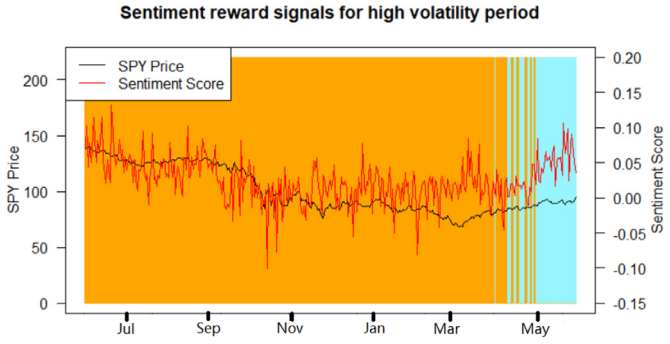

Fig. 10. Sentiment reward with every step update of trading signals of high volatility period (from the beginning of 2008 to the June 1, 2009). Note: The blue background areas indicate the long periods; The orange background areas indicate the short periods. The sentiment score is based on 4.5 h interval. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

quently scattered transactions generated from the trading system during the low volatility period (see Figs. 8 and 9). Overall, we argue the investor sentiment has a pronounced effect on market movements during the high volatility period for two reasons: a) news sentiment measure is more accurate in the high volatility period than that in the low volatility period due to the increased news coverage (see Fig. 9); b) this observation is consistent with the general findings in the behavioral finance literature that negative news have bigger impact on market during high volatility period (Smales, 2014; 2015; Yu & Yuan, 2011). As we can see from Figs. 8 and 9, the sentiment score is relatively low and is close to negative territory during the high volatility period. At the same time, the correlation between the sentiment and market price is relatively high (see Fig. 8). However, the correlation between the

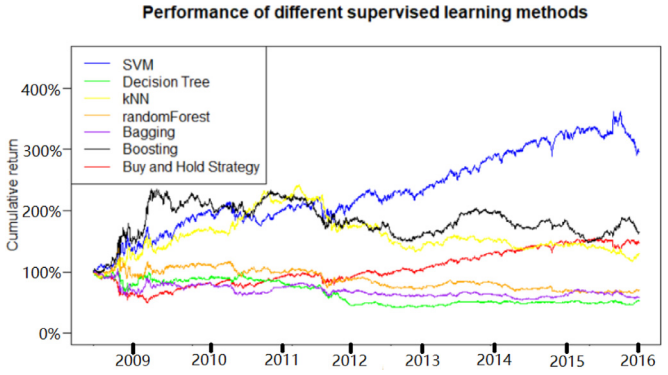

Fig. 11. Comparison of different supervised learning techniques based on sentiment reward in 2008–2015 period. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index.

sentiment and market price start to reduce during the low volatility period (see Fig. 8). All these might contribute to why all the sentiment based trading strategies perform particularly better than the buy-and-hold strategies on ETFs during the volatile market period, and the sentiment reward based trading strategy is doing well compared with other sentiment signal based strategies (see Table 1).

4.2. Learning method comparison

There are many popular and successful supervised learning techniques in machine learning literature. In this section we will construct several sentiment reward trading strategies based on different supervised learning techniques including SVM, decision tree, kNN (k-nearest neighbors), random forest, bagging and boosting. Then we will compare the performance of these strategies and see which supervised learning technique can achieve the highest performance based on sentiment reward. The decision tree algorithm uses a decision tree as a predictive model to analyze the relationship between input feature and the label of a sample. There are three main components in the decision tree structure, the internal node, the branch node and the leaf node. Each internal node represents a judgment on an attribute of input feature. Each branch represents the outcome of this judgment, and each leaf node represents one of the total classification labels. The kNN algorithm is one of the popular instance based learning algorithm. The input to this model is k training samples. The model will predict the classification label of new test sample by a majority vote of its neighbors according to the distance metric of their input features. The random forest algorithm is an ensemble learning method for classification by growing many classification trees. To predict the classification label of a sample, this algorithm will use every tree it generates to predict. The forest will output the final label by a majority vote. Bagging is a popular method used to decrease the variance of a prediction by randomly generating additional training data from the original training dataset. Also as one kind of ensemble meta-algorithms, boosting algorithm is a method used to convert many weak classifiers to a strong one. After a weak classifier is added, misclassified samples gain more weight and future weak classifiers focus more on the samples that are misclassified before. In the following experiment we will update the training set to the latest 150 examples at each time step for every trading strategy.

From Fig. 11 and Table 3 we can see that the SVM achieves the highest performance. Boosting, successfully beating the naive buy-and-hold strategy is ranked at second place.

Table 3
Performance of different supervised learning techniques. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index.

Strategy	Max. drawdown	Annualized performance			
		Mean return	Volatility	Sharpe ratio	Sterling ratio
SVM	13.02%	15.48%	20.43%	0.76	0.67
Boosting	13.02%	6.86%	20.44%	0.34	0.30
Buy and hold	8.18%	5.31%	20.44%	0.26	0.29
kNN	8.18%	3.47%	20.45%	0.17	0.19
Random forest	8.18%	−4.63%	20.45%	−0.23	−0.25
Bagging	13.02%	−6.93%	20.45%	−0.34	−0.30
Decision tree	10.73%	−8.04%	20.45%	−0.39	−0.39

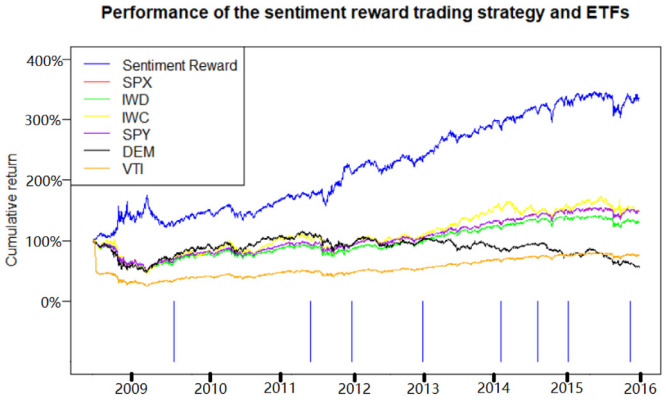

Fig. 12. Performance of sentiment reward trading strategy with retraining mechanism. *Notes:* The bottom vertical line shows the retraining position/transaction of the system. We trade on the SPY ETF using the signals generated from news sentiment and SPX Index.

4.3. Trading system

We examine two specific aspects of the proposed trading system. First, we analyze the system performance with respect to the passive investment in S&P 500 related ETFs. And then we compare the sentiment reward based adaptive retraining system with three other sentiment based retraining systems. Given the same retraining mechanism and the classification method, we aim to show the relative performance comparison of the four different sentiment measures, i.e. sentiment reward, sentiment trend, sentiment shock, and sentiment score. Finally, we want to examine the cost effect of the proposed sentiment reward based adaptive retraining system.

4.3.1. Trading system performance

In this section, we combine the support vector machine (SVM), a supervised learning technique, with an adaptive retraining mechanism to show whether this sentiment reward trading strategy has superior performance to the standard market benchmarks. The training set contains 150 sample trajectories. Then we use the model trained based on these 150 samples to predict the direction of the sentiment reward trajectories in the testing set. The trading system would update the training set to the latest 150 samples only if the investment performance exceeds the threshold as described by Eq. (20). We then employ a naive buy-and-hold strategy on the popular market ETFs (i.e. IWD, IWC, SPY, DEM and VTI) and the *S&P*500 index as the benchmarks for comparison.

Fig. 12 shows the performance of sentiment reward trading strategy with the retraining mechanism and naive buy-and-hold trading strategies based on SPX, IWD, IWC, SPY, DEM and VTI. We set the parameter n of Eq. (20) to 10. By trying different values of threshold m from 1 to 10, we obtain the optimal performance of this retraining system when $m = 8$. Fig. 12 shows the retraining position/transaction of the trading system. Table 4 shows the investment result statistics of this novel sentiment reward trading strategy are much better than SPX index and other ETFs. From performance comparisons, we can see that the sentiment reward trading strategy is consistently superior to the passive market benchmarks. The sentiment reward strategy obtained an average of 17.39% return, 0.85 Sharpe ratio and 0.76 Sterling ratio which is about 3 times all the market benchmarks.

Table 4
Comparison of sentiment reward strategy and ETFs performance in 2008–2015 period. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index. We execute buy-and-hold strategy on all the ETFs.

Strategy	Max. drawdown	Annualized performance			
		Mean return	Volatility	Sharpe ratio	Sterling ratio
Total backtest period					
Senti. reward	13.02%	17.39%	20.43%	0.85	0.76
IWC	11.98%	5.45%	25.79%	0.21	0.25
SPY	8.18%	5.31%	20.44%	0.26	0.29
IWD	8.60%	3.58%	21.79%	0.16	0.19
VTI	50.65%	−3.65%	27.86%	−0.13	−0.06
DEM	7.92%	−7.17%	24.73%	−0.29	−0.40
High volatility regime					
Senti. reward	13.02%	23.31%	41.42%	0.56	1.01
IWC	8.66%	−32.18%	47.04%	−0.68	−1.72
SPY	8.18%	−31.70%	41.42%	−0.77	−1.74
IWD	8.60%	−35.06%	44.91%	−0.78	−1.88
VTI	50.65%	−66.01%	65.91%	−1.00	−1.09
DEM	7.92%	−25.66%	43.52%	−0.59	−1.43
Low volatility regime					
Senti. reward	3.40%	12.80%	11.99%	1.07	0.95
IWC	11.98%	11.82%	18.73%	0.63	0.54
SPY	3.40%	13.09%	11.99%	1.09	0.98
IWD	3.53%	10.77%	12.04%	0.89	0.80
VTI	3.21%	12.88%	12.18%	1.06	0.97
DEM	4.89%	−17.76%	18.45%	−0.96	−1.19

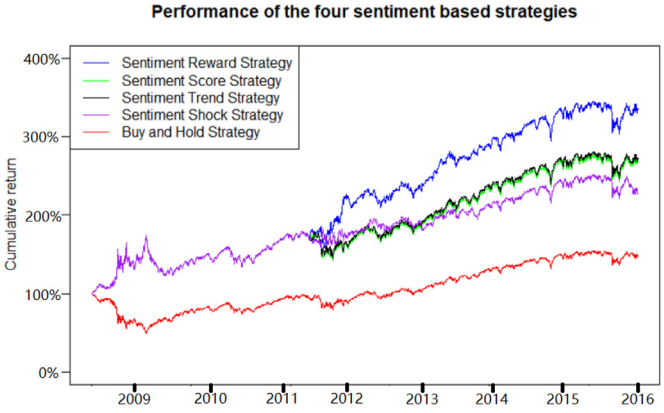

Fig. 13. Sentiment based retraining systems comparison. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index.

Next we design a retraining system by using sentiment score, sentiment trend and sentiment shock as comparisons with our proposed trading system. Initially we regard the first 150 samples as our training set. If the retraining condition (Eq. (20)) is met, we update the training set using the latest samples from $t - 150$ to $t - 1$ according to step ④ in Fig. 5 to generate trading signal. This time point t is the "retraining position" depicted in Fig. 14. If the condition does not hold, we keep on using the existing model parameters for trading.

From Fig. 13 and Table 5 we can see that the sentiment reward retraining system is optimal. Only the sentiment trend retraining system's performance is a little worse than the every step updating system. All the other three sentiment based retraining sys-

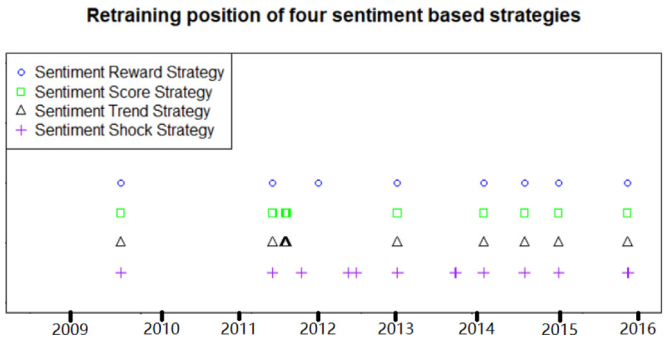

Fig. 14. Retraining position/transaction of four sentiment based strategies. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index.

Table 5
Performance of four sentiment based retraining systems. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index. The same max drawdown of the four sentiment based trading systems is due to the reason that they all predict the wrong direction for that certain time point.

Strategy	Max. drawdown	Annualized performance			
		Mean return	Volatility	Sharpe ratio	Sterling ratio
Senti. reward	13.02%	17.39%	20.43%	0.85	0.76
Senti. trend	13.02%	14.21%	20.43%	0.70	0.62
Senti. score	13.02%	14.00%	20.43%	0.68	0.61
Senti. shock	13.02%	11.43%	20.44%	0.56	0.50
Buy and hold	8.18%	5.31%	20.44%	0.26	0.29

tems achieve better profitability. We also examine the retraining signals generated during the entire period, and we find that the four sentiment based trading systems generate the similar retraining signals. This result shows consistency of the sentiment signals (see Fig. 14). However, the sentiment shock based system generates more retraining signals than the other sentiment based systems. This might explain its underperformance in terms its forecast accuracy (see Table 2).

4.3.2. Trading cost effect analysis

In practice, the transaction costs are inevitable and of significance if a trading strategy involves multiple changes in holding positions. Since the focus of the paper is to identify market entry and exit points, we do not discuss the choice of execution strategies. We assume a highly liquid market, which is the case for the SPY ETF, and adopt a fixed transaction cost estimation based on empirical studies rather than the bid-ask spread or price impact based cost estimation methods. Based on empirical studies, an average round-trip execution cost (commission cost and market impact cos) of large-cap stocks on NYSE is at least 20 bps (Chan & Lakonishok, 1997; Mittermayer, 2004; Tetlock, 2007; Yang, Mo et al., 2017). In this transaction cost sensitivity analysis, we experiment with a range of fixed transaction costs and show their impact on the proposed and benchmark strategies. We examine the cost effect of 5, 10, 15, 20 and 25 bps for a single transaction. Here, we define market return and strategy return with the transaction cost calculation as:

$$r_t = \frac{p_t}{p_{t-1}} - 1 \tag{26}$$

and

$$R_t = F_{t-1} r_t - \delta |F_t - F_{t-1}| \tag{27}$$

where F_{t-1} is the signal generated at the time point $t-1$. If it is a long signal, $F_{t-1} = 1$; If it is a short signal, $F_{t-1} = -1$. R_t is the return of period $(t-1, t]$, which includes the gain or loss of this period according to the trading signal F_{t-1} and the transaction cost incurred due to the difference of the signal F_{t-1} and F_t.

Table 6
Transaction cost sensitivity analysis of sentiment based retraining systems. Note: We trade on the SPY ETF using the signals generated from news sentiment and SPX Index. The same max drawdown of the four sentiment based trading systems is due to the reason that they all predict the wrong direction for that certain time point. At the same time this time point is not related to changing trading position. Thus the transaction cost does not affect the Max. drawdown.

Transac. cost	Max. drawdown	Annualized performance			
		Mean return	Volatility	Sharpe ratio	Sterling ratio
Sentiment reward					
0 bps	13.02%	17.39%	20.43%	0.85	0.76
5 bps	13.02%	13.89%	20.43%	0.68	0.60
10 bps	13.02%	10.50%	20.45%	0.51	0.46
15 bps	13.02%	7.21%	20.48%	0.35	0.31
20 bps	13.02%	4.01%	20.53%	0.20	0.17
25 bps	13.02%	0.91%	20.59%	0.04	0.04
Sentiment trend					
0 bps	13.02%	14.21%	20.43%	0.70	0.62
5 bps	13.02%	14.18%	20.43%	0.69	0.62
10 bps	13.02%	14.15%	20.43%	0.69	0.61
15 bps	13.02%	14.11%	20.43%	0.69	0.61
20 bps	13.02%	14.08%	20.43%	0.69	0.61
25 bps	13.02%	14.05%	20.44%	0.69	0.61
Sentiment score					
0 bps	13.02%	14.00%	20.43%	0.68	0.61
5 bps	13.02%	13.75%	20.43%	0.67	0.60
10 bps	13.02%	13.51%	20.43%	0.66	0.59
15 bps	13.02%	13.27%	20.43%	0.65	0.58
20 bps	13.02%	13.03%	20.44%	0.64	0.57
25 bps	13.02%	12.79%	20.44%	0.63	0.56
Sentiment shock					
0 bps	13.02%	11.43%	20.44%	0.56	0.50
5 bps	13.02%	8.46%	20.45%	0.41	0.37
10 bps	13.02%	5.56%	20.48%	0.27	0.24
15 bps	13.02%	2.74%	20.52%	0.13	0.12
20 bps	13.02%	0.00%	20.57%	0.00	0.00
25 bps	13.02%	−2.68%	20.63%	−0.13	−0.12

In the experiment, we test the transaction cost sensitivity of the four sentiment based retraining systems by setting the transaction cost rate δ in a range from 5 bps to 25 bps. From Table 6, we observe that the proposed sentiment reward retraining system is relatively sensitive to the increase of transaction cost compared with other sentiment based retraining systems. But the sentiment trend based retraining systems have relatively high resilience to the transaction cost increase, which is understandable because the sentiment trend based system has a smoothing mechanism built in. In the meantime, we also observe that the sentiment shock retraining system exhibits the similar cost sensitivity as the sentiment reward retraining system. Together with the forecasting accuracy analysis (see Table 2), we see that the sentiment reward system generates more high accuracy signals, and the number of such signals is almost double the number generated from the sentiment trend system. However, because we only forecast the market directions and some of these signals do not generate much profit, yet additional executions will certainly increase the total costs. It is the same as the sentiment shock based system, but the accuracy of its signals is not as good as that from the sentiment reward based system. Overall, this analysis provides a guideline for building a practical trading system. In reality, the transaction costs are quite different for different markets, but the cost factor has to be considered when designing a profitable trading system.

5. Conclusion

The main contributions of this study can be summarized as follows: (i) We model the interaction mechanism between investor sentiment and market return using Gaussian process inverse reinforcement learning method with a preference graph to fit the situation in which market states respond to investor sentiment shocks differently. (ii) We propose an inverse reinforcement learning method to extract market rewards toward investor sentiment and show that the sentiment rewards provide high quality signals to predict future market directions. (iii) We compare the sentiment reward-based trading system with the most popular passive market based strategies as well as other news sentiment signals based trading strategies, and we demonstrate that the proposed strategy outperforms them all. (iv) Finally, we design a trading system based on the proposed sentiment reward signals along with the SVM learning method and market condition retraining mechanism, and we show that the proposed trading system outperforms all the benchmark strategies especially during the high volatility market conditions. However, the sentiment reward based retraining system is transaction cost sensitive.

Overall, the sentiment reward-based trading system proposed generates superior performance through filtering out the unresponsive news sentiment signals. Its outperformance is especially pronounced under a very volatile market condition, such as the 2008–09 financial crisis period. From the academic research perspective, this study provides a new way of revealing the inherent interaction mechanism between financial market and investor sentiment. From a practical point of view, this sentiment reward trading strategy is a superior and robust strategy and it beats the passive market based investment strategies and other sentiment based strategies consistently.

While the proposed trading system is focused on market direction prediction only, we observe that it is sensitive to transaction cost effect which is also true with other benchmark sentiment signals because some sentiment signals are not big enough to generate robust profit. We suggest future studies consider a mechanism to filter insignificant sentiment signals in that although markets respond to certain sentiment signals, gains often will not be enough to overcome the costs incurred resulting in negative effect to the overall performance.

In this study, we focus on the systematic features of the whole market and design a system for the S&P500 ETF trading. However, we do see the potential that the same methodology can be applied to capture individual stock idiosyncratic features for superior trading performances. This means that reward learning will be focused on individual stock returns and news sentiment analysis, which would require a large number of news articles published for a particular stock market. Due to the scope of the current study, we recommend future studies to examine large-cap stocks with sufficient news reports according to Song, Liu, Yang, Deane, and Datta (2015) to construct a portfolio and implement this sentiment reward based trading strategy. Similarly, this approach can also be applied to a particular industry sector.

Moreover, given the extreme outperformance of the sentiment reward signals under high volatility market conditions, we suggest researchers and practitioners to develop a sentiment regime based trading system where the system can automatically switch between different sentiment signals and other trading signals based on market volatility so that the system performance will be optimized.

References

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on machine learning* (p. 1). ACM.

Almahdi, S., & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*.

Antoniou, C., Doukas, J. A., & Subrahmanyam, A. (2013). Cognitive dissonance, sentiment, and momentum. *Journal of Financial and Quantitative Analysis, 48*(01), 245–275.

Antweiler, W., & Frank, M. Z. (2004). Is all that talk just noise? The information content of Internet stock message boards. *Journal of Finance, 59*(3), 1259–1294.

Baker, M., & Wurgler, J. (2006). Investor sentiment and the cross-section of stock returns. *The Journal of Finance, 61*(4), 1645–1680.

Barber, B. M., & Odean, T. (2012). All that glitters: The effect of attention and news on the buying behavior of individual and institutional investors. *The Review of Financial Studies, 21*(2), 785–818.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science, 2*(1), 1–8.

Chan, L. K. C., & Lakonishok, J. (1997). Institutional equity trading costs: NYSE versus nasdaq. *The Journal of Finance, 52*(2), 713–735.

Checkley, M., Higón, D. A., & Alles, H. (2017). The hasty wisdom of the mob: How market sentiment predicts stock market behavior. *Expert Systems with Applications, 77*, 256–263.

Chen, Y., Mabu, S., Hirasawa, K., & Hu, J. (2007). Genetic network programming with sarsa learning and its application to creating stock trading rules. In *Evolutionary computation, 2007. cec 2007. ieee congress on* (pp. 220–227). IEEE.

Day, M.-Y., & Lee, C.-C. (2016). Deep learning for financial sentiment analysis on finance news providers. In *Advances in social networks analysis and mining (asonam), 2016 ieee/acm international conference on* (pp. 1127–1134). IEEE.

De Long, J. B., Shleifer, A., Summers, L. H., & Waldmann, R. J. (1990). Noise trader risk in financial markets. *Journal of political Economy, 98*(4), 703–738.

Feuerriegel, S., Heitzmann, S. F., & Neumann, D. (2015). Do investors read too much into news? How news sentiment causes price formation. In *System sciences (hicss), 2015 48th hawaii international conference on* (pp. 4803–4812). IEEE.

Feuerriegel, S., & Neumann, D. (2013). News or noise? How news drives commodity prices. (pp. 119–135).

Feuerriegel, S., & Prendinger, H. (2016). News-based trading strategies. *Decision Support Systems, 90*, 65–74.

Gao, X., & Chan, L. (2000). An algorithm for trading and portfolio management using q-learning and sharpe ratio maximization. In *Proceedings of the international conference on neural information processing* (pp. 832–837).

Ho, C.-S., Damien, P., Gu, B., & Konana, P. (2017). The time-varying nature of social media sentiments in modeling stock returns. *Decision Support Systems, 101*, 69–81.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the econometric society*, 263–291.

Kuremoto, T., Obayashi, M., & Kobayashi, K. (2007). Forecasting time series by sofnn with reinforcement learning. In *Proceedings of the 27th annual international symposium on forecasting, neural forecasting competition (nn3), new york, ny, usa* (pp. 24–27).

Kurov, A. (2010). Investor sentiment and the stock market's reaction to monetary policy. *Journal of Banking & Finance, 34*(1), 139–149.

Lee, J. W. (2001). Stock price prediction using reinforcement learning. In *Industrial electronics, 2001. proceedings. isie 2001. ieee international symposium on: 1* (pp. 690–695). IEEE.

Lo, A. W., & Remorov, A. (2017). Stop-loss strategies with serial correlation, regime switching, and transaction costs. *Journal of Financial Markets, 34*, 1–15.

Mittermayer, M.-A. (2004). Forecasting intraday stock price trends with text mining techniques. In *system sciences, 2004. proceedings of the 37th annual hawaii international conference on* (pp. 10–pp). IEEE.

Muhammad, A., Wiratunga, N., & Lothian, R. (2016). Contextual sentiment analysis for social media genres. *Knowledge-Based Systems, 108*, 92–101.

Nasseri, A. A., Tucker, A., & de Cesare, S. (2015). Quantifying stocktwits semantic terms' trading behavior in financial markets. *Expert Systems with Applications: An International Journal, 42*(23), 9192–9210.

Neuneier, R. (1996). Optimal asset allocation using adaptive dynamic programming. In *Advances in neural information processing systems* (pp. 952–958).

Neuneier, R. (1998). Enhancing q-learning for optimal asset allocation. In *Advances in neural information processing systems* (pp. 936–942).

Ng, A. Y., & Russell, S. J. (2000). Algorithms for inverse reinforcement learning. In *Icml* (pp. 663–670).

Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications, 73*, 125–144.

Pröllochs, N., Feuerriegel, S., & Neumann, D. (2016). Negation scope detection in sentiment analysis: decision support for news-driven trading. *Decision Support Systems, 88*, 67–75.

Qiao, Q., & Beling, P. A. (2011). Inverse reinforcement learning with gaussian process. In *American control conference (acc), 2011* (pp. 113–118). IEEE.

Ramachandran, D., & Amir, E. (2007). Bayesian inverse reinforcement learning. *Urbana, 51*(61801), 1–4.

Shiller, R. J. (2003). From efficient markets theory to behavioral finance. *Journal of Economic Perspectives, 17*(1), 83–104.

Shiller, R. J., Fischer, S., & Friedman, B. M. (1984). Stock prices and social dynamics. *Brookings papers on economic activity, 1984*(2), 457–510.

Smales, L. A. (2014). News sentiment in the gold futures market. *Journal of Banking and Finance, 49*, 275–286.

Smales, L. A. (2015). Asymmetric volatility response to news sentiment in gold futures. *Journal of International Financial Markets, Institutions and Money, 34*, 161–172.

Song, Q., Almahdi, S., & Yang, S. Y. (2017). Entropy based measure sentiment analysis in the financial market. In *Computational intelligence, 2017 ieee symposium series on* (pp. 301–305). IEEE.

Song, Q., Liu, A., & Yang, S. Y. (2017). Stock portfolio selection using learning-to-rank algorithms with news sentiment. *Neurocomputing, 264*(Supplement C), 20–28. doi:10.1016/j.neucom.2017.02.097. Machine learning in finance

Song, Q., Liu, A., Yang, S. Y., Deane, A., & Datta, K. (2015). An extreme firm-specific news sentiment asymmetry based trading strategy. In *Computational intelligence, 2015 ieee symposium series on* (pp. 898–904). IEEE.

Sun, L., Najand, M., & Shen, J. (2016). Stock return predictability and investor sentiment: A high-frequency perspective. *Journal of Banking & Finance, 73*, 147–164.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance, 62*(3), 1139–1168.

Yang, S. Y., Liu, A., Chen, J., & Hawkes, A. (2017). Applications of a multivariate hawkes process to joint modeling of sentiment and market return events. *Quantitative Finance*, 1–16.

Yang, S. Y., Mo, S. Y. K., Liu, A., & Kirilenko, A. A. (2017). Genetic programming optimization for a sentiment feedback strength based trading strategy. *Neurocomputing, 264*(Supplement C), 29–41. doi:10.1016/j.neucom.2016.10.103. Machine learning in finance

Yang, S. Y., Mo, S. Y. K., & Zhu, X. (2014). An empirical study of the financial community network on twitter. In *Computational intelligence for financial engineering & economics (cifer), 2104 ieee conference on* (pp. 55–62). IEEE.

Yang, S. Y., Qiao, Q., Beling, P. A., Scherer, W. T., & Kirilenko, A. A. (2015). Gaussian process-based algorithmic trading strategy identification. *Quantitative Finance, 15*(10), 1683–1703.

Yang, S. Y., Song, Q., Mo, S. Y. K., Datta, K., & Deane, A. (2015). The impact of abnormal news sentiment on financial markets. *Journal of Business and Economics, 6*(10), 1682–1694.

Yu, J., & Yuan, Y. (2011). Investor sentiment and the mean–variance relation. *Journal of Financial Economics, 100*(2), 367–381.

Zhai, J., Liu, Q., Zhang, Z., Zhong, S., Zhu, H., Zhang, P., & Sun, C. (2016). Deep q-learning with prioritized sampling. In *International conference on neural information processing* (pp. 13–22). Springer.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Aaai: 8* (pp. 1433–1438). Chicago, IL, USA.