Neurocomputing 142 (2014) 228–238

Contents lists available at ScienceDirect

Neurocomputing

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

Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information

Xiaodong Li [a], Xiaodi Huang [b], Xiaotie Deng [c,e], Shanfeng Zhu [d,e,*]

[a] Department of Computer Science, City University of Hong Kong, Hong Kong
[b] School of Computing and Mathematics, Charles Sturt University, Albury, NSW 2640, Australia
[c] AIMS Lab., Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China
[d] School of Computer Science, Fudan University, Shanghai 200433, China
[e] Shanghai Key Lab. of Intelligent Information Processing, Fudan University, Shanghai 200433, China

ARTICLE INFO

Article history:
Received 21 August 2013
Received in revised form
29 January 2014
Accepted 3 April 2014
Communicated by P. Zhang
Available online 6 June 2014

Keywords:
Multiple kernel learning
Stock return prediction
News analysis

ABSTRACT

The interaction between stock price process and market news has been widely analyzed by investors on different markets. Previous works, however, focus either on market news purely as exogenous factors that tend to lead price process or on the analysis of how past stock price process can affect future stock returns. To take a step forward, we quantitatively integrate information from both market news and stock prices in order to improve the accuracy of prediction on stock future price return in an intra-day trading context. In this paper, we present the design and architecture of our approach for market information fusion. By means of multiple kernel learning, the hidden information behind the two sources is effectively extracted, and more importantly, seamlessly integrated rather than simply combined by a single kernel approach. Experiments on comprehensive comparisons between our approach and three baseline methods (which use only one type of information, or naively combine the two sources) have been conducted on the intra-day tick-by-tick data of the Hong Kong Stock Exchange and market news archives of the same period. It has been shown that for both cross-validation and independent testing, our approach is able to achieve the best results.

1. Introduction

Stock market is one of the most important and active parts of financial markets. The most important role of stock market is to determine the prices of stocks. Therefore, a fair and open price determination process is critical to the well functioning of the financial markets. While determining the *fair value* of a stock, investors have to analyze a large amount of information, which can be in the form of specific news about the underlying company or liquidity-specific data, such as recent trading activities. Among all the forms of information, intra-day market news and tick prices are critical to the trading decisions an investor makes. Tetlock [44] and Schumaker and Chen [37] show in their recent papers that market news has certain predictability on the stock price returns. It has also been shown by many market microstructure researchers that stock prices, especially the intra-day tick-by-tick prices, are closely related to the embedding of public and private information and the formation of prices [7].

With the advancement of the algorithmic trading [18,22], the reporting speed and the volume of market data have been increasing significantly.[1] In particular, more data on both news and stock prices make it increasingly difficult to process them manually. Therefore, how to use computer algorithms to process and model market information has become a challenging problem in the practice of both computer science and finance.

Computer science researchers have studied the problem by approaches that can be mainly classified as two categories: classification and regression. The classification approaches cast the problem as two-class or multi-class classification by making directional predictions in the form of a label that tags a market event[2] [12,13,35–38,49,50].

One major issue with the classification approach is *signal strength bias*, which affects how to quantify, interpret and compare

* Corresponding author at: School of Computer Science, Fudan University, Shanghai 200433, China.
E-mail address: zhusf@fudan.edu.cn (S. Zhu).

[1] Bloomberg (http://www.bloomberg.com/) and ThomsonReuters (http://thomsonreuters.com/) publish their real-time price tickers (quotes and trades) and news tickers (titles, bodies and tags) through networks to the world within a few milliseconds.

[2] Market events can be patterns that are extracted from information sources, such as a "trending" or a "reverting" in prices.

http://dx.doi.org/10.1016/j.neucom.2014.04.043
0925-2312/

the strength of the prediction labels. For example, for a specific data set, in one setting we may use $\pm 1\%$ simple return as two thresholds which determine three classes, i.e., *positive*, *neutral*, and *negative*, or in another setting we use $\pm 0.5\%$ and $\pm 1.5\%$ as four thresholds which determine five classes, namely, *extreme positive*, *positive*, *neutral*, *negative* and *extreme negative*. Thus, the following scenarios could happen:

- Assuming that in the first setting, we receive two output prediction labels that are eventually the same (e.g. *positive*), it is difficult to tell whether both of them indicate the same price return, or it predicts $+1\%$ in the first case and $+2\%$ in the second case.
- Assuming that we receive the same class label (e.g. *positive*, again) in both settings, it is hard to determine whether the label in the first setting is stronger than the one in the second setting or vice versa, especially when there are overlaps between the two labeling methods.
- The labeling method itself has largely determined the final accuracy of a system. In an extreme case, if we choose a large threshold, where nearly all of the simple returns do not exceed this threshold (e.g. 10 times return for a short period of time), then every sample would be labeled with *neutral*, and the final accuracy would therefore become 100% regardless of any classification models used.

Differing from the directional predictions by classification approaches, regression approaches make numerical forecastings [5,42,51]. Because regression does not have the signal strength bias, we adopt regression approaches rather than classification approaches. To be specific, we use regression models in our experiments to predict short-term stock price returns in order to overcome the signal strength bias.

Another issue with aforementioned approaches, which is also the main problem we deal with in this paper, is that models using either of the market information sources would lead to *information bias*. To formalize this, we denote the news information set as **N** and the price information set as **P**. Approaches with information bias model the problem as either of

$$\pi : \mathbf{N} \mapsto \mathbf{L} \quad \text{or} \quad f : \mathbf{N} \mapsto \mathbf{R},$$
$$\pi : \mathbf{P} \mapsto \mathbf{L} \quad \text{or} \quad f : \mathbf{P} \mapsto \mathbf{R}, \tag{1}$$

where **L** denotes a set of nominal labels and **R** is a set of estimated numerical values. Fig. 1 illustrates market scenarios that could happen. At time point t_0, the subsequent price movement is affected by both market news and short-term prices. However, if a model uses only partial information, it cannot explain the mapping from the input combinations to the outputs:

- If a model uses only **P**, a rational prediction in the left figure would be "trending down without reverting", and in the right figure it would be "trending up without reverting". Thus, "reverting and trending up" in the left figure and "reverting and trending down" in the right figure would increase the error rate of the model.
- If a model uses only **N**, derived f would not be able to explain why the price is still "trending down" when good news is released, as shown in the left figure, as well as why the price is still "trending up" when bad news is released, as shown in the right figure.

Considering **N** and **P** together, we believe that both information sources play important roles in driving the future trend of the prices.

In summary, to overcome both signal strength bias and information bias, we cast the problem of stock price prediction as

$$f : \{\mathbf{N}, \mathbf{P}\} \mapsto \mathbf{R}, \tag{2}$$

where both **N** and **P** are used. We adopt Multi-Kernel Support Vector Regression (MKSVR) as f in our system. The reasons are twofold:

1. MKSVR is a regression approach. The outputs of MKSVR are numerical values rather than user-defined *categories*. This would overcome the signal strength bias.
2. MKSVR integrates two information sources in an effective way. Since **N** and **P** have heterogenous features, simply combining those features is insufficient. MKSVR, which could have multiple sub-kernels, uses one sub-kernel to handle one of the information sources. MKSVR learns the weights of sub-kernels and determines which information source is more effective in prediction, where the weights could be interpreted as the extent to which one information source contributes to the prediction.

We design and implement a system for stock market predictions using MKSVR that combines both news articles and Hong Kong Stock Exchange (HKEx) tick prices. In particular, MKSVR has two sub-kernels: one is responsible for news articles, while the other accepts short-term historical prices. After learning the weight of each sub-kernel, the derived model makes numerical forecasting on stock short-term returns. Compared with three other baseline models implemented in the experiments, MKSVR has achieved the best performance with the smaller regression error.

The contributions of this paper are summarized as follows:

1. We build up a system with work flows that use MKSVR to integrate both news articles and stock tick prices, rather than only one of them, to quantitatively make intra-day short-term stock return predications.
2. Our approach by multiple kernel learning (MKL) can effectively integrate both news articles and stock tick prices, outperforming

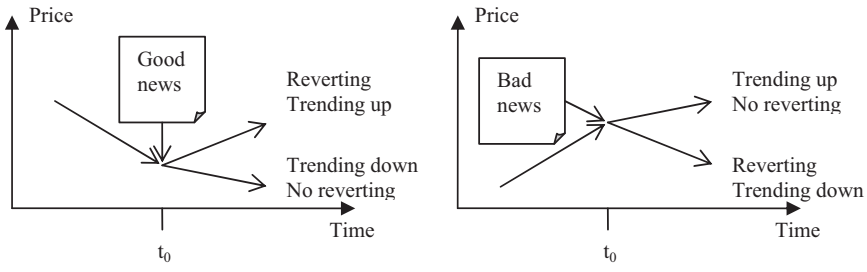


Fig. 1. Possible scenarios of price movements based on the impact of news articles and short-term prices. At time point t_0, the inputs are news article and short-term price trend before t_0. In the left figure, the outputs are either "reverting and trending up" or "trending down without reverting". In the right figure, the outputs are either "trending up without reverting" or "reverting and trending down".

three baseline methods (which use only one information source or use a naive combination of the two sources) on real tick-by-tick data.

The rest of this paper is organized as follows. Section 2 reviews the literature on traditional approaches for stock market prediction, and the formulation of MKL and its applications. Section 3 presents our proposed system and describes the design of the experiments. Section 4 shows the experimental results and discussions. Section 5 gives our conclusion and future work directions.

2. Related work and background

2.1. Traditional approaches

Some useful observations have been made in the finance domain. Ederington and Lee [8] observe that there is always a big increment of the standard deviation of five-minutes returns on a day when a government announcement on a financial relevant policy is released at 8:30 am. Tetlock [44] analyzes the content of the "Abreast of the Market" column in *Wall Street Journal*, and finds that pessimistic words predict low stock returns. Tetlock et al. [45] also find that firm-specific future earnings and returns could be predicted by news when they analyze the tone of firm-specific news.

Analysis of news articles has been reported in the literature of computer sciences. Following the approach of text mining on news articles, Seo et al. [39] build a multi-agent system for intelligent portfolio management, which can assess the risk associated with companies by analyzing news articles. Yu et al. [52] propose a four-stage Support Vector Machine (SVM) based on the multi-agent ensemble learning approach for credit risk evaluation. Fung et al. [13] classify news articles into different categories and predict the directional impact of newly released news articles. The AZFinText system, built by Schumaker and Chen [37], makes not only directional prediction but also quantified estimation of prices.

As illustrated in Fig. 2, the processing pipeline of the approaches that use the news information source could be summarized below:

1. *Representation of news articles.* A piece of news is usually represented as a term vector by using the Vector Space Model after the removal of stop words and feature selection [11,21,25,26,46,48]. Sentiment analysis is occasionally employed to analyze news at a semantic level [15,24,30,31].

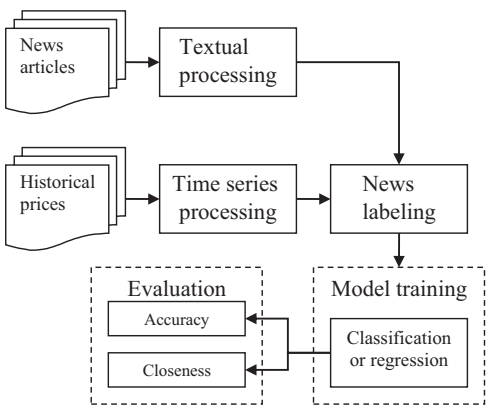


Fig. 2. Architecture of traditional approaches.

2. *Labeling.* Each piece of news is then assigned with a label. In a classification model, pieces of news are sorted by their time stamps, and labeled with nominal values, such as *positive/neutral/negative*. While in regression approaches, news is simply labeled with a continuous value, such as daily stock price return. In this paper, we use short-term price return for intra-day prediction.
3. *Model training and evaluation.* Machine learning models are employed in this step. Except for evaluating models by means of standard metrics, such as Precision, Recall, and Mean Square Error (MSE), some researchers conduct preliminary simulations [12,36], where strategies trade stocks in a virtual market using historical real data or data that was generated by simulated agents. *Return rate* is the measure of the trading performance. However, how to design a functional trading strategy that could make full use of the predictions of a model (not strategies that simply buy-and-hold) is beyond the scope of this paper.

Besides the work on news, there are also many published papers on mining signals from stock prices. Guo et al. [17] focus on the architecture of the neural network and develop a sparsely connected network model, which achieves the better performance than traditional neural networks with respect to three data sets (Microata, Finance Data and Telecommunication Data). For forecasting the changes of long-term index, Hung and Lin [20] develop an intuitionistic fuzzy least-squares support vector machine with genetic algorithms (IFLS-SVRGAs). Lin et al. [29] recently apply FLS-SVR-GA to forecast the seasonal revenue of a company. Gestel et al. [47] use a Bayesian evidence framework and apply it to least-squares support vector machine regression for price and volatility prediction. Tay et al. [42,43] and Cao et al. [3] modify the SVM objective function and make C an adaptive parameter for non-stationary time series. For predicting index prices, Kim [23] concludes that the performance of SVM is better than that of back-propagation neural networks. Applying SVM to predict S&P 500 daily prices, Cao et al. [4,5] also find that SVM has the better performance based on the metrics of normalized mean square error and mean absolute error. Huang et al. [19] predict the price directional movement of NIKKEI 225 index using SVM. After comparing SVM with linear discriminant analysis, quadratic discriminant analysis, and back-propagation neural networks, they reach the same conclusion. To summarize, the steps of the approaches in this category are (1) Preprocessing raw prices. To make historical prices more indicative, prices in absolute price levels are sometimes translated into price indicators. (2) Patten classification. Patterns of prices are then classified (or regressed) by models into predetermined categories (or estimated values).

Note that previous work made use of only one type of information sources, i.e., either finance news or historical prices. In contrast, our proposed approach use both information sources which are integrated for market prediction by MKL.

2.2. Multiple Kernel learning

In order to have a deep understanding of the problem and to interpret the derived decision function in a simple way, we describe MKL used in our system in this section.

The objective function of a single kernel SVR is

$$\min_{\mathbf{w},b} \ \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + C\sum_{i=1}^{N}(\xi_i+\xi_i^*), \text{s.t.} \quad (\langle \mathbf{w},\phi(\mathbf{x}_i)\rangle + b) - y_i \le \varepsilon + \xi_i,$$
$$y_i - (\langle w,\phi(\mathbf{x}_i)\rangle + b) \le \varepsilon + \xi_i^*, \quad \xi_i,\xi_i^* \ge 0, \quad i=1,\ldots,N, \tag{3}$$

where N is the number of training instances, and C is a penalty term which balances between training error and model complexity. $\xi_i^{(*)}$ ($\xi_i^{(*)}$ refer to ξ_i and ξ_i^*) are slack variables, where ξ_i means

that the estimated value is more than y_i at least ε and ξ_i^* means that the estimated value is lower than y_i at least ε. The derived regression function is

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*)\mathbf{K}(\mathbf{x}_i, \mathbf{x}) + b, \tag{4}$$

where $\alpha_i^{(*)}$ are the Lagrangian multipliers (dual variables), $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ is a kernel function, and $b = y_k + \varepsilon - \sum_{i=1}^{N} (\alpha_i - \alpha_i^*)\mathbf{K}(\mathbf{x}_i, \mathbf{x}_k)$ with any $\alpha_k^{(*)} \in (0, C)$.

MKL uses multiple sub-kernels with their convex combinations

$$\tilde{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^{l} \beta_s \mathbf{K}_s(\mathbf{x}_i, \mathbf{x}_j), \tag{5}$$

where l is the number of sub-kernels, $\beta_s \geq 0$ and $\sum_{s=1}^{l} \beta_s = 1$. Each sub-kernel handles a set of features. The kernel fusion of the MKSVR could be written as

$$\Phi(x) = [\sqrt{\beta_1}\phi_1(x), \ldots, \sqrt{\beta_l}\phi_l(x)]. \tag{6}$$

Eq. (6) maps the input space into the feature space. The objective function and constraints for MKSVR become

$$\begin{aligned} &\min_{\mathbf{w},b} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{N} (\xi_i + \xi_i^*), \\ &\text{s.t.} (\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) - y_i \leq \varepsilon + \xi_i, \\ &\quad y_i - (\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \leq \varepsilon + \xi_i^*, \xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, N, \\ &\quad \beta_s \geq 0, \quad s = 1, \ldots, l, \quad \sum_{s=1}^{l} \beta_s = 1, \end{aligned} \tag{7}$$

where Φ is the vector of function mappings in Eq. (6). The regression function is

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*)\tilde{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}) + b, \tag{8}$$

where we have $\tilde{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, and $b = y_p + \varepsilon - \sum_{i=1}^{N} (\alpha_i - \alpha_i^*)\tilde{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_p)$, which is obtained from any $\alpha_p^{(*)} \in (0, C)$.

Sonnenburg et al. [41] recast the MKL problem [27] as a semi-infinite linear program (SILP), which can be efficiently solved using an off-the-shelf LP solver and a standard SVR implementation. To be specific,

$$\begin{aligned} &\max_{\theta, \beta_s} \quad \theta, \text{s.t.} \quad \sum_{s=1}^{l} \beta_s S_s(\alpha) \geq \theta, \quad \beta_s \geq 0, \\ &\quad \sum_{s=1}^{l} \beta_s = 1, \quad s = 1, \ldots, l, \\ &\quad \text{for all } \alpha \in (0, C), \end{aligned} \tag{9}$$

where we have

$$\begin{aligned} S_s(\alpha) = &-\frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathbf{K}_s\langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &- \varepsilon \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*). \end{aligned} \tag{10}$$

The MKL model has been implemented for making financial predictions. Using about 40 sub-kernels, Yeh et al. [51] apply MKSVR to stock historical prices for making inter-day close price forecasting. Fletcher et al. [10] also use MKSVR on foreign currency market prices and volumes for high frequency predictions. Note that both of them use features derived from market price data only. In contrast, this paper, which extends our preliminary work on market directional prediction [28], integrates news articles and market prices in order to improve the performance of quantitative stock prediction.

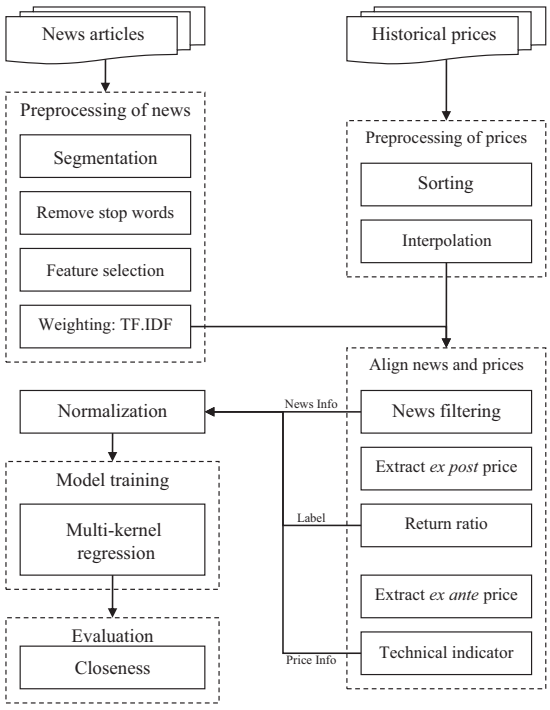


Fig. 3. Detailed processing pipeline of our system.

3. System architecture and design

The detailed processing pipeline is shown in Fig. 3. Section 3.1 describes the information sources that are fed into the system. Sections 3.2 and 3.3 present the processing of news articles and historical prices. Section 3.4 presents how to align news and prices. Sections 3.5 and 3.6 discuss data normalization and model training.

3.1. Information sources

Two information sources are integrated in the system: (1) news articles and (2) historical tick prices. News and prices data have following properties:

- *Time stamped*. Each news article and price (bid, ask and trade) are associated with a time stamp that records their release time. By means of the time stamps, the system can align news with prices accordingly.
- *High frequency*. Being designed to make the intra-day short-term prediction, the system requires that the data should at least be published at the minute interval in order to make possible predictions and produce convincing results. High frequency data have following characteristics which need to be handled during preprocessing:
  - *Big volume*. Tick data have much more records than daily data over the same time period. For one trading day, daily data have only *open*, *high*, *low* and *close* prices, while the tick data, take second-based tick data for example, have 3600 ticks × 4 trading hours = 14,400 entries, which are thousand times of the daily data.
  - *Variant interval*. Time intervals between different tick data entries are different. As quotations and transactions may happen at any time point, the time intervals between consecutive entries are not always the same.

3.2. Preprocess news articles

News articles need to be preprocessed, and the main steps are listed below:

1. *Chinese segmentation*.[3] News articles are segmented by a Chinese segmentation software.[4] Although the segmentation software can produce outputs of high quality, many financial terminologies cannot be identified correctly. A finance dictionary is thus employed to refine the segmentation results.
2. *Word filtering*. This step filters unimportant words, such as stop words,[5] and extracts representative words, such as nouns, verbs, and adjectives [35–38].[6]
3. *Weighting*. We calculate the widely used $TF \cdot IDF$ (term frequency $\times$ inverse document frequency) value for each word as its weight. Each news article is thus represented by a vector of non-negative numbers.
4. *Feature selection*. Feature selection is employed to reduce the feature dimensions. Since the word list extracted from the news articles is very long, feature selection is needed to filter the words that are not useful for predicting the labels. Feldman [9, Chapter IV.3.1] selects about top 10% words with high occurrences as the features. In our system, we use χ^2 feature selection. χ^2 evaluates features individually by measuring their χ^2 statistics with respect to the classes. For example, denote $p(t_k)$ as the percentage of articles in which word t_k occurs, its complementary as $p(\overline{t_k})$, and class labels as $p(+)$ and $p(-)$. Then, we have

$$\chi^2(t_k, l) = \frac{(ad-bc)^2 N}{(a+b)(a+c)(b+d)(c+d)}, \tag{11}$$

where l is the label vector, and

$$\begin{aligned} a &= N \cdot p(\overline{t_k}, -), \\ b &= N \cdot p(\overline{t_k}, +), \\ c &= N \cdot p(t_k, -), \\ d &= N \cdot p(t_k, +). \end{aligned} \tag{12}$$

A higher χ^2 score indicates that feature t_k is more informative for prediction. We discretize the continuous label with cutoff $\pm 0.3\%$ (which is commonly considered as the transaction cost of the market), and select top 1000 words with high χ^2 scores.

3.3. Preprocess historical prices

Tick data are preprocessed through the following steps:

1. *Sorting*. Since the quotations and transactions may arrive in different orders, we must first sort the price data by their time stamps.
2. *Interpolation*. Since the time intervals between consecutive price records are not equal, there can be no record over some time periods. This raises the question of what prices should be during these time periods. This problem can be solved in two ways: (1) linear time-weighted interpolation proposed by Dacorogna et al. [6] and (2) nearest closing price. The second method splits tick data series along time axis with window size τ, and samples the closing price $p(\tau)$ of each τ in an iterative way:

$$p(\tau_i) = \begin{cases} p_{\max\{t\}} & \text{if } \{t\} \subseteq \tau_i, \ \{t\} \neq \phi, \\ p(\tau_{i-1}) & \text{if } \{t\} = \phi, \end{cases} \tag{13}$$

where set $\{t\}$ contains all the time stamps that are in τ. If there is no record in period of τ_i, $p(\tau_{i-1})$ is taken as $p(\tau_i)$. Although both methods are sound, we adopt the second one because of its fast speed (commonly implemented in industry).

3.4. Align news articles and prices

3.4.1. Filter news

Following the approach proposed by Schumaker et al. [36], the news should be filtered for two reasons: (1) *Trading hour limitation*. Take HKEx for example, according to the regulation of HKEx year 2001, 10:00am–12:30pm and 14:30pm–16:00pm are the continuous trading sessions. Since we make short-term predictions, only the news articles released during the trading hours are assumed to have impact on stock prices. Besides the limitation of trading hours, the opening 20 min in the morning session and opening 20 min in the afternoon sessions are also eliminated in order to absorb the impact of news that is released during the night and lunch break.[7] (2) *News impact overlapping*. As illustrated in Fig. 4, two news articles, denoted as d_1 and d_2, are about the same stock within the prediction time horizon Δ. In this scenario, it is hard to tell whether the change of the price at time $t_{+\Delta}$ is caused by d_1, or d_2, or both. In order to avoid the conflict and follow the same method in Schumaker et al. [36], d_1 is eliminated in our system.

3.4.2. Extract and process ex post prices

Price movement after the news is released (termed as *ex post* prices) are used as the labels in the system. Gidofalvi [14] claims that the impact of news reaches the greatest level within 20 min after it is released. Without the knowledge of how to precisely calculate how long the impact will last, we label the news articles by future 5, 10, 15, 20, 25, and 30 min *returns*. This also extends the work [36] by expanding estimating one future time point to six.

More precisely, suppose that news is released at t_0, and the current price is p_0. Thus, the corresponding prices of future 5, 10, 15, 20, 25, and 30 min, denoted as p_{+5}, p_{+10}, p_{+15}, p_{+20}, p_{+25}, and p_{+30}, can also be extracted. As such, we convert *ex post* prices into *return* by

$$R = \frac{p_i - p_0}{p_0}, \tag{14}$$

where R is used as the label of a news article. Note that, if $t_{+\Delta}$ violates the requirement of the *trading hour limitation*, the news will be eliminated.

3.4.3. Extract and process ex ante prices

Prices from 30 min to 1 min before a piece of news is released are termed as *ex ante* prices in our system.[8] *ex ante* prices are

[3] This step can be skipped for English news.
[4] Software can be downloaded from http://ictclas.org/
[5] Baidu stop word list is adopted.
[6] The segmentation software tags each word with Part Of Speech (POS). We keep the nouns, verbs and adjectives following many previous researchers.

[7] From trading point of view, (1) overnight news articles are usually taken into consideration when analyzing mid- or long-term signals, such as daily, weekly and even monthly; (2) for high frequency trading, systems usually use short-term intra-day signals. Since HKEx has an auction session in the morning before the continuous session starts, overnight news impact is assumed to be *absorbed* and has been reflected in the auction prices. Thus, at the prediction frequency of our system, only intra-day news is considered.

[8] The length of the *ex ante* time window is selected based on two principles: (1) The length of the window should be at least as long as the length of the prediction time horizon. The *ex ante* prices could be considered as the information set that is used to do the regression for the future price movement prediction. Since the longest prediction time horizon is 30 min in our setting, the length of the window should be greater or equal to 30 min. (2) The length of the window should

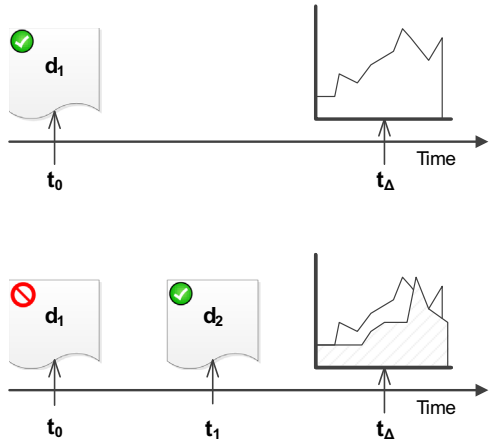


Fig. 4. Example of news filtering. (1) When only d_1 appears in $[t_0, t_\Delta]$, we assume that the price movement at t_Δ is related to d_1. (2) If both d_1 and d_2 are in $[t_0, t_\Delta]$, we assume that the price movement at t_Δ is related to d_2.

sampled at a 1 minute interval. Therefore, there are 30 points for each piece of news. If we simply take the 30 points as 30 features, the machine learning model, however, will assume that the 30 features are independent of each other, which implies that the sequential dependency of the price series $p_{-30}, p_{-29}, \ldots, p_{-1}$ is not preserved. Thus, the model learns no sequential information from the series. Cao and Tay [5,42] convert the series into RDP indicators. Following their approach, we use the same formulae of RDPs, which are listed in Table 1, to convert the *ex ante* prices.

In addition to RDPs, we employ some other market indicators from stock technical analysis. The formulae and descriptions of the technical indicators are listed in Tables 2 and 3, where p_i is the price at time point i, and q is the number of *ex ante* price sample points that are used in the formulae.

By means of the conversions, 30 *ex ante* price points become 6 RDPs and 5 technical indicators, all of which are simply referred to as indicators in the following sections.

3.5. Normalization

After the preprocessing in the previous steps, we have obtained (1) a group of news article instances (a set of vectors, denoted as N); (2) a group of technical indicator instances (a set of vectors, denoted as I); and (3) a vector R containing numerical labels. Each vector of N and I corresponds to one piece of news and its *ex ante* price indicators, and each entry in R corresponds to the future price return.

The matrix N contains all the instances of news. As shown in Section 3.2, all the news are represented by vectors of $TF \cdot IDF$ values, which are non-negative by definition. We denote non-negative valued features as f_+ and use Eq. (15) to normalize them

$$norm(f_+^{k,i}) = \frac{f_+^{k,i} - \min\{f_+^k\}}{\max\{f_+^k\} - \min\{f_+^k\}}, \tag{15}$$

where k indicates the k-th non-negative feature, and $f_+^{k,i}$ is the i-th element of feature f_+^k. The range of values after the normalization is $[0, 1]$. In contrast, the features in I can take both positive and negative values, which are denoted simply as f and normalized by

$$norm(f^{k,i}) = \frac{f^{k,i}}{\max\{|f^k|\}}. \tag{16}$$

After the normalization, values range is $[-1, 1]$.

Table 1
The formulae of RDPs.

RDP	Formula
RDP-5	$100*(p_i - p_{i-5})/p_{i-5}$
RDP-10	$100*(p_i - p_{i-10})/p_{i-10}$
RDP-15	$100*(p_i - p_{i-15})/p_{i-15}$
RDP-20	$100*(p_i - p_{i-20})/p_{i-20}$
RDP-25	$100*(p_i - p_{i-25})/p_{i-25}$
RDP-30	$100*(p_i - p_{i-30})/p_{i-30}$

Table 2
Stock technical indicators.

Indicator	Formula
RSI(q)	$100*UpAvg/(UpAvg+DownAvg)$ $UpAvg = \sum_{p_i > (\sum_i p_i)/q}(p_i - (\sum_i p_i)/q)$ $DownAvg = \sum_{p_i < (\sum_i p_i)/q}(p_i - (\sum_i p_i)/q)$
RSV(q)	$100*(p_0 - \min_q(p_i))/(\max_q(p_i) - \min_q(p_i))$
R(q)	$100*(\max_q(p_i) - p_0)/(\max_q(p_i) - \min_q(p_i))$
BIAS(q)	$100*(p_0 - (\sum_i p_i)/q)/((\sum_i p_i)/q)$
PSY(q)	$100*(\sum 1\{p_i > p_{i-1}\})/q$

Table 3
Indicator description.

Indicator	Description
RSI	Relative strength index
RSV	Raw stochastic value
R	Williams index
BIAS	Bias
PSY	Psychological line

3.6. Model training

Three groups of models have been setup for comparison: (1) the model is based only on one information source; (2) the model is based on two information sources, and uses the simple feature combination method; and (3) the model is based on two information sources, and uses the multiple kernel method. The details about the setup of the models are described as follows:

1. *News articles only*. This model takes labeled news instances as the input of SVR. It tests the prediction performance when only news articles are available (see Fig. 5(1)).
2. *ex ante prices only*. This model takes labeled indicator instances as the input of SVR. It tests the prediction performance when only historical prices are available (see Fig. 5(2)).
3. *Naive combination of news articles and ex ante prices*. This model uses the simple combination of the features of news articles and prices. The naive combination refers to the concatenation of the 1000 features from news and 11 features from indicators to form a feature vector with 1011 dimensions. Since the instances in N and the instances in I are one–one

(*footnote continued*)
not be too long. The total trading hour of a trading day is 4 h. As described in Section 3.4.1, there are some extra steps in the system that will eliminate some pieces of news which are near the market open/close and lunch break. If the length of the window is long, e.g. 1 h, many pieces of news will be eliminated. Based on these two considerations, we empirically choose 30 min for the window.

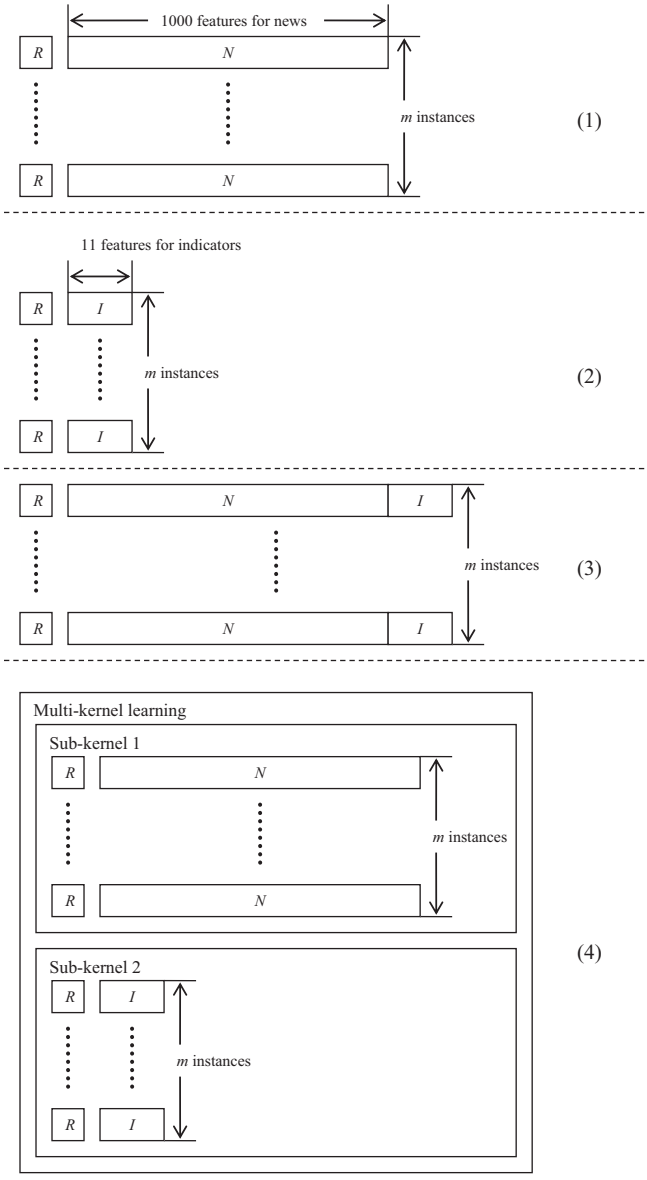


Fig. 5. Model setup: (1) news articles only; (2) *ex ante* prices only; (3) naive combination; and (4) MKSVR.

correspondence, the label of instances after feature combination is the same as the one in (1) and (2) (see Fig. 5(3)).

4. *MKSVR*. MKSVR uses multiple kernels to combine the features of news articles and *ex ante* prices. SHOGUN [40] Ver. 0.10.0 with MATLAB interface, an implementation of MKSVR, is used in our experiments (key settings are listed in Table 4). We set up two sub-kernels, as shown in Fig. 5 (4), one handles the features of *N* and the other contains the features of *I*.

For the naive combination using SVR, we have

$$f(\mathbf{x}) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*)\mathbf{K}_{naive}(\mathbf{x}_i, \mathbf{x}) + b, \tag{17}$$

where $\alpha_i^{(*)}$ are Lagrange multipliers, and $\mathbf{x}_i$ $(i = 1, 2, \ldots, m)$ are labeled training samples of the 1011 feature vector. For MKSVR, the similarities are calculated among the instances of news, and among the instances of indicators. The derived similarity matrices are taken as two sub-kernels of the MKSVR, and

Table 4
Some key settings of SHOGUN used.

SHOGUN parameters	Setup
Interleaved optimization	Yes
Solver	Elasticnet
Kernel combination	Combined

weights β_N and β_I are learnt for sub-kernels

$$\tilde{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s \in \{N,I\}} \beta_s \mathbf{K}_s(\mathbf{x}_i^s, \mathbf{x}_j^s), \tag{18}$$

where $\beta_N,\ \beta_I \geq 0$ and $\beta_N + \beta_I = 1$.

4. Experimental results and discussions

4.1. Data sets

Both news articles and market prices of HKEx serve as the data sets of the experiments.

- *News articles*. The news articles of year 2001 used in our experiments are bought from Caihua.[9] The news contains the firm-specific reports and also the announcements that are related to the macro-economic environment in Hong Kong and around the world. All the news articles are written in Traditional Chinese. Each piece of news is associated with a time stamp (second-basis) indicating the release time of the news.
- *Market prices*. The market prices contain all the stocks' tick prices of HKEx in year 2001.

HKEx has thousands of stocks, but not all of them are traded actively in the market. We focus mainly on the constituents of Hang Seng Index[10] (HSI), which are liquid stocks. HSI includes 33 stocks in year 2001. According to the HSI change history, the constituents were changed twice in year 2001, which happened on 1st June and 31th July. Due to the *tyranny of indexing* [34], the price movement of newly added constituent is not rational and usually mispriced during the first few months. Therefore, we only select the stocks that had been constituents through the whole year. This reduces the total number of stocks to 23.

There are 28,885 pieces of news in total. In order to evaluate the performance of our system and remove the impact of September 11, the original whole year data set is split into August, September, and October, respectively, forming three training/testing sets: (I) 8/4 months, (II) 9/3 months, and (III) 10/2 months.[11] The numbers of samples before preprocessing are listed in Table 5, while the numbers of samples after preprocessing are listed in Table 6.

4.2. Parameter selection

During the training phase, parameters of the model are determined by grid search in 5-fold cross-validation. The RBF kernel is used in the experiments. We initially set $\epsilon = 0.01$, 0.001 and 0.0001, and find that (1) in the case of $\epsilon = 0.01$, the predictions give almost the same outputs because of too much tolerance;

[9] Caihua, http://www.finet.hk/
[10] Hang Seng Index, http://www.hsi.com.hk/
[11] The number before symbol '/' is the number of months that are used for cross-validation, while the number after '/' is the number of months for independent testing.

Table 5
Cross-validation/testing instances before preprocessing.

	Start	End	Amount
Cross-validation sets			
Set I	2001-01	2001-08	19,225
Set II	2001-01	2001-09	22,143
Set III	2001-01	2001-10	24,394
Testing sets			
Set I	2001-09	2001-12	9660
Set II	2001-10	2001-12	6742
Set III	2001-11	2001-12	4491

Table 6
Cross-validation/testing instances after preprocessing.

	5 m	10 m	15 m	20 m	25 m	30 m
Set I						
CV	2057	1964	1867	1754	1680	1603
Test	1273	1237	1207	1168	1137	1107
Set II						
CV	2525	2415	2308	2175	2086	1993
Test	805	786	766	747	731	717
Set III						
CV	2834	2717	2606	2461	2363	2264
Test	496	484	468	461	454	446
Total	3330	3201	3074	2922	2817	2710

and (2) in cases of $\epsilon = 0.001$ and $\epsilon = 0.0001$, the results are close if we preserve three decimal points. Therefore, we set ϵ as 0.001 in the experiments. For parameter C, we try $C = 1$, 10, and 100, and find that when C is large, both single kernel and multi-kernel regression models tend to over-fit the data and consume too much CPU time. In order to balance the bias and variance, and also to run the experiments in reasonable time, we set $C=1$ in the cross-validation. The same approach and setting are used in some pioneer work, such as Yeh et al. [51], where $C=1$ and $\epsilon = 0.001$. For γ, we set $\gamma = 0.01 \times 2^k$ for the nonlinear search, where k starts from 0 to 15 with a step size of 1. Thus, there are 16 combinations of parameters at each time point in total.

For each combination of the parameters, 5-fold cross-validation is conducted to validate the trained model. When applying cross-validation on time series data, special care has to be taken [1,2]. In contrast to simply randomly picking a certain subset from the data set, the temporal dependencies in time series data need to be *de-overlapped* while re-sampling, which will otherwise cause the "look-into-future" issue. Instead of sampling at the instance level, we sample at the trading day level, i.e., each day is taken as the minimal unit for sampling. Since the system only makes intra-day prediction, instances from different days are considered independent. We equally split the training set into 5 parts, and 4 of the 5 parts are used to train the model while the left 1 part is for validation. Among all the parameter combinations, the one with the best performance is selected to configure the final model for independent testing.

4.3. Experimental results and findings

Regression error, which indicates how close (small error) a predicted value is to its true value, is measured by Root Mean Square Error (Eq. (19)) and Mean Absolute Error (Eq. (20)) in our experiments

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{19}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{20}$$

With the selected parameters, we perform the 5-fold cross-validation 10 times for each of prediction time window, using the average RMSE and MAE as the performance measurement. Tables 7 and 9 report the RMSE and MAE results for cross-validation, respectively. Their independent testing results are listed in Tables 8 and 10. The best scores are highlighted in the bold fonts and the underlined numbers are the second best results.

Compared with the results reported by Schumaker [36], which includes large regression errors (greater than 1, even 10), our results are smaller than 1. The reason is that we use price *Return* in our experiments instead of raw prices. For analyzing different stocks, the conversion of the stock prices into *Return* may smooth their different price levels.

Examining the experimental results, we can reach the following conclusions:

1. From Tables 7 and 8, it is clear that MKSVR outperforms three other baseline models. The MKSVR has achieved 15 best results

Table 7
RMSE results of cross-validation.

	5 m	10 m	15 m	20 m	25 m	30 m
Set I						
Indicator	0.107	0.109	0.133	0.153	0.132	**0.147**
News	0.100	0.114	0.157	0.191	0.158	0.196
Naive	0.084	0.122	0.164	0.177	0.157	0.183
MKSVR	**0.079**	**0.087**	**0.128**	**0.147**	**0.122**	0.150
Set II						
Indicator	0.097	0.102	0.135	0.124	0.090	**0.101**
News	0.096	0.104	0.147	0.131	0.093	0.110
Naive	0.092	0.108	0.146	0.131	0.092	0.106
MKSVR	**0.087**	**0.096**	**0.129**	**0.117**	**0.086**	0.106
Set III						
Indicator	**0.086**	0.141	0.140	**0.121**	0.105	0.108
News	0.110	0.127	0.159	0.151	0.116	0.127
Naive	0.112	0.121	0.150	0.141	0.094	0.127
MKSVR	0.089	**0.101**	**0.129**	**0.121**	**0.088**	**0.101**

Table 8
RMSE results of independent testing.

	5 m	10 m	15 m	20 m	25 m	30 m
Set I						
Indicator	0.223	0.278	0.171	**0.137**	0.173	**0.120**
News	0.131	0.160	0.199	0.201	0.181	0.198
Naive	**0.110**	0.174	0.198	0.186	0.175	0.187
MKSVR	0.112	**0.142**	**0.151**	0.138	**0.105**	0.122
Set II						
Indicator	0.160	0.138	0.146	0.196	0.199	0.194
News	0.120	0.131	0.169	0.166	0.168	0.144
Naive	0.112	0.126	0.166	0.166	**0.157**	**0.143**
MKSVR	**0.110**	**0.124**	**0.140**	**0.148**	0.160	0.145
Set III						
Indicator	**0.107**	0.281	0.269	**0.149**	0.459	0.178
News	0.142	0.190	0.241	0.203	0.190	0.173
Naive	0.169	0.164	0.249	0.204	**0.150**	0.187
MKSVR	0.110	**0.161**	**0.199**	0.151	0.159	**0.139**

and 3 second best results among 18 points of cross-validation. In addition, the MKSVR has produced 10 best results and 7 second best results among 18 points of independent tests. Similar conclusions could be drawn from MAE results reported in Table 9 and 10. As 5-fold cross-validation is performed 10 times in our experiments, we also calculate *t*-tests between the baseline models and the MKSVR using the 50 results (5-fold $\times$ 10 times). The test results are reported in Tables 11, 12 and 13. From the *p*-values, we can see that the MKSVR outperforms the other three models significantly (marked in bold fonts), except for 3 points that the MKSVR cannot achieve the best results and 2 points that the MKSVR achieves the same performance (which are marked as '–').

2. Naive combinations do not have any advantage over model 1 and model 2. From either cross-validation results or independent testing results, naive combinations cannot significantly improve what a single information source based model has achieved. Recall that the naive combination model constructs the feature vector in the form of $\langle N, I\rangle$, where the features of N and the features of I are actually two different types: the values of N features measure the *statistical soundness* of words in text documents, while the values of I features measure the historical price movement. The calculation of the similarity between vectors is neither comprehensive nor interpretable by naively combining N and I. Besides the potential inaccurate similarity calculation, the naive combination has the drawback of the *feature bias*. Since the length of the news features vector is nearly 10 times that of the indicator features vector, the features in the naive combinations strongly bias towards news features. As a consequence, the changes in news features are more likely to *contribute* to the changes of the similarities.
   Considering the performance of both the Naive combination and the MKSVR, we can conclude that the greater number of information sources cannot guarantee better performance. To achieve better results, the selection of an effective approach that can appropriately integrate multiple information sources is the key point.
3. In order to make sure that every sub-kernel in MKSVR actually participates in the prediction, the sub-kernels' weights learned by the MKSVR are plotted in Fig. 6. It can be seen that all sub-kernels have positive weights, implying that every sub-kernel in MKSVR contributes to the final prediction.
   Fig. 6 also illustrates that the weights of I are larger than N before 5 m, while the weights of N are larger than I from 10 m to 30 m. By following Gönen and Alpaydın's interpretation about the sub-kernels' weights [16], one reasonable interpretation here is as follows: in the MKSVR, the price information source contributes more than the news information source to short-term prediction. In other words, the MKSVR extracts more information from prices at 5 m predictions. As the model continues learning, the impact of news information becomes more important and the weights of N increase accordingly. This observation is consistent with the common knowledge that the information hidden in prices is more important for short-term forecasting, while for longer term, news impact may have more power.

Table 9
MAE results of cross-validation.

	5 m	10 m	15 m	20 m	25 m	30 m
Set I						
Indicator	0.053	0.065	0.100	0.131	**0.081**	0.139
News	0.063	0.076	0.107	0.131	0.111	0.138
Naive	**0.041**	0.084	0.115	0.126	0.113	0.132
MKSVR	0.044	**0.054**	**0.086**	**0.095**	**0.081**	**0.100**
Set II						
Indicator	0.072	0.060	**0.085**	0.086	0.059	0.069
News	0.070	0.078	0.115	0.104	0.075	0.075
Naive	0.068	0.073	0.121	0.106	0.057	0.070
MKSVR	**0.051**	**0.059**	0.086	**0.077**	**0.055**	**0.068**
Set III						
Indicator	0.068	0.079	0.082	**0.075**	0.063	**0.066**
News	0.070	0.084	0.110	0.103	0.080	0.089
Naive	0.075	0.074	0.104	0.098	**0.058**	0.091
MKSVR	**0.051**	**0.063**	**0.081**	0.078	0.059	**0.066**

Table 10
MAE results of independent testing.

	5 m	10 m	15 m	20 m	25 m	30 m
Set I						
Indicator	0.145	0.202	0.104	0.136	0.114	0.119
News	0.086	0.109	0.143	0.148	0.131	0.147
Naive	**0.058**	0.126	0.145	0.137	0.128	0.140
MKSVR	0.067	**0.090**	**0.103**	**0.090**	**0.069**	**0.080**
Set II						
Indicator	0.159	0.126	0.111	0.194	0.196	0.190
News	0.094	0.101	0.151	0.133	0.132	0.091
Naive	0.097	0.084	0.160	0.151	**0.095**	**0.087**
MKSVR	**0.067**	**0.077**	**0.095**	**0.096**	0.101	0.091
Set III						
Indicator	0.100	0.210	0.177	**0.091**	0.249	0.125
News	0.098	0.139	0.176	0.141	0.133	0.124
Naive	0.123	**0.102**	0.184	0.148	**0.089**	0.138
MKSVR	**0.067**	0.104	**0.132**	0.098	0.104	**0.088**

5. Conclusion

Stock market prediction is always an interesting research topic. In this paper, we have presented a system for forecasting stock price returns. By using MKSVR, the system quantitatively analyzes and integrates intra-day market news and stock tick price. Experiments have been conducted by using the market news and tick data of the Hong Kong stock market over one whole year. Results have demonstrated that the MKSVR is capable of making the better use of hidden information in news articles and historical stock prices than the models that simply use news articles to forecast stock prices. It has also been shown that the MKSVR outperforms those models that use just one information source.

MKSVR in the system currently makes use of two selected information sources, but the system could easily scale up and

Table 11
Set I *p*-value.

Set I	5 m	10 m	15 m	20 m	25 m	30 m
MKSVR/Indicator	**3.77E−17**	**1.50E−15**	**3.16E−02**	**9.44E−03**	**5.91E−03**	–
MKSVR/News	**6.83E−19**	**6.95E−30**	**4.21E−27**	**4.74E−37**	**2.09E−32**	**4.96E−41**
MKSVR/Naive	**2.19E−02**	**2.59E−32**	**1.71E−34**	**8.00E−26**	**1.89E−34**	**5.92E−32**

Table 12
Set II *p*-value.

Set II	5 m	10 m	15 m	20 m	25 m	30 m
MKSVR/indicator	**2.15E−07**	**1.22E−03**	**3.37E−03**	**1.53E−05**	**3.73E−02**	–
MKSVR/news	**9.40E−08**	**5.07E−06**	**8.08E−17**	**9.73E−13**	**3.19E−05**	**1.30E−02**
MKSVR/naive	**2.12E−03**	**1.49E−11**	**7.02E−18**	**1.11E−13**	**1.44E−03**	–

Table 13
Set III *p*-value.

Set III	5 m	10 m	15 m	20 m	25 m	30 m
MKSVR/indicator	–	**2.85E−24**	**2.95E−06**	–	**1.43E−14**	**5.68E−04**
MKSVR/news	**1.03E−21**	**3.54E−30**	**2.22E−33**	**2.87E−36**	**9.45E−37**	**3.61E−35**
MKSVR/naive	**4.68E−29**	**4.19E−18**	**8.24E−24**	**5.06E−24**	**1.20E−05**	**1.04E−38**

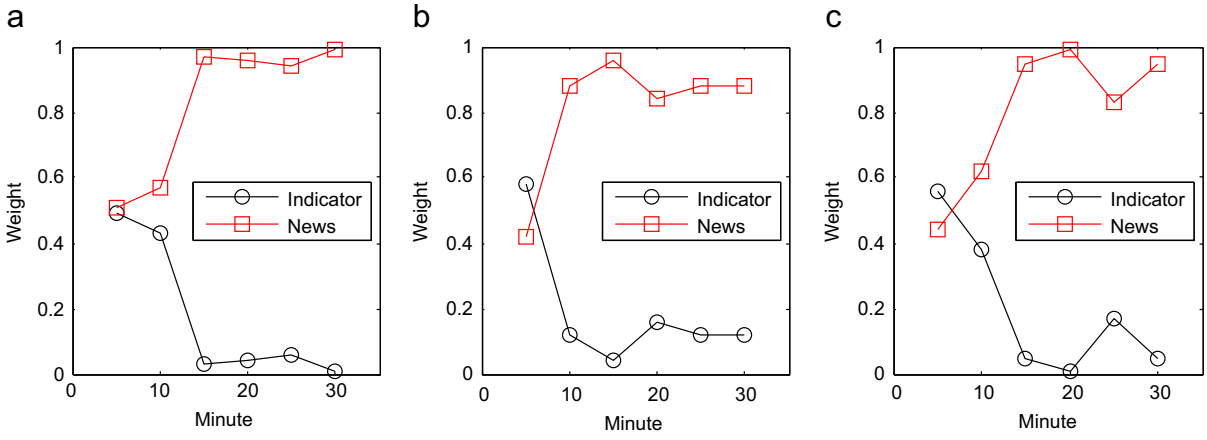


Fig. 6. MKSVR sub-kernel weights of data sets. (a) Set I, (b) set II and (c) set III.

accept more information sources. For the future work, the question as to what kind of information source can provide complementary information without redundancy, and how to convert data into a format that can be easily used by MKSVR, are worth being investigated. Furthermore, current information sources could be analyzed to form more sub-kernels. For example, positive and negative news could be classified by using sentiment analysis. As such, we can use different sub-kernels for the news in different sentiment categories. Finally, ensemble learning methods, such as Learn++ [32,33], which can deal with heterogeneous features separately and combine them at a classifier level, might also be a good approach to this problem.

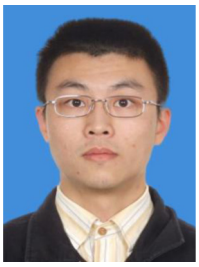
Acknowledgments

Xiaotie Deng is supported by the National Science Foundation of China (Grant no. 61173011) and a Project 985 Grant of Shanghai Jiaotong University. Shanfeng Zhu is supported by the National Science Foundation of China (Grant no. 61170097), and Scientific Research Starting Foundation for Returned Overseas Chinese Scholars, Ministry of Education, China.

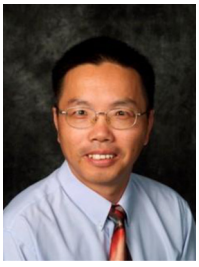
References

[1] Christoph Bergmeir, José M. Benítez, On the use of cross-validation for time series predictor evaluation, Inf. Sci. 191 (2012) 192–213.
[2] Felix Bießmann, Jens-Michalis Papaioannou, Mikio Braun, Andreas Harth, Canonical trends: detecting trend setters in web data, in: ICML '12, 2012.
[3] Lijuan Cao, Qingming Gu, Dynamic support vector machines for non-stationary time series forecasting, Intell. Data Anal. 6 (1) (2002) 67–83.
[4] Lijuan Cao, Francis E.H. Tay, Financial forecasting using support vector machines, Neural Comput. Appl. 10 (2) (2001) 184–192.
[5] Lijuan Cao, Francis E.H. Tay, Support vector machine with adaptive parameters in financial time series forecasting, IEEE Trans. Neural Netw. 14 (6) (2003) 1506–1518.
[6] Michel M. Dacorogna, An Introduction to High-Frequency Finance, Academic Press, 2001.
[7] David Easley, Nicholas M. Kiefer, Maureen O'Hara, One day in the life of a very common stock, Rev. Financ. Stud. 10 (3) (1997) 805–835.
[8] Louis H. Ederington, Jae Ha Lee, How markets process information: news releases and volatility, J. Finance 48 (4) (1993) 1161–1191.
[9] Ronen Feldman, James Sanger, The Text Mining Handbook, Cambridge University Press, 2007.
[10] Tristan Fletcher, Zakria Hussain, John Shawe-Taylor, Multiple kernel learning on the limit order book, J. Mach. Learn. Res. 11 (2010) 167–174.
[11] Tak-chung Fu, Fu-lai Chung, Vincent Ng, Robert Luk, Pattern discovery from stock time series using self-organizing maps, in: Workshop Notes of KDD2001 Workshop on Temporal Data Mining, ACM, 2001, pp. 26–29.
[12] Gabriel Fung, Jeffrey Yu, Wai Lam, News sensitive stock trend prediction, Adv. Knowl. Discov. Data Min. 2336 (2002) 481–493.
[13] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Hongjun Lu, The predicting power of textual information on financial markets, IEEE Intell. Inf. Bull. 5 (1) (2005) 1–10.
[14] Gyozo Gidófalvi, Using News Articles to Predict Stock Price Movements, Department of Computer Science and Engineering, University of California, 2001.
[15] Namrata Godbole, Srinivasaiah Manjunath, Skiena Steven, Large-scale sentiment analysis for news and blogs, in: ICWSM '07, 2007.
[16] Mehmet Gönen, Ethem Alpaydın, Multiple kernel learning algorithms, J. Mach. Learn. Res. 12 (2011) 2211–2268.
[17] Z.X. Guo, Wai Keung Wong, Min Li, Sparsely connected neural network-based time series forecasting, Inf. Sci. 193 (2012) 54–71.
[18] Larry Harris, Trading and Exchanges: Market Microstructure for Practitioners, Oxford University Press, USA, 2002.
[19] Wei Huang, Yoshiteru Nakamori, Shou-Yang Wang, Forecasting stock market movement direction with support vector machine, Comput. Oper. Res. 32 (10) (2005) 2513–2522.

[20] Kuo-Chen Hung, Kuo-Ping Lin, Long-term business cycle forecasting through a potential intuitionistic fuzzy least-squares support vector regression approach, Inf. Sci. 224 (2013) 37–48.
[21] Min Jiang, Changle Zhou, Shuo Chen, Embodied concept formation and reasoning via neural-symbolic integration, Neurocomputing 74 (1) (2010) 113–120.
[22] Barry Johnson, Algorithmic Trading DMA: An Introduction to Direct Access Trading Strategies, Myeloma Press, 2010.
[23] Kyoung-jae Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1–2) (2003) 307–319.
[24] Soo-Min Kim, Eduard Hovy, Determining the sentiment of opinions, in: Proceedings of the 20th International Conference on Computational Linguistics, COLING'04, Stroudsburg, PA, USA, Association for Computational Linguistics, 2004, pp. 1367–1373.
[25] Teuvo Kohonen, Self-organized formation of topologically correct feature maps, Biol. Cybern. 43 (1982) 59–69.
[26] Teuvo Kohonen, Panu Somervuo, Self-organizing maps of symbol strings, Neurocomputing 21 (1–3) (1998) 19–30.
[27] Gert R.G. Lanckriet, Tijl De Bie, Nello Cristianini, Michael I. Jordan, William Stafford Noble, A statistical framework for genomic data fusion, Bioinformatics 20 (16) (2004) 2626–2635.
[28] Xiaodong Li, Chao Wang, Jiawei Dong, Feng Wang, Xiaotie Deng, Shanfeng Zhu, Improving stock market prediction by integrating both market news and stock prices, in: Abdelkader Hameurlain, Stephen Liddle, Klaus-Dieter Schewe, Xiaofang Zhou (Eds.), Database and Expert Systems Applications, Lecture Notes in Computer Science, vol. 6861, Springer, Berlin/Heidelberg, 2011, pp. 279–293.
[29] Kuo-Ping Lin, Ping-Feng Pai, Yu-Ming Lu, Ping-Teng Chang, Revenue forecasting using a least-squares support vector regression model in a fuzzy environment, Inf. Sci. 220 (2013) 196–209.
[30] Bo Pang, Lillian Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval, vol. 2(1–2), 2008, pp. 1–135.
[31] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, EMNLP '02, Stroudsburg, PA, USA, vol. 10, 2002, Association for Computational Linguistics, pp. 79–86.
[32] Robi Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45.
[33] Robi Polikar, L. Upda, S.S. Upda, Vasant Honavar, Learn++: an incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 31 (4) (2001) 497–508.
[34] Jay R. Ritter, Behavioral finance, Pac-Basin Finance J. 11 (4) (2003) 429–437.
[35] Robert P. Schumaker, Hsinchun Chen, Textual analysis of stock market prediction using financial news articles, in: American Conference on Information Systems, AMCIS '06, 2006.
[36] Robert P. Schumaker, Hsinchun Chen, A quantitative stock prediction system based on financial news, Inf. Process. Manag. 45 (5) (2009) 571–583.
[37] Robert P. Schumaker, Hsinchun Chen, Textual analysis of stock market prediction using breaking financial news: the azfin text system, ACM Trans. Inf. Syst. 27 (2) (2009) 12:1–12:19.
[38] Robert P. Schumaker, Hsinchun Chen, A discrete stock price prediction engine based on financial news, Computer 43 (1) (2010) 51–56.
[39] Young-Woo Seo, Joseph Giampapa, Katia Sycara, Financial news analysis for intelligent portfolio management, Thesis, 2004.
[40] Sören Sonnenburg, Gunnar Rätsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio deBona, Alexander Binder, Christian Gehl, Vojtěch Franc, The shogun machine learning toolbox, J. Mach. Learn. Res. 99 (2010) 1799–1802.
[41] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, Bernhard Schölkopf, Large scale multiple kernel learning, J. Mach. Learn. Res. 7 (2006) 1531–1565.
[42] Francis E.H. Tay, Lijuan Cao, Application of support vector machines in financial time series forecasting, Omega 29 (4) (2001) 309–317.
[43] Francis E.H. Tay, Lijuan Cao, Modified support vector machines in financial time series forecasting, Neurocomputing 48 (1) (2002) 847–861.
[44] Paul C. Tetlock, Giving content to investor sentiment: the role of media in the stock market, J. Finance 62 (3) (2007) 1139–1168.
[45] Paul C. Tetlock, Maytal Saar-Tsechansky, Sofus Macskassy, More than words: quantifying language to measure firms' fundamentals, J. Finance 63 (3) (2008) 1437–1467.
[46] Alfred Ultsch, Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series, Kohonen Maps 46 (1999) 33–46.
[47] Tony Van Gestel, Johan A.K. Suykens, Dirk-Emma Baestaens, Annemie Lambrechts, Gert Lanckriet, Bruno Vandaele, Bart De Moor, Joos Vandewalle, Financial time series prediction using least squares support vector machines within the evidence framework, IEEE Trans. Neural Netw. 12 (4) (2001) 809–821.
[48] Feng Wang, Cheng Yang, Zhiyi Lin, Yuanxiang Li, Yuan Yuan, Hybrid sampling on mutual information entropy-based clustering ensembles for optimizations, Neurocomputing 73 (7) (2010) 1457–1464.
[49] Di Wu, Gabriel Fung, Jeffrey Yu, Zheng Liu, Integrating multiple data sources for stock prediction, in: Web Information Systems Engineering – WISE 2008, vol. 5175, 2008, pp. 77–89.
[50] Di Wu, Gabriel Fung, Jeffrey Yu, Qi Pan, Stock prediction: an event-driven approach based on bursty keywords, Front. Comput. Sci. China 3 (2009) 145–157.
[51] Chi-Yuan Yeh, Chi-Wei Huang, Shie-Jue Lee, A multiple-kernel support vector regression approach for stock market price forecasting, Expert Syst. Appl. 38 (2011) 2177–2186.
[52] Lean Yu, Wuyi Yue Yue, Shouyang Wang, Kin Keung Lai, Support vector machine based multiagent ensemble learning for credit risk evaluation, Expert Syst. Appl. 37 (2) (2010) 1351–1360.

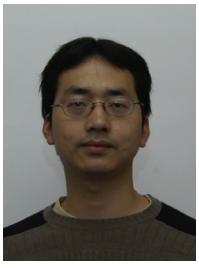


Xiaodong Li received the B.Sc. degree in Computer Science from the Nanjing University, China. He is currently a Ph.D. student at the City University of Hong Kong. His research interests include information retrieval, text mining and machine learning. He is also interested in the domain of market microstructure and algorithmic trading.



Xiaodi Huang is a Senior Lecturer in the School of Computing and Mathematics at Charles Sturt University, Australia. He received his Ph.D. degree in 2004. His research areas include visualization, data mining, and Web services. He has published over 100 scholar papers in international journals and conferences. He is a Regular Reviewer for several international journals, and serves as the committee members of international conferences. He is a Member of the ACM and IEEE Computer Society. For details, visit his homepage at http://www.csusap.csu.edu.au/~xhuang/.

Xiaotie Deng is a Chair Professor in the Department of Computer Science, Shanghai Jiaotong University. His research focus is on algorithmic game theory, which deals with computational issues on fundamental economic problems such as Nash equilibrium, resource pricing and allocation protocols such as auction and market equilibrium, as well as theory and practice in Internet market design.



Shanfeng Zhu received his B.S. and M.Phil. degrees in Computer Science from the Wuhan University, China, in 1996 and 1999, respectively, and his Ph.D. in Computer Science from the City University of Hong Kong in 2003. He is currently an Associate Professor with School of Computer Science, and Shanghai Key Lab. of Intelligent Information Processing, Fudan University, China. Before joining Fudan University in July 2008, he was a Postdoctoral Fellow at Kyoto University, Japan. His research focuses on developing and applying machine learning, data mining and algorithmic methods for Information Retrieval, Algorithmic Trading and Bioinformatics. He is a Member of the CCF and the ACM.