

Huang, Shian-Chang

Article

A big data analysis system for financial trading

Global Business & Finance Review (GBFR)

Provided in Cooperation with:

People & Global Business Association (P&GBA), Seoul

Suggested Citation: Huang, Shian-Chang (2017) : A big data analysis system for financial trading, Global Business & Finance Review (GBFR), ISSN 2384-1648, People & Global Business Association (P&GBA), Seoul, Vol. 22, Iss. 3, pp. 32-44,
<https://doi.org/10.17549/gbfr.2017.22.3.32>

This Version is available at:

<https://hdl.handle.net/10419/224376>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by-nc/4.0/>



A Big Data Analysis System for Financial Trading

Shian-Chang Huang^a

^a*Department of Business Administration, National Changhua University of Education, Taiwan*

A B S T R A C T

Big data analysis and cloud computing are becoming increasingly involved in the area of finance. The high computation capability enables one to apply complicated analysis utilizing large amounts of financial data. Big data analysis can find hidden patterns in large amounts of data. This capability can help investors in derivatives pricing, risk management and financial forecasting, and profitable trading. Owing to the high risk associated with trading financial options, this study aims to develop an intelligent option trading support system, where nonlinear or kernel canonical correlation analysis (KCCA) is used to extract the hidden forces that drive the price movement of an option, and a generalized dynamic kernel based predictors are employed to generate trading signals. Comparing with conventional feature extractions and pure regression models, the performance improvement of the new method is significant and robust. The cumulated trading profits are substantially increased. The resultant intelligent trading support system can help investors, fund managers and investment decision-makers make better and profitable decisions.

Keywords: Big Data Analysis, Kernel Canonical Correlation Analysis, Support Vector Machine, Financial Trading, Financial Option

1. Introduction

For financial investment and trading, data represents the ultimate challenge and the ultimate opportunity. The ability to manage and analyze data effectively can lead to better business decisions and lasting competitive advantages. To achieve the objectives, recent studies have put emphasis on exploiting big data techniques to manage and use datasets that are too large and complex to process with conventional

methods. Financial trading of securities using technical and quantitative analysis has been traditionally modelled by statistical techniques. In contrast to these statistical approaches, complex models coming from big data analysis or machine learning have emerged as new solutions. The extensive literature (Yoo, et al., 2007) has shown how some machine learning techniques have demonstrated that they are well-suited for quantitative analysis within the financial industry, as their capabilities of finding hidden patterns in large amounts of financial data may help in derivatives pricing, risk management and financial forecasting.

Financial investment is knowledge-intensive. Due to the high risk associated with trading financial derivatives, an online trading support system for option trading is important for investors in controlling

Received: Jan. 22, 2017; Revised: Jul. 21, 2017; Accepted: Sep. 4, 2017

† Shian-Chang Huang

Department of Business Administration, National Changhua University of Education, Taiwan

Tel. +886-953092968 E-mail: shuang@cc.ncue.edu.tw

and hedging their risk. Recently, many big data analysis or machine learning methods have been employed for algorithmic trading. However, their performance is not satisfactory, because these models do not reflect key features of financial information. Financial price is known to have a series of change-points, and good prediction for its movements is the key to successful trading. However, to make the correct trading decision is a very difficult problem because of the influence of the embedded noise and price fluctuations that confuse the price trend interpretation. The study employs wavelet analysis to resolve the financial big data, and uses kernel canonical correlation analysis (KCCA, Fyfe and Lai, 2001) to extract the hidden features that drive the price movement of an option, and a generalized dynamic kernel based predictors are employed to generate trading signals.

In recent years, with the advances in electronic transactions, vast amounts of data have been collected. In this context, the emergence of data mining technology enables researchers to build a financial trading support system (Boginski, Butenkob, and Pardalos, 2006; Bose and Mahapatra, 2001; Enke and Thawornwong, 2005; Hu et al., 2015; Irma, Stelios, and Steven, 2002; Patel et al., 2015; Wang and Weigend, 2004). Prior studies may use the charting heuristics of technical analysis to identify the bull flag, or focus on the forecast of financial prices. They often employ a statistical or artificial intelligence (AI) approach to facilitate the trading strategy-making. Typical decision models include decision tree (Wang and Chan, 2006; Wu, Lin, and Lin, 2005), case based reasoning (CBR) (Chun and Park, 2005; Oh and Kim, 2007), neural networks (Armano, Marchesi, and Murru, 2005; Chen, Leung, and Daouk, 2003; Thawornwong and Enke, 2004), fuzzy system (Wang, 2003), support vector machines (Żbikowski, 2015), and hybrid ones (Cervelló-Royo, Guijarro, and Michniuk, 2015; Chun and Kim, 2004; Fang and Xu, 2003; Zhang, 2003).

Option is a kind of financial derivatives. Its price is nonlinearly determined by future value of the underlying stock. Its equilibrium price is also

determined by the arbitrage opportunity in the market. Consequently, charting heuristic and traditional decision models are not suitable for option trading. In order to effectively predict the future movement of option price, high dimensional option data of the same strike price and major stock indices related to the market are used in this study for analyzing the price co-movement. In financial big data analysis, the most important is the feature extraction. Low-dimensional representative features can prevent the problem of the curse of dimensionality, and alleviate computation burden of the predictor. In feature extraction, KCCA is a relatively new methodology in big data analysis. KCCA is a nonlinear supervised method that finds maximal correlated input features for the outputs. What distinguishes KCCA from other methods (such as kernel principal component analysis) is that KCCA extracts a representative subspace that maximally correlated with the output. Based on the more discriminative subspace (or features), the performance of next stage predictor can be improved.

Although KCCA has found good applications in many fields, traditional KCCA operates on time domain is not very effective for the analysis of financial data. Market investors ranging from short-term traders (hedging strategists, market makers), medium-term traders (international portfolio managers), to long-term investors (central banks) are a diverse group. They operate on very different time scales. Consequently, traditional KCCA could not reveal key features of each group. On the contrary, wavelet analysis (Percival and Walden, 2000; Gençay et al., 2002) is a time-frequency technique which overcomes the limitations of traditional KCCA. Wavelet multi-resolution analysis is increasingly popular for financial time series analysis because it enables the practitioner to focus on particular time scales where trading patterns are considered important. The basic idea is to apply KCCA in wavelet domain to find hidden patterns. According to the arbitrage pricing theory (APT) of Ross (1976). If one could find useful hidden factors or sources that drive the market. The target asset could be price more correctly by these driving sources.

For trading decisions, the support vector machine (SVM) method (e.g., Vapnik, 1995; Cristianini and Shawe-Taylor, 2000; Schoelkopf, Burges, and Smola, 1999), is a good candidate for trend forecasting and buy/sell decisions. SVM and kernel machine (KM) models (Chen, 2006; Schölkopf and Smola, 2002) attract considerable interest. These techniques have been gaining more and more in popularity and are regarded as the state-of-the-art technique for regression and classification problems, with hugely successful application in many areas. The theoretical basis of SVM is the structural risk minimization principle, which gives excellent generalization properties. However, it has been shown that the standard SVM technique is not always able to construct parsimonious models in system identification (Drezet and Harrison, 1998). This shortcoming encourages the exploration of new methods for the parsimonious models under the framework of both SVM and KM. Tipping (2001) first introduced the relevance vector machine (RVM) method, which can be viewed from a Bayesian learning framework of the kernel machine and produces an identical functional form to the SVM/KM. The results given in Tipping (2001) demonstrate that the RVM has a generalization performance comparable to the SVM but requires dramatically fewer kernel functions or model terms. A drawback of the RVM algorithm is a significant increase in computational complexity, compared with the SVM method.

Recently, Rosipal and Trejo (2001) derived a novel method for constructing sparse kernel models based on partial least square regressions (PLSR, Wold et al., 1984; Rosipal and Kramer, 2006). Their kernel-based PLSR (KPLSR) algorithm extends traditional PLSR to high dimensional Reproducing Kernel Hilbert Space (RKHS). PLSR creates score vectors (components, latent vectors) by using the existing covariances between input and output variables while keeping most of the variance of both data sets. PLSR has proven to be useful in situations where the dimension of input variables is much greater than that of observations or high multi-collinearity among the input variables exists. This situation is also quite common in the case of kernel-based learning where

the original data are mapped to a high-dimensional feature space which dimension is much higher than the sample size.

The major innovation of this paper lies in integrating wavelet-domain KCCA with generalized dynamic kernel regressions (KPLSR) for option trading. In the first stage, wavelet-domain KCCA finds the hidden significant features suitable for financial forecasting and trading decisions. In the second stage, a dynamic KPLSR is constructed for price forecasting and buy/sell decisions. This study conducts a series of experiments on the proposed model using option data on the Taiwan Weighted Stock Index (TAIEX) to validate its trading profits. Empirical results indicated that comparing with other feature extraction methods and pure regressors, the proposed system performs best. The cumulated trading profits are significantly increased. The resultant intelligent investment decision support system can help investors make profitable decisions.

The remainder of the paper is organized as follows. Section 2 introduces prior research on financial trading, the neural network and support vector regression models. Section 3 describes wavelet-domain KCCA algorithm used in the new algorithmic trading system. Section 4 introduces the KPLSR algorithm. Section 5 describes the data used in the study, and discusses the experimental findings. Conclusions are given in Section 6.

2. Prior Research

2.1 Multi-layer Neural Networks

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way (Haykin, 1999). Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function defined by

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

as an activation function. Multi-layer networks use a variety of learning techniques, the most popular being back-propagation. The output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques (e.g., gradient descent), the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to train the model.

2.2 Support Vector Regressions(SVR)

The support vector machines (SVM) were proposed by Vapnik (1995). Based on the structured risk minimization (SRM) principle, SVMs seek to minimize an upper bound of the generalization error instead of the empirical error as in other neural networks. The SVM regression function is formulated as follows:

$$\hat{y} = \mathbf{w}^T \phi(\mathbf{x}) + b, \quad (2)$$

where \mathbf{x} is the input vector. ϕ is a nonlinear mapping from the input space to the feature space. T stands for the matrix transposition. The coefficients \mathbf{w} and b are estimated by minimizing

$$R(C) = C \frac{1}{N} \sum_{i=1}^N L_{\varepsilon}(y_i, \hat{y}_i) + \frac{1}{2} \|\mathbf{w}\|^2, \quad (3)$$

where

$$L_{\varepsilon}(y_i, \hat{y}_i) = \begin{cases} |y_i - \hat{y}_i| - \varepsilon & |y_i - \hat{y}_i| \geq \varepsilon, \\ 0 & \text{others,} \end{cases} \quad (4)$$

where y_i is the actual price in the i th period, and \hat{y}_i is the model prediction. Both C and ε are prescribed parameters. The first term $L_{\varepsilon}(y_i, \hat{y}_i)$ is called the ε -intensive loss function. This function indicates that errors below ε are not penalized. The term $\frac{1}{N} \sum_{i=1}^N L_{\varepsilon}(y_i, \hat{y}_i)$ is the empirical error. The

second term, $\frac{1}{2} \|\mathbf{w}\|^2$, measures the smoothness of the function. C evaluates the trade-off between the empirical risk and the smoothness of the model. Introducing positive slack variables ξ and ξ^* , which represent the distance from the actual values to the corresponding boundary values of ε -tube. Equation (3) is transformed to the following constrained formation:

$$\min_{\mathbf{w}, b, \xi, \xi^*} R(\mathbf{w}, \xi, \xi^*) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \left(\sum_{i=1}^N (\xi_i + \xi_i^*) \right) \quad (5)$$

Subject to

$$\mathbf{w}^T \phi(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i^*, \quad (6)$$

$$y_i - \mathbf{w}^T \phi(\mathbf{x}_i) - b \leq \varepsilon + \xi_i, \quad (7)$$

$$\xi_i, \xi_i^* \geq 0. \quad (8)$$

After taking the Lagrangian and conditions for optimality, one can get the model solution in dual representation,

$$\hat{y} = f(\mathbf{x}, \alpha, \alpha^*) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(\mathbf{x}, \mathbf{x}_i) + b, \quad (9)$$

where α_i and α_i^* are nonzero Lagrangian multipliers, which are the solution to the dual problem, and $K(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x}) \phi(\mathbf{x}_i)$ is the kernel function.

3. Wavelet-domain Kernel Canonical Correlation Analysis

A long-standing problem in statistics and related areas is how to find a suitable representation of multivariate data. Representation means that we somehow transform the data so that its essential structure is made more visible or accessible. Kernel canonical correlation analysis is a method for finding maximal correlated input features for the outputs.

Traditional KCCA operates on time domain is not

very effective for the analysis of financial time series. Market investors are diverse groups. They operate on very different time scales. Wavelet multi-resolution analysis (Percival and Walden, 2000; Gençay et al., 2002) enables the practitioner to focus on particular time scales where trading patterns are considered important. Consequently, wavelet-domain KCCA is better than pure KCCA in analyze financial time series, which could reveal richer characteristic features of each time scale.

First, we decompose time series $y(t)$ by a sequence of projections onto the wavelet basis. The wavelet representation of the signal or time series can be written as

$$y(t) = \sum_k s_{j,k} \phi_{j,k}(t) + \sum_k d_{j,k} \psi_{j,k}(t) + \sum_k d_{j-1,k} \psi_{j-1,k}(t) + \dots + \sum_k d_{1,k} \psi_{1,k}(t),$$

where ϕ is the father wavelet and ψ the mother wavelet. $\phi_{j,k}$ and $\psi_{j,k}$ are scaling and translation of ϕ and ψ , defined as

$$\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j}t - k) = 2^{-j/2} \phi\left(\frac{t - 2^j k}{2^j}\right) \tag{10}$$

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k) = 2^{-j/2} \psi\left(\frac{t - 2^j k}{2^j}\right). \tag{11}$$

In the representation J is the number of multiresolution components, and $s_{j,k}$ are called the smooth coefficients, and $d_{j,k}$ are called the detailed coefficients. They are defined by

$$s_{j,k} = \int y(t) \phi_{j,k}(t) dt \tag{12}$$

$$d_{j,k} = \int y(t) \psi_{j,k}(t) dt. \tag{13}$$

If we define

$$S_j(t) = \sum_k s_{j,k} \phi_{j,k}(t) \tag{14}$$

$$D_j(t) = \sum_k d_{j,k} \psi_{j,k}(t) \quad \text{for } j = 1, 2, \dots, J. \tag{15}$$

$S_j(t)$ and $D_j(t) \Big|_{j=1}^J$ are called the smooth signal and the detail signals, respectively, which constitute a decomposition of a signal into components at different scales.

Second, an KCCA is applied on the decomposed signals of inputs and the output. Notationally, the basic canonical correlation analysis (CCA) algorithm consider that we are given a set of pairs $\{x_i, y_i\}_{i=1}^l$, with $x_i \in R^N, y_i \in R^M$. Let us now introduce matrices $X = [x_1, \dots, x_l]^T$ and $Y = [y_1, \dots, y_l]^T$, where the T superscript denotes matrix or vector transposition. Let us also denote by $X' = XU$ and $Y' = YV$ two matrices, each one containing n_p projections of the original input and output data, U and V being the projection matrices of sizes $N \times n_p$ and $M \times n_p$, respectively. The goal of CCA is to find the directions (U, V) of maximum covariance between the projected input and output data:

$$\max Tr\{U^T C_{XY} V\} \tag{16}$$

$$\text{subject to } U^T C_{XX} U = I, V^T C_{YY} V = I \tag{17}$$

where I is the identity matrix of size n_p , and C_{XY} represents the covariance between the input and output datasets; namely, $C_{XY} = \frac{1}{l} \tilde{X}^T \tilde{Y}$, where \tilde{X}, \tilde{Y} are the centered versions of X and Y . Similarly, $C_{XX} = \frac{1}{l} \tilde{X}^T \tilde{X}$, $C_{YY} = \frac{1}{l} \tilde{Y}^T \tilde{Y}$.

All previous methods assume that there exists a linear relation between the latent variables of X and of Y . However, this might not necessarily hold, and thus non-linear versions have become necessary to solve this problem. Kernel methods are a promising approach to formulate non-linear versions from linear algorithms. Notationally, consider $\phi(x): R^N \rightarrow H$ a function that maps the input data into some Reproducing Kernel Hilbert Space (RKHS), usually referred to as feature space, of very large or even infinite dimension. Let $\Phi = [\phi(x_1), \dots, \phi(x_l)]^T$ and $Y = [y_1, \dots, y_l]^T$, and denote by $\Phi' = \Phi U$ the projection containing

n_p features of the original input data, U being a projection matrix. With $\tilde{\Phi}$ and \tilde{Y} the centered versions of Φ and Y , KCCA can be formulated as

$$\max \text{Tr}\{U^T \tilde{\Phi}^T \tilde{Y} V\} \quad (18)$$

$$\text{subject to: } U^T \tilde{\Phi}^T \tilde{\Phi} U = I, V^T C_{YY} V = I. \quad (19)$$

Making use of the Representer's Theorem, which states that all projection vectors U can be expressed as a linear combination of the training data; namely, $U = \tilde{\Phi}^T A$, where $A = [\alpha_1, \dots, \alpha_{np}]$ and α_i is a column vector containing the coefficients for the i th projection vector. The maximization problem of KCCA can be reformulated as follows:

$$\max \text{Tr}\{A^T \mathbf{K}_X \tilde{Y} V\} \quad (20)$$

$$\text{subject to: } A^T \mathbf{K}_X \mathbf{K}_X A = I, V^T C_{YY} V = I. \quad (21)$$

where $\mathbf{K}_X = \tilde{\Phi} \tilde{\Phi}^T$ is the centered kernel matrices, and $\mathbf{K}_Y = \tilde{Y} \tilde{Y}^T$.

4. Generalized Kernel Regressions

Kernel Partial Least Squares is a nonlinear extension of the Partial Least Squares (PLS) method, commonly used in chemometrics. PLS is a method based on the projection of input and output variables to the latent variables (components), while keeping maximum covariances between input and output latent variables. Because the PLS technique is not widely known, first, a description of linear PLS is provided. This will simplify the next description of its nonlinear kernel-based variant.

Consider a general setting of the linear PLS algorithm to model the relation between two data sets. Given a set of input samples $x_i |_{i=1}^n$ (where each $x_i \in \mathbb{R}^m$) and the corresponding set of outputs $y_i |_{i=1}^n$ (where $y_i \in \mathbb{R}$). PLS models the relations between these two data

sets by means of score vectors. In matrix notation, PLS decomposes the input variable X and the output variable Y into the following form:

$$X = TP^T + F \quad (22)$$

$$Y = UQ^T + G \quad (23)$$

where the T, U are matrices of the extracted score vectors (components, latent vectors). P and Q represent matrices of loadings, while F and G are the matrices of residuals. The aim of PLS method is to find T and U by maximizing their covariance. Then T and U are the best subspaces to represent X and Y .

For kernel PLS, with the use of a kernel, a nonlinear transformation maps the original input space into a feature space F , i.e. $\phi : x_i \in \mathbb{R}^m \rightarrow \phi(x_i) \in F$. By constructing a linear PLS regression model in the kernel-induced feature space F , effectively a nonlinear kernel PLS regression in the original input space is obtained and the mutual orthogonality of the score vectors can be retained.

Let Φ be the $n \times m'$ matrix of input samples in the feature space F , and its i th row be the vector $\varphi(x_i)^T$. Let m' be the dimensionality of $\varphi(x_i)$, which can be infinite. Let t is the score vector (component) which is obtained in the following way: the process starts with random initialization of the Y -score u and then iterates the following steps:

1. randomly initialize u ;
2. $t = \phi \phi^T u, t \leftarrow t / \|t\|$;
3. $c = Y^T t$;
4. $u = Yc, u \leftarrow u / \|u\|$;
5. Repeat steps 1-4 until convergence;
6. Deflate $\phi \phi^T, Y$ matrices:

$$\phi \phi^T \leftarrow (\phi - t(t^T \phi))(\phi - t(t^T \phi))^T, Y \leftarrow Y - t(t^T Y).$$

The kernel PLS regression is an iterative process; i.e. after extraction of one component the algorithm starts again using the deflated matrices $\phi \phi^T$ and Y computed in step 6 to extract next component. Thus we can achieve the sequence of the models up to the point when the rank of $\phi \phi^T$ is reached.

Assume that the process is iterated for F_{ac} times. Denote the sequence of t 's and u 's obtained $t_1, t_2, \dots, t_{F_{ac}}$ and $u_1, u_2, \dots, u_{F_{ac}}$, respectively. Moreover, let $T = [t_1, t_2, \dots, t_{F_{ac}}]$ and $U = [u_1, u_2, \dots, u_{F_{ac}}]$. The "kernel trick" can then be utilized instead of explicitly mapping the input data, and results in: $K = \phi\phi^T$, where K stands for the $n \times n$ kernel matrix: $K(i, j) = k(x_i, x_j)$, where k is the kernel function. K can now be directly used in the deflation instead of ϕ , as

$$K \leftarrow (I_n - t t^T) K (I_n - t t^T). \tag{24}$$

Given a set of test samples $z_i |_{i=1}^m$ (where $z_i \in \mathbb{R}^m$), the predictions made on testing points is

$$\hat{Y}_i = K_i U (T^T K U)^{-1} T^T Y, \tag{25}$$

where K_i is the $n_i \times n$ kernel matrix defined on the test set such that $K_i(i, j) = k(z_i, x_j)$.

5. Empirical Data Analysis

The empirical data used in this research are the option prices on the Taiwan composite stock index (TWSI) traded on the Taiwan Futures Exchange (TWIFEX). The transaction data for call and put options from January 2, 2007 to June 20, 2007 with expiration on June 20, 2007 are used for the study. The data comprise a total of 109 observations. It's well known that TWSI is highly correlated with major Asian stock indices, such as NK225(Japan), KOSPI(South Korea), HSI(Hong Kong), and TSI(Singapore). Moreover, TWSI is also heavily influenced by NASDAQ(U.S.) and S&P500(U.S.). Consequently, we include all these indices together with the daily high, low, open, close prices, trading volumes, and uncovered contract positions for daily trading decisions.

This study selects all types of options to test

forecasting performance, namely, in-the-money, at-the-money and out-of-the-money options. For call options, data near $K = 7500$ approximates the at-the-money options in the sample period; data below $K = 7500$ represents the in-the-money options, while data above $K = 7500$ represents the out-of-the-money options. The call options with strike price $K = 7000, K = 7200, K = 7400, K = 7600, K = 7800, K = 8000$ are analyzed in this study. Put options with similar strike prices are also analyzed for comparison.

5.1 The Trading Signals and Cumulated Profits

First, this study compares three basic models: SVR, KPLSR, and FFNN. Secondly, this study compares KCCA and PCA (principal component analysis, Jolliffe, 2002) for feature extractions. In KCCA and PCA, the eigenvectors corresponding to the first five largest eigenvalues are used. We list the results of cumulative profits (in %) and trading signals for every model. If the prediction on next day exceeds a profit of 5%, the automated trading system will take a long position on this option. Similarly, when the prediction exceeds a loss of 5%, the automated trading system will make a short position on the option. The structure of the feed-forward backpropagation network (FFBPN) is two layers with five neurons. The parameters used in the SVR and KPLSR models are optimized by grid search. For SVM, they are

Table 1. Cumulated trading profits (%) of call options under different models

	K=7000	K=7200	K=7400
SVR	22.0741	28.6068	22.7995
KPLSR	13.6116	28.6165	67.1632
FFNN	-26.2771	-38.0781	-2.0962
	K=7600	K=7800	K=8000
SVR	29.6664	36.5542	85.7189
KPLSR	60.3635	-6.8489	32.6066
FFNN	-8.0082	-2.6998	12.6068

Table 2. Cumulated trading profits (%) of put options under different models

	K=7000	K=7200	K=7400
SVR	117.2673	-41.3225	-229.0310
KPLSR	-130.7604	40.1962	-191.7443
FFBPN	-153.3545	-117.6523	-304.2971
	K=7600	K=7800	K=8000
SVR	70.2156	-79.7672	172.7433
KPLSR	-74.8175	240.7303	99.9757
FFBPN	-182.6939	-125.0797	119.2406

set as follows: $C = 10^4$, $\varepsilon = 0.01$, and $\sigma = 0.1$ for the Gaussian Kernel. The KPLSR also uses Gaussian Kernel with the same parameter. The training of SVR (or KPLSR) is in a dynamic manner, namely, the data window of the training set slides with the current prediction.

Table 1 lists the performances of FFBPN, SVR, and KPLSR models for call options with different strike prices, while Table 2 provides similar results on put options.

For trading profits, Tables 1 and 2 indicate KPLSR and SVR have similar performance, and they are significantly better than FFBPN. For pure models no matter SVR or KPLSR their performances are not stable. For trading signals, the trading frequency of SVR is higher than KPLSR. This will induce very high transaction costs which heavily erode the profits. Moreover, the training of KPLSR is faster than SVR, and KPLSR produces sparser model than SVR. The overall performance of KPLSR is more stable and robust than SVR.

Tables 3, 4 compare various models with or without feature extraction. Figure 1 is the bar chart of Table 3, and Figure 2 is that of Table 4. Tables 3 and 4 indicate that the new model (wavelet-domain KCCA) is better than time-domain PCA. The subspace or features extracted by PCA is not sufficiently to represent the latent structure of the input data set. Consequently, the performance of PCA+SVR is poor and unstable. Sometimes PCA+SVR is even poor than the pure SVR model. Contrarily, due to the superior

Table 3. Cumulated trading profits (%) of call options under the new model

	K=7000	K=7200	K=7400
The new model	101.2550	149.7855	275.4207
PCA+SVR	15.6657	1.5593	7.5426
PCA+KPLSR	18.1733	28.0473	64.5697
SVR	22.0741	28.6068	22.7995
KPLSR	13.6116	28.6165	67.1632
	K=7600	K=7800	K=8000
The new model	322.5194	478.1150	657.9662
PCA+SVR	55.9362	31.9186	89.1505
PCA+KPLSR	40.7616	-6.8489	32.6066
SVR	29.6664	36.5542	85.7189
KPLSR	60.3635	-6.8489	32.6066

Table 4. Cumulated trading profits (%) of put options under the new model

	K=7000	K=7200	K=7400
The new model	593.6436	334.7599	371.9502
PCA+SVR	-273.6423	-252.9529	-225.5476
PCA+KPLSR	-147.3769	40.1962	-40.4817
SVR	117.2673	-41.3225	-229.0310
KPLSR	-130.7604	40.1962	-191.7443
	K=7600	K=7800	K=8000
The new model	209.8987	397.4558	587.3596
PCA+SVR	-51.5558	-44.7000	-169.4751
PCA+KPLSR	27.0277	58.1332	125.3836
SVR	70.2156	-79.7672	172.7433
KPLSR	-74.8175	240.7303	99.9757

capability of wavelet-domain KCCA to infer the independent hidden driving forces of market data, the performance of the new model is substantially improved, and thus the cumulated trading profits are significantly increased. Figures 2-14 plot the trading signals and cumulative profits of the new model under different strike prices for comparison. It can be seen that the new system is stable and robust no matter in call or put options tradings.

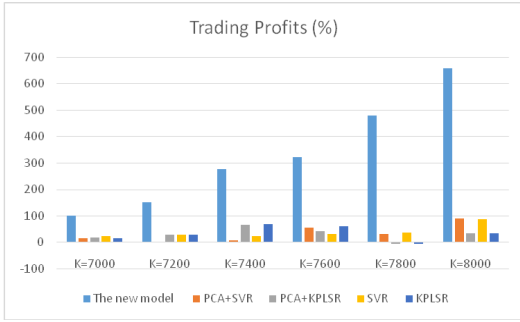


Figure 1. Cumulated trading profits (%) of call options under the new model

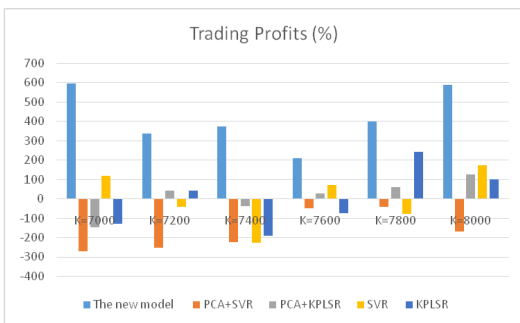


Figure 2. Cumulated trading profits (%) of put options under the new model

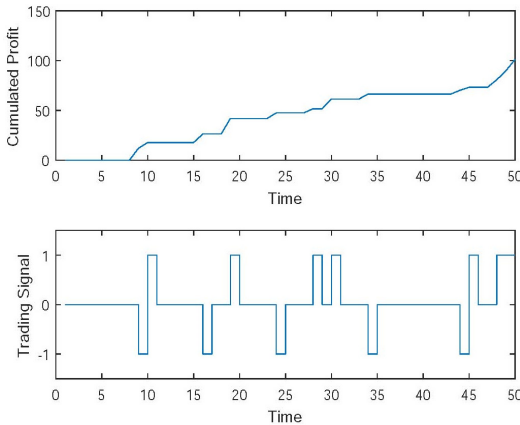


Figure 3. Cumulated profit and trading signal of the new model under call option of K=7000.

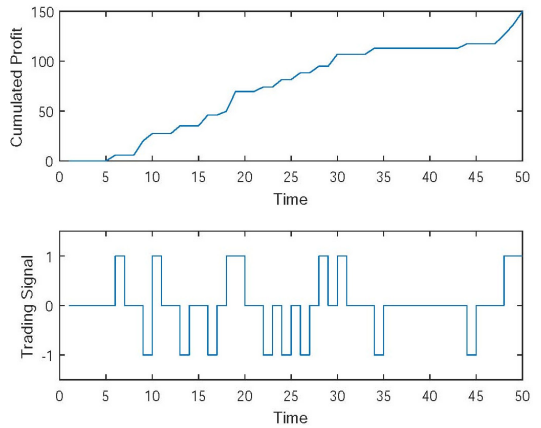


Figure 4. Cumulated profit and trading signal of the new model under call option of K=7200.

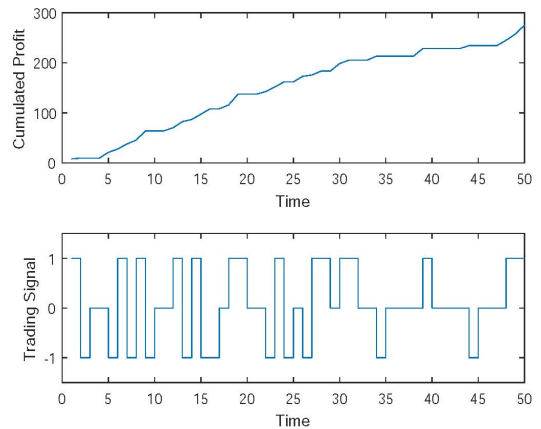


Figure 5. Cumulated profit and trading signal of the new model under call option of K=7400.

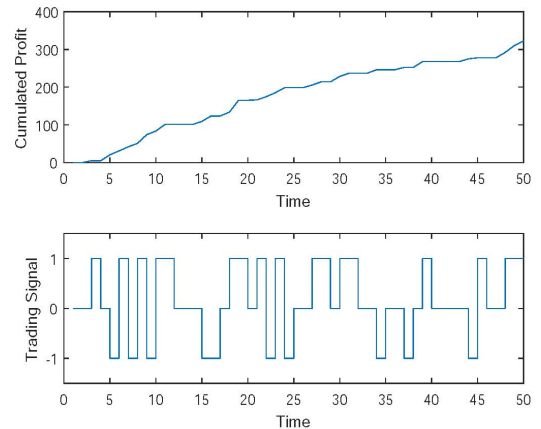


Figure 6. Cumulated profit and trading signal of the new model under call option of K=7600.

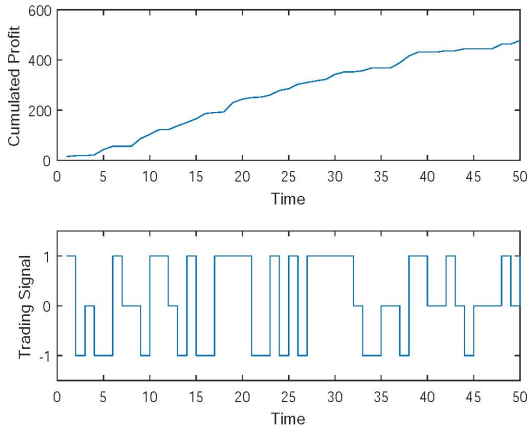


Figure 7. Cumulated profit and trading signal of the new model under call option of $K=7800$.

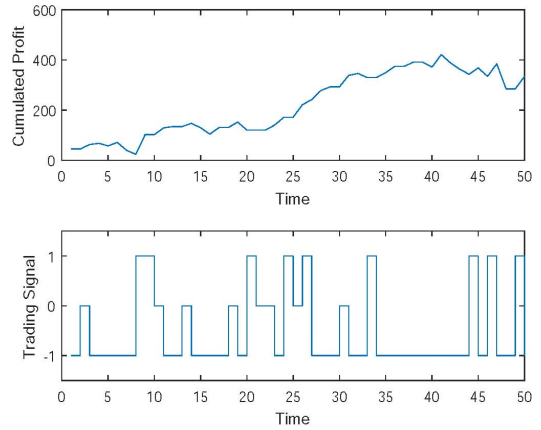


Figure 10. Cumulated profit and trading signal of the new model under put option of $K=7200$.

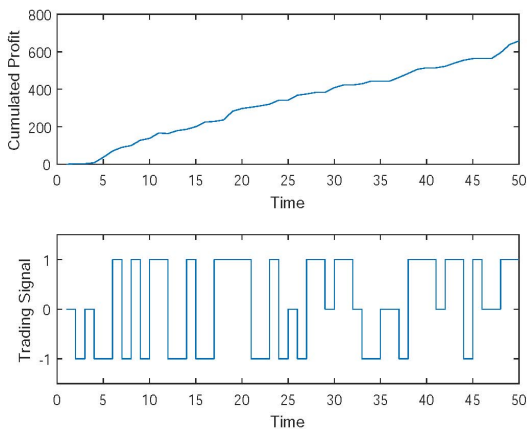


Figure 8. Cumulated profit and trading signal of the new model under call option of $K=8000$.

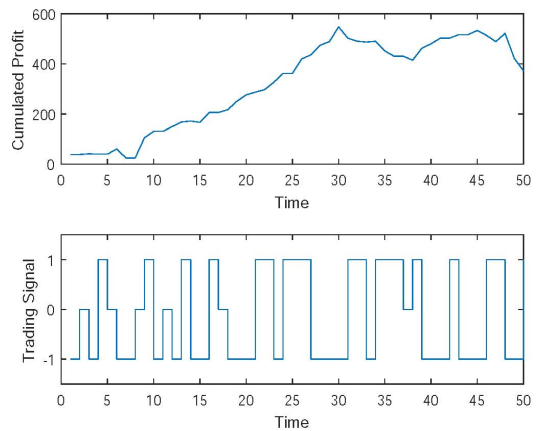


Figure 11. Cumulated profit and trading signal of the new model under put option of $K=7400$.

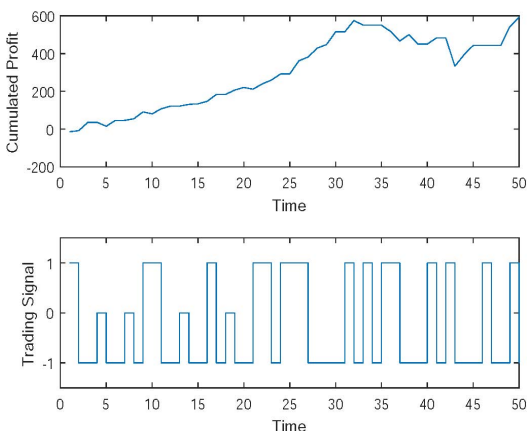


Figure 9. Cumulated profit and trading signal of the new model under put option of $K=7000$.

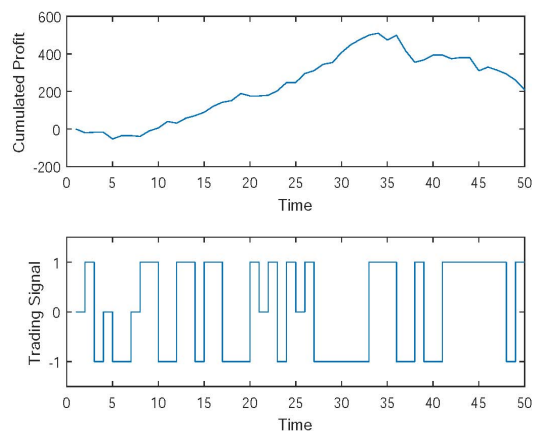


Figure 12. Cumulated profit and trading signal of the new model under put option of $K=7600$.

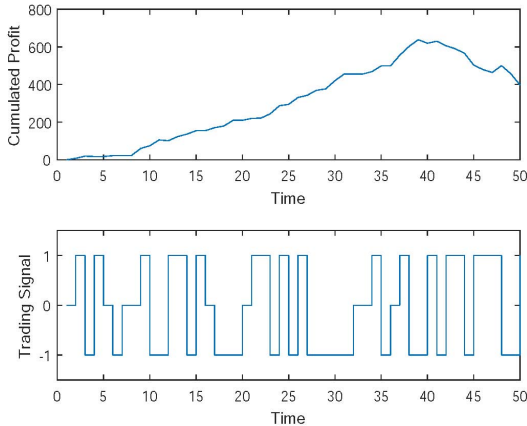


Figure 13. Cumulated profit and trading signal of the new model under put option of $K=7800$.

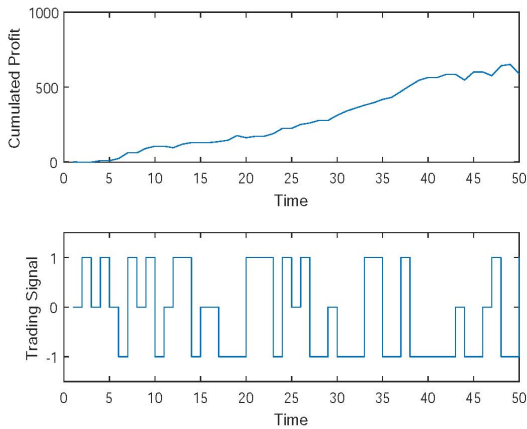


Figure 14. Cumulated profit and trading signal of the new model under put option of $K=8000$.

In sum, by score components KPLSR is superior to SVR in modeling the input and output relationship in our high dimensional data set. Integrating wavelet-domain KCCA with KPLSR is even better in the financial input/output modeling, because wavelet-domain KCCA could transform original input space to a better subspace which uncovers the most important hidden driving forces of the market, and thus help KPLSR to generate more accurate buy/sell signals.

Moreover, the non-stationarity of financial time series is successfully solved by our sliding training algorithm which driven by data adaptively adjusts the SVR and KPLSR models. The traditional pure SVR or KPLSR models operate on original data space

can not capture characteristic financial features for trading, and thus their performances are poor.

6. Conclusions

What drives the movements of a financial time series? This surely is a question of interest to many people, ranging from researchers who wish to understand financial markets, to traders who will benefit from such knowledge. Can modern big data analysis or data mining techniques help discover some of the underlying forces? In this paper, we focus on a new technique-wavelet-domain kernel canonical correlation analysis to uncover the hidden driving forces. Wavelet-domain KCCA enables the investors to focus on particular time scales where trading patterns are considered important and highly correlated with future movements of the market.

By integrating wavelet-domain KCCA with dynamic kernel based regressions, this paper develops a novel automated trading decision system. Conducting a series of experiments using option data on the Taiwan Weighted Stock Index (TAIEX), the new system outperformed other feature extraction methods and pure regressors. The cumulated trading profits are significantly increased. The resultant intelligent investment trading support system can help investors, fund managers and investment decision-makers make good profits.

The highly effective decision support system can also be applied to other problems involving financial trading. Results of this study can also be used to perform a good hedge on global investments.

References

Armano, G., Marchesi, M., Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences*, 170(1), 3-33.

- Boginski, V., Butenkob, S., & Pardalos, P. M. (2006). Mining market data: a network approach. *Computers and Operations Research*, 33, 3171-3184.
- Bose, I., & Mahapatra, R. K. (2001). Business data mining: a machine learning perspective. *Information and Management*, 39(3), 211-225.
- Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems with Applications*, 42(14), 5963-5975.
- Chen, S. (2006). Local regularization assisted orthogonal least squares regression. *Neurocomputing*, 69, 559-585.
- Chen, A. S., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: forecasting and trading the Taiwan stock index. *Computers and Operations Research*, 30, 901-923.
- Chun, S. H., & Kim, S. H. (2004). Data mining for financial prediction and trading: application to single and multiple markets. *Expert Systems with Applications*, 26, 131-139.
- Chun, S. H., & Park, Y. J. (2005). Dynamic adaptive ensemble case-based reasoning: application to stock market prediction. *Expert Systems with Applications*, 28, 435-443.
- Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines, Cambridge University Press.
- Drezet, P. M. L., & Harrison, R. F. (1998). Support vector machines for system identification, in: Proceedings of the UKACC International Conference on Control'98, Swansea, UK, September 1-4, 688-692.
- Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29, 927-940.
- Fang, Y., & Xu, D. (2003). The predictability of asset returns: an approach combining technical analysis and time series forecasts. *International Journal of Forecasting*, 19, 369-385.
- Fyfe, C., & Lai, P. L. (2001). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10, 365-374.
- Gençay, R., Selçuk, F., & Whitcher, B. (2002). *An introduction to wavelets and other filtering methods in finance and economics*. Academic Press, London.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd edition ed.). Upper Saddle River, NJ: Prentice Hall.
- Hu, Y., Feng, B., Zhang, X., Ngai, E., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications*, 42, 212-222.
- Irma, B. F., Stelios, H. Z., & Steven, W. (2002). Knowledge discovery techniques for predicting county investment risk. *Computers and Industrial Engineering*, 43, 787-800.
- Jolliffe, I.T. (2002). *Principal Component Analysis, Series: Springer Series in Statistics*, 2nd ed., Springer, NY.
- Oh, K. J., & Kim, T. Y. (2007). Financial market monitoring by casebased reasoning. *Expert Systems with Applications*, 32(3), 789-800.
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42, 259-268.
- Percival, D., & Walden, A. (2000). *Wavelet Methods for Time Series Analysis*, Cambridge University Press, Cambridge.
- Rosipal, R., & Kramer, N. (2006). Overview and Recent Advances in Partial Least Squares. In *Subspace, Latent Structure and Feature Selection Techniques*, Saunders C., Grobelnik M., Gunn S., Shawe-Taylor J. (eds.), Springer, 34-51.
- Rosipal, R., & Trejo, L. J. (2001). Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. *Journal of Machine Learning Research*, 2, 97-123.
- Ross, S. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3), 341-360.
- Schölkopf, B., Burges, C. J. C., & Smola, A. J. (1999). *Advances in kernel methods - support vector learning*, MIT Press, Cambridge, MA.
- Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, Massachusetts: The MIT Press.
- Thawornwong, S., & Enke, D. (2004). The adaptive selection of financial and economic variables for use with artificial neural networks. *Neurocomputing*, 56, 205-232.
- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1, 211-244.
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.
- Wang, Y. F. (2003). Mining stock price using fuzzy rough set system. *Expert Systems with Applications*, 24, 13-23.
- Wang, J. L., & Chan, S. H. (2006). Stock market trading rule discovery using two-layer bias decision tree. *Expert Systems with Applications*, 30, 605-611.
- Wang, H., & Weigend, A. S. (2004). Data mining for financial decision making. *Decision Support Systems*, 37, 457-460.
- Wold, S., Ruhe, H., Wold, H., & Dunn, W. J. (1984). The collinearity problem in linear regression. The PLS approach to generalized inverse. *SIAM Journal of Scientific and Statistical Computations*, 5, 735-743.
- Wu, M. C., Lin, S. Y., & Lin, C. H. (2005). An effective application of decision tree to stock trading. *Expert Systems with Applications*, 31(2), 1-5.
- Yoo, P. D., Kim, M. H., & Jan, T. (2007). Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, 2, 835-841.
- Żbikowski, K. (2015). Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Systems with Applications*, 42, 1797-1805.

Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159-175.