Huseyin INCE (Corresponding author)
Theodore B. TRAFALİS
Gebze Technical University, Faculty of Business Administration
Gebze/Kocaeli, TURKEY
E-mail: h.ince@gtu.edu.tr

A HYBRID FORECASTING MODEL FOR STOCK MARKET PREDICTION

Abstract. *Stock market predictions have been studied by academics and practitioners. In this paper, a hybrid model is proposed to predict the stock market movement. We have combined the independent component analysis (ICA) and kernel methods. ICA is used to select the important indicators. After determining the inputs, kernel methods are employed to predict the direction of the stock market. We have used the Dow-Jones, Nasdaq and S&P500 indices for experiments. Technical indicators of the indices are used as input variables for the proposed model. According to the analysis of the experimental results, kernel methods are capable of producing satisfactory forecasting accuracies and gain rates for Dow-Jones, Nasdaq and S&P 500 indices. The trading experiment shows that the kernel methods obtain higher rate of returns than the other investment strategies.*

Keywords: *Hybrid Model, Kernel Methods, Stock Market Forecasting, Support Vector Machines, Minimax Probability Machines.*

JEL Classification: C45, E37

1. Introduction

Prediction of stock price movement has been regarded as one of the most challenging problems since the stock market is a complex, dynamic, non-stationary, and chaotic system in nature. Stock price movements are not random and highly non-linear . Therefore, several artificial intelligence and time series models have been proposed to predict the stock prices and returns (Ince and Trafalis 2006; Leigh, Purvis and Ragusa 2002). Developing an investment strategy based on fundamental indicators earns significant abnormal returns (Piotroski 2000). In addition to this, technical indicators, also known as charting, have been used to explore the dynamics of stock price movement by analyzing the past sequence of stock prices. There are recurring patterns in the market behavior, which can be identified and predicted. Technical analysts have used number of statistical parameters called technical indicators and charting patterns from historical data .

Recently, researchers have proposed artificial intelligence (AI) techniques for stock price prediction. Most of the studies have focused on stock market index and individual stock prediction (Atsalakis and Valavanis 2009; Ince and Trafalis 2006; Tay and Cao 2002). More recent studies have presented encouraging results on stock selection using kernel methods, support vector machines, neural networks, fuzzy logic, swarm intelligence and hybrid techniques (Chavarnakul and Enke 2008; Ince and Trafalis 2006; Teoh, Chen, Cheng and Chu 2009). Noisy datasets may torture the prediction accuracy of the learning algorithms. This will decrease the generalization capability of the methods. In order to increase the generalization capability, feature selection (extraction) techniques have been used before training the classification/ forecasting training models. There are many well-known feature selection techniques, which are decision trees, data envelopment analysis (DEA), principal component analysis (PCA), fuzzy ranking analysis, independent component analysis (ICA) among others.

Although some artificial intelligence techniques, such as independent component analysis, self-organizing map, genetic algorithms, and decision trees can be applied for selecting the representative features (Kao, Chiu, Lu and Yang 2013; Wang, Wang, Zhang and Guo 2012), they are not widely considered in the business domain, especially for predicting the direction of the stock market. To increase the generalization capability, we propose an integrated approach by using ICA and kernel methods to predict the direction of stock market indices as well as individual stock prices. The ICA method is applied for selecting appropriate features and further improves the performance of the kernel methods. First, ICA is used to estimate the independent components and mixing matrix from the stock market data. The ICs are used to construct the forecasting variables. Then, kernel methods are applied the reconstructed forecasting variables to build the classification model. Nasdaq, S&P 500 and Dow-Jones indices are used to evaluate the performance of the proposed approach.

The paper is organized as follows; in section 2 the basics of ICA is presented. Then in section 3 kernel methods are explained. Also a hybrid model is proposed for predicting the direction of stock market index. Experimental results are presented in section 4. Finally section 5 concludes the paper.

2. Independent Component Analysis

ICA is a feature extraction technique for extracting independent sources from observed data. The ICA aims to find independent sources from their mixtures that are mixtures of unknown sources without knowing any specific knowledge of mixing mechanism (Lu, Lee and Chiu 2009). The nonlinear ICA assumes that the observed data are nonlinear combination of ICs. In many real applications, the data is a nonlinear mixture of latent signals. Thus, the nonlinear ICA technique is more practical. Several researchers proposed NLICA to solve problems in the machine learning literature (Kao, Chiu, Lu and Yang 2013; Wang, Wang, Zhang and Guo 2012). .

The ICA technique is defined in (Hyvarinen 1999), and assumes that *m* observed variables, **X** = [x_1, x_2, …, x_m] are the linear mixture of n statistically independent components, S = [s_1, s_2, …, s_n]

$$\mathbf{X} = \mathbf{AS} \tag{1}$$

where **A** is the mixture matrix having a full rank and $m \geq n$. The vector **s** represents independent components. The basic ICA aims to estimate the **S** and mixture matrix **A** from **X**. The ICA solution is obtained by finding the de-mixing matrix **W** such that

$$\mathbf{Y} = \mathbf{WX} \tag{2}$$

where **Y** =[y_1, y_2, .., y_n]^T called independent component vector, is the estimation of **S**, and **W**, de-mixing matrix, is an estimation of $\mathbf{A}^{-1}$. If the observed data are nonlinear combination of the latent sources, then this can be formulated as nonlinear ICA model as follows :

$$\mathbf{x} = \mathbf{F(s)} \tag{3}$$

where **x** and **s** are the data and source vector and F is an unknown nonlinear transformation function. The nonlinear ICA tries to find a map G: $\Re^n \to \Re^n$ under the assumption that the number of independent components equals to the number of mixtures. Then, the nonlinear ICA finds a mapping that yields components which are statistically independent.

$$\mathbf{y} = \mathbf{G(s)} \tag{4}$$

In this study, Fast ICA (Hyvarinen and Oja 1997) algorithm is adopted to solve the independent components. It is a computationally efficient and robust fixed-point algorithm for independent component analysis.

3. Kernel Methods

Kernel methods transform or map an input ***x*** from the input space *X* into a higher dimensional feature space $\mathscr{F}$ through a map ϕ: ***x*** → ϕ(***x***) so that the nonlinear problems can be solved linearly in the feature space $\mathscr{F}$ (Vapnik 2000). Kernel methods have become a popular tool for solving classification and prediction problems. They exhibit good generalization performance on many real-life datasets. Next, we explain the support vector machines (SVM), twin support vector machines (TWSVM), minimax probability machines (MPM), and kernel fisher discriminant analysis(KFDA).

3.1. Support Vector Machines

Support vector machines are a novel approach for pattern classification. SVMs, based on statistical learning theory, are proposed by (Vapnik 2000) to solve classification problems. In a two-class classification problem, SVMs try to find a linear optimal hyperplane so that the margin of separation between two classes is maximized. In a non-linear separable, one may transform the input space via a non-linear mapping into a higher dimension feature space so that linear hyperplane can be found in this space (Ince and Trafalis 2006). Kernel functions are used to transform the data set from input space to feature space. Since the training of a SVM is done by solving a linearly constrained quadratic problem, the solution is unique, optimal and global (Burges 1998; Vapnik 2000).

Given a training set $D = \{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\} \in \Re^n \times \{\pm 1\}, \text{and } y_i = \{+1, -1\}$ represent the positive and negative classes respectively. Classification is to find an optimal separating hyperplane (decision function) *f(**x**)* to determine *y* according to ***x***. That mean we find a rule to separate the point in $\Re^n$ into two parts. When the training set is not linear separable, the slack variables $\xi_i \geq 0$ are introduced to *i^th^* training example ($\mathbf{x_i}$,y_i) and the corresponding constraint are relaxed to $y_i(\mathbf{w^T x_i} + b) + \xi_i \geq 1$. The objective is to maximize the margin and minimize the classification error $C\sum_{i=1}^{l} \xi_i$. This can be formulated as a QP problem as follows:

$$\min_{\mathrm{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \tag{5}$$
$$\text{s.t. } y_i(\mathbf{w^T x_i} + b) + \xi_i \geq 1, \quad i = 1,2,...,l.$$

where C > 0 is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the training error. For nonlinear classification case, SVM maps the training samples into a high dimensional feature space via kernel function. Once we choose the kernel function $K(x_i, x_j)$, the hyperplane is determined by the dual of the problem (5). The dual problem is to maximize the objective function

$$\begin{aligned}&\max_{\alpha}\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)\\ &\text{s.t.}\sum_{i=1}^{l}\alpha_i y_i=0\quad i=1,2,...,l\\ &\quad 0\le\alpha_i\le C\quad i=1,2,...,l.\end{aligned}\tag{6}$$

Let $\boldsymbol{\alpha}^*=(\alpha_1^*,\alpha_2^*,...,\alpha_l^*)$ and $b^*=y_i-\sum_{j=1}^{l}y_i\alpha_i^*K(x_i,x_j)$ the optimal solution of (6), then the decision function is given by

$$f(\mathbf{x})=\text{sign}\left(\sum_{i=1}^{l}\alpha_i^* y_i K(\mathbf{x},\mathbf{x}_i)+b^*\right)\tag{7}$$

In order to obtain good results, it is important to set the free parameters such as regularization parameter (C), kernel function and its parameter. Several studies have been conducted to determine the optimal kernel and regularization parameter (For example see (Chapelle, Vapnik, Bousquet and Mukherjee 2002)).

3.2. Twin Support Vector Machines

Jayadeva and Chandra(Jayadeva and Chandra 2007), proposed twin support vector machines (TWSVM) that is a binary classifier that does classification using two non-parallel hyperplanes instead of a single hyperplane as in the case of pure SVM approach. The two nonparallel hyperplanes are obtained by solving two quadratic programming problems (QPPs) of smaller size compared to a single large QPP solved by conventional SVMs. The idea is to solve two QPPs with objective function corresponding to one class and constraints corresponding to the other class . Consider a two-class classification problem of classifying m_1 data points belonging to class +1 and m_2 data points belonging to class -1 in the *n* dimensional real space $\Re^n$. Let matrix A in $\Re^{m1xn}$ represents the data points in class +1 and matrix B in $\Re^{m2xn}$ represent the data points in class -1. Given the above stated binary classification problem, linear TSVM seeks two non-parallel hyperplanes in $\Re^n$ by solving the following two pair QPPs (Jayadeva and Chandra 2007):

$$\begin{aligned}&\min_{\mathbf{w}^{(1)},b^{(1)},\mathbf{q}}\frac{1}{2}(\mathbf{Aw}^{(1)}+\mathbf{e_1}b^{(1)})^T(\mathbf{Aw}^{(1)}+\mathbf{e_1}b^{(1)})+c_1\mathbf{e}_2^{\mathbf{T}}\mathbf{q}\\ &s.t.\quad \textbf{-}(\mathbf{Bw}^{(1)}+\mathbf{e}_2b^{(1)})+\mathbf{q}\ge\mathbf{e}_2,\ \ \mathbf{q}\ge\mathbf{0},\text{ and}\end{aligned}\tag{8}$$

$$\begin{aligned}&\min_{\mathbf{w}^{(2)},b^{(2)},\mathbf{q}} \frac{1}{2}(\mathbf{Bw}^{(2)}+\mathbf{e}_2 b^{(2)})^T(\mathbf{Bw}^{(2)}+\mathbf{e}_2 b^{(2)})^T + c_2\mathbf{e}_1^{\mathbf{T}}\mathbf{q}\\ &s.t.\quad (\mathbf{Aw}^{(2)}+\mathbf{e}_1 b^{(2)})+\mathbf{q}\geq \mathbf{e}_1,\ \ \mathbf{q}\geq \mathbf{0},\end{aligned} \tag{9}$$

where c1, c2 are regularization parameters, $\mathbf{e_1} \in \Re^{m_1}$ and $\mathbf{e}_2 \in \Re^{m_2}$ are vectors of ones.

Using the Lagrangian for (8) and (9), and the Karush-Kuhn-Tucker (K.K.T) conditions, we obtain the Wolfe dual for (8) and (9) as follows:

$$\begin{aligned}&\max_{\alpha} \mathbf{e}_2^{\mathbf{T}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}(\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}\mathbf{G}^{\mathbf{T}}\boldsymbol{\alpha}\\ &\mathrm{s.}t.\quad \mathbf{0}\leq\boldsymbol{\alpha}\leq\mathbf{c_1}\mathbf{e_1},\end{aligned} \tag{10}$$

where ***H*=[*A e*₁]** and ***G*** = [***B e*₂**].

$$\begin{aligned}&\max_{\gamma} \mathbf{e}_1^{\mathbf{T}}\boldsymbol{\gamma} - \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\alpha}^{\mathbf{T}}\mathbf{P}(\mathbf{Q}^{\mathbf{T}}\mathbf{Q})^{-1}\mathbf{P}^{\mathbf{T}}\boldsymbol{\gamma}\\ &\mathrm{s.}t.\quad \mathbf{0}\leq\boldsymbol{\gamma}\leq\mathbf{c}_2\mathbf{e}_2,\end{aligned} \tag{11}$$

where ***P*=[*A e*₁]** and ***Q*** = [***B e*₂**].

According to (Jayadeva and Chandra 2007) , two hyperplanes are obtained for two classes, points are classified according to which hyperplane a given point is closest to. The non-parallel hyperplanes can be obtained from the solution of QPPS given in (10) and (11) as follows:

$$\begin{aligned}&\mathbf{u} = -(\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}\mathbf{G}^{\mathbf{T}}\boldsymbol{\alpha}, \qquad \text{where}\ \ \mathbf{u}=[\mathbf{w}^{(1)}\ b^{(1)}]^T\\ &\mathbf{v} = (\mathbf{Q}^{\mathbf{T}}\mathbf{Q})^{-1}\mathbf{P}^{\mathbf{T}}\boldsymbol{\gamma}, \qquad \text{where}\ \ \mathbf{v}=[\mathbf{w}^{(2)}\ b^{(2)}]^T\end{aligned} \tag{12}$$

After computing u and v as shown in (12), separating hyperplanes

$$\mathbf{x}^T\mathbf{w}^{(1)}+b^{(1)}=0 \quad \text{and} \quad \mathbf{x}^T\mathbf{w}^{(2)}+b^{(2)}=0 \tag{13}$$

are obtained. A new data is assigned to class r (r =1,2), depending on which of the two planes given by (13) it lies to closest to,

$$\mathbf{x}^T\mathbf{w}^{(r)}+b^{(r)} = \min_{l=1,2}|\,\mathbf{x}^T\mathbf{w}^{(l)}+b^{(l)}\,|, \tag{14}$$

where $|\cdot|$ is the perpendicular distance of point x from the plane $\mathbf{x}^T\mathbf{w}^{(l)}+b^{(l)}=0, l=1{,}2.$

TWSVM can be extended to nonlinear classifier cases by considering the following two kernel-generated surfcases

$$\mathbf{K}(\mathbf{x}^{\mathbf{T}},\mathbf{C}^{\mathbf{T}})\mathbf{u}^{(1)} + b^{(1)} = 0, \text{ and} \\ \mathbf{K}(\mathbf{x}^{\mathbf{T}},\mathbf{C}^{\mathbf{T}})\mathbf{u}^{(2)} + b^{(2)} = 0, \qquad (15)$$

where $\mathbf{C}^T$ =[**A B**]^T and ***K*** is the kernel function. According to (Jayadeva and Chandra 2007), linear classifier is a special case of (15) by using linear kernel function. Optimization problem is constructed for the hypersurfaces given in (15) as follows:

$$\min_{\mathbf{u}^{(1)},b^{(1)},\mathbf{q}} \frac{1}{2}\left\|\mathbf{K}(\mathbf{A},\mathbf{C}^{\mathbf{T}})\mathbf{u}^{(1)} + \mathbf{e_1}b^{(1)}\right\|^2 + c_1\mathbf{e_2^T q} \\ s.t. \quad -(\mathbf{K}(\mathbf{B},\mathbf{C}^{\mathbf{T}})\mathbf{u}^{(1)} + \mathbf{e}_2 b^{(1)}) + \mathbf{q} \geq \mathbf{e_2}, \quad \mathbf{q} \geq \mathbf{0}, \text{ and} \qquad (16)$$

$$\min_{\mathbf{u}^{(2)},b^{21)},\mathbf{q}} \frac{1}{2}\left\|\mathbf{K}(\mathbf{B},\mathbf{C}^{\mathbf{T}})\mathbf{u}^{(2)} + \mathbf{e}_2 b^{(2)}\right\|^2 + c_2\mathbf{e}_1^{\mathbf{T}}\mathbf{q} \\ s.t. \quad (\mathbf{K}(\mathbf{A},\mathbf{C}^{\mathbf{T}})\mathbf{u}^{(2)} + \mathbf{e}_1 b^{(2)}) + \mathbf{q} \geq \mathbf{e}_1, \quad \mathbf{q} \geq \mathbf{0}, \qquad (17)$$

where c1, c2>0 are parameters.

Using the Lagrangian for (16) and (17), and the Karush-Kuhn-Tucker (K.K.T) conditions, the Wolfe dual for (16) and (17) are obtained as follows:

$$\max_{\alpha} \mathbf{e_2^T}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{R}(\mathbf{S^TS})^{-1}\mathbf{R^T}\boldsymbol{\alpha} \\ \text{s.}t. \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{c_1 e_1}, \qquad (18)$$

where ***S*=[*K(A,C*^{T}*) e*$_1$]** and ***R*** = [***K(B,C***^{T}***) e***$_2$].

$$\max_{\gamma} \mathbf{e_1^T}\boldsymbol{\gamma} - \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\alpha}^{\mathbf{T}}L(N^{\mathbf{T}}N)^{-1}L^{\mathbf{T}}\boldsymbol{\gamma} \\ \text{s.}t. \quad \mathbf{0} \leq \boldsymbol{\gamma} \leq \mathbf{c}_2\mathbf{e}_2, \qquad (19)$$

where ***L*** = [***K(A,C***^{T}***) e***$_1$] and ***N*** = [***K(B,C***^{T}***) e***$_2$]. The augmented vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ can be obtained as follows.

$$\mathbf{z}_1 = -(\mathbf{S^TS})^{-1}\mathbf{R^T}\boldsymbol{\alpha}, \qquad \text{where} \quad \mathbf{z}_1 = [\mathbf{u}^{(1)}\ b^{(1)}]^T \\ \mathbf{z}_2 = (\mathbf{N^TN})^{-1}\mathbf{L^T}\boldsymbol{\gamma}, \qquad \text{where} \quad \mathbf{z}_2 = [\mathbf{u}^{(2)}\ b^{(2)}]^T \qquad (20)$$

Once problem (18) and (19) are solved to obtain the surfaces (15), a new data point $\mathbf{x} \in \Re^n$ is assigned to class 1 or class -1 in a similar manner to the linear case.

3.3. Kernel Fisher Discriminant Analysis

Kernel Fisher discriminant analysis (Mika, Ratsch and Muller 2001), implements the well-known Fisher linear discriminant in a feature space induced by kernel functions, has been applied to many pattern recognition problems and demonstrates an impressive level of performance on a range of benchmark data sets (Cawley and Talbot 2003; Saadi, Talbot and Cawley 2007). KFDA's basic idea can be described that through some nonlinear mapping the input space can be mapped implicitly into a high-dimensional kernel feature space where nonlinear pattern now appears linear. Note that in KFDA any explicit mapping is not necessary, because kernel trick is introduced (Mika, Ratsch and Muller 2001).

Suppose we are given training set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\} \in \Re^n \times \{\pm 1\}$, and $y_i = \{+1, -1\}$ represent the positive and negative classes respectively. Let $\boldsymbol{D_1} = \{\boldsymbol{x_1^1}, \boldsymbol{x_2^1}, \dots, \boldsymbol{x_{l_1}^1}\}$ and $\boldsymbol{D_2} = \{\boldsymbol{x_1^2}, \boldsymbol{x_2^2}, \dots, \boldsymbol{x_{l_2}^2}\}$, ($l = l_1 + l_2$), be samples from two classes with class label +1 and -1. Then, linear discriminant analysis attempts to find a linear combination of input variables, $\boldsymbol{w \cdot x}$, that maximizes the average separation of the projections of points belonging to negative and positive classes, while minimizing the within class variance of the projections of those points. Fisher discriminant finds the vector ***w*** by maximizing

$$\boldsymbol{J(w)} = \frac{\boldsymbol{w^T S_B w}}{\boldsymbol{w^T S_W w}} \tag{21}$$

where $\boldsymbol{S_B}$ is the between class scatter matrix and $\boldsymbol{S_W}$ the within class scatter matrix. The matrices $\boldsymbol{S_B}$ and $\boldsymbol{S_W}$ are given by

$$\begin{aligned} \boldsymbol{S_B} &= \boldsymbol{(m_1 - m_2)(m_1 - m_2)^T} \\ \boldsymbol{S_W} &= \sum_{i \in \{1,2\}} \sum_{j=1}^{l_i} \boldsymbol{(x_j^i - m_i)(x_j^i - m_i)^T} \end{aligned} \tag{22}$$

where $\boldsymbol{m_i} = \frac{1}{\boldsymbol{l_i}} \sum_{j=1}^{l_i} \boldsymbol{x_j^i}$.

When the data is not linearly separable in input space, it can be mapped into feature space *F* by using the kernel functions. To find the Fisher's linear discriminant in the feature space *F*, equation (21) has to be formulated in terms of only dot products of the training data, $\phi(\boldsymbol{x}_i)\cdot\phi(\boldsymbol{x}_j)$, induced by a positive definite kernel $K = \boldsymbol{X}\times\boldsymbol{X}\rightarrow\Re$ defining inner product $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)\cdot\phi(\boldsymbol{x}_j)$ (Cawley and Talbot 2003; Mika, Ratsch and Muller 2001; Saadi, Talbot and Cawley 2007). The kernel matrices for entire data set, ***K***, and for each class, ***K***$_1$ and ***K***$_2$ is defined as follows:

$$\boldsymbol{K} = [k_{ij} = K(\boldsymbol{x}_i,\boldsymbol{x}_j)]_{i,j=1}^{l},$$
$$\boldsymbol{K_i} = [k_{jk}^{i} = K(\boldsymbol{x}_j,\boldsymbol{x}_k^{i})]_{j,k=1}^{j=l,k=l_j}. \qquad (23)$$

According to the theory of reproducing kernels, any solutions **w**∈F must line in the span of all training sample in F. So, w can be formulated by

$$\boldsymbol{w} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{l}} \boldsymbol{x}_i\phi(\boldsymbol{x}_i). \qquad (24)$$

To find Fisher's linear discriminant in *F* we need to maximize

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha^T M\alpha}}{\boldsymbol{\alpha^T N\alpha}} \qquad (25)$$

where ***α*** =(*α*$_1$, *α*$_2$,…, *α*$_l$), , $\boldsymbol{M} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$, $\boldsymbol{m}_i = \boldsymbol{K}_i\boldsymbol{u}_i, \boldsymbol{u}_i$ is the column vector containing l_i elements with a common value of l_i^{-1}, and $\boldsymbol{N} = \sum_{i\in\{1,2\}} \boldsymbol{K}_i(\boldsymbol{I}-\boldsymbol{U}_i)\boldsymbol{K}_i^T$, where I is the identity matrix and ***U***$_i$ is a matrix with all elements equal to l_i^{-1}. The leading eigenvector of ***N***$^{-1}$***M*** gives the coefficients, ***α,*** of equation (25). The problem with this setting is that N is likely to be singular, or ill-conditioned. In order to avoid this, a regularized solution is obtained by substituting ***N***$_\mu$= ***N*** + *μ**I***, where *μ* is a regularization constant. The kernel Fisher's discriminant classifier can be written as

$$f(\boldsymbol{x}) = \boldsymbol{w}\phi(\boldsymbol{x}) + b,$$
$$(\boldsymbol{w}\phi(\boldsymbol{x})) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{l}} \alpha_i \mathrm{k}(\boldsymbol{x_i},\boldsymbol{x}) \qquad (26)$$
$$b = -\boldsymbol{\alpha}\frac{l_1\boldsymbol{M}_1 + l_2\boldsymbol{M}_2}{l}$$

In addition to this approach, KFD classifier can be determined by solving the following systems of linear equations (Mika, Ratsch and Muller 2001):

$$\begin{bmatrix} \boldsymbol{KK}+\mu\boldsymbol{I} & \boldsymbol{K1} \\ (\boldsymbol{K1})^T & l \end{bmatrix}\begin{bmatrix} \boldsymbol{a} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{K} \\ \boldsymbol{1} \end{bmatrix}\boldsymbol{y}, \tag{27}$$

This formulation shows the similarities between the Fisher's discriminant analysis and least squares support vector machines.

3.4. Minimax Probability Machines

The MPM is a recently proposed binary classifier that tries to minimize the probability of misclassification for two-class classification problem. The MPM model minimizes the worst case probability of misclassification of future data points under all possible choices of class densities (Lanckriet, Ghaoui, Bhattacharyya and Jordan 2003).

In order to formulate the MPM, training data, **x** and **y,** are assumed to be generated from two classes distributions with means and covariance matrices given by $\boldsymbol{x} \sim \{\bar{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}}\}$ and $\boldsymbol{y} \sim \{\bar{\boldsymbol{y}}, \boldsymbol{\Sigma}_y\}$. Note that **x** and **y** also denotes the two classes. The objective is to determine the hyperplane $H(\mathbf{w},b) = \{\mathbf{z} \mid \mathbf{w}^T\mathbf{z} = b\}$, where $\mathbf{w} \in \Re^n \setminus \{0\}$ and $b \in \Re$, which separates the two classes with maximum probability. The generalization error is minimized by finding the hyperplane for which the worst case probabilities $\Pr(\mathbf{w}^T\mathbf{x} \le b)$ and $\Pr(\mathbf{w}^T\mathbf{y} \ge b)$ are minimized. This can be solved with the following optimization problem

$$\begin{aligned} &\max_{\alpha, w\neq 0, b} \ \alpha \\ &s.t. \\ &\inf_{x\sim(\bar{x},\Sigma_x)} \Pr(\boldsymbol{w^T x} \ge b) \ge \alpha, \\ &\inf_{y\sim(\bar{y},\Sigma_y)} \Pr(\boldsymbol{w^T y} \le b) \ge \alpha \end{aligned} \tag{28}$$

The quantity (1−α) is an upper bound on the generalization error, which is called the worst-case misclassification probability. The optimization problem (28) can be transformed to a Second Order Cone Programming (SOCP) problem.

According to (Ng, Zhong and Yang 2007), the optimization objective in MPM is not the optimal in the sense of minimizing the Bayes error. Therefore, an extension, minimum error minimax probability machine (MEMPM) was proposed and derived. Corresponding hyperplane, $H(\mathbf{w},b) = \{\mathbf{z} \mid \mathbf{w}^T\mathbf{z} = b\}$, is obtained by minimizing the worst-case Bayes error as follows:

$$\begin{aligned}
&\max_{\alpha,\beta,\mathbf{w}\neq 0,b} \theta\alpha + (1-\theta)\beta \\
&s.t. \\
&\inf_{x\sim(\bar{x},\Sigma_{\mathbf{x}})} \Pr\left(\mathbf{w}^{\mathbf{T}}\mathbf{x} \geq b\right) \geq \alpha, \\
&\inf_{y\sim(\bar{y},\Sigma_{\mathbf{y}})} \Pr\left(\mathbf{w}^{\mathbf{T}}\mathbf{y} \leq b\right) \geq \beta
\end{aligned} \tag{29}$$

where α, β are the worst-case classification accuracies of future data for the class ***x*** and ***y***, respectively. θ represents the prior probability of the class ***x*** and (1- θ) is the prior probability of the class ***y***.

The MPM and MEMPM models are formulated in input space and we assumed that two classes are linearly separable. The nonlinear classification problems can be solved by mapping the problem to a higher dimensional feature space by using the kernel functions that satisfy the Mercer's condition. The MPM and MEMPM can be extended to find a hyperplane in a feature space, which is nonlinear in input space (Ince and Trafalis 2006; Lanckriet, Ghaoui, Bhattacharyya and Jordan 2003; Ng, Zhong and Yang 2007). Also, the nonlinear decision function can be formulated for training samples as

$$H(\boldsymbol{z}) = \sum_{i=1}^{N_x} \boldsymbol{w}_i K(\boldsymbol{z}, \boldsymbol{x}_i) + \sum_{j=1}^{N_y} \boldsymbol{w}_j K(\boldsymbol{z}, \boldsymbol{y}_j), \tag{30}$$

where $\boldsymbol{w}_i$ is obtained by solving kernelized MPM and MEMPM models as in (Lanckriet, Ghaoui, Bhattacharyya and Jordan 2003), $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are the training data for class ***x*** and class ***y***, respectively; while $K(\cdot,\cdot)$ is a kernel function. Since The MPM is formulated as second order cone program (SOCP) problem, general-purpose programs such as SeDuMi, Mosek can be used efficiently to find optimal solution.

3.5 A Hybrid Model for Stock Market Prediction

Predicting the direction of the stock indices, and individual stock price can be formulated as a two-class classification problem. Several artificial intelligence techniques, namely kernel methods, neural networks, decision trees, swarm intelligence etc., have been developed and used successfully.

Stock market prediction requires capturing and modeling actions of stock market players, while observing and evaluating historical data. Stock prices increase or decrease reacting to several factors including “inside” and publicly available information. The methodology considers historical stock prices as inputs (predictors) to create a prediction model that forecasts next day’s trend of a stock market index. Figure 1 shows the prediction model proposed in this paper. The proposed model consists of two main stages. In the first stage, the ICA technique is used to estimate the independent components from the stock market data. Then these independent components were integrated into the kernel methods to build a stock market prediction model. The detail of the proposed approach is as follows:

In step 1, we obtain the stock market index data (daily open, high, low, close, and volume). In step 2, technical indicators (see Table 1) are computed by using the stock market index data. Technical indicators can be used to estimate (predict) the possibility of current trend reversal and then making buy/sell decision. We have used 12 technical indicators as input in the proposed approach. Definitions of these indicators are given in Table 1. Step 3 is the data smoothing steps. The forecasting variables (technical indicators) have to be smoothed with a suitable preprocess (Atsalakis and Valavanis 2009). Then, ICA algorithm chooses the important indicators that can be used as input in the next stage.

In the second stage (step 5), the ICs are used as input variables to construct kernel methods for predicting stock market direction. The dataset is divided into two groups, training and validation set. Kernel methods are applied to training dataset, and validation set is used to compute some performance statistics. Then, we use these statistics for comparison of kernel methods.

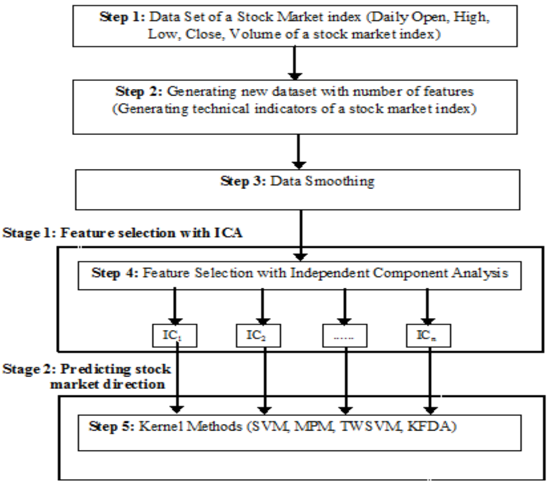
Step 1: Data Set of a Stock Market index (Daily Open, High, Low, Close, Volume of a stock market index)
Step 2: Generating new dataset with number of features (Generating technical indicators of a stock market index)
Step 3: Data Smoothing
Stage 1: Feature selection with ICA
Step 4: Feature Selection with Independent Component Analysis
IC$_1$
IC$_2$
......
IC$_n$
Stage 2: Predicting stock market direction
Step 5: Kernel Methods (SVM, MPM, TWSVM, KFDA)

Figure 1: Stock prediction model

4 Experimental Results and Discussion

4.1. Dataset and performance criteria

For evaluating the performance of the proposed model, the daily Dow-Jones, Nasdaq 100, and S&P500 indices are used in this study. Since we attempt to forecast the direction of stock market indices, technical indicators are used as forecasting variables . Our goal is to predict the directions of daily change of the stock price index. The direction is defined as “1” or “-1”. If the next day’s index value is greater than today’s index value, then direction is defined as “1”, otherwise it is defined as “-1”. Dataset consists of 2037 trading days, from August 3, 2007 to May 23, 2015. 80% of the data is used as training sample and 20% of the data is used for holdout sample. The number of the training sample is 1600 and that of the holdout sample is 437. The holdout sample is used to evaluate the prediction performance. The input data are normalized into the range of [−1:0; 1:0]. The goal of scaling is to independently normalize each feature component to the specified range.

Table 1: Initially selected technical indicators as input

Technical Indicators	**Formula**
Exponential Moving Average (*EMA 10* and *EMA 50*)	$(C_t \times a) + \left(EMA_{t-1} \times (1-a)\right)$ $a = 2/(n+1)$
Stochastic Oscillator %K	$\%K = \frac{(C_t - LL_{t-n})}{(HH_{t-n} - LL_{t-n})} \times 100$ where LL_t and HH_t are lowest low and highest high in the last t days, respectively
Price Rate of Change	$\frac{C_t - C_{t-n}}{C_{t-n}} \times 100$
Relative Strength Index (RSI)	$RSI = 100 - \frac{100}{1+(U/D)}$; U = Total gain in the last n days; D = Total Loss in last n days
Accumulation Distribution Oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
MACD	$MACD(t) = EMA_{12}(t) - EMA_{26}(t)$
Williams *%R*	$\%R = \frac{HH_{t-n} - C_t}{HH_{t-n} - LL_{t-n}} \times 100$
High Price Acceleration	$\frac{H_t - HH_{t-n}}{H_n} \times 100$
Disparity 5	$\frac{C_t}{MA_5} \times 100$
Disparity 10	$\frac{C_t}{MA_{10}} \times 100$

C_t is the closing price at time t, L_t the low price at time t, H_t high price at time t, LL_{t-n} lowest low in the last t-n days, HH_{t-n} highest high in the last t-n days, MA_t the simple moving average of t days.

The prediction performance (hit ratio) is calculated as follows:

$$P = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{31}$$

where R_i is the prediction result for i^{th} trading day and defined by

$$\mathrm{R_i} = \begin{cases} 1 & \text{if } \mathrm{PO_i} = \mathrm{AO_i} \\ 0 & \text{otherwise,} \end{cases}$$

PO_i is the predicted output from the model for the i^{th} trading day, AO_i is the actual output for the i^{th} trading day and n is the number of the examples.

The Sharpe ratio is another criteria that is used as performance criteria. It can be defined as the mean return of the trading strategy by its standard deviation. Another words, The Sharpe ratio measures return to the risk taken; higher positive values are preferred.

4.2. Prediction results

ICA is first applied to filter out the noise contained in the dataset. The filtered data are then used in kernel methods (SVM, MPM, TSSVM, KFDA). When using ICA for de-noising, the basic ICA model is first utilized to the mixture matrix X of size $m\times n$ combined from m forecasting variables (x_i) of size $1\times n$ for estimating a demixing matrix (W) of size $m\times m$ and independent components (y_i) of size $1\times n$. To find the ICs representing the noise, the Testing-and-Acceptance (TnA) method is used. Specifically, the Relative Hamming Distance (RHD) reconstruction error is adapted to order the ICs. The smaller the RHD value is the higher is the similarity between the data (see (Lu, Lee and Chiu 2009) for more information about RHD). After obtaining the de-noised data, we used them in building the prediction model. The performances of kernel methods are mainly affected by the setting of the parameters C and γ (Lu, Lee and Chiu 2009). In this study, we used a radial basis (RBF) kernel function. The grid search algorithm is used to determine the best C and γ for SVM, C_1 , C_2 and γ for TWSM. Once these optimal parameters are determined, the whole training sample is trained again.

It is also of interest to compare the performance of the hybrid kernel methods with that of the SVM, TWSVM, MPM, KFDA and random walk model. The random walk model assumes that the best forecast is equal to the most recently observable observation. Thus, the prediction using the random walk model would be expressed as $y_{t+1} = y_t$.

After determining the parameters of each model, comparison of the performance among hybrid kernel methods, SVM, TWSVM, MPM, KFDA and random walk model is carried out. Table 2 shows the number of correct prediction and the hit ratio for DJIA, Nasdaq and S&P 500 indices. For Dow-Jones index, hit ratio is between 0.53 and 0.86 with a mean hit ratio 0.76. For Nasdaq, the hit ratio changes between 0.50 and 0.73 with a mean ratio 0.65. Finally, the hit ratio is between 0.54 and 0.85 with a mean hit ratio 0.77 for S&P 500 index. The predictive effectiveness was tested by binomial test. The predictive effectiveness was evaluated by conducting a one-sided test of *H_0: p = 0.5 against H_a: p >0.50*. Hit ratios with an asterisk(*) indicate that they are significantly different from 0.5 at a 95% confidence level. This result confirms that the sign (direction) predicted by proposed models is better than random. Furthermore, it implies that the random walk model cannot be used to forecast the direction of stock index return.

Table 2: Forecasting performance of different models

	DJIA		**NASDAQ**		**S&P500**	
	Number	Hit Ratio	Number	Hit Ratio	Number	Hit Ratio
ICA-SVM	**378**	**0.86******	**321**	**0.73***	**372**	**0.85***
ICA-TWSVM	332	0.76*	297	0.68*	345	0.79*
ICA-MPM	351	0.80*	316	0.72*	364	0.83*
ICA- KFDA	372	0.85*	280	0.65*	368	0.84*
SVM	315	0.72*	278	0.64*	330	0.76*
TWSVM	275	0.63*	270	0.62*	315	0.72*
MPM	302	0.69*	272	0.62*	320	0.73*
KFDA	309	0.71*	265	0.61*	318	0.73*
RW	232	0.53	219	0.50	236	0.54

[a] The table shows the number of times a forecasting model correctly predict the direction of index return for holdout sample. A ratio marked with an asterisk(*) indicates that a 95% significance level based on a one side test of *H_0: p = 0.5* against *H_a: p >0.50*.

The trading performance of the proposed model is evaluated by simulation. Before we present the trading performance of the proposed model, we explain the operational details of the trading simulation. The trading simulation assumes that in the beginning of each period the investor makes an asset allocation decision of whether to shift assets stock index funds (Dow-Jones, Nasdaq and S&P 500 index fund) or stay in cash. It should be noted that stock index fund depends on stock index level. It is also assumed that the money that has been invested in stock index fund becomes illiquid and remains 'locked up' until the end of the period. In the beginning of each period the investor has to make a decision (to purchase stock index fund or stay in cash), based on the predictions generated by the forecasting models. This strategy implies full investment in either stock index fund or stay in cash for the whole period. Transaction cost, dividends, short selling, and leveraging are not allowed. Based on these assumptions, decision rules are given as follows:

If (C_{t+1} = +1), then invest in stock index fund and receive the stock return for period t+1 (R_{t+1})

Else if (C_{t+1} = -1) then stay in cash for the period t+1

where C is the sign of return predicted by the models. Using these decision rules, we can obtain the excess return over the simulation period for each forecasting models. Table 3 reports excess return and Sharpe Ratio of the forecasting models for each indices. For each index, excess return and Sharpe ratio are computed and given in Table 3. According to results, ICA-SVM method outperforms other forecasting methods for DJIA index. Excess return for DJIA index is 23.68% and corresponding Sharpe ratio is 0.94 which is the highest value among the other methods. For NASDAQ and S&P 500 indices, ICA-SVM has the highest excess return. The results demonstrate clearly that the return on investment and Sharpe ratio for the proposed forecasting models (hybrid models) outperform by far the pure forecasting models.

Table 3: Trading performances of the techniques

	DJIA		**NASDAQ**		**S&P500**	
	Return	Sharpe Ratio	Return	Sharpe Ratio	Return	Sharpe Ratio
ICA-SVM	**23.68**	**0.94**	**22.93**	0.91	**25.61**	0.81
ICA-	18.88	0.68	13.66	0.47	17.17	0.42
ICA-MPM	22.65	0.81	19.09	**0.93**	18.28	0.79
ICA-	17.35	0.9	18.05	0.73	15.77	**0.88**
SVM	14.32	0.35	10.78	0.25	15.53	0.43
TWSVM	9.52	0.17	9.73	0.14	11.24	0.36
MPM	13.25	0.28	9.25	0.08	13.75	0.28
KFDA	12.5	0.23	9.38	0.09	8.32	0.13
RW	-2.25	-0.12	-1.35	-0.05	-3.26	-0.17

5. Conclusions

With the inherent high volatility, complexity, and turbulence of stock markets, the prediction of stock market index is a challenging task. Also, the diversity and complexity of domain knowledge existing in the financial market makes it very difficult for investors to make the right decisions. This paper introduces a hybrid model that combines the ICA with kernel methods (SVM, TWSVM, MPM, and KFDA). In proposed model, the knowledge discovery process is mainly composed of feature representation by technical indicators, feature extraction by ICA, and modeling by kernel methods. According to the experimental results, ICA-SVM, ICA-TWSVM, ICA-MPM and ICA-KFDA methods are capable of producing satisfactory forecasting accuracies and excess returns for Dow-Jones, Nasdaq and S&P 500 indices.

We showed that the prediction accuracy can be significantly enhanced by using the two-stage model in comparison with a single-stage model. Since financial time series are non-stationary and, the two-stage model can better capture the characteristics of the time series. The results suggest that the proposed prediction model provides a promising alternative for financial time series forecasting.

REFERENCES

[1]. **Atsalakis, G. S. and K. P. Valavanis (2009),** ***Forecasting Stock Market Short-Term Trends Using a Neuro-Fuzzy Based Methodology****; Expert Systems with Applications* 36, 10696-10707;

[2]. **Burges, C. J. C. (1998),** ***A Tutorial on Support Vector Machines for Pattern Recognition****; Data Mining and Knowledge Discovery* 2, 121-167;

[3]**Cawley, G. C. and N. L. C. Talbot (2003),** ***Efficient Leave-one-out Cross-validation of Kernel Fisher Discriminant Classifiers;*** *Pattern Recognition* 36, 2585-2592;

[4]C**hapelle, O., V. Vapnik, O. Bousquet and S. Mukherjee (2002),** ***Choosing Multiple Parameters for Support Vector Machines****; Machine Learning* 46, 131-159;

[5]**Chavarnakul, T. and D. Enke (2008),** ***Intelligent Technical Analysis Based Equivolume Charting for Stock Trading Using Neural Networks;*** *Expert Systems with Applications* 34, 1004-1017;

[6]**Hyvarinen, A. (1999),** ***Fast and Robust Fixed-Point Algorithms for Independent Component Analysis****; Ieee Transactions on Neural Networks* 10, 626-634;

[7]**Hyvarinen, A. and E. Oja (1997),** ***A Fast Fixed-Point Algorithm for Independent Component Analysis****; Neural Computation* 9, 1483-1492;
[8]**Ince, H. and T. B. Trafalis (2006), Kernel** ***Methods for Short-Term Portfolio Management;*** *Expert Systems with Applications* 30, 535-542;
[9]**Jayadeva, K. R. and S. Chandra (2007), Twin** ***Support Vector Machines for Pattern Classification****; Ieee Transactions on Pattern Analysis and Machine Intelligence* 29, 905-910;
[10]**Kao, L. J., C. C. Chiu, C. J. Lu and J. L. Yang (2013),** ***Integration of Nonlinear Independent Component Analysis and Support Vector Regression for Stock Price Forecasting****; Neurocomputing* 99, 534-542;
[11]**Lanckriet, Gert RG, Laurent El Ghaoui, Chiranjib Bhattacharyya and Michael I. Jordan (2003),** ***A Robust Minimax Approach to Classification*****;** *The Journal of Machine Learning Research* 3, 555-582;
[12]**Leigh, W., R. Purvis and J. M. Ragusa (2002),** ***Forecasting the NYSE Composite Index with Technical Analysis, Pattern Recognizer, Neural Network, and Genetic Algorithm: A Case Study in Romantic Decision Support****; Decision Support Systems* 32, 361-377;
[13]**Lu, C. J., T. S. Lee and C. C. Chiu (2009),** ***Financial Time Series Forecasting Using Independent Component Analysis and Support Vector Regression;*** *Decision Support Systems* 47, 115-125;
[14]**Mika, S., G. Ratsch and K. R. Muller (2001),** ***A Mathematical Programming Approach to The Kernel Fisher Algorithm;*** *Advances in Neural Information Processing Systems 13* 13, 591-597;
[15]**Ng, J. K. C., Y. Z. Zhong and S. Q. Yang (2007),** ***A Comparative Study of Minimax Probability Machine-Based Approaches for Face Recognition****; Pattern Recognition Letters* 28, 1995-2002;
[16]**Piotroski, J. D. (2000),** ***Value Investing: The Use of Historical Financial Statement Information to Separate Winners From Losers****; Journal of Accounting Research* 38, 1-41;
[17]**Saadi, K., N. L. C. Talbot and G. C. Cawley (2007),** ***Optimally Regularised Kernel Fisher Discriminant Classification*****;** *Neural Networks* 20, 832-841;
[18]**Tay, F. E. H. and L. J. Cao (2002),** ***Modified Support Vector Machines in Financial Time Series Forecasting****; Neurocomputing* 48, 847-861;
[19]**Teoh, H. J., T. L. Chen, C. H. Cheng and H. H. Chu (2009),** ***A Hybrid Multi-Order Fuzzy Time Series for Forecasting Stock Markets*****;** *Expert Systems with Applications* 36, 7888-7897;
[20]**Vapnik, Vladimir Naumovich (2000)** ***The Nature of Statistical Learning Theory*****;** 2nd edition.(*Springer, New York);*
[21]**Wang, J. J., J. Z. Wang, Z. G. Zhang and S. P. Guo (2012),** ***Stock Index Forecasting Based on a Hybrid Model****; Omega-International Journal of Management Science* 40, 758-766.