Financial Data Analysis by Manifold Clustering and Kernel Machines

Shian-Chang Huang[1] and Tung-Kuang Wu[2]

Abstract**—**The challenge of modern financial forecasting comes from the high-dimensionality, nonlinearity, and non-stationarity of financial data. To address the problem, this study employs wavelet analysis to map time domain inputs to time-frequency (or wavelet) domain, and a sparse multi-manifold clustering (SMMC) to partition the high-dimensional feature space into several disjointed regions according to their dynamics. In the final stage, hierarchical multiple kernel machines (HMKM) are employed to perform the high-dimensional forecasting and trading. In our system, SMMC can effectively cluster data on multiple manifolds that are very close to each other, manifolds with non-uniform sampling and holes. HMKM embeds basis kernels in a directed acyclic graph, and optimized them by a graph-adapted sparsity-inducing norm, which performs the feature selection in polynomial time in the number of selected kernels. The empirical results demonstrate that the proposed model outperforms traditional neural networks, support vector machines, statistical models, and significantly reduces the forecasting errors.

Keywords**—**Manifold Clustering, Multiple Kernel Machine, Manifold Learning, Wavelet Analysis, Time Series Forecasting

I. Introduction

FINANCIAL forecasting and trading are essential for the success of financial institutions. The need for cutting edge technology to support financial trading is also urgent. Hence, numerous models have been developed to maximize forecast accuracy, but they are usually linear, parametric in nature and operate in time domain (Atsalakis and Valavanis [1][2]). Hence, their performance is unsatisfactory. On the other hand, with the continued liberalization of cross-border cash flow, international financial markets have become increasingly interdependent. Investors are highly susceptible to exchange risk and fluctuations in equity prices throughout the world.

However, the tight correlations among financial markets provide investors with valuable information to make accurate forecasts regarding the co-movements of stock indices. International investors are a diverse group, operating on very different time scales. As a result, the correlation pattern between international market indices are not fixed between each time scale. Consequently, this study aims to address the problem by a new model in wavelet domain that fully exploits time-frequency features from high-dimensional financial time series. Namely, this study will implement a new forecasting strategy which extracts key features from worldwide financial markets to enable more accurate predictions.

Financial time series are regarded as the most difficult signal for prediction. The difficulties arise from inherent nonlinearity and non-stationarity (time-varying dynamics) in financial time series. To reduce the non-stationary, one needs to transfer the stock index into its returns, and then predicts the returns, or employs classifiers to predict its future movements for optimal trading. The main streams of financial forecasting can be summarized as follows (Atsalakis and Valavanis [1] [2]; Bahrammirzaee [6]; Krollner et al. [22]): (1) classical statistical or econometric models include Auto-Regressive (AR), Auto-Regressive Moving Average (ARMA), Auto-Regressive Integrated Moving Average (ARIMA), Generalized Auto-Regressive Conditional Heteroscedasticity (GARCH), and Stochastic Volatility (SV) models. (2) Soft computing techniques (Zadeh [41]) include Neural Networks (NNs), Genetic Algorithms (GAs), Genetic Programming (GP), Case Based Reasoning (CBR), Support Vector Machines (SVMs, Vapnik [34]), and advanced kernel methods (Schoelkopf et al. [33]). In general, soft computing techniques have shown to provide better performance than statistical approaches.

Statistical methods are often limited by strict assumptions of normality, linearity, variable independence etc. Their models are usually stationary, parametric, and linear. However, financial time series are nonlinear with time-varying dynamics; namely, its statistical properties change with time. Consequently, most parametric statistical models are not flexible enough to compete with soft computing techniques which are usually nonlinear, non-parametric, and adaptive (Krollner et al. [22]). Although soft computing techniques are flexible enough, they tend to overfit in original time domain. The key to effectively solve this problem is by controlling the complexity of soft computing models. One solution is to regularize these models by some smoothness criterions. However, the effectiveness of regularization is limited, because most financial time series are not smooth in time domain. When an exceptional event or information occurs, financial markets become unstable and evolve to a new equilibrium state. Forecasting models operating in time domain are hard to track the sudden transients in stock prices. Therefore, the best solution is to transform financial time series to wavelet domain capable of identifying these key features with compact representation. Wavelet transformations are techniques for addressing the problem (Gençay et al. [17]; Boashash [7]).

The power of wavelet analysis comes from its multiple time-scale wavelets that can localize data in the time-frequency

Shian-Chang Huang[1] is with the Department of Business AdministrationNational Changhua University of Education, Taiwan.

Tung-Kuang Wu[2], is with Department of Information Management National Changhua University of Education, Taiwan.

space. At high scales (shorter time intervals), the wavelet has a small time support, and thus can better focus on short lived, strongly transient features like discontinuities, ruptures and singularities. At low scales (longer time intervals), the wavelet's time support is large, making it good at identifying long periodic features such as long-run trends and patterns (Aussem et al. [3]) . In contrast to sine or cosine basis waves in Fourier transformation, a wavelet basis function oscillates around zero and damps rapidly down to zero. Thus it is localized both in time and space, as opposed to the sine and cosine functions that have constant amplitude over the entire real line. Hence, wavelets are good at handling non-stationary signals and analyzing dynamic patterns that may change rapidly over time. Recent financial applications of wavelet analysis include Ramsey and Zhang [31], Davidson et al. [13], Pan and Wang [29], Ramsey and Lampart [32], Gençay et al. [16][17][18][19], Lee [25], Yamada [35], Yousefi and Weinreich [37], and In and Kim [20].

Another problem in forecasting is that financial time series are usually non-stationary; namely, time series switch their dynamics between different regions. This leads to changes in the dependency structure between input and output variables. Consequently, it is difficult for a single predictor to capture such a switching input-output relationship. Inspired by the so-called "divide-and-conquer" principle that is often used to attack complex problems, the approaches of local modeling have emerged as one of the promising methods of time series prediction (Oh [28]). This study employed a sparse multi-manifold clustering (SMMC) algorithm (Elhamifar and Vidal [14]) for partitioning the feature space into several disjointed regions for different time series dynamics. We then employed an architecture involving multiple experts to overcome the problem, namely, using different experts for different feature regions.

Sparse manifold clustering is outstanding at partitioning data points non-linearly distributed on multiple manifolds. In contrast to traditional nearest neighbors-based methods for manifold modeling, which fix the number of neighbors or the neighborhood radius and then compute the weights between points in each neighborhood, SMMC finds both the neighbors and the weights automatically. SMMC automatically choose a sparse solution for neighbors of the given data point, which approximately span a low-dimensional affine subspace at that point. The size of the optimal neighborhood of a data point, which can be different for different points, provides an estimate of the dimension of the manifold to which the point belongs. Consequently, SMMC can effectively handle multiple manifolds that are very close to each other, manifolds with non-uniform sampling and holes, as well as estimate the intrinsic dimensions of the manifolds.

Prior studies in financial forecasting such as Zhang and Hu [39], Yao and Tan [36], Zimmermann et al. [40], and Kamruzzaman and Sarker [21] employed neural networks as the nonlinear predictor. They demonstrated that NNs usually outperform statistical ARIMA models. However, traditional NN models suffer several disadvantages, including: dependency on a large number of model parameters, possibility of being trapped into local minima, and over-fitting on training data resulting in poor generalization ability. The above problems are partial addressed by the technique of support vector machine (Vapnik [34]; Cristianini and Shawe-Taylor [11]). SVM embodies the structural risk minimization principle for regularizing model complexity, and thus possesses excellent generalization properties. Although SVM is very successful in various applications, the success of SVMs is often dependent on the choice of a good kernel and features-ones that are typically hand-crafted and fixed in advance. However, hand-tuning kernel parameters can be difficult as can selecting and combining appropriate sets of features. Moreover, SVMs are based on a single kernel. In practice data come from multiple sources, it is often desirable to base on combinations of multiple kernels. Multiple Kernel Machines (MKM, Lanckriet et al., [23]) seeks to address this issue by learning the kernel from training data. In particular, it focuses on how the kernel can be learnt as a linear combination of given basis or local kernels.

Many MKM formulations have been proposed in the literature. Recent applications have also shown that using multiple kernels instead of a single one can enhance the interpretability of the decision function and improve performances (Lanckriet et al. [23]). However, one major difficulty in MKM is that the number of these basis kernels is usually exponential in the dimension of the input space and applying multiple kernel learning directly in this decomposition would be intractable. To select basis kernels more efficiently, Bach [4][5] proposed a hierarchical multiple kernel machines (HMKM) which use the natural hierarchical structure of the problem to extend the multiple kernel learning framework to kernels that can be embedded in a directed acyclic graph (DAG). Bach [5] showed that it is possible to perform high-dimensional kernel selection through a graph-adapted sparsity-inducing norm in polynomial time in the number of selected kernels.

The major innovation of this paper lies in the combination of SMMC with wavelet-based HMKM for global stock index forecasting. In the first stage, wavelet analysis is used to transform the input space (raw data) to a time scale feature space suitable for financial forecasting, whereupon a SMMC algorithm is used to partition the feature space according to each time scale dynamics. In the second stage of the new method, multiple HMKMs that best fit partitioned regions are constructed for the final forecasting. Empirical results show that the proposed model outperforms neural networks, SVMs, and traditional GARCH models by significantly reducing root-mean-squared forecasting error.

Performing regression in the efficient manifold makes the proposed model more sparse and parsimonious than traditional SVM models, and reduces the risk of overfitting. Similar works related to this study are Zhang et al. [38] and Li and Kuo [26]. Zhang et al. [38] implemented a neural network predictor in wavelet domain, which also suffers the drawbacks of NNs. Li and Kuo [26] used a two-level SOM (self-organizing map) to detect price patterns for forecasting and trading strategy. SOM is not a manifold clustering algorithm, which does not consider the multiple manifolds formed by data, and therefore is less effective.

II. The Proposed System

A. Wavelet Transformations

Wavelet analysis is good at analyzing localized variations of power within a time series. By decomposing a time series into time-scale (or time-frequency) space, one is able to determine both the dominant modes of variability and how those modes vary in time. For the details and applications of wavelet analysis we refer to Chui [10], Daubechies [12], Percival and Walden [30], Lee [24], Lee [25], Gençay et al. [17], Bruce and Gao [9].

Any function $y(t)$ in $L^2(R)$ can be decomposed by a sequence of projections onto the wavelet basis:

$$s_{J,k} = \int y(t)\Phi_{J,k}(t)dt \quad (1)$$

$$d_{j,k} = \int y(t)\Psi_{j,k}(t)dt, \quad (2)$$

where J is the number of multiresolution; Φ is the father wavelet and Ψ is the mother wavelet. $s_{J,k}$ is the smooth coefficients, while $d_{j,k}$ is the detailed coefficients. $\Phi_{j,k}$ and $\Psi_{j,k}$ are scaling and translation of Φ and Ψ, defined by

$$\Phi_{j,k}(t) = 2^{-j/2}\Phi(2^{-j}t-k) = 2^{-j/2}\Phi\left(\frac{t-2^j k}{2^j}\right) \quad (3)$$

$$\Psi_{j,k}(t) = 2^{-j/2}\Psi(2^{-j}t-k) = 2^{-j/2}\Psi\left(\frac{t-2^j k}{2^j}\right). \quad (4)$$

The above projections are called wavelet transformation. $s_{J,k}$, and $d_{j,k}$ are signal representation in wavelet domain.

To recover the signal, the inverse wavelet transformation constructs smooth and detail signals from $s_{J,k}$, $d_{j,k}$, and synthesizes the signal as follows:

$$y(t) = \sum_k s_{J,k}\Phi_{J,k}(t) + \sum_k d_{J,k}\Psi_{J,k}(t)$$

$$+ \sum_k d_{J-1,k}\Psi_{J-1,k}(t) + \ldots + \sum_k d_{1,k}\Psi_{1,k}(t),$$

B. Sparse Multi-Manifold Clustering

According to Elhamifar and Vidal [14], assuming we are given a collection of N data points $\{\mathbf{x}_i \in R^D\}_{i=1}^N$ lying in n different manifolds $\{\mathbb{M}_l\}_{l=1}^n$ of intrinsic dimensions $\{d_l\}_{l=1}^n$. We consider the problem of simultaneously clustering the data according to the underlying manifolds and obtaining a low-dimensional representation of the data points within each cluster.

To do clustering, we wish to connect each point to other points from the same manifold. We address this problem by formulating an optimization algorithm based on sparse representation. The underlying assumption behind the proposed method is that each data point has a small neighborhood in which the minimum number of points that span a low-dimensional affine subspace passing near that point is given by the points from the same manifold.

Consider a point $\mathbf{x}_i$ in the d_l-dimensional manifold $\mathbb{M}_l$ and consider the set of points $\{\mathbf{x}_j\}_{j\neq i}$. Among these points, the ones that are neighbors of $\mathbf{x}_i$ in $\mathbb{M}_l$ span a d_l-dimensional affine subspace of R^D that passes near $\mathbf{x}_i$. In other words,

$$\| [\mathbf{x}_1 - \mathbf{x}_i, \ldots, \mathbf{x}_N - \mathbf{x}_i]\mathbf{c}_i \|_2 \leq \varepsilon \text{ and } \mathbf{1}^T\mathbf{c}_i = 1 \quad (5)$$

has a solution $\mathbf{c}_i$ whose $d_l + 1$ nonzero entries corresponds to $d_l + 1$ neighbors of $\mathbf{x}_i$ in $\mathbb{M}_l$.

Following Elhamifar and Vidal [14], we normalize the vectors $\{\mathbf{x}_j - \mathbf{x}_i\}_{j\neq i}$ and let

$$\mathbf{X}_i = \left[\frac{\mathbf{x}_1 - \mathbf{x}_i}{\| \mathbf{x}_1 - \mathbf{x}_i \|_2}, \ldots, \frac{\mathbf{x}_N - \mathbf{x}_i}{\| \mathbf{x}_N - \mathbf{x}_i \|_2}\right]. \quad (6)$$

In this way, the locations of the nonzero entries of any solution $\mathbf{c}_i$ of $\| \mathbf{X}_i\mathbf{c}_i \|_2 \leq \varepsilon$, $\mathbf{1}^T\mathbf{c}_i = 1$ do not depend on whether the selected points are close to or far from $\mathbf{x}_i$. Among all the solutions, we look for the one that uses a few closest neighbors of $\mathbf{x}_i$. For this aim, points that are closer to $\mathbf{x}_i$ get lower penalty than points that are farther away. Consequently, we consider the following weighted L_1-optimization program

$$\min \| \mathbf{Q}_i\mathbf{c}_i \|_1 \text{ subject to } \| \mathbf{X}_i\mathbf{c}_i \|_2 \leq \varepsilon, \mathbf{1}^T\mathbf{c}_i = 1, \quad (7)$$

where the L_1-norm promotes sparsity of the solution. $\mathbf{Q}_i$ is the proximity inducing matrix, which is a positive-definite diagonal matrix with diagonal elements $\frac{\| \mathbf{x}_j - \mathbf{x}_i \|_2}{\sum_{t\neq i} \| \mathbf{x}_t - \mathbf{x}_i \|_2}$.

For real implementation, we use the following formulation

$$\min \lambda \| \mathbf{Q}_i\mathbf{c}_i \|_1 + \frac{1}{2} \| \mathbf{X}_i\mathbf{c}_i \|_2^2 \text{ subject to } \mathbf{1}^T\mathbf{c}_i = 1, \quad (8)$$

where the parameter λ sets the trade-off between the sparsity of the solution and the affine reconstruction error.

By solving the proposed optimization programs for each data point, we obtain the necessary information for clustering. This is because the solution $\mathbf{c}_i^T = [c_{i1}, \ldots, c_{iN}]$ of the proposed optimization programs satisfies

$$\sum_{j\neq i} \frac{c_{ij}}{\| \mathbf{x}_j - \mathbf{x}_i \|_2}(\mathbf{x}_j - \mathbf{x}_i) \approx 0. \quad (9)$$

We can rewrite $\mathbf{x}_i$ as combination other components, namely $\mathbf{x}_i \approx [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]\mathbf{w}_i$, where the weight vector $\mathbf{w}_i = [w_{i1}, \ldots, w_{iN}]$ associated to the i-th data point is defined as

$$w_{ii} = 0, w_{ij} = \frac{c_{ij} / \| \mathbf{x}_j - \mathbf{x}_i \|_2}{\sum_{t \neq i} c_{it} / \| \mathbf{x}_t - \mathbf{x}_i \|_2} (\mathbf{x}_j - \mathbf{x}_i), j \neq i. \quad (10)$$

The indices of the few nonzero elements of $\mathbf{w}_i$, ideally, correspond to neighbors of $\mathbf{x}_i$ in the same manifold.

Next, the weights $\mathbf{w}_i$ are used to perform clustering. First we build a similarity graph $G = (V, E)$ whose nodes represent the data points. We connect each node i, corresponding to $\mathbf{x}_i$, to the node j, corresponding to $\mathbf{x}_j$, with an edge whose weight is equal to $[w_{ij}]$. While, potentially, every node can get connected to all other nodes, because of the sparsity of $\mathbf{w}_i$, each node i connects itself to only a few other nodes that correspond to the neighbors of $\mathbf{x}_i$ in the same manifold. Such neighbors are called sparse neighbors. In addition, the distances of the sparse neighbors to $\mathbf{x}_i$ are reflected in the weights $[w_{ij}]$. The similarity graph built in this way has ideally several connected components, where points in the same manifold are connected to each other and there is no connection between two points in different manifolds. Then we can cluster the data by applying spectral clustering (Ng [27]).

C. Hierarchical Multiple Kernel Learning

Classical kernel machines are based on a single kernel. In practice data come from multiple sources, it is often desirable to base learning machines on combinations of multiple kernels. Following Bach [4][5], this study also considers a positive definite kernel that can be expressed as a large sum of positive definite basis or local kernels. This exactly corresponds to the situation where a large feature space is the concatenation of smaller feature spaces, and we aim to do selection among these many kernels, which may be done through multiple kernel learning.

However, one major difficulty is that the number of these smaller kernels is usually exponential in the dimension of the input space and applying multiple kernel learning directly in this decomposition would be intractable. In order to perform selection efficiently, we follow the assumption of Bach [4][5] that these small kernels can be embedded in a directed acyclic graph (DAG).

Let's consider the problem of predicting a random variable Y from a random variable X, where $\mathbb{X}$ and $\mathbb{Y}$ may be general spaces. We assume that we are given n observations $(\mathbf{x}_i, y_i), i = 1,..., n$. We define the empirical risk of a function f from X to R as $\frac{1}{n}\sum_{i=1}^{n} l(y_i, f(\mathbf{x}_i))$, where l is a loss function.

Assume that we are given a positive definite kernel $k : \mathbb{X} \times \mathbb{X} \to R$, and that this kernel can be expressed as the sum, over an index set V, of basis kernels $k_v, v \in V$; namely, for all $\mathbf{x}, \mathbf{x}'$, we have $k(\mathbf{x}, \mathbf{x}') = \sum_{v \in V} k_v(\mathbf{x}, \mathbf{x}')$. For each $v \in V$, we denote by $\mathbb{F}_v$ and ϕ_v the feature space and feature map of k_v, i.e., $k_v(\mathbf{x}, \mathbf{x}') = (\phi_v(\mathbf{x}), \phi(\mathbf{x}'))$. The feature map $\phi(\mathbf{x})$ and feature space $\mathbb{F}$ for k is the concatenation of the feature maps $\phi_v(\mathbf{x})$ for each kernel k_v, i.e, $\mathbb{F} = \prod_{v \in V} \mathbb{F}_v$ and $\phi(\mathbf{x}) = (\phi_v(\mathbf{x}))_{v \in V}$. The multiple kernel learning looking for a certain $\beta \in \mathbb{F}$ to form a predictor function $f(\mathbf{x}) = \langle \beta, \phi(\mathbf{x}) \rangle$ is equivalent to looking jointly for $\beta_v \in \mathbb{F}_v, \forall v \in V$, and $f(\mathbf{x}) = \sum_{v \in V} \langle \beta_v, \phi_v(\mathbf{x}) \rangle$.

The goal of this paper is to perform kernel selection among the kernels $k_v, v \in V$. Following Bach [4][5], we use a graph to limit the search to specific subsets of V. Namely, instead of considering all possible subsets of active (relevant) vertices, we are only interested in estimating correctly the hull of these relevant vertices.

Assume that the input space $\mathbb{X}$ factorizes into p-components $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_p$ and that we are given p sequences of length $q+1$ of kernels $k_{ij}(\mathbf{x}_i, \mathbf{x}_i'), i \in \{1,..., p\}$, $j \in \{0,..., q\}$, such that $k(\mathbf{x}, \mathbf{x}') = \sum_{j_1,...,j_p=0}^{q} \prod_{i=1}^{p} k_{ij_i}(\mathbf{x}_i, \mathbf{x}_i') = \prod_{i=1}^{p} (\sum_{j_i=0}^{q} k_{ij_i}(\mathbf{x}_i, \mathbf{x}_i'))$. We thus have a sum of $(q+1)^p$ kernels, that can be computed efficiently as a product of p sums. In this context, products of kernels correspond to interactions between certain variables, and our DAG implies that we select an interaction only after all sub-interactions were already selected. The DAGs are especially suited to nonlinear variable selection, in particular with the polynomial and Gaussian kernels.

Let's consider the linear kernel, $k_{ij}(\mathbf{x}_i, \mathbf{x}_i') = C_j^q \langle \mathbf{x}_i, \mathbf{x}_i' \rangle^j$, where $\langle , \rangle$ stands for inner product. The full kernel is then equal to $k(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{p} \sum_{j=0}^{q} C_j^q \langle \mathbf{x}_i, \mathbf{x}_i' \rangle^j = \prod_{i=1}^{p} (1 + \mathbf{x}_i \mathbf{x}_i')^q$. Note that this is not exactly the usual polynomial kernel. Typical polynomial kernel, $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}\mathbf{x}')^q$, are multivariate polynomials of total degree less than q. Another example is the product of Gaussian kernel, $\sum_{J \subset \{1,...,p\}} \prod_{i \in J} e^{-b(\mathbf{x}_i - \mathbf{x}_i')^2} = \sum_{J \subset \{1,...,p\}} e^{-b\|\mathbf{x}_J - \mathbf{x}_J'\|^2}$. It's also known as all-subset Gaussian kernel. Our framework will select the relevant subsets for the Gaussian kernels.

The optimal hierarchical multiple kernel learning could be formulated as the following minimization problem:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \sum_{v} \langle \beta_v, \phi_v(\mathbf{x}_i) \rangle) + \frac{\lambda}{2} (\sum_{v} d_v \| \beta_{D(v)} \|)^2, \quad (11)$$

where $\sum_v d_v \| \beta_{D(v)} \| = \sum_v d_v (\sum_{w \in D(v)} \| \beta_w \|^2)^{1/2}$ is the structured block L_1-norm; d_v are positive weights and $D(v)$ is the descendant set of v [1]. Penalizing by such a norm will indeed impose that some of the vectors $\beta_{D(v)} \in \prod_{w \in D(v)} F_w$ are exactly zero, and leads to sparse solutions.

III. Experimental Results

This section uses real financial data to test our system. In the first stage, the SMMC provides a effective mean for separating different dynamic regimes according to the key wavelet features. In the second stage, HMKM maps data to high-dimensional multiple reproducing kernel Hilbert spaces (RKHS). Multiple RKHSs are rich in topological structures for modeling nonlinear relationships between input and output. Consequently, the hybrid system is flexible enough for representing nonlinear non-stationary financial time series.

The first data set used in this study comprise daily stock indices of six industrial countries including: CAC40(France), FTSE100(UK), DAX30(Germany), MIB40(Italy), TSX60(Canada), and NK225(Japan). The data cover the period from Jan. 2004 to Dec. 2005, comprising 435 observations. These stock market indices are transformed into daily returns via the following formula:

$$r(t) = \log\left(\frac{P(t)}{P(t-1)}\right), \tag{12}$$

where $P(t)$ represents the actual index value at time t.

The second data set comprises major Asian stock indices, including: NASDAQ (US), NK225 (Japan), TWSI (Taiwan) and KOSPI (South Korea). The data period is from Jan. 2003 to Dec. 2004 with 435 observations.

In this study, one-step-ahead forecasting is considered for trading decisions. One-step-ahead forecasting is suitable for the construction of an online trading system. The lagged returns of each index serve as the inputs for prediction. These returns are decomposed by wavelet basis into five mutually orthogonal periodicity series, ranging from the shortest-periodicity series to the longest-periodicity series. This study uses Daubechies least asymmetric filters with length 8 for the decomposition. Due to decomposition, key time-scale features are captured. These features cannot be revealed clearly in the original input space.

To conquer the problem of non-stationarity inherent in financial time series, after multi-resolution decomposition a SMMC algorithm is employed to partition the feature space. According to their dynamics, the space is partitions into six distinct regimes. In each regime, a HMKM model is trained for forecasting. The training is in a dynamic and adaptive manner, for example, 300 data points before the day of prediction are treated as the training data set, and the window of the training data set slides with the current prediction. Thus our system are dynamic and adaptive to online input data. The daily returns in the last 135 days of the data series are used as the test data set to evaluate the performances of the new model and other prediction models. Two lagged returns of each index serve as the explanatory variables for prediction. These returns are transformed to wavelet domain via Daubechies least asymmetric filter of length 8 (for other wavelet filters the performance for prediction is similar).

A. Performance Measurements and Model Settings

Traditional performance indices such as MSE (mean square error), RMSE (root mean squared error), MAE (mean absolute error), and MAPE (mean absolute percent error), can be used to measure forecasting accuracy. These measures are defined as follows:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(r_t - \hat{r}_t)^2,$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(r_t - \hat{r}_t)^2},$$

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|r_t - \hat{r}_t|,$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{r_t - \hat{r}_t}{r_t}\right|,$$

where N denotes the number of forecasting periods, r_t represents the actual return at time t, and $\hat{r}_t$ is the forecast return at time t. MAE and MAPE have been realized their drawbacks and shortcomings in measuring performance, and MSE is the square of RMSE. Therefore they are excluded in this study.

This study compares the new model with three traditional predictors, the GARCH ([8][15]), RBFNN (radial basis neural network), and SVM models. GARCH models are good at describing volatility clustering. For model settings, the RBFNN used in this study has three layers: an input layer, a hidden layer with a non-linear RBF (radial basis function) activation function, and a linear output layer. All parameters are optimized during the training stage. The RBFNN is trained using the following settings: the number of centers (or neuros) is five, and the centers are initialized via a k-means algorithm. The maximum iteration is 500, and the stopping criterion is 1e-5.

The SVM parameters are optimized by a genetic algorithm which uses the inverse of forecasting error as the fitness function. Table 1 is the performance of every model for the first data set. Table 2 is the performance of every model for the second data set.

[1] Since we are only interested in the hull of the selected elements $\beta_v \in F_v$, the hull of a set I is characterized by the set of v, such that $D(v) \subset I^c$, i.e., hull(I) = $\{v \in V, D(v) \subset I^c\}^c$. In our context, we are hence looking at selecting vertices $v \in V$ for which $\beta_{D(v)} = (\beta_w)_{w \in D(v)} = 0$.

TABLE I
FORECASTING PERFORMANCE OF FOUR MODELS ON THE FIRST DATA SET

RMSE	CAC 40	FTSE 100	DAX 30
GARCH Predictions	0.2970	0.2853	0.3335
NN Predictions	0.3206	0.2963	0.3728
Pure SVM Predictions	0.3213	0.3023	0.3708
The Hybrid Model	0.1360	0.1394	0.1589
RMSE	MIB 40	TSX 60	NK 225
GARCH Predictions	0.2541	0.2582	0.3944
NN Predictions	0.2827	0.3264	0.3825
Pure SVM Predictions	0.2788	0.3394	0.3790
The Hybrid Model	0.1299	0.1621	0.1509

TABLE II
PERFORMANCE OF FOUR MODELS ON THE SECOND DATA SET

RMSE	NK225	TWSI	KOSPI
GARCH Predictions	0.4725	0.5557	0.2673
NN Predictions	0.4604	0.7600	0.2701
pure SVM Predictions	0.4426	0.5672	0.2883
The Hybrid Model	0.3133	0.3542	0.2022

B. Performance Comparison

The results clearly demonstrate that only the new model instantly effectively tracks the return movements. Traditional models have serious time delays. RBFNN and SVM models tend to over-react, while the GARCH model tends to under-react. On all of the data sets, the new model outperforms three traditional models. It substantially reduces the RMSE errors, and there is no significant difference among the three traditional models.

Another interesting phenomenon is that SVM always outperforms GARCH; namely, nonlinear non-parametric models are typically better than parametric ones. However, the performance of RBFNN is unstable and not always outperforms GARCH. Although SVM and RBFNN are all nonlinear and non-parametric regressors, their performance is very different. The difference comes from the model regularization. SVM addresses the drawbacks of overfitting of RBFNN by adding a regularization term in the model objective function, but the performance improvement is limited. Consequently, regularization is not the key to improve forecasting performance because most financial time series are not smooth in time domain. There are jumps in price when a exceptional event or information occur. Forecasting models operating in time domain are hard to track these sudden jumps. For significant performance improvement, one should consider transforming the time series to wavelet domain which is excellent in capturing these features with compact representation. Our proposed model combines all the advantages mentioned above. Therefore, it performs best.

IV. CONCLUSION

Global investors are a diverse group that operates on very different time scales. Moreover, financial markets switch their dynamics between bull and bear markets. Thus, their correlation patterns and dynamics changes over time. Combining wavelet analysis, sparse multi-manifold clustering (SMMC), and hierarchical multiple kernel machines (HMKM), this research developed a novel prediction system, which operates on the space of multiple resolutions and multiple dynamic regimes. Using multiple HMKMs, the new system is flexible enough to predict the future evolutions of worldwide stock indices, and makes trading decisions. Convectional models such as neural networks, GARCH, and SVM only incorporate information regarding the average correlations and dynamics among the indices. Therefore, their performance is poor.

The success of the proposed model can be attributed to the following reasons. First, the correlation patterns of our data sets are complex. Wavelet analysis provides a solid foundation for extracting the key features of these indices. Second, the non-stationarity of financial time series is successfully partitioned by the sparse multi-manifold clustering algorithm, and adaptively solved by our sliding training algorithm which dynamically adjusts the HMKM model. Third, the approach of local modeling is a new and promising method to handle financial time series. This study employs an architecture involving multiple experts, which uses different experts for different feature regions.

The highly effective framework for forecasting and trading decision in this study can also be applied to other problems involving financial investments. Results of this study can also be used to perform a good hedge on global investments.

REFERENCES

[1] Atsalakis, G. and Valavanis, K. (2009), Surveying stock market forecasting techniques - Part II: Soft computing methods, Expert Systems with Applications, 36(3), 2009, 5932-5941.

[2] Atsalakis, G. and Valavanis, K. (2010), Surveying Stock Market Forecasting Techniques - Part I: Conventional Methods, Journal of Computation Optimization in Economics and Finance, 2(1), Article 4.

[3] Aussem, A., Campbell, J., and Murthagh, F. (1998), Wavelet-based feature extraction and decomposition strategies for financial forecasting, *Journal of Computational Intelligence in Finance*, 6(2), 5-12.

[4] Bach, F. (2008), Exploring Large Feature Spaces with Hierarchical Multiple Kernel Learning, Advances in Neural Information Processing Systems (NIPS 2008).

[5] Bach, F. (2009), High-Dimensional Non-Linear Variable Selection through Hierarchical Kernel Learning, Technical report, HAL 00413473.

[6] Bahrammirzaee, A. (2010), A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems, Neural Computing and Applications, 19(8), 1165-1195. http://dx.doi.org/10.1007/s00521-010-0362-z

[7] Boashash, B. (2003), Time Frequency Signal Analysis and Processing: A Comprehensive Reference, Elsevier.

[8] Bollerslev, T. (1986) Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics*, 31, 307-327. http://dx.doi.org/10.1016/0304-4076(86)90063-1

[9] Bruce, A. and Gao, H. Y. (1996) Applied Wavelet Analysis with SPLUS, *Springer-Verlag*, New York.

[10] Chui, C. K. (1992) An Introduction to Wavelets, *Academic Press*, Boston, MA.

[11] Cristianini, N. and Shawe-Taylor, J. (2000) An Introduction to Support Vector Machines, Cambridge University Press.

[12] Daubechies, I. (1992) Ten Lectures on Wavelets, SIAM, Philadelphia, PA. http://dx.doi.org/10.1137/1.9781611970104

[13] Davidson, R., Labys, W. C., and Lesourd, J. B. (1998) Wavelet analysis of commodity price behavior, *Computational Computational Economics*, 11, 103-128. http://dx.doi.org/10.1023/A:1008666428579

[14] Elhamifar, E. and Vidal, R. (2011), Sparse Manifold Clustering and Embedding, Neural Information Processing Systems (NIPS), 2011.

[15] Glosten, L., Jagannathan, R., and Runkle, D. (1993), Relationship between the Expected Value and the Volatility of the Nominal Excess Return on Stocks, *Journal of Finance*, 48, 1779-1801. http://dx.doi.org/10.1111/j.1540-6261.1993.tb05128.x

[16] Gençay, R., Selçuk, F., and Whitcher, B. (2001) Scaling properties of foreign exchange volatility, *Physica*, 289, 249-266. http://dx.doi.org/10.1016/S0378-4371(00)00456-8
[17] Gençay, R., Selçuk, F., and Whitcher, B. (2002) An introduction to wavelets and other filtering methods in finance and economics. *Academic Press*, London.
[18] Gençay, R., Selçuk, F., and Whitcher, B. (2003) Systematic risk and time scales, *Quantitative Finance*, 3, 108-116. http://dx.doi.org/10.1088/1469-7688/3/2/305
[19] Gençay, R., Selçuk, F., and Whitcher, B. (2005) Multiscale systematic risk, *Journal of International Money and Finance*, 24, 55-70. http://dx.doi.org/10.1016/j.jimonfin.2004.10.003
[20] In, F. and Kim, S. (2006), The hedge ratio and the empirical relationship between the stock and futures markets: A new approach using wavelet analysis, *Journal of Business*, 2006, 79, 799-820. http://dx.doi.org/10.1086/499138
[21] Kamruzzaman J. and Sarker, R. (2003), Forecasting of currency exchange rate using ANN: a case study, to appear in *Proc. IEEE International Conference on Neural Networks & Signal Processing*, (ICNNSP'03), Nanjing, China, pp793-797, 2003.
[22] Krollner, B., Vanstone, B., and Finnie, G. (2010), Financial time series forecasting with machine learning techniques: A survey, European symposium on artificial neural networks: Computational and machine learning, Bruges, Belgium, April 2010.
[23] Lanckriet, G. R. G., Cristianini, N., Ghaoui, L. E., Bartlett, P., and Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. J. Machine Learning Research, 5, 27-72.
[24] Lee, G. H. (1998) Wavelets and wavelet estimation: a review, *Journal of Economic Theory and Econometrics*, 4, 123-158.
[25] Lee, S. H. (2004) International transmission of stock market movements: a wavelet analysis, *Applied Economics Letters*, 11, 197-201. http://dx.doi.org/10.1080/1350485042000203850
[26] Li, S. T. and Kuo, S. C. (2008), Knowledge discovery in financial investment for forecasting and trading strategy through wavelet-based SOM networks, Expert Systems with Applications, 34(2), 935-951. http://dx.doi.org/10.1016/j.eswa.2006.10.039
[27] Ng, A.Y., Jordan, M.I., and Weiss, Y. (2002), On spectral clustering: analysis and an algorithm, Advances in Neural Information Processing Systems, MIT Press, 2002, pp. 849-852.
[28] Oh, S.K., Kim, M.S., Eom, T.D., and Lee, J.J. (2005), Heterogeneous local model networks for time series prediction, Applied Mathematics and Computation, 168(1), 164-177. http://dx.doi.org/10.1016/j.amc.2004.08.018
[29] Pan, Z., and Wang, X. (1998) A stochastic nonlinear regression estimator using wavelets, *Computational Economics*, 11, 89-102. http://dx.doi.org/10.1023/A:1008670529487
[30] Percival, D. and Walden, A. (2000) Wavelet Methods for Time Series Analysis, *Cambridge University Press*, Cambridge. http://dx.doi.org/10.1017/CBO9780511841040
[31] Ramsey, J. B. and Zhang, Z. (1997) The analysis of foreign exchange data using waveform dictionaries, *Journal of Empirical Finance*, 4, 341-372. http://dx.doi.org/10.1016/S0927-5398(96)00013-8
[32] Ramsey, J. B. and Lampart, C. (1998) The decomposition of economic relationships by time scale using wavelets: Expenditure and income, *Studies in Nonlinear Dynamics and Econometrics*, 3, 23-42. http://dx.doi.org/10.2202/1558-3708.1039
[33] Schoelkopf, B., Burges, C. J. C., and Smola, A. J.(1999) Advances in kernel methods - support vector learning, MIT Press, Cambridge, MA.
[34] Vapnik, V. N. (1995) The Nature of Statistical Learning Theory. New York: Springer-Verlag. http://dx.doi.org/10.1007/978-1-4757-2440-0
[35] Yamada, H. (2005) Wavelet-based beta estimation and Japanese industrial stock prices, *Applied Economics Letters*, 12, 85-88. http://dx.doi.org/10.1080/1350485042000307152
[36] Yao, J. and Tan, C. L. (2000) A case study on using neural networks to perform technical forecasting of forex, *Neurocomputing*, 34, 79-98. http://dx.doi.org/10.1016/S0925-2312(00)00300-3
[37] Yousefi, S. and Weinreich, I. (2005), Dominik Reinarz Wavelet-based prediction of oil prices, *Chaos, Solitons and Fractals*, 25, 265-275. http://dx.doi.org/10.1016/j.chaos.2004.11.015
[38] Zhang, B. L., Coggins, R., Jabri, M. A., Dersch, D., and Flower, B. (2001), Multiresolution forecasting for futures trading using wavelet decompositions, IEEE Transactions on Neural Networks, 12(4), 765-775. http://dx.doi.org/10.1109/72.935090
[39] Zhang, G. and Hu, M. Y. (1998) Neural network forecasting of the British Pound/US Dollar exchange rate, *OMEGA: Int. Journal of Management Science*, 26, 495-506.
[40] Zimmermann, H., Neuneier, R. and Grothmann, R. (2001) Multi-agent modeling of multiple FX-markets by neural networks, *IEEE Trans. Neural Networks*, 12(4), 735-743. http://dx.doi.org/10.1109/72.935087
[41] Zadeh, L. (1994), Fuzzy Logic, Neural Networks, and Soft Computing, Communications of the ACM, 37(3), 77-84. http://dx.doi.org/10.1145/175247.175255