Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neurocomputing 55 (2003) 285–305

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Volatility forecasting from multiscale and high-dimensional market data

Valeriy V. Gavrishchaka*, Supriya B. Ganguli

Science Applications International Corporation, McLean, VA 22102, USA

Received 6 March 2002; accepted 31 March 2003

Abstract

Advantages and limitations of the existing volatility models for forecasting foreign-exchange and stock market volatility from multiscale and high-dimensional data have been identified. Support vector machines (SVM) have been proposed as a complimentary volatility model that is capable of effectively extracting information from multiscale and high-dimensional market data. SVM-based models can handle both long memory and multiscale effects of inhomogeneous markets without restrictive assumptions and approximations required by other models. Preliminary results with foreign-exchange data suggest that SVM can effectively work with high-dimensional inputs to account for volatility long-memory and multiscale effects. Advantages of the SVM-based models are expected to be of the utmost importance in the emerging field of high-frequency finance and in multivariate models for portfolio risk management.

Keywords: Volatility models; Support vector machines; High-frequency finance

1. Introduction

Predictive capabilities of the data-driven models of the systems with complex multiscale dynamics depend on the quality and amount of the available data and on the algorithm used to extract generalized mappings. Availability of the real-time, high-resolution data constantly increases in many fields of practical interest. However, the majority of advanced statistical and machine learning algorithms, including neural networks (NN), can encounter a set of problems called "dimensionality curse" when applied to

* Corresponding author.
E-mail addresses: valeriyg@alexandra.net, gavrish@apo.saic.com (V.V. Gavrishchaka), supriya@apo.saic.com (S.B. Ganguli).

0925-2312/03/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0925-2312(03)00381-3

high-dimensional data [4]. Nonstationarity of the system can also impose significant limitations on the size of a training set which leads to poor generalization ability of the model.

A very promising algorithm that can tolerate high-dimensional and incomplete data is support vector machine (SVM) [43,44]. SVMs have recently been receiving significant interest due to excellent results in various applications [10]. SVM combines the training efficiency and simplicity of linear algorithms with the accuracy of the best nonlinear techniques, and systematic approach for optimal generalization. In many practical applications SVMs can tolerate high-dimensional and/or incomplete data and often demonstrate performances superior to the best available techniques including NNs [10]. Note that in this article majority of the comparative references to NNs would imply multilayer perceptron (MLP) or similar algorithms and architectures [4,33]. Recent successful applications of SVM-based adaptive systems include image/object classification [32], face detection and recognition [30], text categorization [22], process identification in high-energy physics [42], cancer diagnostic and prognosis [26], gene classification [8], as well as many other scientific, engineering, medical, and biological applications.

Recently we have also applied SVM to a challenging problem of real-time space weather forecasting [20]. It has been shown that the performance of the SVM-based model for geomagnetic substorm prediction can be comparable (or superior) to that of the best existing models including NNs [19]. The advantages of the SVM-based techniques are expected to be much more pronounced in the next generation of the space-weather forecasting models, which will incorporate many types of high-dimensional, multiscale input data once real-time availability of this information becomes technologically feasible.

Financial time-series forecasting is another challenging area where advantages of the SVM-based systems could be very important. Although some financial applications of the SVM have been reported [13,16] the full range of potential SVM applications in finance remains largely unexplored. For example, there are no comprehensive studies of the SVM applications to volatility forecasting from multiscale and high-dimensional market data. Exception is a recent work by Van Gestel et al. [41] where new SVM formulation is introduced and applied to financial time series. Encouraging results of the volatility modeling of the daily DAX30 closing pricing have been reported [41].

Volatility of the foreign exchange and stock markets is a very important quantity for option pricing, value-at-risk (VaR) calculations used in portfolio risk management, and for general decision making in real-time trading systems. The empirically confirmed existence of volatility long memory (up to several months) may require high-dimensional inputs in the volatility models. Multivariate structure of the volatility and covariance models used in portfolio risk management further increases dimensionality of the model. Emerging new field of the high-frequency finance [11], dealing with multiscale market data (from several minutes to several months), imposes even more demanding requirements on the dimensionality of the multiscale volatility models.

In this paper we review stylized facts and features of the market data and existing volatility models. Limitations of the known volatility models, especially their ability to handle long memory and multiscale nature of the market data, are identified. SVM-based system is proposed as a complimentary multiscale volatility model.

Advantages and potential applications of the new model are discussed. Encouraging preliminary results of the SVM model application to volatility forecasting of foreign exchange market are reported. Although only foreign exchange market examples are considered in this paper, almost all discussion is relevant to stock market data as well. When necessary, specific differences between stock and foreign exchange markets are mentioned.

2. Data description: stylized facts of financial data

In this section we define the main measures used to characterize financial time series and describe their universal properties revealed in numerous empirical studies. A typical daily \$US/DM exchange rate that will be used in this paper is shown in Fig. 1a. Nonstationarity of the moving average of the time series is clear from this figure. The more practical quantity is the logarithm of return given by

$$r_i = \ln(X_i/X_{i-1}), \tag{1}$$

where i is an index of a homogeneous time sequence (e.g., the end of each trading day) and X_i is an exchange rate (or stock price) at time t_i. The daily return time series corresponding to Fig. 1a is shown in Fig. 1b. The moving average of the return time series is almost stationary and close to zero. Another important quantity of the financial time series is volatility. Optimal definition of the realized volatility depends on the particular application and properties of the time series of interest. In many cases realized volatility at time t_i is defined as a standard deviation of returns in some interval $[t_{i-n}, t_i]$. For purposes of this paper we consider realized volatility to be $v_i = |r_i|$ or $v_i = r_i^2$ that is a reasonable choice in many other applications as well [11].

Extensive empirical studies of the market data revealed several universal or stylized facts. Returns have been found to have only very short-range correlation with typical characteristic time of just a few trading minutes [7,11,27]. This absence of linear correlation is illustrated in Fig. 2 where dotted line represents the autocorrelation function of daily returns computed from \$US/DM exchange rate from 1/1/1980 to 1/1/2000. On the other hand volatilities (e.g., represented by absolute values of returns) are clustered and have long-range memory (up to several months) [7,11,27]. The volatility autocorrelation function exhibits hyperbolic (power-law) behavior. This is illustrated in Fig. 2 for \$US/DM exchange data (solid line). To avoid strong seasonal variations we took into account only week-day data in Fig. 2 and all subsequent examples in this paper. This is a transformation from physical to business time which is a simplified version of a more sophisticated transform called θ-time necessary for higher frequency data [11]. The other important fact is that probability density function (pdf) of returns is fat-tailed and leptokurtic at small time scales (from several minutes to several days) and approaches Gaussian at larger scales [7,11,27]. Volatilities have also been found to be negatively correlated with corresponding returns. This fact (called leverage effect) is more pronounced in stock markets (for both individual stocks and indices) and has been clarified in recent detailed empirical studies [6].

A number of recent studies with high-frequency data [2,11] revealed essential heterogeneity of the market and multiscale nature of the market dynamics. Discovered

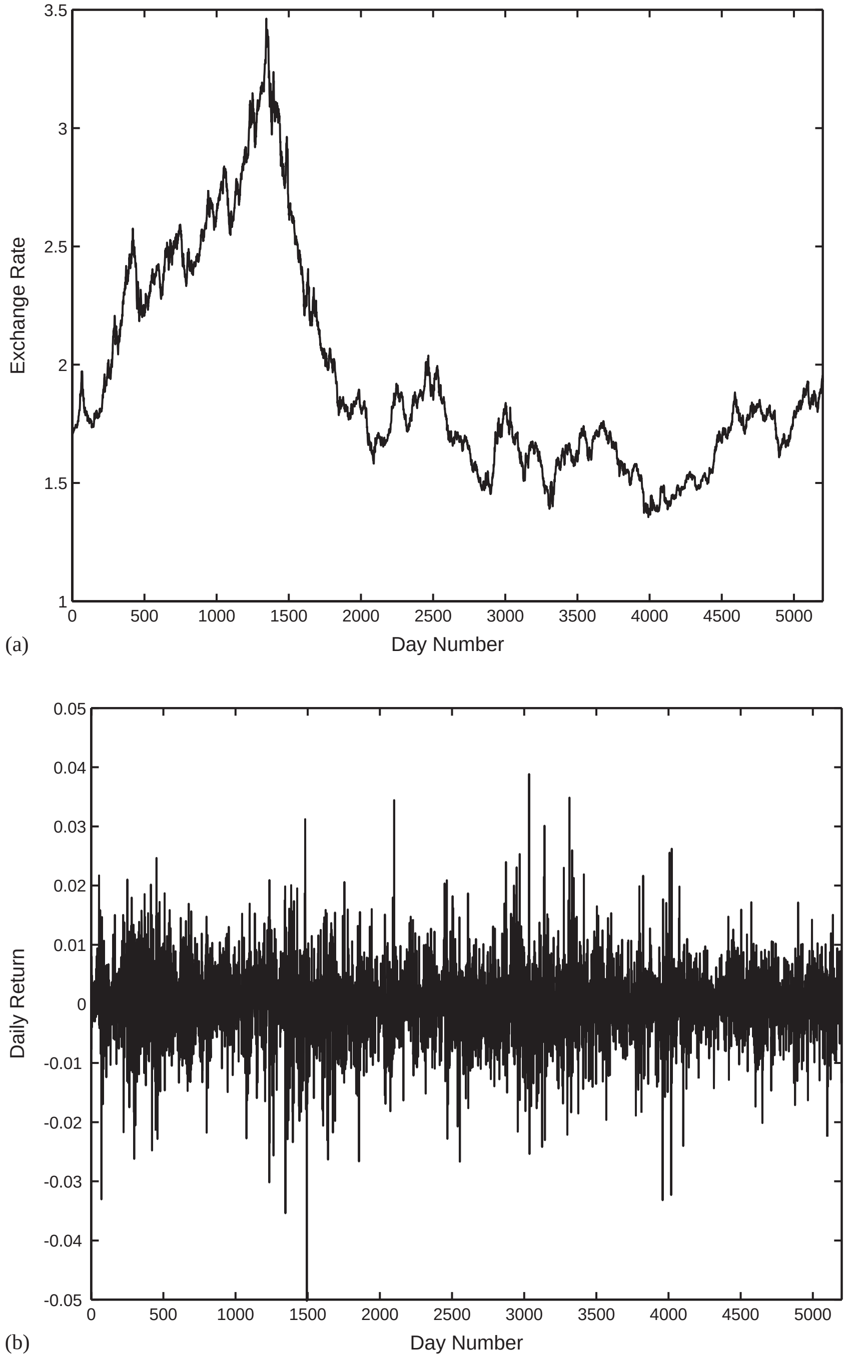


Fig. 1. (a) $US/DM exchange rate and (b) daily returns from 1/1/1980 to 1/1/2000 (weekends excluded).

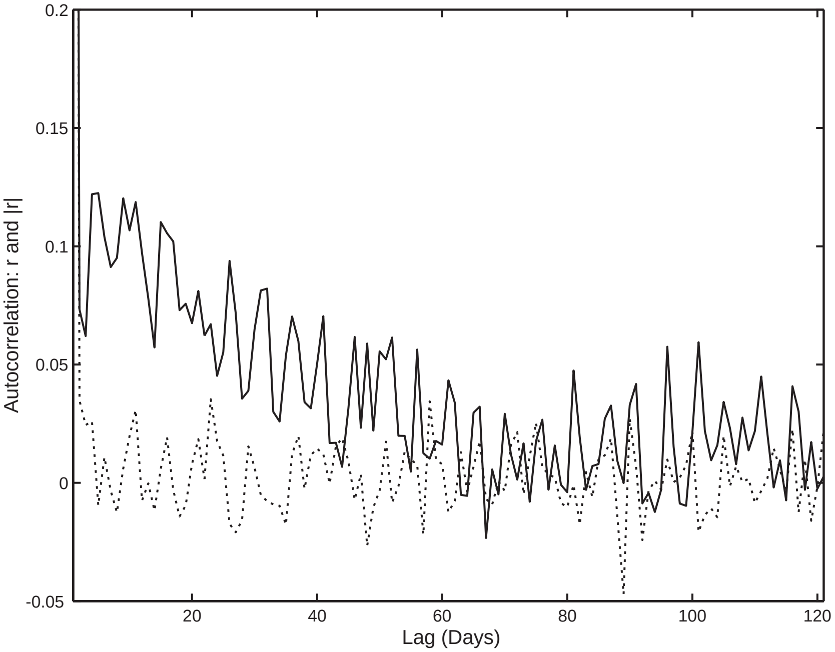


Fig. 2. Autocorrelation fuction of daily returns (dotted line) and absolute returns (solid line) as a function of lag in days. $US/DM exchange rate data from 1/1/1980 to 1/1/2000 are used (weekends excluded).

properties of multiscale volatilities have been found to be very important to understand market structures. Contrary to the assumption of a homogeneous market where all participants interpret news and react to them in the same way, the hypothesis of a heterogeneous market assumes that different market agents have different time horizons and dealing frequencies (from intraday dealers or market makers to central banks and large commercial organization). In this framework, hyperbolic decline of the volatility correlations is interpreted as a superposition of exponential memories of the market components with a wide range of time constants. Also in a heterogeneous market different agents are likely to settle for different prices and decide to execute their transactions in different market situations, i.e., they create volatility. This is unlike the homogeneous market where more agents mean faster price convergence and smaller volatility. Empirical studies clearly indicate positive correlation of volatility and market presence that supports heterogeneous market hypothesis [11].

Heterogeneous market hypothesis suggests that traders with different time horizons are interested in the volatility on different time grids. A coarse time grid reflects the view of a long-term trader and a fine time grid that of a short-term trader. The "coarse" (v^c) and "fine" (v^f) volatilities can be defined as

$$v^c(t_i) = \left| \sum_{j=1}^{n} r(\Delta t^*, t_{i-1} + j\Delta t^*) \right|, \tag{2}$$

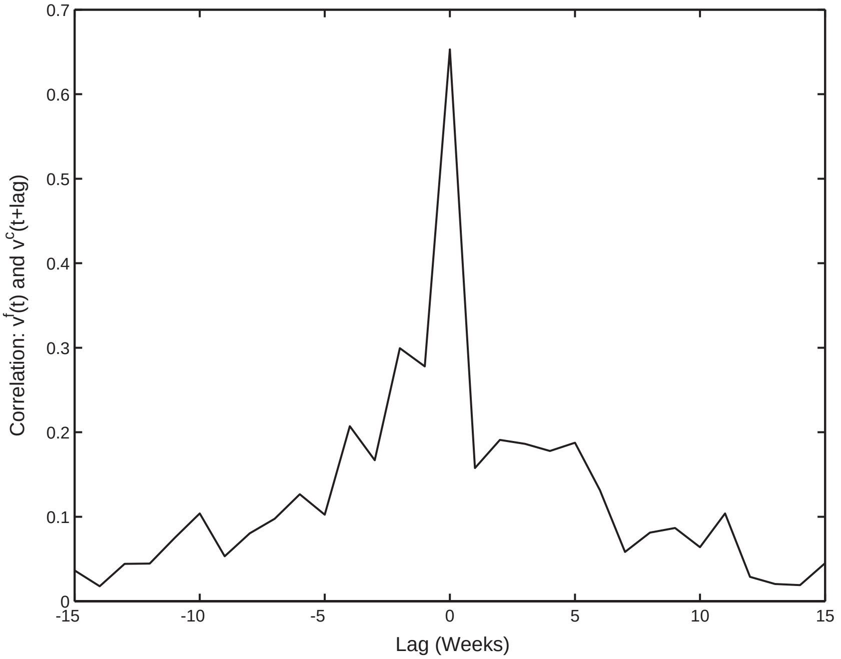


Fig. 3. Lead-lagged correlations of the fine and coarse-grain volatilities as a function of positive/negative lag in weeks. Volatilities are computed on a weekly interval from daily returns. $US/DM exchange rate data from 1/1/1980 to 1/1/2000 are used (weekends excluded).

$$v^f(t_i) = \sum_{j=1}^{n} |r(\Delta t^*, t_{i-1} + j\Delta t^*)|, \tag{3}$$

where $\Delta t^* = \Delta t/n$, $\Delta t = t_i - t_{i-1}$, the first return argument is the time scale over which return is computed, and the second argument is the time of this return measurement. For example, if we consider weekly volatility measures (on business time scale), then v^c is given by $|\sum_{i=1}^{5} r_i|$ and v^f by $\sum_{i=1}^{5} |r_i|$, where r_i is a daily return at the ith day.

An important effect found for both foreign exchange [11] and stock [2] markets is asymmetric lead-lag correlation of volatilities. Lagged correlation is a linear correlation of the two time series one of which is shifted (lagged) in time. Lagged correlation reveals causal relations and information flow structures in the market. To illustrate effect of asymmetric volatility correlations we consider fine volatility defined by averaged absolute returns over five working days and coarse volatility defined as absolute return over a full (working) week (i.e., $n = 5$ in Eqs. (2) and (3)). Lead-lagged correlations of these volatilities obtained from $US/DM exchange rate (from 1/1/1980 to 1/1/2000) are shown in Fig. 3. We see a clear asymmetry: the coarse volatility predicts fine volatility better than the other way around, i.e., information flows from large to small scales. This is consistent with heterogeneous market hypothesis since short-term traders can react to clusters of coarse volatility, while the level of fine volatility does not affect strategies of long-term traders.

In the next section we review some of the existing volatility models. Although there is no universal volatility model that incorporates or explains all of the stylized market facts, different models focus on different set of features that finally determines their accuracy and applicability scope.

3. Existing volatility models and their limitations

There are two general classes of volatility models in widespread use: deterministic and stochastic models. Deterministic models consider volatility (conditional variance) to be a deterministic function of the past returns (and/or other observables) that are described by some stochastic process (e.g., Wiener process). Stochastic volatility models describe volatility by its own stochastic process. Below we give a short overview of the mentioned volatility models and their limitations. Less universal volatility models such as recently introduced "model-free" approach based on realized volatility computed with high-frequency data [1] will not be discussed here.

A common example of the deterministic volatility models is autoregressive conditional heteroskedastic (ARCH)-type models [5,14]. These models assume a particular stochastic process for the returns and a simple functional form for the volatility. Volatility in these models is unobservable (latent) variable. The most widely used model of this family is generalized ARCH (GARCH) process [5]. GARCH(p,q) process defines volatility as

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2, \tag{4}$$

where return process defined as

$$r_t = \sigma_t \varepsilon_t. \tag{5}$$

Here ε_t is an identically and independently distributed (i.i.d.) with zero mean and variance 1. The most common choice for the return stochastic model (ε_t) is a gaussian (Wiener process). However, to take into account realistic fat-tailed return distributions, GARCH model is also used with Student-t distribution of returns. Parameters α_i and β_i from equation are estimated from historical data by maximizing the likelihood function (LF) which depends on the assumed return distribution. A typical implementation of the GARCH model uses maximization of the LF by local gradient methods or by their combination with robust nonlocal optimization techniques such as genetic algorithms (GA) [11].

GARCH and other ARCH-type processes is the most common choice of the volatility model for both option pricing and portfolio risk management (VaR calculations). GARCH process can reproduce a number of known stylized volatility facts including mean reverting. Explicit specification of the stochastic process and simplified (linear) functional form for the volatility allows to do simple analysis of the model properties and its asymptotic behavior. However assumptions of the ARCH-type models also impose significant limitations. For example, GARCH(p,q) model does not cover leverage and general nonlinear effects. Model parameter calculation from the market data is practical only for low-order models (small p and q), i.e., in general, it is difficult to capture direct long memory effects. Volatility multiscale effects are not covered (see discussion in the next section). Finally, the model gives unobservable quantity that leads to difficulty in quantifying the prediction accuracy and comparison with other models. Some of these restrictions are relaxed in the GARCH model extensions. For example, in threshold GARCH (TGARCH) [45] leverage effect is taken into account in a simplified form. However majority of the mentioned limitations cannot be resolved in a self-consistent fashion.

A number of nonlinear extensions of the ARCH-type framework have been proposed. One of the advantages of the true nonlinear volatility model is an adequate modeling of the leverage effect that is not modeled accurately by the GARCH extensions. Donaldson and Kamstra [12] proposed NN-based volatility model. They found that a proper modeling of non-linearities captures volatility effects that are overlooked by traditional models like GARCH and its extensions.

Schittenkopf et al. [34,35] added a detailed analysis of the distributional assumptions underlying NN-based volatility models. They found that models with non-gaussian distributions (mixture of gaussians or Student-t) are superior to those with gaussian distributions. This is due to heteroskedastic nature of the financial time series and fat-tail nature of return distribution. Non-gaussian (mixture of gaussians) models have been formulated as mixture density NNs [4] where appropriate generalization of a simple gaussian loss function (mean squared error) is made. In some regimes, mixture density NNs have been shown to perform significantly better than GARCH-type models. Potential limitations of the NN-based models can be related to high-dimensional inputs ("dimensionality curse" [4]) in such applications as small-scale volatility forecasting.

In a stochastic (latent) volatility models, volatility is not function purely of observables (returns) and is described by its own stochastic process. Stochastic volatility approach gives more flexibility than ARCH-type models that becomes important when true volatility is high and ARCH-type model cannot provide enough mixing. Stochastic volatility provides more mixing because observable and volatility shocks are imperfectly correlated with one another. One of the simplest and effective approaches models volatility as Ornstein–Ulenbeck process which correlates with return stochastic process to account for leverage effect [28].

As discussed in the previous section the volatility autocorrelation function is characterized by hyperbolic (power-law) decay. A stochastic process that exhibits a hyperbolic decay in its autocorrelation function is fractional Brownian motion introduced by Mandelbrot and Van Ness [25]. This process is an extension of a Brownian motion (Wiener process). Unlike Wiener process fractal Brownian motion has memory which makes it attractive for stochastic volatility modeling. For example stochastic volatility framework where volatility is described by Ornstein–Ulenbeck process driven by fractional Brownian motion has been developed and applied to options on futures contracts [23].

Fractional Brownian motion mentioned above is an example of a mono-fractal process, since the scaling exponent of this process is a linear function of the moment order. However, recent studies of the market data suggest that this is not always the case, i.e., function is nonlinear [36,37]. This empirical fact and analogy with multiplicative cascade theory of the developed turbulence [18] resulted in volatility models based on multi-fractal processes in the form of multiplicative cascades [21,29]. Another multifractal approach is based on using simple mono-fractal process such as fractional Brownian motion on a multi-fractal trading time [17,24]. The latter approach is inspired by empirical analysis of the transaction times.

Stochastic volatility models are much more flexible than ARCH-type and similar deterministic models. They can account for more empirical properties of the volatility dynamics. However it is significantly more difficult to analyze these models and make reliable estimation of all their free parameters from the available market data. Therefore stochastic volatility models are not yet widely accepted in real business applications.

In the following sections we will discuss only deterministic volatility models that can provide significant advantage over the standard ARCH-type approaches.

4. Multiscale volatility models for heterogeneous market

One of the most significant limitations of the existing ARCH-type and similar deterministic volatility models is their inability to capture the heterogeneity of traders acting at different time horizons. For example, if the empirical data can be described as generated by one GARCH process at one particular data frequency, the dynamics of the data sampled at any other frequency is theoretically determined by temporal aggregation (or disaggregation) of the original process. These derived processes at different frequencies can be compared to empirically estimated processes at the same frequencies. Significant deviation between theoretical and empirical results reject hypothesis of only one GARCH process responsible for data generation [11,15]. In other words model parameters obtained for the data of different frequencies are significantly different. It means that there is more than one relevant frequency in the volatility generation. This is manifestation of the presence of many independent volatility components in the data, i.e., the signature of market heterogeneity.

As discussed earlier there is asymmetry in the interaction between volatilities measured at different frequencies (see Fig. 3). A coarsely defined volatility predicts a fine volatility better than the other way around. This effect is not present in a simple GARCH model. More complex types of ARCH models have to be developed to account for the heterogeneity that is especially pronounced in high-frequency data. One of such approaches is the Heterogeneous Autoregressive Conditional Heteroskedasticity (HARCH) model [11].

The HARCH process has a variance equation based on multiscale returns, i.e., returns computed over time intervals of different sizes

$$\sigma_t^2 = c_0 + \sum_{j=1}^{n} c_j \left(\sum_{i=1}^{j} r_{t-i} \right)^2 , \tag{6}$$

where return process is still given by (5) and c_j are parameters of the model. The terms of (6) reflect the component structure of the market in a natural way. HARCH model is rather different from the typical ARCH model. For example HARCH(2) model can be rewritten in two forms:

$$\sigma_t^2 = c_0 + c_1 r_{t-1}^2 + c_2(r_{t-1} + r_{t-2})^2, \tag{7}$$

or

$$\sigma_t^2 = c_0 + (c_1 + c_2) r_{t-1}^2 + c_2 r_{t-2}^2 + 2c_2 r_{t-1} r_{t-2}. \tag{8}$$

The last form (8) can be identified as ARCH(2) model plus an important mixed term $r_{t-1}r_{t-2}$, i.e., signs of returns matter.

HARCH can reproduce empirical behavior of lagged correlations as well as the long memory of volatility. This is a qualitative difference between GARCH model and its variations. For example fractionally integrated GARCH (FIGARCH) process [3] has been designed to model the long memory but cannot reproduce the lead-lag correlations since it is still based on returns measured over one time scale.

Although HARCH model is able to capture multiscale nature of volatility, application of the HARCH model in its original form can be computationally prohibitive especially for high-frequency volatility. This is due to many free parameters (corresponding to different market components) that need to be estimated from the market data. For example modeling of the intraday volatility can easily result in hundreds of free parameters since small-scale volatility depends on many larger scale volatilities. To make HARCH model practical, additional restriction on the number of independent market components has to be applied. This is done by clustering adjacent components and assuming the coefficients c_j to be equal across the same cluster. No more than 7 clusters (components) are usually considered [11].

In the next section we describe our SVM-based approach as a complimentary multiscale volatility model that can relax a number of restrictive assumptions of the ARCH/HARCH models including limitation on the number of independent market components.

5. Multiscale volatility model based on support vector machine

SVMs developed by Vapnik [43,44] have recently been receiving significant interest due to excellent results in various applications [10]. We do not intend to give a detailed introduction to SVM in this paper and refer readers to excellent books and papers on this topic (e.g., see [10], and references therein). Also we provided a short introduction to the main ideas used in SVM in our recent paper [20]. Here we give only a brief description of the SVM and its main advantages.

SVM is a combination of a kernel-based approach and a structural risk minimization (SRM) principle [43,44]. First step is a nonlinear mapping from the input to a higher-dimensional feature space. Kernel-based approach allows to represent the discriminant function in high-dimensional feature space without explicit dependence on the feature space dimensionality. Kernel-based machine decouples the number of free parameters (related to the machine capacity) from the size of the input space which can be very large or even infinite. SRM provides solid theoretical grounds for optimization of the SVM generalization ability that is often superior to other approaches used in machine learning algorithms.

In general, training of the SVM for classification and support vector regression (SVR) reduces to the minimization problem with constraints that is a typical quadratic programming problem [9]. Application of the SVR also involves finding adequate loss function. Loss function should not only be able to correctly approximate noise distribution of the modeled data but also have a suitable form for optimization algorithm used in a particular SVR implementation. The most common choice is the original ε-insensitive loss function (ε-ILF) [10,43,44] which is similar to loss functions used in the field of robust statistics. It has been shown [31] that the use of ε-ILF is justified under assumption that the noise is a superposition of Gaussian processes. This noise model is quite suitable for heteroskedastic market data we are interested in, and ε-ILF will be used in our volatility model.

The optimization problem for the ε-SVR is given by

$$\min_{\alpha,\alpha^*}\left[\frac{1}{2}(\alpha-\alpha^*)^{\mathrm{T}}Q(\alpha-\alpha^*)+\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)\right],$$

$$\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)=0,$$

$$0\leqslant\alpha_i,\quad \alpha_i^*\leqslant C,\quad i=1,\ldots,l. \tag{9}$$

Here $C>0$ is a regularization parameter (soft margins), $(y_i,\mathbf{x}_i)$ is a training set, l is a number of training samples, $Q_{ij}\equiv y_iy_jK(\mathbf{x}_i,\mathbf{x}_j)$ is a positive semidefinite matrix, and K is a kernel function representing inner product of the feature vectors. ε-ILF is given by $L^{\varepsilon}(x,y,f)=|y-f(x)|_{\varepsilon}=\max(0,|y-f(x)|-\varepsilon)$, where $f=\sum_{i=1}^{l}(-\alpha_i+\alpha_i^*)K(x_i,x)+b$, $x\in R^n$, $y\in R^1$, and $b\in R^1$ is a constant. Approximation function f is equivalent to the hyperplane in the feature space implicitly defined by the kernel K that solves the optimization problem (9).

Although SVM training is a typical quadratic programming problem, due to the specifics of SVM applications such as large data sets and high density of the Q-matrix, standard algorithms can become impractical. Recent developments mainly include algorithms that employ various decomposition techniques [10], where at any time a fixed size subset of α_i is updated, while others are kept constant. Various heuristics are used for choosing a working set at each step. Here we use an algorithm described by Chang et al. [9] and implemented as LIBSVM library (www.csie.ntu.edu.tw/cjlin/libsvm).

Applicability of the SVM (SVR) model to our problem is based on the assumption that volatility σ can be described as a nonlinear function F of a time series of returns r:

$$\sigma_i^2=F[r_{i-1},r_{i-2},\ldots,r_{i-n}], \tag{10}$$

where index $i-j$ correspond to time $(t_i-j\,\mathrm{d}t)$, $\mathrm{d}t$ is a time lag interval and $T=n\,\mathrm{d}t$ is a total length of the memory for previous inputs. Since F could be any nonlinear function this framework automatically covers multiscale dependencies in a more general form than HARCH model. A framework similar to original HARCH framework can be useful to directly study influence of different market components. This is given by

$$\sigma_i^2 = F\left[r_{i-1}, (r_{i-1} + r_{i-2}), \ldots, \left(\sum_{j=1}^{n} r_{i-j}\right)\right], \tag{11}$$

where n is a number of the market multiscale components.

Practical usage of the described SVM models requires specification of the volatility σ in (10) and (11). In this paper we adopt the most common choice as $\sigma_i^2 = r_i^2$. However, in general, other volatility measures can also be easily used in the described framework and will be considered in our future work. For example, SVM can be trained on σ_i time series that is calculated using intraday return data from day i [1].

We need also to ensure that trained SVM model will always output non-negative numbers for σ^2. This is achieved by choosing mapping function as

$$\sigma_i^2 = exp(F[r_{i-1}, r_{i-2}, \ldots, r_{i-n}]). \tag{12}$$

In operation terms it means that SVM is trained on $\ln(r^2)$ instead of r^2, and *exp* mapping is applied to the SVM output.

Since the main advantage of the SVM is its ability to handle high-dimensional data, SVM-based volatility model can model long memory and multiscale effects without restrictive assumptions required by other model. For example, unlike HARCH model SVM will not require strict limitations on the number of independent market components.

6. Results

Building a full featured SVM-based volatility model that can be useful in real trading infrastructure is beyond the scope of this paper. Therefore extensive comparison of the SVM model and other available volatility models will be done in our future articles. Here we illustrate the ability of the SVM-based volatility model to handle challenge of the long-memory and multiscale effects of the real market data and present preliminary comparison with two basic models.

As an example we still use $US/DM exchange rate. Steps of our analysis include the choice of a 670 day time window from the exchange data covering the period from 1/1/1980 to 1/1/2000. To demonstrate sensitivity of the model performance to training and test data, we also consider several time windows shifted from the base window with a step of 5 business days. First 540 days of data in each window are used for training and validation in a standard 5-fold cross-validation procedure. Remaining 130 days of data are used as test sets. Cross-validation procedure is used to optimize SVM parameters such as regularization parameter C, ε-parameter of the loss function, coefficients of the kernel function, and the type of the kernel function itself. Optimization is performed with respect to a linear correlation coefficient between model outputs and corresponding real data. Final conclusions of the model performance are based on the results obtained on the test set.

As mentioned in the previous section SVM model in form (10) incorporates both long memory and multiscale effects. To demonstrate that SVM can efficiently extract

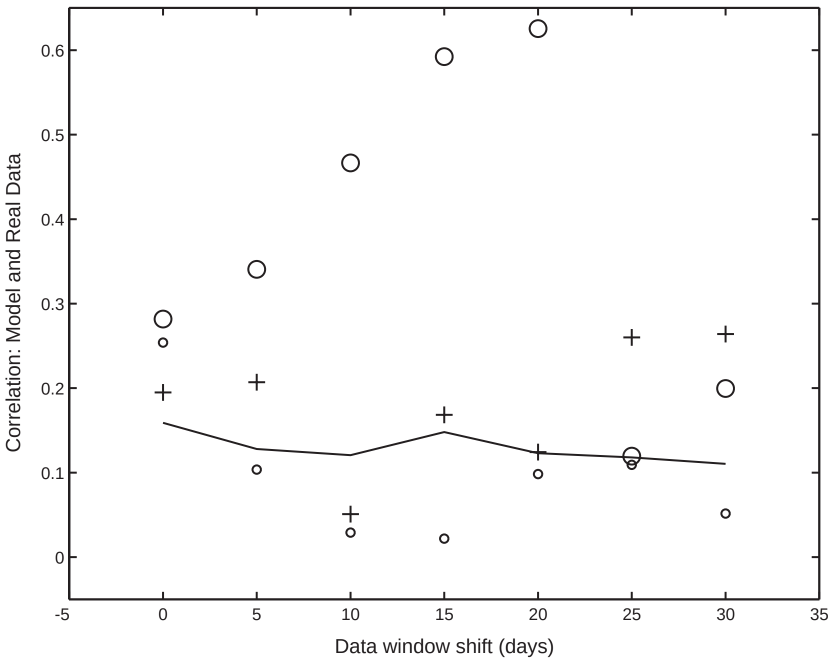


Fig. 4. Linear correlation of real and model volatilities for different data windows shifted from a base window by a variable number of business days. Large and small circles represent SVM model with 15 and 4 inputs, respectively. Benchmark models are shown by a solid line (naive model) and crosses (GARCH(1,1) model).

information from lagged return vectors of high-dimension we train SVMs with small and large number of lagged returns as inputs. In Fig. 4 correlation measure of real and model volatilities for SVM model with 4 (small circles) and 15 (large circles) inputs is shown for several data sets (shifted time windows). Here we used radial basis function kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2)$, where γ is a constant. Optimal values for parameters C, ε, and γ, obtained from 5-fold cross-validation procedure, vary with data set. For data sets (windows) considered in Fig. 4, optimal values are the following: $10 < C < 20, 0 < \varepsilon < 0.01$, and $0.1 < \gamma < 0.7$. Large values of C indicate large noise level that is typical for market data.

It is clear that SVM with large number of inputs demonstrates superior performance. Due to existence of long term memory and multiscale effects, an algorithm, whose ability of efficient information extraction does not significantly change with input dimensionality, should produce noticeable improvement in forecasting when number of used lagged returns is increasing. Fig. 4 demonstrate that this effect is observed in our case. This illustrates that SVM can effectively extract information from high-dimensional inputs to improve volatility forecasting with respect to the models of lower dimensionality.

In Fig. 4 we also compare SVM performance to that of naive and GARCH(1,1) models (solid line and crosses, respectively). Naive model uses previous step as prediction (i.e., $r_i^2 = r_{i-1}^2$). GARCH coefficients in (4) are estimated in a standard manner

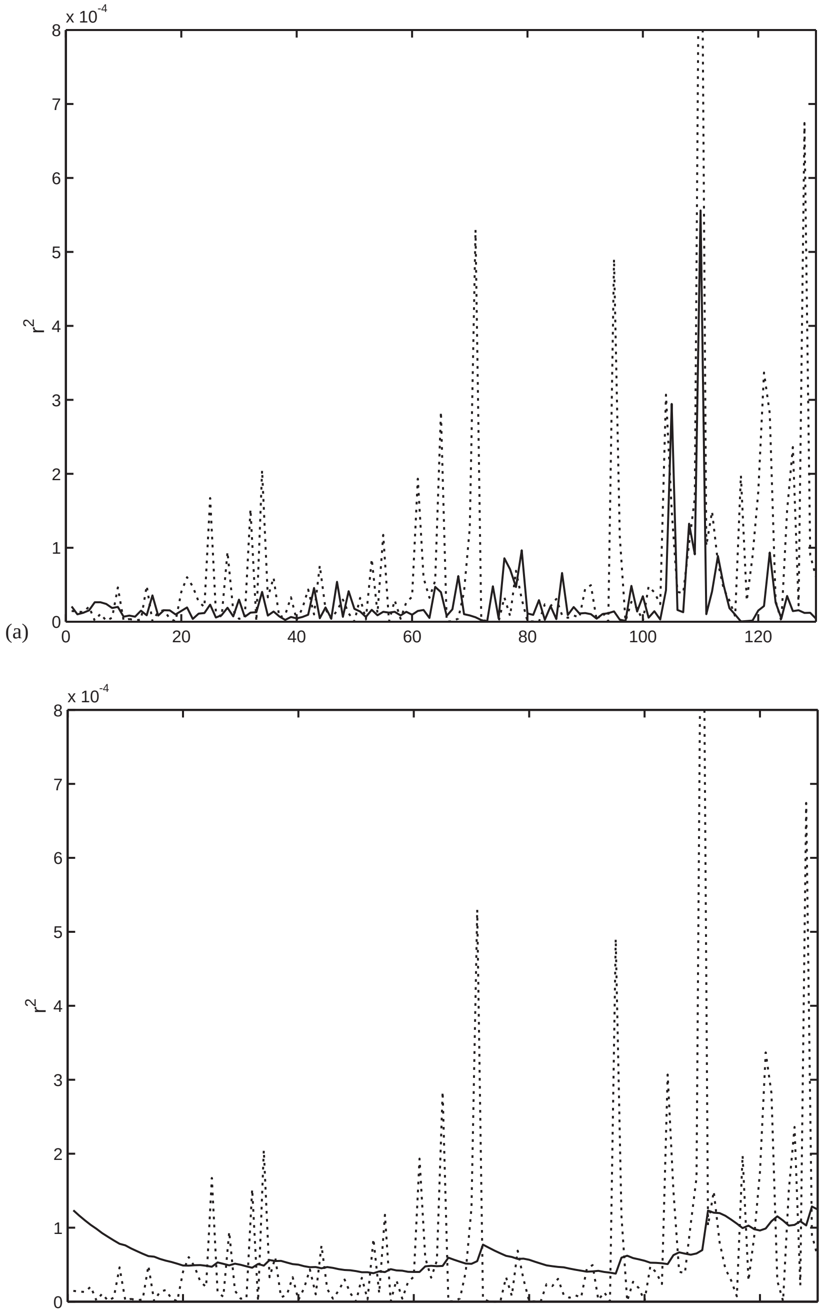


Fig. 5. Original (dotted line) and predicted (solid line) r^2 time series obtained from (a) SVM and (b) GARCH(1,1) models. Shift from the base data window is 10 business days.

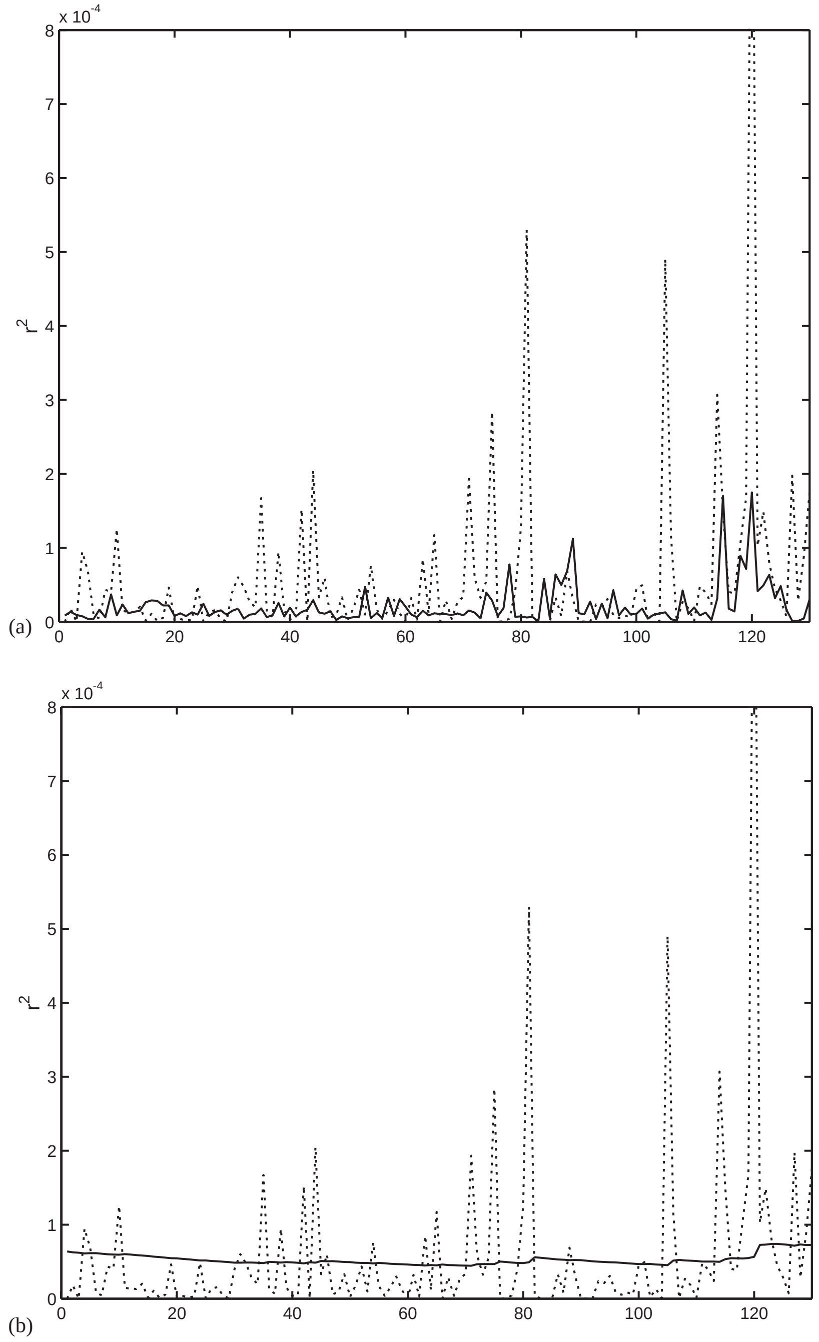


Fig. 6. Original (dotted line) and predicted (solid line) r^2 time series obtained from (a) SVM and (b) GARCH(1,1) models. Base data window is used.

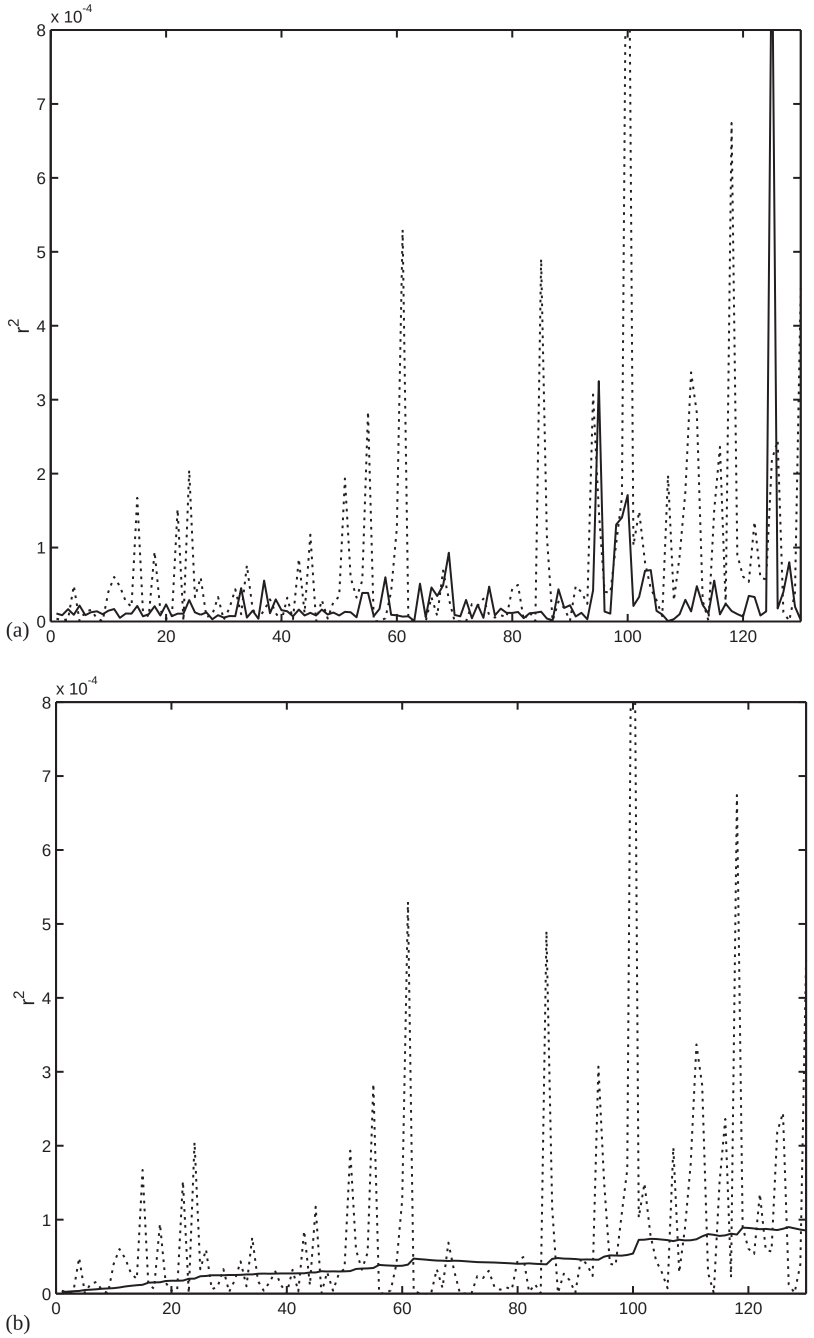


Fig. 7. Original (dotted line) and predicted (solid line) r^2 time series obtained from (a) SVM and (b) GARCH(1,1) models. Shift from the base data window is 20 business days.

by maximizing LF calculated from training data [40]. It should be mentioned that GARCH outputs latent (unobservable) variable. Therefore measuring GARCH model performance with respect to realized r^2 time series is an approximation frequently used in practice [40].

It is clear that 15-input SVM consistently outperforms naive model and significantly outperform GARCH model in many cases. It is also clear that performance of both SVM and GARCH model are quite sensitive to the data set used for training and testing. We also found that when, instead of 5-fold cross-validation procedure, test data are directly used as validation set to optimize SVM parameters, SVM performance becomes more stable. It suggests that a better validation procedure may further improve performance of the SVM volatility model. Similar effects are observed with SVM given by (11). This configuration is less general but directly relevant to multi-scale HARCH model. The detailed comparison of this model with HARCH and related models will be given in a future article.

To understand details of the prediction capabilities of the SVM volatility model it is useful to compare predicted r^2 time series with the real one. Three typical cases are shown in Figs. 5(a), 6(a), and 7(a). The dotted and solid lines represent real and predicted r^2 time series, respectively. For comparison, corresponding time series from the GARCH(1,1) model (solid line) are shown in Figs. 5(b), 6(b), and 7(b).

Fig. 5(a) illustrate a typical case where SVM shows exceptional ability to model large-amplitude volatility fluctuations. This is especially impressive compared to GARCH time series in Fig. 5(b). Accurate prediction of the large-amplitude volatility events is one of the most important requirements in many financial applications including risk management and optimization of trading strategies. Fig. 6(a) illustrate other typical case where modeling of the large-amplitude events are less accurate but still significantly better than GARCH model predictions. Fig. 7(a) illustrates a case where SVM demonstrates quite accurate prediction of some large fluctuations while significantly overestimates other events. Note that majority of the volatility models (including GARCH) usually underestimates large volatility fluctuations.

7. Discussion and conclusion

In this paper, we addressed the problem of volatility forecasting from high-dimensional and multiscale market data. SVM-based model was proposed as a possible complimentary approach to volatility forecasting. SVM combines the learning effectiveness of linear machines with the classification/regression power of the best nonlinear algorithms. Unlike typical nonlinear techniques such as NNs, the size of the SVM input space is decoupled from the number of free parameters and allows one to process high-dimensional data without encountering the “high-dimensionality curse”. This makes SVM a possible model for processing real-time multiscale and high-frequency market data. SVM tolerance to incomplete data is another advantage of the SVM-based volatility model that can address the problem of the market data nonstationarity.

We reviewed the most important features of the foreign exchange and stock market data and existing volatility models. Model limitations in describing volatility dynamics and ability to extract information from high-dimensional and multiscale historical market data have been identified. Adequate description of such important volatility features as long term memory and asymmetric lead-lag correlation of volatilities (i.e., asymmetric information flow from large to small scales) leads to increasing dimensionality of the model and is one of the most challenging problems. Most of the existing approaches address this problem with rather restrictive assumptions to make the model computationally practical. These restrictions include limiting memory size, disregarding multiscale volatility effects, and limiting number of independent market components in some multiscale volatility models.

SVM's ability to handle high-dimensional and incomplete data allows to significantly relax those restrictions in the SVM-based volatility model introduced in this paper. Since this model imposes no significant restrictions on the length of the lagged vector of input parameters (memory size) and on the number of independent multiscale volatilities (market components), SVM model will allow to study parameter regimes where other existing models become computationally unrealistic. Besides that SVM models can automatically include such effects as volatility dependence on the sign of the return (which is required to cover leverage effect) and general nonlinear effects that are not covered by the models currently used in practice.

Our preliminary results with $US/DM exchange rate indicate that SVM model can efficiently extract information from the inputs with large number (up to 30) of multi-scale volatilities and/or high-dimensional vector of lagged returns that is computationally prohibitive for the most of the existing models. Our preliminary benchmark tests indicate that SVM can perform significantly better or comparable to both naive and GARCH(1,1) models. The advantages of the SVM-based techniques are expected to be much more pronounced in modeling small-scale (intraday) volatilities and high-frequency financial data.

Our future work will include more detailed studies of the SVM-based volatility model using larger data sets, both foreign exchange and stock market data, and extensive comparison of the SVM model accuracy with that of other models. We will also perform an extensive search of the optimal SVM algorithm and more advanced validation procedures. For example, in this paper we used standard ε insensitive loss function. However this may not be an optimal loss function for this application. Recently Edelman [13] proposed a new loss function that performs better on a high-noise data typical for financial time series. Van Gestel et al. [41] introduced new SVM formulation that can also be an effective approach for volatility modeling.

Although proposed SVM-based volatility model may be superior to other approaches in a number of practically important cases, the standalone usage of this model in a trading or risk management system may not be realistic. This is because many financial practitioners prefer to deal with models whose operations can be understood analytically (at least in asymptotic limits) and this is not possible in a "black-box" machine learning system. Although recent research efforts to provide explanation facilities by generating decision trees or rule bases from NN-based or other systems is quite successful [38,39], simple analytical models are still the most popular ones.

Practical application of the SVM volatility model as a complimentary approach is more realistic. For example, it can be used in parallel with one of the popular volatility models (e.g., GARCH). When the difference in the forecasted volatility value of the two models exceeds prespecified threshold, the value from the simple base model will not be used but instead more sophisticated decision will be made. There is also a possibility of SVM volatility model usage as a component in multiple experts framework or committee machine [4].

Acknowledgements

This work is supported by Science Applications International Corporation. We thank all referees evaluated this paper for valuable comments and suggestions.

References

[1] T.G. Andersen, T. Bollerslev, F.X. Diebold, P. Labys, Exchange rate returns standardized by realised volatility are (nearly) Gaussian, NBER Working Paper No: 7488.

[2] A. Arneodo, J.-F. Muzy, D. Sornette, Direct causal cascade in the stock market, Eur. Phys. J. B 2 (1998) 277.

[3] R.T. Baillie, T. Bollerslev, H.-O. Mikkelsen, Fractionally integrated generalized autoregressive conditional heteroskedasticity, J. Econometrics 74 (1996) 3.

[4] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[5] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, J. Econometrics 31 (1986) 307.

[6] J.-P. Bouchaud, A. Matacz, M. Potters, Leverage effect in financial markets: the retarded volatility model, Phys. Rev. Lett. 87 (2001) 228701.

[7] J.-P. Bouchaud, M. Potters, Theory of Financial Risk: From Statistical Physics to Risk Management, Cambridge University Press, Cambridge, 1999.

[8] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, M. Ares Jr., D. Haussler, Support vector machine classification of microarray gene expression data, Technical report UCSC-CRL-99-09, University of California, Santa Cruz.

[9] C.-C. Chang, C.-W. Hsu, C.-J. Lin, The analysis of decomposition methods for support vector machines, IEEE Trans. Neural Networks 11 (2000) 1003.

[10] N. Cristianini, J. Shawe-Taylor, Introduction to Support Vector Machines and other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, 2000.

[11] M.M. Dacorogna, R. Gencay, U. Muller, R.B. Olsen, O.V. Pictet, An Introduction to High-Frequency Finance, Academic Press, San Diego, 2001.

[12] R.G. Donaldson, M. Kamstra, An artificial neural network-GARCH model for international stock return volatility, J. Empirical Finance 4 (1997) 17.

[13] D. Edelman, Enforced-denial support vector machines for noisy data with applications of financial time series forecasting, in: Proceedings of the International Conference on Statistics, Combinatorics and Related Areas and the Eighth International Conference of Forum for Interdisciplinary Mathematics, School of Mathematics and Applied Statistics, University of Wollongong, Wollongong, NSW 2522, Australia, 19–21 December 2001.

[14] R.F. Engle, Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation, Econometrica 50 (1982) 987.

[15] R.F. Engle, A.J. Patton, What good is a volatility model? Quantitative Finance 1 (2001) 237.

[16] A. Fan, D. Hong, M. Palanaswami, C. Tan, A support vector machine approach to bankruptcy prediction: a case study, The 6th International Conference on Computational Finance, New York, January 1999.

[17] A. Fisher, L. Calvet, B.B. Mandelbrot, Multifractality of the DM/US dollar exchange rate, Cowles Foundation Discussion Paper, 1997.

[18] U. Frish, Turbulence, Cambridge University Press, Cambridge, 1995.

[19] V.V. Gavrishchaka, S.B. Ganguli, Optimization of the neutral-network geomagnetic model for forecasting large-amplitude substorm events, J. Geophys. Res. 106 (2001) 6247.

[20] V.V. Gavrishchaka, S.B. Ganguli, Support vector machine as an efficient tool for high-dimensional data processing: application to substorm forecasting, J. Geophys. Res. 106 (2001) 29911.

[21] S. Ghashghaie, W. Breymann, J. Peinke, P. Talkner, Y. Dodge, Turbulent cascades in foreign exchange markets, Nature 381 (1996) 767.

[22] T. Joachims, Text categorization with support vector machines, in: Proceedings of European Conference on Machine Learning (ECML), 1998.

[23] N. Lordkipanidze, Modeling and estimation of long memory in stochastic volatility: application to options on futures contracts, Working Paper, Cornell University, 2001.

[24] B.B. Mandelbrot, Sci. Am. 280 (1999) 70.

[25] B.B. Mandelbrot, J.W. Van Ness, Fractional brownian motions, fractional noises and applications, SIAM Rev. 10 (1968) 422.

[26] O.L. Mangasarian, W.N. Street, W.H. Wolberg, Breast cancer diagnosis and prognosis via linear programming, Oper. Res. 43 (4) (1995) 570.

[27] R.N. Mantegna, H.E. Stanley, An Introduction to Econophysics: Correlation and Complexity in Finance, Cambridge University Press, Cambridge, 2000.

[28] J. Masoliver, J. Perello, A correlated stochastic volatility model measuring leverage and other stylized facts, cond-mat/0111334 v1, November 2001.

[29] J.F. Muzy, J. Delour, E. Bacry, Modelling fluctuations of financial time series: from cascade process to stochastic volatility model, cond-mat/0005400, May 2000.

[30] E. Osuna, R. Freund, F. Girosi, Training support vector machines: an application to face detection, in: Proceedings of Computer Vision and Pattern Recognition, 1997, p. 130.

[31] M. Pontil, S. Mukherjee, F. Girosi, On the Noise Model of Support Vector Machine Regression, CBCL Paper 168, AI Memo 1651, Massachusetts Institute of Technology, Cambridge, MA, 1998.

[32] M. Pontil, A. Verri, Object recognition with support vector machines, IEEE Trans. Pattern Anal. Machine Intell. 20 (1998) 637.

[33] J.C. Principe, N.R. Euliano, W.C. Lefebvre, Neural and adaptive systems, Wiley, New York, 2000.

[34] C. Schittenkopf, G. Dorffner, E.J. Dockner, Volatility prediction with mixture density networks, in: L. Niklasson, M. Boden, T. Ziemke (Eds.), ICANN '98—Proceedings of the 8th International Conference on Artificial Neural Networks, Springer, Berlin, 1998, p. 929.

[35] C. Schittenkopf, G. Dorffner, E.J. Dockner, Fat tails and non-linearity in volatility models: what is more important?, in: Proc. IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering (CIFEr), NY, USA, 1999, p. 259.

[36] F. Schmitt, D. Schertzer, S. Lovejoy, Multifractal analysis of foreign exchange data, Appl. Stochastic Models Data Anal. 15 (1999) 29.

[37] F. Schmitt, D. Schertzer, S. Lovejoy, Multifractal fluctuations in finance, Int. J. Theor. Appl. Finance 3 (2000) 361.

[38] G.P.J. Schmitz, C. Aldrich, F.S. Gouws, ANN-DT: an algorithm for extraction of decision trees from artificial neural networks, IEEE Trans. Neural Networks 10 (1999) 1392.

[39] A.B. Tickle, R. Andrews, M. Golea, J. Diederich, The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks, IEEE Trans. Neural Networks 9 (1998) 1057.

[40] R.S. Tsay, Analysis of Financial Time Series, Wiley, New York, 2002.

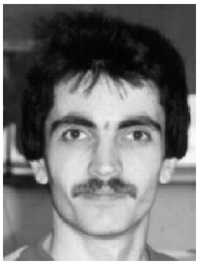
[41] T. Van Gestel, J. Suykens, D. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, J. Vandewalle, Financial time series prediction using least squares support vector machines within the evidence framework, IEEE Trans. Neural Networks (Special Issue on Neural Networks in Financial Eng.) 12 (2001) 809.

[42] P. Vannerem, K.-R. Mller, B. Schlkopf, A. Smola, S. Sldner-Rembold, Classifying LEP data with support vector algorithms, hep-ex/9905027, 1999.

[43] V. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.

[44] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[45] J.-M. Zakoian, Threshold heteroskedastic models, J. Econ. Dynamics Control 18 (1994) 931.



Dr. Valeriy V. Gavrishchaka received his MS and Ph.D. degrees in physics from Moscow Institute of Physics and Technology (Russia) and from West Virginia University (USA) in 1989 and 1996, respectively. From 1997 to 2001 he worked as a research scientist and since 2001 as a consultant at Science Applications International Corporation. His research interests include analysis and simulation of fundamental multiscale processes in space and laboratory plasmas as well as new multidisciplinary approaches for complex system modeling in finance, medicine, and other fields.

Dr. Supriya B. Ganguli is the Chief Information Officer at the Command, Control, Communication and Information Technology Group at SAIC. She holds a Ph.D. in Physics. Her research interests include modeling and simulations of wide range of complex systems with applications in space physics and space weather forecasting, engineering, and other fields.