Gaussian Process Regression Stochastic Volatility Model for Financial Time Series

Jianan Han, Xiao-Ping Zhang, *Senior Member, IEEE*, and Fang Wang

***Abstract*—Traditional economic models have rigid-form transition functions when modeling time-varying volatility of financial time series data and cannot capture other time-varying dynamics in the financial market. In this paper, combining the Gaussian process state-space model framework and the stochastic volatility (SV) model, we introduce a new Gaussian process regression stochastic volatility (GPRSV) model building procedures for financial time series data analysis and time-varying volatility modeling. The GPRSV extends the SV model. The flexible stochastic nature of the Gaussian process state description allows the model to capture more time-varying dynamics of the financial market. We also present the model estimation methods for the GPRSV model. We demonstrate the superior volatility prediction performance of our model with both simulated and empirical financial data.**

***Index Terms*—Financial time series, Gaussian process, Gaussian process regression stochastic volatility model (GPRSV), Gaussian process state-space models, Monte Carlo method, particle filtering, volatility modeling.**

I. Introduction

THE problem of analyzing financial time series data is an important task for both financial research and investment. In the past decades, many researchers take the modeling approach to describe financial data. Modeling provides us a way of discovering knowledge from data and making predictions [1]. From this point, modeling financial time series data is very similar to modeling signals in engineering applications. For example, in the presence of noise, filtering methods such as Kalman filters and particle filters can be applied to financial data [2], [3]. With the recent development of Bayesian nonparametric modeling in signal processing community, we can model financial data with more flexible tools and modeling methods, such as Gaussian process (GP) [4] and copula process [5], etc.

Volatility modeling has been one of the most active financial time series research areas in the past decade [6]. It is of great importance for both finance market practitioners and academic researchers. Volatility can be expressed as the standard deviation of an asset return and it is widely used to describe the variability of financial time series data [7]. There are two main classes of time changing variance models: the generalized autoregressive conditional heteroscedasticity (GARCH) model and the stochastic volatility (SV) model. Autoregressive conditional heteroscedasticity (ARCH) model was first introduced by Nobel laureate Engle [8]. Bollerslev extended the model to GARCH [9]. Parameters of GARCH class models can be learned/estimated using maximum likelihood methods. Although ARCH and GARCH are good to represent some properties of financial asset return series, such as volatility clusters, they are not good to capture some other properties, such as the asymmetric effect. Extensions of GARCH model such as GJR-GARCH [10] are proposed to fix this problem.

SV models are powerful alternatives of widely used GARCH family [10]–[17]. They differ from GARCH models on the process of how the conditional volatility evolves over time. For SV models, the volatility equation is expressed as a stochastic process, which means the value of volatility at time t is latent and unobservable. The first discrete time-varying SV model was introduced by Taylor [11]. Unlike GARCH models, which model the conditional expectation of the volatility, SV models model the volatility process itself separately. The SV model offers more flexibilities than GARCH models. However, the inference of SV model parameters is not as straightforward as the corresponding simple GARCH typed model. In [17], Shephard reviews SV models and inference methods like methods of moments (MM) and quasi-maximum likelihood (QML).

Both GARCH and SV models can be viewed as instances of state-space models (SSMs), which are widely used models for effective modeling of time series data and dynamical systems [1]. The essential idea is that for an observed time series y_t there is an underlying process x_t which itself is evolving through time in a way that reflects the structure of the system. The model consists of two parts: a hidden state x_t and an observation variable y_t. For volatility modeling, the observation variable is return of asset time series, and the volatility can be modeled as the hidden system state.

For above traditional econometric volatility models, model prediction performance is limited by the rigid linear state transition function form because in the financial market, the parameters of a function themselves may change over time. One possible solution to solve this problem is to take a dynamic stochastic function form for the state transition, which can be achieved by using Bayesian nonparametric tools. Nonparametric models are more natural to describe financial time series dynamic behaviors.

Manuscript received October 13, 2015; revised March 21, 2016 and May 09, 2016; accepted May 10, 2016. Date of publication May 19, 2016; date of current version August 12, 2016. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN239031. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Dmitry M. Malioutov.

J. Han is with the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: jhan@ee.ryerson.ca).

X.-P. Zhang is with the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada, and also with the School of Accounting and Finance, the Ted Rogers School of Management, Ryerson University Toronto, ON M5B 2K3, Canada (e-mail: xzhang@ee.ryerson.ca).

F. Wang is with the School of Business and Economics, Wilfrid Laurier University, Waterloo, ON N2L 3C5, Canada (e-mail: fwang@wlu.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2016.2570738

GPs can be used to extend SSMs to Gaussian process state-space models (GP-SSMs), which are Bayesian nonparametric models. The GP-SSM is proved to be a powerful tool to describe the nonlinear dynamic systems in many areas [18], [19]. GPs are widely used as dimensionality reduction technique in the machine learning community. In [20], Lawrence introduces the Gaussian process latent variable model (GPLVM) for the principal component analysis. In Lawrence's model, GP prior is used to map from a latent space to the observed data-space that is high dimensional. In [21], Ko and Fox propose a GP based Bayesian Filter, a nonparametric way to recursively estimate the state of a dynamical system. Wang *et al.* propose the Gaussian process dynamical model (GPDM) in [22]. The GPDM enriches the GPLVM to capture temporal structure by incorporating a GP prior over the dynamics in the latent space. Frigola *et al.* point out that GP can represent functions of arbitrary complexity and provide a straightforward way to specify assumptions about the unknown function in [18].

For the GP-SSM inference, both the hidden states and the GP dynamics are unknown. Direct estimation of hyper-parameters, hidden states and GP function values is a challenging task.

Monte Carlo methods become more and more popular for model estimation and identification because of their accuracy and flexibility of handling complicated models. Two main methods are sequential Monte Carlo (SMC) and particle Markov chain Monte Carlo (particle MCMC) methods. The SMC method is also called particle filter in some applications [23], [24]. Ever since its introduction, the SMC method has been widely used in many areas to solve the problem of inference complex nonlinear models. In economics study, economists introduced many dynamic stochastic general equilibrium (DSGE) models to real-world time series, which often exhibit strong non-Gaussian and time-varying behaviors. In this scenario, SMC methods are used to estimate nonlinear, non-Gaussian SSMs.

The particle MCMC method was first introduced in [25]. The idea of particle MCMC is to use of a certain SMC sampler to construct a Markov kernel leaving the joint smoothing distribution invariant. In [26], Lindsten *et al.* propose the PGAS algorithm. Frigola *et al.* apply the PGAS algorithm to the problem of GP-SSMs inference [19]. Their results show that the PGAS algorithm is suitable to estimate a non-Markovian SSM. In [26], a novel particle particle MCMC algorithm, particle Gibbs with ancestor sampling (PGAS) was proposed. In [19], Frigola *et al.* apply the algorithm to learn hidden states of a GP-SSM and GP dynamics jointly.

For volatility modeling research, Kim *et al.* first estimate a SV model using particle filter in [27]. Recently, in [28], Wu *et al.* propose a GP based GARCH model and a regularized auxiliary particle chain filter (RAPCF) algorithm to estimate the model.

In this paper, with the GP-SSM framework combined with the SV modeling concept, we present a novel nonparametric model—Gaussian process regression stochastic volatility (GPRSV) model to solve the problem of modeling and predicting time-varying variance of financial time series data. GPRSV models usually are more difficult to estimate than parametric volatility models. We apply the recent development of Bayesian nonparametric methods to improve the prediction performance of volatility models. We estimate the hidden states or system variable distribution by taking a full Bayesian nonparametric approach. We can use two estimation methods for the new model. The first one is the RAPCF algorithm [28] based on a SMC inference algorithm for computational efficiency and the second one is PGAS algorithm [19] based on a MCMC method for more accuracy. We demonstrate the superior volatility prediction performance of the new GPRSV model and inference methods with both simulated and empirical financial data.

Our main contribution is to introduce a novel nonparametric model—GPRSV model to solve the problem of modeling and predicting time-varying variance of financial time series data. The new GPRSV model uses a GP-SSM framework combined with the SV modeling concept, different from the GPVM by Wu *et al.* [28]. Furthermore, our GPRSV model incorporates the asymmetric stochastic volatility (ASV) and therefore is more flexible and generic in that it can now handle the well-known volatility effect—leverage effect.

The second contribution of work is that we provided a solution to learn the proposed model. We demonstrate that both SMC algorithms such as RAPCF [28] and MCMC algorithms such as PGAS [19] can be adjusted to estimate the GPRSV models. We also demonstrated through experiments that the new GPRSV model performs better than corresponding GARCH and SV models. In addition, our contribution also lies in the evaluation of the performance of different methods on the prediction of realized volatility. Previous work did not compare on the prediction of realized volatility. These experimental results are significant to help identify the strengths and weaknesses of different methods.

The paper is organized as follows. Section II discusses the volatility modeling. Section III introduces the new GPRSV model. The learning algorithms of the GPRSV model are described in Section IV. Section V presents extensive simulation and experimental results for the real financial data. Finally, we conclude the paper and discuss the various aspects of future work in Section VI.

II. Volatility Modeling

Time series data are collected through time. A time series is a sequence of data points of measurement $z_t \in \mathbb{R}$ index by time t. Financial time series analysis is a highly empirical discipline. People concern more with how asset valuation changes over time. In financial time series research, we usually analyze assets return instead of price [29]. The net return series is defined as

$$\tilde{r}_t = \frac{p_t - p_{t-1}}{p_{t-1}} \tag{1}$$

where r_t is the net return at time t, p_t is the asset price at time t. The logarithm of the total return is also often used due to asymmetry of the net return. The log return series is defined as

$$r_t = \log(1 + \tilde{r}_t) = \log p_t - \log p_{t-1}. \tag{2}$$

It is not hard to see that they are essentially the same when the net return is small. The log return is more commonly used in empirical research.

The idea behind volatility modeling is to express the relationship of the return and the volatility and how these two processes evolve over time. Volatility is a forward-looking concept, we often model the financial time series return variance conditioned on all the relevant information I_{t-1}, defined as

$$\sigma_t^2 = \text{var}(r_t|I_{t-1}) = E((r_t - \mu_t)^2|I_{t-1}) \tag{3}$$

where μ_t is expected value of the asset return r_t.

There are some characteristics commonly observed in asset return series, and all volatility models should capture these characteristics [7], [30].

1) *Heteroscedastic*: The volatility of asset return is not constant through time. It is also called heteroskedasticity. For asset returns, the value of this conditional volatility is time-varying.
2) *Volatility Clustering*: It is widely accepted that the volatilities of asset returns tend to cluster. It also means there are some periods that the market are with high volatilities and some other periods with lower volatilities.
3) *Asymmetric Effect*: Based on rich empirical observations of financial asset returns, volatilities tend to react differently on positive and negative returns.
4) *Heavier Tails*: Rich evidences show that financial asset returns exhibit heavy tails and high-peakiness. Volatility models should explain that the asset returns are not normally distributed.

III. GPRSV Model

We introduce a new GPRSV model to solve the problem of financial time series volatility modeling and predicting. Similar to GARCH models and basic SV models, we model the financial asset return and volatility in state-space modeling framework. The logarithm of variance is modeled as the unobserved latent variable of the system in our model. We use GP to sample unknown hidden states transition function. A GPRSV model can be viewed as an instance of GP-SSM applying to SV models.

A. GARCH Models

Standard GARCH(1, 1) model assumes the asset return follows a Gaussian distribution. Assume the mean μ is zero and the variance is time-varying:

$$r_t \sim \mathcal{N}(0, \sigma_t^2) \tag{4a}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \beta\sigma_{t-1}^2, \tag{4b}$$

where α and β are model parameters, $\alpha_0 > 0, \alpha_1 \geq 0, \beta \geq 0$, and $\alpha_1 + \beta \leq 1$. As can be seen, there is no noise term in the above equations and that the volatility σ_t^2 depends on the observed return r_{t-1}.

B. SV Models

The logarithm of variance is modeled by a latent AR(1) process. Taylor's stochastic model can be presented as

$$r_t = \mu_t + a_t = \mu_t + \sigma_t\epsilon_t \tag{5a}$$

$$\log(\sigma_t^2) = \alpha_0 + \alpha_1 \log(\sigma_{t-1}^2) + \sigma_n\eta_t \tag{5b}$$

where α_1 is a parameter which controls the persistence of logarithm variance and the value of α_1 is between $(-1, 1)$. There are two independent and identically distributed random variables ϵ_t and η_t. The original SV model assumes these two noise parts to be independent identically distributed (i.i.d.) standard normally distributed.

The leverage effect is a well-known phenomenon in financial time series data. ASV models are proposed to extend the original SV model [31], [32]. In an ASV model, a negative correlation between return of time t and volatility of time $t + 1$ is added. An ASV model can be expressed as

$$r_t = \mu_t + a_t = \mu_t + \sigma_t\epsilon_t \tag{6a}$$

$$\log(\sigma_t^2) = \alpha_0 + \alpha_1 \log(\sigma_{t-1}^2) + \tau\eta_t \tag{6b}$$

$$\begin{bmatrix} \epsilon_t \\ \eta_t \end{bmatrix} \sim \mathcal{N}(0, \Sigma), \tag{6c}$$

$$\Sigma = \begin{bmatrix} 1 & \rho\tau \\ \rho\tau & \tau^2 \end{bmatrix}. \tag{6d}$$

It can be seen that the SV model is an unconditional approach in that the time-varying volatility process does not depend on the observable return variables and can parsimoniously model the volatility process itself [12].

The inference of SV model parameters is not as straightforward as the corresponding simple GARCH type model. Inference methods like MM and QML are commonly used [17]. Monte Carlo simulation-based methods to estimate SV models become more and more popular because of their accuracy and flexibility of handling complicated models.

C. GP-SSMs

1) SSM: The general form of standard SSM can be summarized as

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \epsilon,\ \mathbf{x}_t \in \mathbb{R}^M \tag{7a}$$

$$\mathbf{y}_t = g(\mathbf{x}_t) + \nu,\ \mathbf{y}_t \in \mathbb{R}^D, \tag{7b}$$

where ϵ and ν are both i.i.d. noise with zero mean and unit variance. The unknown function f describes the system dynamics and function g links the observation and the system hidden state. Both functions f and g can be either linear or non-linear. The hidden state $\mathbf{x}_t$ follows a Markov chain process.

2) Gaussian Process: A GP can be viewed as an extension of a multivariate Gaussian distribution to infinite dimensions [4]. Any finite subset of samples from the process follows a multivariate Gaussian distribution. Also, a GP can be considered as a normal distribution over function, and it is determined by the mean function $m(x)$ and the covariance function $k(x, x')$:

$$f(x) = \mathcal{GP}(m(x), k(x, x')). \tag{8}$$

All values of $f(x)$ at any location x are jointly Gaussian distributed.

3) GP-SSM: We can now combine the GP and the SSM together. The way of combining the two is to use the SSM structure and apply GP to describe the hidden state transition function. The essence of the GP-SSM is to change the rigid

form of states transition function of traditional SSMs with a GP prior. Financial data exhibits many dynamics because the market is changing all the time and a lot of small changes of the involved factors can result in significant fluctuations. The rigid form of the state transition function in traditional SSMs cannot capture such time-varying dynamics of the model itself. And as more and more data become available, stochastic GP-SSMs become feasible to better represent such time-varying dynamics of the financial market. We assume the hidden state transition function f is sampled from a GP. The SSM is extended to a GP-SSM. Compared with standard SSM, the GP-SSM is a more flexible and powerful tool to model time series data. We can take advantage of this tool to more accurately predict time-varying volatilities.

D. GPRSV Models Framework

In the presented new GPRSV model, the conditional volatility is modeled in a Bayesian nonparametric way. We assume that the hidden system state process is governed by a stationary stochastic process. The main difference between the GPRSV model and traditional SV models is the driving force for the stochastic process. In traditional SV models, the state transition process is assumed to follow a rigid linear autoregressive form, see (4) and (5). In GPRSV models, the state transition process is not limited to a rigid form but a GP prior is placed over the state transition function. The basic framework of a GPRSV model can be represented by the following equations:

$$a_t = r_t - \mu = \sigma_t \epsilon_t, \tag{9a}$$

$$v_t = \log(\sigma_t^2) = f(v_{t-1}) + \tau \eta_t, \tag{9b}$$

$$f \sim \mathcal{GP}(m(x), k(x, x')), \tag{9c}$$

$$\begin{bmatrix} \epsilon_t \\ \eta_t \end{bmatrix} \sim \mathcal{N}(0, \Sigma), \tag{9d}$$

$$\Sigma = \begin{bmatrix} 1 & \rho\tau \\ \rho\tau & \tau^2 \end{bmatrix}, \tag{9e}$$

where r_t is the asset return at time t and μ is the mean of r_t, a_t is the innovation of the return series; v_t is the logarithm of variance at time t, ϵ_t and η_t are i.i.d. standard Gaussian distributed noises $\sim \mathcal{N}(0, 1)$, respectively. Also, we consider the well-known phenomenon called financial leverage of a negative correlation between today's return and tomorrow's volatility [31], [32], i.e., asymmetric effect. This leverage effect can be captured by the correlation ρ between ϵ_t and η_t. Therefore, our model is also an ASV model similar to [33], [34]. Note that τ and ρ is unknown scaling parameters to be estimated. A special case is $\rho = 0$, i.e., the correlation between ϵ_t and η_t is zero. Such zero correlation GPRSV model can be used when there is no leverage effect such as in exchange rates or when the leverage effect is small since it has fewer parameters to estimate [35].

Note that (9b) represents the SV modeling concept as in (5b) and is fundamentally different from the GARCH modeling based GP process in [28].

The function f is the hidden state transition function. Here we assume function f follows a GP, which is defined by the mean function $m(x)$ and covariance function $k(x, x')$. The parameters with $m(x)$ and $k(x, x')$ are called hyper-parameters. We can put all hyper-parameters in a vector $\boldsymbol{\theta}$. For an example, if the mean function is defined as $m(x) = cx$, then we have c as hyper-parameter for mean function. If the exponential covariance function is $k(x, x') = \gamma \exp(-0.5|x - x'|^2/l^2)$, we have γ, l as the covariance function hyper-parameters. In this case, we have $\boldsymbol{\theta} = (c, \gamma, l)$. We use logarithm of variance instead of standard deviation directly in our model. This is same as Taylor's SV model [11] and Nelson's EGARCH model [36].

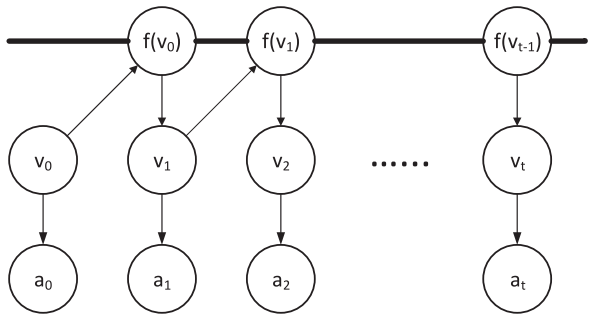


Fig. 1. Graphical model representation of a Gaussian process regression stochastic volatility (GPRSV) model, where a_t is the observation variable at time t, and v_t are the hidden variable (logarithm of volatility) at time t, f_t is the Gaussian process sampled function value at time t, and the thick horizontal line represent fully connected nodes. Hyper-parameters of the Gaussian process are omitted in the figure.

In the GP, the mean function $m(x)$ can encode prior knowledge of system dynamics. For example, we may encode the asymmetric effect in the mean function by adding term of previous positive terms of a_t. The covariance function $k(x, x')$ is defined by covariance between function values $Cov(f(v_t), f(v_{t'}))$, so the covariance function is used to describe the correlation relationship of the time-varying volatility values. Fig. 1 shows the graphical model representation of a GPRSV model.

Financial time series data are changing all the time, and it does not follow the same pattern to change. The rigid linear auto regression function form is limited in the traditional SV models. In the GPRSV model, we do not confine the function form to a fixed form. With different mean and covariance function forms and hyper-parameters, we can sample from a rich class of the state transition functions defined by a stochastic GP.

E. Model Building Process

We can build a GPRSV model in a four step process similar to Tsay's procedures in [7] of building a traditional conditional volatility model. We show the flowchart of this process in Fig. 2.

1) Specify Mean Equation: First we need to test the serial dependence in the return series. If the series are linear dependent, we should use an econometric model (e.g. an ARMA model) to remove the linear dependence in the return series [7], [37]. Depending on the data we want to model, we can use different methods to remove the linear dependence. After doing that, we can specify the distribution the return variable. In (9a), we simply normalize the return series to remove the linear dependence

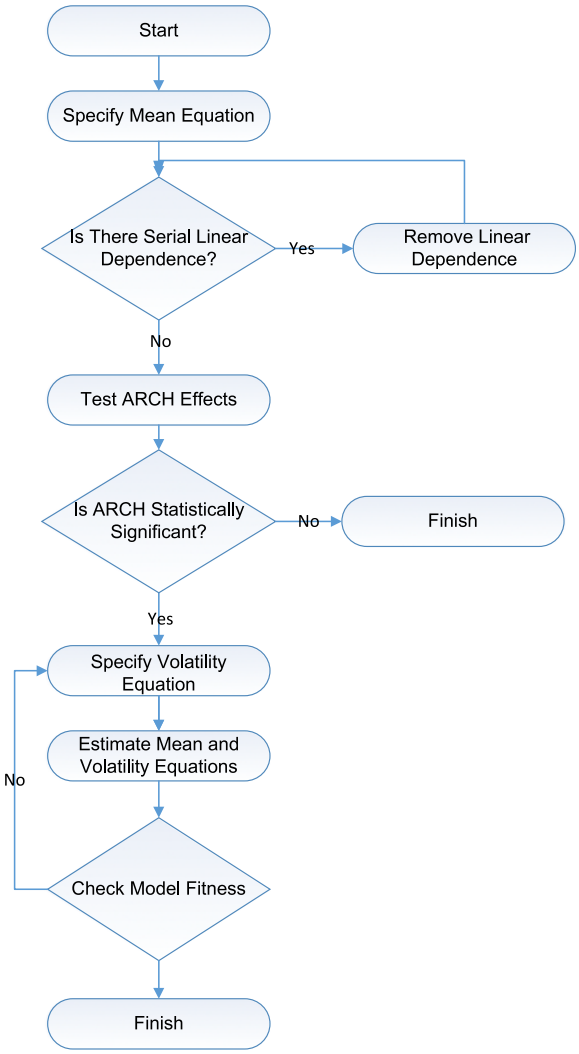


Fig. 2. Flowchart of GPRSV model building process.

part. If the mean of the return series is not significantly different from zero, we can use the return series directly. Otherwise we model the innovation or residuals a_t, and we specify ϵ_t as Gaussian distribution.

Note that in a GP-SSM framework, the hidden state transition function is unknown and it is sampled from a GP defined the molder. The GP has its unknown hyper parameters to be estimated. Together with unknown hidden states (volatilities), parameters in mean and variance equations, there are there parts to be learned. In practice, due to weak serial correlations in asset return series data [7], we prefer to remove linear dependence first to reduce the number of parameters to be estimated in the GRSV model. Note that such approach is also used in [18] and [19].

2) Test ARCH Effect: The residuals of the asset return a_t expressed in (9a) are often used to test conditional heteroskedasticity of the series data. This conditional heteroskedasticity is also known as the ARCH effect [7]. There are two kinds of test for ARCH effect, the first one is to apply the Ljung-Box statistics $Q(m)$ to a_t^2 [38], and the second test is the Lagrange multiplier (LM) test [8]. The null hypothesis of Ljung-Box test is that the first m lags of autocorrelation function (ACF) of the testing series are zero. For the Lagrange multiplier test, we assume in the linear regression form:

$$a_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \cdots + \alpha_m a_{t-m}^2 + c_t, \tag{10}$$

where $t = m+1, \ldots, T$, c_t is the noise term and T is the sample size. We define

$$\mathrm{SSR}_0 = \sum_{t=m+1}^{T} (a_t^2 - \bar{\omega})^2, \tag{11a}$$

$$\mathrm{SSR}_1 = \sum_{t=m+1}^{T} \hat{c}_t^2, \tag{11b}$$

$$F = \frac{(\mathrm{SSR}_0 - \mathrm{SSR}_1)/m}{\mathrm{SSR}_1/(\mathrm{T}-2m-1)}, \tag{11c}$$

where

$$\bar{\omega} = (1/T)\sum_{t=1}^{T} a_t^2 \tag{12}$$

is the sample mean of a_t^2; F is asymptotically distributed as a chi-squared distribution χ_m^2 under null hypothesis and m is the degree of freedom. The null hypothesis H_0 is $\alpha_1 = \cdots = \alpha_m = 0$. The decision rule is to reject H_0 if $F > \chi_m^2(\alpha)$(here $\chi_m^2(\alpha)$ is the upper $100(1-\alpha)$th percentile of χ_m^2), or type-I error: the p value of F is less than α (see [7] for details).

Also we can use sample autocorrelation function (ACF) and sample partial autocorrelation function (PACF) to see the ARCH effect of financial time series data. If both ACF and PACF are not significant but the squared returns are significantly autocorrelated, we can model the data using a conditional volatility model. If we do not observe the autocorrelation of the squared returns, there is no need to use a conditional volatility model, i.e., all such time-varying conditional volatility models, including ARCH and our model, are not applicable.

3) Specify Volatility Equation: The key of volatility modeling is to specify how the hidden volatility or logarithm of variance evolves over time. In GPRSV models, this part is modeled using the flexible Bayesian nonparametric tool, GP regression. For GARCH and SV models this part is modeled with a linear regression approach. Once we estimate the model parameters, those parametric models are determined. When the hidden variable is modeled using GP regression, we need to specify both the mean and covariance functions. Besides these functions forms, the initial value of hyper-parameters (the parameters in mean and covariance functions are called hyper-parameters) associated with them need to be specified as well. Note that with the same hyper-parameters, the function form is not constant and is a random function sampled from a GP determined by the hyper-parameters.

4) Estimate Model Parameters and Check Model Fitness: After specifying both the mean and volatility equations, and associated hyper-parameters and in Steps 2 and 3, we can use training data to estimate unknown parameters. Once we obtain estimated parameters, we can use testing data to test the esti-

mated model. And it is necessary to check the fitness of model we obtained so far. We can examine the model fitness using the diagnostics of the SV model described by Kim *et al.* in [27]. If necessary we need to go back to Step 3 to modify the GP mean and covariance function forms or hyper-parameters.

IV. Inference for the GPRSV model

The linear SSMs with Gaussian noise can be inferred using Kalman Filters [39], but linear Gaussian SSMs can only model a limited set of phenomena. GP-SSMs provide us a flexible framework for time series analysis, but this great descriptive power comes with the expense of computational cost. However, it is impossible to obtain analytic solution for our nonlinear GP-SSMs using the Kalman filter algorithm. We need simulation-based methods like SMC and MCMC methods to solve the problem of inference our nonlinear GP-SSMs. Our solution to this problem is applying the Monte Carlo method to simulate the unknown densities. The core idea of Monte Carlo methods is to draw a set of i.i.d. samples (particles) from a target distribution density, and use the samples to approximate the target density with the point-mass function [40]

$$p_N(x) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x^{(i)}}(x), \tag{13}$$

where $x^{(i)}$ is the ith sample, N is the number of samples, and $\delta_{x^{(i)}}(x)$ denotes the Delta-Dirac mass function value at $x^{(i)}$. Furthermore we can approximate integrals of f which is function of interest. $I(f)$ can be achieved with tractable sums $I_N(f)$:

$$\begin{aligned} I_N(f) &= \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \xrightarrow[N\to\infty]{a.s.} I(f) \\ &= \int f(x)p(x)dx. \end{aligned} \tag{14}$$

To estimate GPRSV models, we can use two Monte Carlo simulated based algorithms: the PGAS [26] and RAPCF algorithms [28]. When applying these two algorithms to GPRSV model estimation problems, the GP regression function value f is marginalized out. Then we can target jointly estimate the hidden states and hyper-parameters. After marginalizing out f, the models become non-Markovian SSMs. Traditional filter and smooth methods are not capable of identifying such models. The Monte Carlo methods based algorithms we present here provide us a powerful tool to solve this problem. Both of the hidden states and parameters can be represented using particles associated with normalized weights.

A. SMC Methods

The first method we can use to estimate a GPRSV model is the RAPCF algorithm [28], which belongs to SMC method [41]. Compared with the original learning approach in [28], our learning algorithms learn both unknown hyper-parameters in the GP and normal parameter using particles. In [28], the hyper-parameters are all within the GP. In our model, besides these hyper-parameters, we also need to learn extra unknown parameters τ and ρ.

We put all unknown GP mean and covariance equation hyper-parameters and τ and ρ in a vector $\boldsymbol{\theta}$ and initialize $\boldsymbol{\theta}$ with a prior $p(\boldsymbol{\theta})$. Besides $p(\boldsymbol{\theta})$, other inputs include: return data $r_{1:T}$, shrinkage parameter λ, and the number of particles N. We have total N particles indexed by i. At the beginning we remove linear dependence from the return series $r_{1:T}$ and obtain $a_{1:T}$. For the first iteration $t = 0$, we can sample N parameter particles from prior $p(\boldsymbol{\theta})$. Also we set initial importance weights $W_0^i = 1/N$. From $t = 1$ to $t = T$, we do the following steps:

1) Remove linear dependence from $r_{1:T}$, and obtain residuals $a_{1:T}$, which is the observation variable in the SSM point of view.
2) The mean of N particles is calculated by
$$\bar{\boldsymbol{\theta}}_{t-1} = \Sigma_{i=1}^{N} W_{t-1}^i \boldsymbol{\theta}_{t-1}^i, \tag{15}$$
and then parameter particles are shrunk towards their empirical means based on
$$\tilde{\boldsymbol{\theta}}_t^i = \lambda \boldsymbol{\theta}_{t-1}^i + (1-\lambda)\bar{\boldsymbol{\theta}}_{t-1}, \tag{16}$$
where λ is the shrinkage parameter. Empirically we use $\lambda = 0.95$ in the experiment. The empirical range is 0.9 to 0.98. As can be seen, λ is a parameter generating some perturbations on top of the mean for a particle.
3) Given all the hidden states $v_{1:t-1}$ until time $t-1$ and the calculated parameter $\tilde{\boldsymbol{\theta}}_t^i$, we have the state transition function f sampled from a GP. With known hidden state transition function f, we can compute the expected value μ_t^i of v_t in (9b) as
$$\mu_t^i = \mathbb{E}\left(v_t | \tilde{\boldsymbol{\theta}}_t^i, v_{1:t-1}^i\right). \tag{17}$$
This is a one-step prediction for hidden state v_t using GP regression. This is different from the traditional parametric models whose state transition function is rigid form.
4) The conditional probability $p(a_t|\mu_t^i, \tilde{\boldsymbol{\theta}}_t^i)$ is computed using (9a). Assuming $\epsilon_t \sim \mathcal{N}(0,1)$, we have
$$a_t \sim \mathcal{N}(0, \sigma_t^2). \tag{18}$$
As we defined $v_t = \log(\sigma_t^2)$, we can compute $p(a_t|\mu_t^i, \bar{\boldsymbol{\theta}}_t^i)$
$$p\left(a_t|\mu_t^i, \bar{\boldsymbol{\theta}}_t^i\right) \sim \mathcal{N}\left(0, e^{\mu_t^i}\right). \tag{19}$$
The importance weights g_t^i are calculated as
$$g_t^i \propto W_{t-1}^i p(a_t|\mu_t^i, \bar{\boldsymbol{\theta}}_t^i). \tag{20}$$
5) After obtaining the important weights, we resample N new particles, and use j for indexing. The jth particle is sampled according to importance weights given by $\{g_t^i, i = 1, \ldots, N\}$.
6) The chain of v_t is propagated forward $\{v_{1:t-1}^j, j = 1, \ldots, N\}$. We add jitter by
$$\boldsymbol{\theta}_{t-1}^j \sim \mathcal{N}(\boldsymbol{\theta}_t^j, (1-\lambda^2)\boldsymbol{Z}_{t-1}), \tag{21}$$

where $\boldsymbol{Z}_{t-1}$ is empirical covariance matrix of $\boldsymbol{\theta}_{t-1}$. $\boldsymbol{Z}_{t-1}$ is computed using

$$\boldsymbol{Z}_{t-1} = \mathbb{E}\left[(\boldsymbol{\theta}_{t-1} - \tilde{\boldsymbol{\theta}}_{t-1})(\boldsymbol{\theta}_{t-1} - \tilde{\boldsymbol{\theta}}_{t-1})^T\right]. \quad (22)$$

7) New states v_t^j are generated according to

$$v_t^j \sim p\left(v_t|\boldsymbol{\theta}_t^j, v_{1:t-1}^j, a_{1:t-1}\right). \quad (23)$$

8) Adjust weights W_t^j according to

$$W_t^j \propto p\left(a_t|v_t^j, \boldsymbol{\theta}_t^j\right)/p\left(a_t|\mu_t^j, \tilde{\boldsymbol{\theta}}_t^j\right). \quad (24)$$

The algorithmic details of the RAPCF procedure can be found in [28].

B. Particle MCMC methods

Besides SMC methods, we can estimate GPRSV models using MCMC methods as well. MCMC plays a significant role in statistics, economics, computing science and physics over the last three decades. In this section we focus on particle MCMC methods to estimate GPRSV models.

We describe the process of estimating a GPRSV model using PGAS algorithm as follows.

1) Remove to linear dependence from $r_{1:T}$, and obtain residuals $a_{1:T}$ which is the observation variable in SSM point of view.
2) In the first iteration, we set $\boldsymbol{\theta}[0]$ and $v_{1:T}[0]$ values randomly. For the rest iterations, we sample particles of $\boldsymbol{\theta}[l]$ conditionally on $v_{1:T}[l-1]$ and $a_{1:T}$.
3) Given that the state trajectory $v_{1:T}[l-1]$ is fixed, we have a GP regression problem where $v_{1:T}[l-1]$ is input, and $v_{1:T}[l]$ is output. Then we can marginalize out the latent dynamics, and sample the hyper-parameters with slice sampling [42].
4) We run conditional particle filter with ancestor sampling (CPF-AS) algorithm. We target at $p(v_{1:T}|\boldsymbol{\theta}[l], a_{1:T})$, conditionally on the previous iteration hidden state trajectory $v_{1:T}[l-1]$. The output of CPF-AS is the new hidden state trajectory $v_{1:T}[l]$ and updated weights w_T^i.
5) Last, we sample k with $p(k=i) = w_T^i$ and set $v_{1:T}[l] = v_{1:T}^k$. The output of PGAS is the hidden volatility $v_{1:T}$ and the hyper-parameter $\boldsymbol{\theta}$.

The key steps in CPF-AS algorithms are:

1) Initialize $N-1$ hidden state v_1^i from the prior $\sim p_1^{\boldsymbol{\theta}}(v_1)$ and leave the last one $v_1^N = v_1'$. Also we initialize the weight $w_1^i = W_1^{\boldsymbol{\theta}}(v_1^i) = 1/N$.
2) Then from $t=2$ until $t=T$, we do resampling and ancestor sampling: we sample $N-1$ times with replacement from $v_{1:t-1}^i$, following $e_t^i \sim \text{Discrete}(\{w_{t-1}^j\}_{j=1}^N)$, for $i = 1, 2, \ldots, N-1$. Then the particle propagation is conducted by resampling $v_t^i \sim p_t^{\boldsymbol{\theta}}(v_t|v_{1:t-1}^{e_t^i})$, for $i = 1, 2, \ldots, N-1$. We set the last particle differently by setting $v_t^N = v_t'$.
3) Combine the two parts together with $v_{1:t}^i = \{v_{1:t-1}^{e_t^i}, v_t^i\}$. Finally the weights are updated by sampling e_t^N with $w_{t-1}^i f_{\boldsymbol{\theta}}(v_t'|v_{t-1}^i)$.

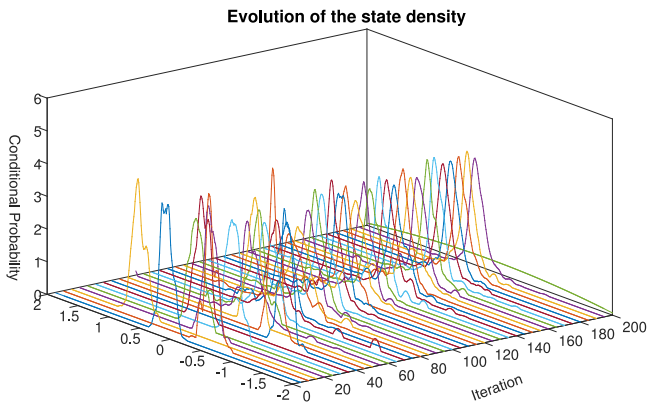


Fig. 3. Estimated hidden state densities of simulated data. There are 200 iteration steps for the simulated data, and we plot every 5 densities in this figure. The densities are generated using particles and weights in RAPCF.

The algorithmic details of the PGAS and CPF-AS algorithms can be found in [19] and [26].

The above two types of methods both can estimate the presented GPRSV models. The particle MCMC method, PGAS, is an offline algorithm that is more accurate than the SMC method, RAPCF, but PGAS is more computationally expensive than RAPCF as shown in [28]. In our experiment, we find that the SMC method, RAPCF, can provide us desired accuracy.

V. Experiments

We apply both the simulated and empirical financial data to demonstrate the new GPRSV model and related inference methods. First, to show that the RAPCF algorithm can be used to estimate GPRSV models, we generate ten sets of simulated data. Then we continue to demonstrate the prediction performance of GPRSV models with real financial data.

A. Simulated Data

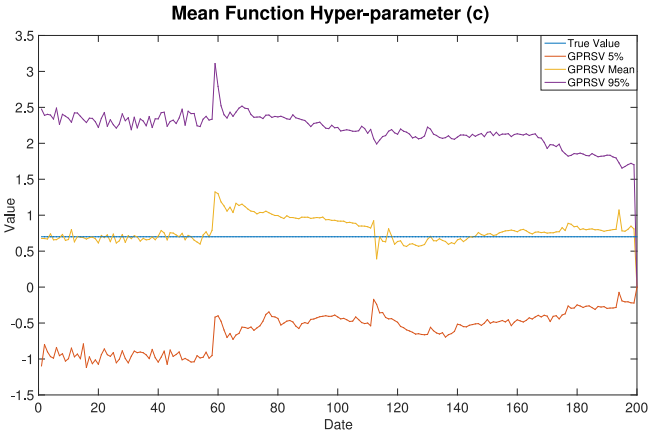
We generate ten synthetic data sets of length $T = 200$ according to (9). We sample the hidden state transition function f from a GP prior. The mean function $m(x_t)$ and the covariance function $k(x, x')$ are specified as follows:

$$m(x_t) = cx_{t-1} \quad (25a)$$

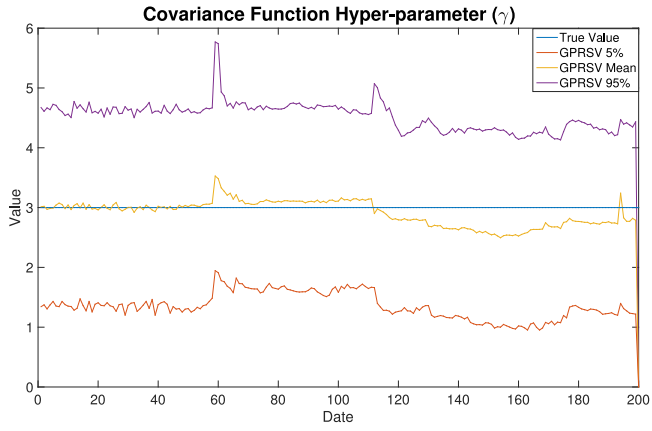
$$k(x, x') = \gamma \exp(-0.5|x - x'|^2/l^2), \quad (25b)$$

where c is the mean equation hyper-parameter, and γ, l are the covariance hyper-parameters. The mean function reflect the pattern how the hidden volatility v_t change with the previous times v_{t-1}, in this case, (25a) representing an auto-regressive fashion as in the traditional SV model. And the covariance function reflects the scale of stochastic deviation of the state transition function from the mean state transition function.

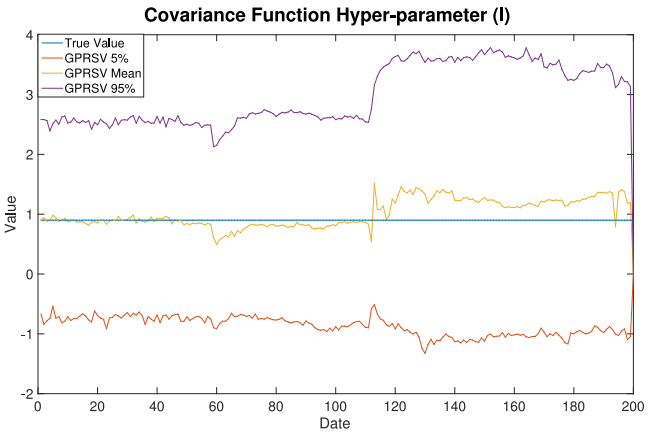
In Fig. 3, we plot the hidden state variable density at every 5 iteration steps to illustrate the convergence of the algorithm. Fig. 4 plots the expected value and 90% posterior intervals for all the hyper-parameters estimated from particles. As can be seen, the hyper-parameters are estimated reasonably well using particles.



(a)



(b)



(c)

Fig. 4. Results of the Gaussian process hyper-parameters. The hyper-parameters are estimated from RAPCF algorithm using particles.

In Fig. 5, we show the results of predictive log-likelihood. At each iteration step, we can calculate the log-likelihood with the estimated hidden state value and the observation value. Compared with the values obtained from the true hidden state and observation, the particle filter based estimates are rather accurate in terms of the log-likelihood. With more particles used, the accuracy of results can improve. Based on our experiment, 800 to 1000 particles are enough to estimate these sets of GPRSV models. With different GP function forms and numbers of hyper-parameters, the more particles may be required.

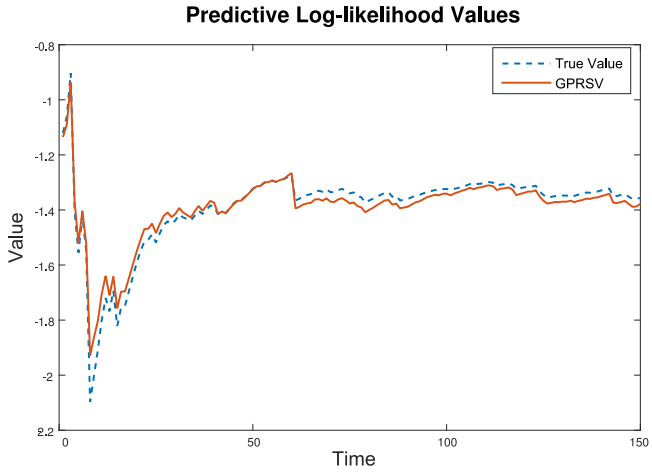


Fig. 5. RAPCF algorithm estimated predictive Log-likelihood value are compared with true value calculate from (9). We discard the first 50 burn in iterations. The predictive log-likelihood results of the RAPCF estimated parameters show that the algorithm can successfully estimate the hidden volatility.

B. Real Data

In this subsection, we apply the new GPRSV model to the real financial data, and compare our model with a class of parametric models and SV models. We use the realized volatility calculated from intraday data as the proxy for the true daily volatility value. The process of the comparing is as follows: first we use in-sample data to train both the two typed models, and then we estimate the volatility values for the out-of-sample period. Finally we use the average loss function values criterion to rank the models.

The evaluation of prediction performance of the model is the key step in the empirical data experiment. In finance study, it is rare to find a method that is consistently superior to predict the price of financial assets. Empirical studies are often inconclusive. The problem of volatility predicting is that we cannot observe the variance directly. The evaluation of volatility prediction can be complicated. One of the most popular evaluation approaches for prediction models is to employ a statistical loss function [6]. We adopt a class of statistical loss functions instead of a particular one. Here we denote the unbiased ex post proxy of conditional variance as σ^2_{t+p} and the p-step predicted value of the model as $\hat{\sigma}^2_{t+p}$. We take the following loss functions [43], [44],

$$\mathrm{MAD}: \mathrm{L}(\hat{\sigma}_{\mathrm{t+p}}, \sigma_{\mathrm{t+p}}) = n^{-1} \sum_{t=1}^{n} |\hat{\sigma}_{t+p} - \sigma_{t+p}| \tag{26}$$

$$\mathrm{MLAE}: \mathrm{L}(\hat{\sigma}_{\mathrm{t+p}}, \sigma_{\mathrm{t+p}}) = n^{-1} \sum_{t=1}^{n} \log(|\hat{\sigma}^2_{t+p} - \sigma^2_{t+p}|) \tag{27}$$

$$\mathrm{QLIKE}: \mathrm{L}(\hat{\sigma}^2_{\mathrm{t+p}}, \sigma^2_{\mathrm{t+p}}) = n^{-1} \sum_{t=1}^{n} (\sigma^2_{t+p}/\hat{\sigma}^2_{t+p} + \log \hat{\sigma}^2_{t+p}) \tag{28}$$

$$\mathrm{HMSE}: \mathrm{L}(\hat{\sigma}^2_{\mathrm{t+p}}, \sigma^2_{\mathrm{t+p}}) = n^{-1} \sum_{t=1}^{n} (\sigma^2_{t+p}/\hat{\sigma}^2_{t+p} - 1)^2. \tag{29}$$

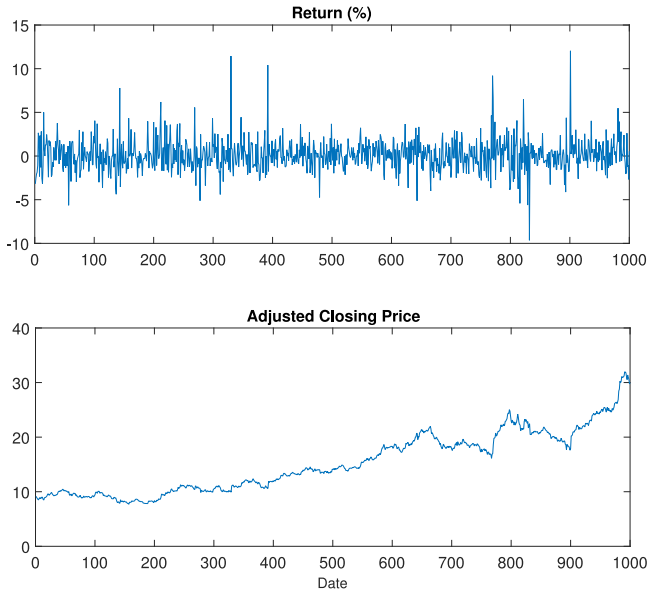


Fig. 6. Both IBM return (in percentage) and price data are plotted. The data period is from January 1, 1988 to September 14, 2003. There are 1000 observations in total.

Another problem with volatility prediction evaluation is that we do not have the true volatility value in the loss function. We have to use some proxy to stand for the real value. Some proxy like the square of return can be quite inaccurate. In our experiment, we use the "realized volatility" calculate by high frequency data [43], [45]. In our experiment, we want to model daily return series volatility, so we can use the daily volatility estimated by high-frequency intra-daily data. Compared with the squared return, realized volatility is considered to be more precise proxy for volatility prediction evaluation.

The data set we analyze is the IBM stock daily adjusted closing price data.[1] We use the daily adjusted closing price as our input to compute the return. The realized volatility data are from [43]. The data period is from January 1, 1988 to September 14, 2003. There are $T = 1000$ observations in total, the first 200 ones (from January 1, 1988 to September 27, 2001) are used as in-sample part for training purposes and the rest observations (from September 28, 2001 to September 14, 2003) are used as out-of-sample for evaluating prediction performance.

We build the basic GPRSV model with the IBM return data. The price and return data are shown in Fig. 6. The in-sample data mean value is quite small and the standard deviation is around one. The detailed statistics are presented in Table I.

To test the ARCH effect as explained in the model building process section, we plot both ACF and PACF for the data in Fig. 7. We can observe that both ACF and PACF are not significant for the returns but the squared returns are significantly autocorrelated. We also conducted the Ljung-Box Q-Test, a standard

[1]The data set can be obtained from YAHOO finance website at http://finance.yahoo.com/. The closing stock price is adjusted for any distributions and corporate actions (such as stock splits, dividends, etc.) that occurred in the stock history to accurately represent the firm's equity value beyond the simple unadjusted market price.

TABLE I
DESCRIPTIVE STATISTICS OF IBM DAILY RETURN DATA. WE ALSO INCLUDE IN PARENTHESIS THE P-VALUES OF THE NULL HYPOTHESES THAT THE MEAN IS ZERO, THE SKEWNESS IS ZERO AND THE KURTOSIS IS BELOW THREE, RESPECTIVELY

Mean	Standard Deviation	Skewness	Kurtosis	Min	Max
0.111 (0.051)	1.798	0.853 (0.00)	9.235 (0.00)	−9.650	12.047

TABLE II
ESTIMATED GPRSV MODEL HYPER-PARAMETERS RESULTS FOR IBM DAILY RETURN DATA

c	γ	l
1.8777	3.3064	1.3044

procedure suggested in Tsay [7] and confirmed the observations. In this case, we can model the data using a conditional volatility model.

Follow the 4-step process discussed in Section III, first we get the return data, and test the ARCH effect. The GP dynamics (mean and covariance function) are specified as in (25). The hyper-parameters include c, γ and l. Using the algorithms we discussed in Section IV, the hyper-parameters and hidden states are estimated. The estimated parameters are presented in Table II. We compare the new GPRSV model with four traditional parametric volatility models: GARCH, GJR-GARCH, SV and ASV. For the GARCH typed models, we use Kevin Sheppard's Oxford MFE Toolbox (http://www.kevinsheppard.com/MFE_Toolbox) to estimate parameters and make prediction. For GPVM and GPRSV models, RAPCF algorithm burn-in period is 200. The number of samples (or particles) is 200. We use shrinkage parameter $\lambda = 0.96$. For GARCH typed models, we use 200 data points to train model parameters.

In Fig. 8, we plot the estimated volatility values of GARCH, GJR-GARCH, SV and GPRSV models along with the return data to illustrate the time-varying volatility and the differences among different models. Table III presents the results of loss function values of all models with realized volatility as proxy. As can be seen in the table, the GPRSV achieves the lowest average loss function values for all functions. The prediction performance the new GPRSV model is the best based on the loss function values.

Besides stock data of IBM, we also apply the experiment to three additional index data. The return and realized volatility data are obtained from Oxford-Man Institute of Quantitative Finance Realized Library.[2] The loss function values arepresented in Tables V, VII and IX. The t-statistics from Diebold–Mariano–West (DMW) tests [46] of equal predictive accuracy for GPRSV model compared with other models are presented in Tables IV, VI, VIII and X, respectively. A t-statistic absolute value greater

[2]The data set can be obtained from Oxford-Man Institute of Quantitative Finance website at http://realized.oxford-man.ox.ac.uk/

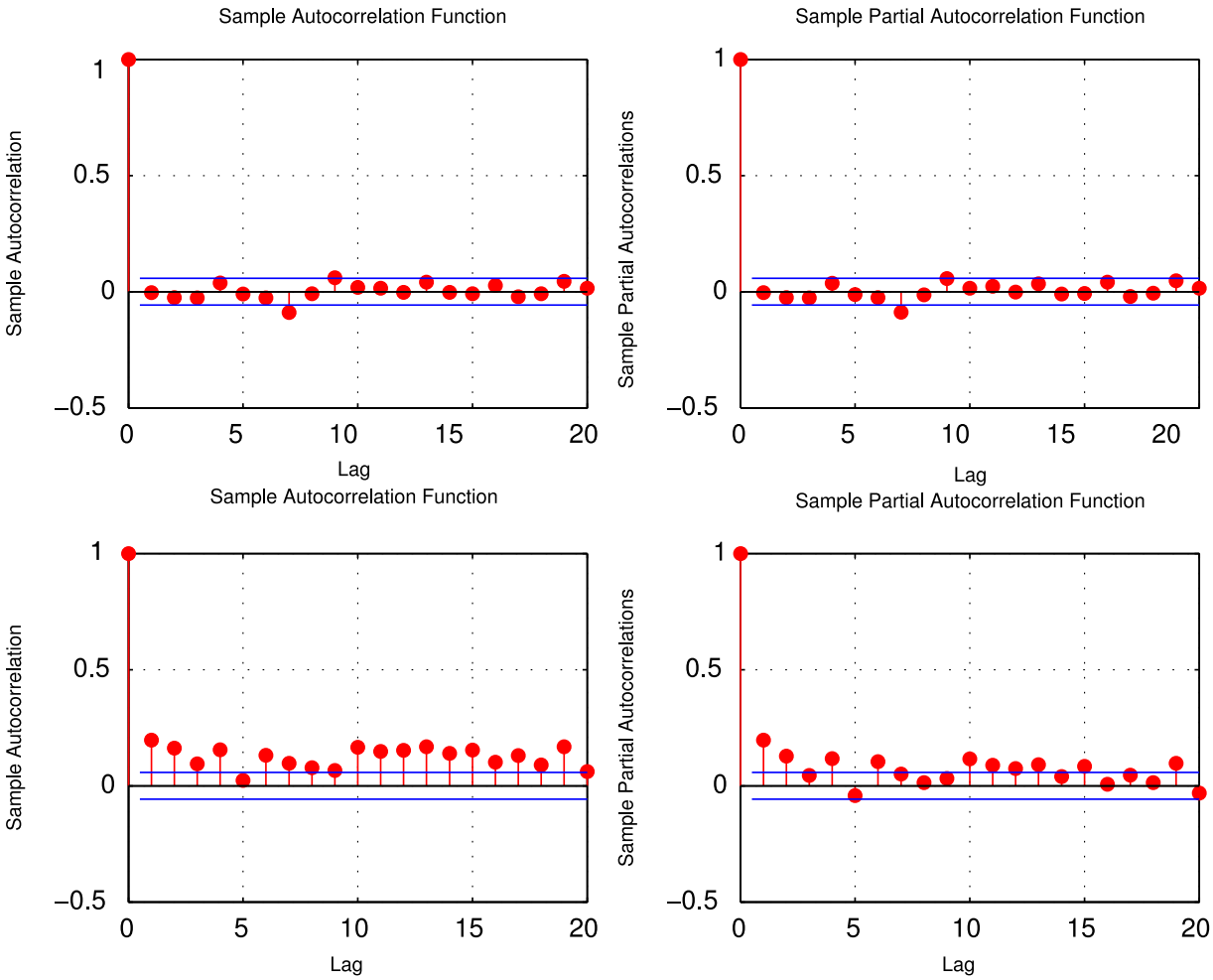


Fig. 7. Sample ACF and PACF functions for IBM daily returns. The first row: ACF and PACF of the returns; the second row: ACF and PACF of the squared returns.

TABLE III
LOSS FUNCTION VALUES OF DIFFERENT MODELS FOR IBM VOLATILITY PREDICTION (ONE-STEP PREDICTION)

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH [9]	3.7867	0.7835	9.6472	2.1311
GJR-GARCH [10]	3.7920	0.7717	6.9648	2.0917
SV [11]	3.8000	0.7944	9.1739	2.1312
ASV [32]	3.7684	0.7619	7.1121	2.0835
GPVM [28]	3.5570	0.4122	3.3894	**1.6916**
GPRSV	**3.1631**	**0.3636**	**1.8617**	1.7976

Note: The lowest loss function values are marked using bold fonts. The volatility proxy is the 65-minutes sampled realized volatility.

than 1.96 indicates a rejection of the null of equal predictive accuracy at the 0.05 significance level. The GP regression based nonparametric models—both GPVM and our GPRSV—perform better than the parametric models. The experimental results show that our GPRSV model has consistently superior forecasting ability to the GPVM.

To better understand the flexible function form of the GP, we show an example of the learned unknown function of SP 500

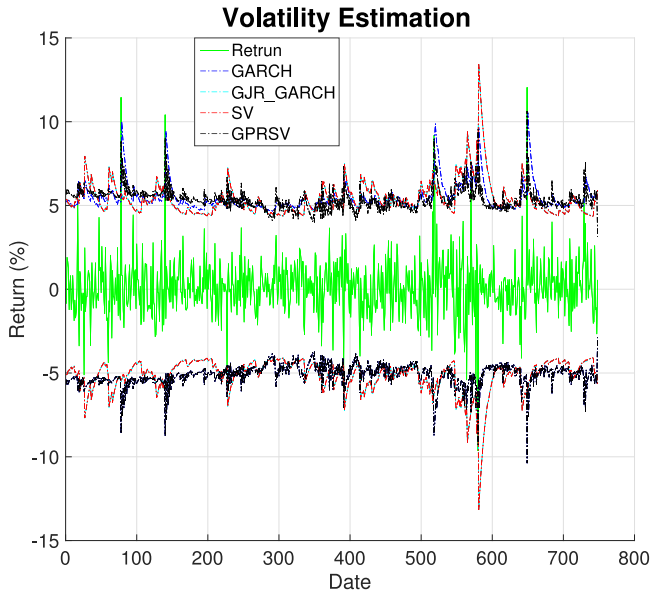


Fig. 8. We plot the return series and predicted $-3\sigma_t$ and $3\sigma_t$ volatility curves based on GARCH, GJR-GARCH, SV and GPRSV models.

TABLE IV
IBM DATA THE T-STATISTICS FROM DIEBOLD-MARIANO-WEST TESTS OF EQUAL PREDICTIVE ACCURACY FOR GPRSV COMPARED WITH OTHER MODELS

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH [9]	−2.4025	−3.5714	−4.8753	−2.7842
GJR-GARCH [10]	−2.3141	−3.4514	−3.2769	−2.8941
SV [11]	−2.4573	−3.5215	−4.5843	−2.7638
ASV [32]	−2.5675	−3.3476	−3.1721	−2.6545
GPVM [28]	−2.0123	−1.9921	−2.8726	2.0354

Note: A t-statistic absolute value greater than 1.96 indicates a rejection of the null of equal predictive accuracy at the 0.05 level. The sign of the t-statistics indicates which forecast performed better for each loss function: a positive t-statistic indicates that the GPRSV model forecast produced larger average loss than the other models, while a negative sign indicates the opposite.

TABLE V
LOSS FUNCTION VALUES OF DIFFERENT MODELS FOR SP 500 INDEX VOLATILITY PREDICTION (ONE-STEP PREDICTION)

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH	8.89E-05	−9.6230	0.5522	−7.3718
GJR-GARCH	8.32E-05	−9.7468	0.5439	−7.3376
SV	7.68E-05	−9.4625	0.6667	−7.3715
ASV	7.71E-05	−9.8052	0.6745	−7.3485
GPVM	6.62E-05	−10.4918	0.3666	**−8.5135**
GPRSV	**6.63E-05**	**−10.3787**	**0.3495**	−8.4971

Note: The lowest loss function values are marked using bold fonts. The volatility proxy is the 5-minutes sampled realized volatility.

TABLE VI
SP 500 DATA THE T-STATISTICS FROM DIEBOLD-MARIANO-WEST TESTS OF EQUAL PREDICTIVE ACCURACY FOR GPRSV COMPARED WITH OTHER MODELS

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH	−5.1423	−3.1438	−2.3852	−3.2731
GJR-GARCH	−4.7879	−3.5493	−2.4583	−3.7885
SV	−4.5412	−4.0127	−2.7782	−3.2899
ASV	−4.6213	−3.6218	−2.8416	−3.7957
GPVM	−1.895	−2.117	−2.0421	2.2023

Note: A t-statistic absolute value greater than 1.96 indicates a rejection of the null of equal predictive accuracy at the 0.05 level. The sign of the t-statistics indicates which forecast performed better for each loss function: a positive t-statistic indicates that the GPRSV model forecast produced larger average loss than the other models, while a negative sign indicates the opposite.

TABLE VII
LOSS FUNCTION VALUES OF DIFFERENT MODELS FOR STOXX 50 INDEX VOLATILITY PREDICTION (ONE-STEP PREDICTION)

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH	9.09E-05	−9.4411	0.5921	−7.1775
GJR-GARCH	8.49E-05	−9.5540	0.4935	−7.3485
SV	8.49E-05	−9.4625	0.4925	−7.3518
ASV	9.72E-05	−9.5357	0.4697	−7.3376
GPVM	4.03E-05	−10.1406	0.3266	**−8.0479**
GPRSV	**2.88E-05**	**−10.2958**	**0.2563**	−8.0383

Note: The lowest loss function values are marked using bold fonts. The volatility proxy is the 5-minutes sampled realized volatility.

TABLE VIII
STOXX 50 DATA THE T-STATISTICS FROM DIEBOLD-MARIANO-WEST TESTS OF EQUAL PREDICTIVE ACCURACY FOR GPRSV COMPARED WITH OTHER MODELS

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH	−5.5514	−3.2574	−2.4752	−3.5674
GJR-GARCH	−5.1782	−3.6727	−2.1835	−3.4834
SV	−5.2013	−3.8423	−2.2314	−3.1127
ASV	−5.8415	−4.0113	−2.1456	−3.0835
GPVM	−3.1472	−2.2431	−1.9906	1.2916

Note: A t-statistic absolute value greater than 1.96 indicates a rejection of the null of equal predictive accuracy at the 0.05 level. The sign of the t-statistics indicates which forecast performed better for each loss function: a positive t-statistic indicates that the GPRSV model forecast produced larger average loss than the other models, while a negative sign indicates the opposite.

TABLE IX
LOSS FUNCTION VALUES OF DIFFERENT MODELS FOR N 2252 VOLATILITY PREDICTION (ONE-STEP PREDICTION)

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH	6.62E-05	−10.3923	0.6085	−7.7209
GJR-GARCH	6.54E-05	−10.4208	0.5962	−7.8094
SV	6.58E-05	−10.4254	0.6138	−7.7871
ASV	6.61E-05	−10.4191	0.6165	−7.6871
GPVM	**5.49E-05**	−10.2146	0.2056	**−8.3709**
GPRSV	5.47E-05	**−10.2396**	**0.2015**	−8.3633

Note: The lowest loss function values are marked using bold fonts. The volatility proxy is the 5-minutes sampled realized volatility.

TABLE X
N 2252 DATA THE T-STATISTICS FROM DIEBOLD-MARIANO-WEST TESTS OF EQUAL PREDICTIVE ACCURACY FOR GPRSV COMPARED WITH OTHER MODELS

Model	MAD	$MLAE$	$HMSE$	$QLIKE$
GARCH	−3.5146	−2.6574	−2.4752	−3.5674
GJR-GARCH	−3.7824	−2.4727	−2.1835	−2.7834
SV	−3.9134	−2.3423	−2.2314	−3.2175
ASV	−3.9135	−2.1323	−2.1456	−2.9835
GPVM	2.1721	−1.9931	−1.9906	1.6916

Note: A t-statistic absolute value greater than 1.96 indicates a rejection of the null of equal predictive accuracy at the 0.05 level. The sign of the t-statistics indicates which forecast performed better for each loss function: a positive t-statistic indicates that the GPRSV model forecast produced larger average loss than the other models, while a negative sign indicates the opposite.

index data. The transition function of hidden variables (v or σ) is assumed to follow a linear auto regression form in traditional parametric models. We do not have such assumption in our model, and the unknown function f is sampled from a GP. In Fig. 9, we plot the transition function samples f used for the learning of the GP. The Y axis in Fig. 9 represents the mean value of the hidden variance computed from multiple particles. The solid line segments show the transition function samples used to learn hyper parameters of the GP in (9c). As can be seen, the sample transition function fitting is unbiased as specified in (9b). More specifically, Fig. 9 shows that the GP transition function is flexible to better capture the time-varying function mapping using the probability distribution of transition functions rather

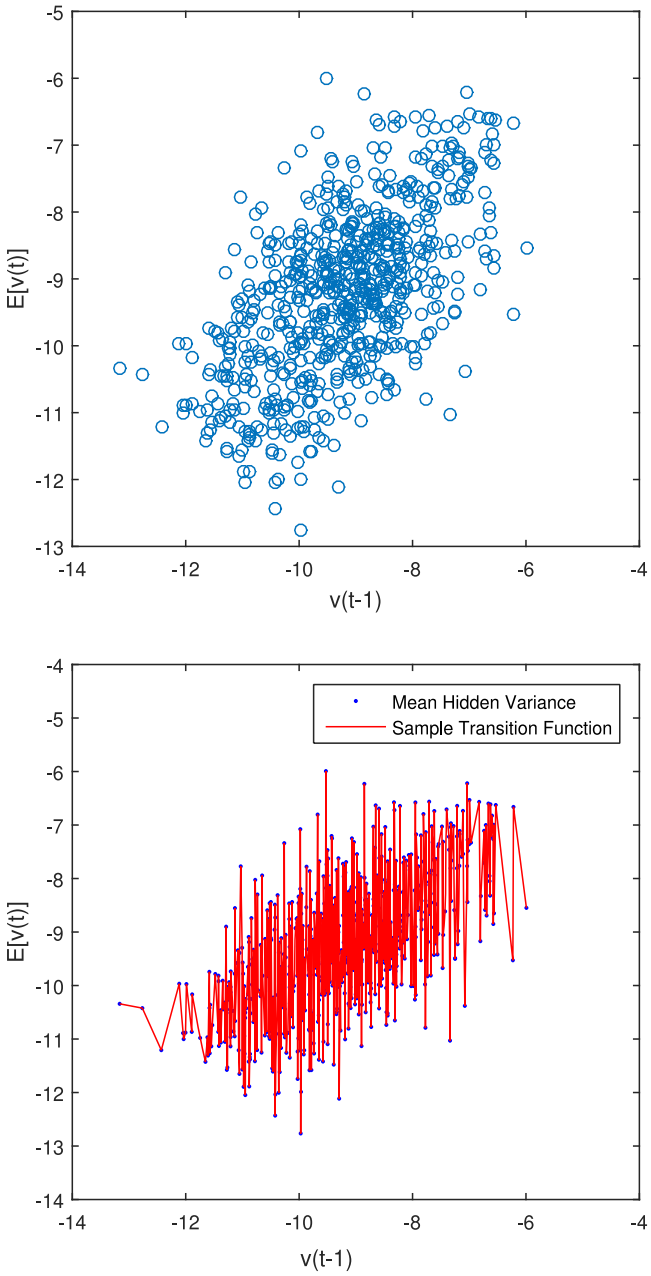


Fig. 9. The example transition function f sampled from a Gaussian process in the GPRSV model for SP500 index data. Top: data point; Bottom: the transition function samples using the mean values of the hidden variance computed from multiple particles as in (9b) and (9c).

than a fixed deterministic function as in traditional GARCH or SV models.

For simulation based algorithms computation cost, RAPCF and PGAS computational cost comparison is discussed in [28]. The cost of applying PGAS is $O(NMT^4)$, RAPCF is $O(NT^3)$. In our experiment, the RAPCF based algorithm is adopted. As an example, the average running time for RAPCF is 5.3342 seconds for case of $N = 200$ particles on N2252 data set on a windows PC with an Intel Core i7-4770 Processor. The average running time for GARCH is 0.9521 seconds, and GJR-GARCH is 0.9528 seconds. For SV model the average running time is 2.5226 seconds, and ASV model is 3.2125 seconds.

VI. Conclusion and Discussions

In this paper, we present a new Gaussian process regression based volatility (GPRSV) model to predict the time-varying volatility of financial time series data based on the combination of the GP-SSM framework and the SV modeling. After we introduce the GPRSV model, we employ a joint estimation algorithm for the hidden volatility states and the Gaussian process dynamics. The flexible stochastic nature of the Gaussian process state description in the GPRSV allows the model to capture more time-varying dynamics of the financial market while the rigid form of traditional parametric modeling such as GARCH models cannot. Note that as more and more data become available, stochastic models such as the GPRSV model and the related MC methods become feasible to better represent time-varying dynamics of the financial market. Our experiment results show that we can successfully estimate the hidden states and hyperparameters of the GPRSV model, and that the GPRSV model can achieve superior volatility prediction performance to traditional economic parametric models.

For future research, on the modeling aspect, we can add exogenous factors to improve the prediction performance. For examples, when modeling one particular energy stock we can use the return data of the energy index or crude oil price as exogenous factors to the stock of interest. We can also apply different covariance functions besides the most common used squared exponential covariance function to adapt better for specific applications. On the application aspect, we can try to apply this model to forecast tail risk measurements, such as the VaR and expected shortfall [47].

Further, we note that realized volatility calculated using intraday data has recently attracted more attention [48]–[50]. Though it is a different problem to use high-frequency (intraday) data to estimated low-frequency (daily) volatility. It is an interesting future work to incorporate the GRPSV model with realized volatility following the realized SV model proposed by Takahashi *et al.* [51].

Besides normal distribution, ϵ_t can follow heavy-tail and skewed distribution as well (see Nakajima and Omori [52]). For heavy tail residuals, we can assume follows a heavy tail or skewed distribution as well. Our model is not limited to Gaussian residuals. With different distribution assumptions, more unknown parameters need to be estimated. Our model may be extended to handle this problem by replacing the Gaussian assumption in (9d) with another heavy tail distribution. The corresponding estimation algorithm can then be modified accordingly.

Acknowledgment

The authors would like to thank the Associate Editor and anonymous reviewers for bringing the ideas to extend the model to handle the leverage effect and numerous constructive suggestions for this paper.

References

[1] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *Int. J. Pattern Recog. Artif. Intell.*, vol. 15, no. 01, pp. 9–42, 2001.

[2] T. Rajbhandary, X.-P. Zhang, and F. Wang, "Piecewise constant modeling and Kalman filter tracking of systematic market risk," in *Proc. IEEE Glob. Conf. Signal. Inf. Process.*, Dec. 2013, pp. 1144–1144.

[3] L. Vo, X.-P. Zhang, and F. Wang, "Multifactor systematic risk analysis based on piecewise mean reverting model," in *Proc. IEEE Glob. Conf. Signal. Inf. Process.*, Dec. 2013, pp. 1142–1142.

[4] C. E. Rasmussen, *Gaussian Processes for Machine Learning. Cambridge*, MA, USA: MIT Press, 2006.

[5] A. Wilson and Z. Ghahramani, "Copula processes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2460–2468.

[6] C. Brownlees, R. Engle, and B. Kelly, "A practical guide to volatility forecasting through calm and storm," *J. Risk*, vol. 14, no. 2, pp. 1–20, 2011.

[7] R. Tsay, *Analysis of Financial Time Series*. New York, NY, USA: Wiley, 2010.

[8] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica, J. Econometric Soc.*, vol. 50, pp. 987–1007, 1982.

[9] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econometrics*, vol. 31, pp. 307–327, 1986.

[10] L. R. Glosten, R. Jagannathan, and D. E. Runkle, "On the relation between the expected value and the volatility of the nominal excess return on stocks," *J. Finance*, vol. 48, no. 5, pp. 1779–1801, 1993.

[11] S. Taylor, *Modelling Financial Time Series. Chichester*, Chichester, U.K.: Wiley, 1986.

[12] M. Fridman and L. Harris, "A maximum likelihood approach for non-Gaussian stochastic volatility models," J. Bus. Econ. Stat., pp. 284–291, 1998.

[13] E. Jacquier, N. G. Polson, and P. Rossi, "Bayesian analysis of stochastic volatility models," *J. Bus. Econ. Statist.*, vol. 12, no. 4, pp. 371–89, 1994.

[14] E. Jacquier, N. G. Polson, and P. E. Rossi, "Bayesian analysis of stochastic volatility models with fat-tails and correlated errors," *J. Econometrics*, vol. 122, no. 1, pp. 185–212, 2004.

[15] A. C. Harvey and N. Shephard, "Estimation of an asymmetric stochastic volatility model for asset returns," *J. Bus. Econ. Statist.*, vol. 14, no. 4, pp. 429–434, 1996.

[16] N. Shephard, "Statistical aspects of arch and stochastic volatility," *Monogr. Statist. Appl. Probability*, vol. 65, pp. 1–68, 1996.

[17] S. Neil and T. Andersen, "Stochastic volatility: Origins and overview," *Univ. Oxford, Dept. Economics, Oxford, U.K., Economics Series Working Papers 389*, 2008.

[18] R. Frigola, Y. Chen, and C. Rasmussen, "Variational Gaussian process state-space models," in *Proc. Adv. Neural Inf. Process. Syst.*,2014, pp. 3680–3688.

[19] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, "Bayesian inference and learning in Gaussian process state-space models with particle MCMC," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3156–3164.

[20] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *J. Mach. Learn. Res.*, vol. 6, pp. 1783–1816, 2005.

[21] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," *Auton. Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[22] J. Wang, A. Hertzmann, and D. Blei, "Gaussian process dynamical models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1441–1448.

[23] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[24] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Stat. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.

[25] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *J. Roy. Statist. Soc. B*, vol. 72, no. 3, pp. 269–342, 2010.

[26] F. Lindsten, M. Jordan, and T. Schön, "Particle Gibbs with ancestor sampling," *J. Mach. Learn. Res.*, vol. 15, pp. 2145–2184, 2014.

[27] S. Kim, N. Shephard, and S. Chib, "Stochastic volatility: Likelihood inference and comparison with ARCH models," *Rev. Econ. Stud.*, vol. 65, no. 3, pp. 361–393, 1998.

[28] Y. Wu, J. M. Hernández-Lobato, and Z. Ghahramani, "Gaussian process volatility model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1044–1052.

[29] J. Campbell, A. W.-C. Lo, and A. C. MacKinlay, *The Econometrics of Financial Markets. Princeton*, Princeton, NJ, USA: Princeton Univ. Press, 1997.

[30] S.-H. Poon and C. W. Granger, "Forecasting volatility in financial markets: A review," *J. Econ. Literature*, vol. 41, no. 2, pp. 478–539, 2003.

[31] A. A. Christie, "The stochastic behavior of common stock variances: Value, leverage and interest rate effects," *J. Financial Econ.*, vol. 10, no. 4, pp. 407–432, 1982.

[32] G. Wu, "The determinants of asymmetric volatility," *Rev. Financial Stud.*, vol. 14, no. 3, pp. 837–59, 2001.

[33] J. Yu, "On leverage in a stochastic volatility model," *J. Econometrics*, vol. 127, no. 2, pp. 165–178, 2005.

[34] Y. Omori, S. Chib, N. Shephard, and J. Nakajima, "Stochastic volatility with leverage: Fast and efficient likelihood inference," *J. Econometrics*, vol. 140, no. 2, pp. 425–449, 2007.

[35] T. Bollerslev, R. Chou, and K. F. Kroner, "Arch modeling in finance: A review of the theory and empirical evidence," *J. Econometrics*, vol. 52, nos. 1/2, pp. 5–59, 1992.

[36] D. Nelson, "Conditional heteroskedasticity in asset returns: A new approach," *Econometrica*, vol. 59, no. 2, pp. 347–70, Mar. 1991.

[37] J. D. Hamilton, *Time Series Analysis. Princeton*, Princeton, NJ, USA: Princeton Univ. Press, 1994.

[38] A. McLeod and W. Li, "Diagnostic checking ARMA time series models using squared-residual autocorrelations," *J. Time Series Anal.*, vol. 4, no. 4, pp. 269–273, 1983.

[39] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[40] C. Andrieu, N. De Freitas, A. Doucet, and M. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, nos. 1/2, pp. 5–43, 2003.

[41] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[42] R. M. Neal, "Slice sampling," *Ann. Statist.*, vol. 31, no. 3, pp. 705–767, 2003.

[43] A. J. Patton, "Volatility forecast comparison using imperfect volatility proxies," *J. Econometrics*, vol. 160, no. 1, pp. 246–256, 2011.

[44] S. J. Koopman, B. Jungbacker, and E. Hol, "Forecasting daily variability of the S&P 100 stock index using historical, realised and implied volatility measurements," *J. Empirical Finance*, vol. 12, no. 3, pp. 445–475, 2005.

[45] A. Torben, T. Bollerslev, F. Diebold, and P. Labys, "Modeling and forecasting realized volatility," *Econometrica*, vol. 71, no. 2, pp. 579–625, 2003.

[46] F. Diebold and R. Mariano, "Comparing predictive accuracy," *J. Bus. Econ. Stat.*, vol. 13, pp. 253–263, 1995.

[47] M. Takahashi, T. Watanabeb, and Y. Omoric, "Volatility and quantile forecasts by realized stochastic volatility models with generalized hyperbolic distribution," *Int. J. Forecasting*, vol. 32, no. 2, pp. 437–457, Apr. 2016.

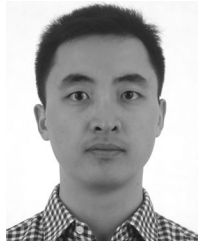
[48] T. G. Andersen, T. Bollerslev, F. X. Diebold, and P. Labys, "Modeling and forecasting realized volatility," *Econometrica*, vol. 71, no. 2, pp. 579–625, Mar. 2003.

[49] T. G. Andersen, T. Bollerslev, and F. X. Diebold, "Roughing it up: Including jump components in the measurement, modeling, and forecasting of return volatility," *Rev. Econ. Statist.*, vol. 89, no. 4, pp. 701–720, Nov. 2007.

[50] F. Corsi, "A simple approximate long-memory model of realized volatility," *J. Financial Econometrics*, vol. 7, no. 2, p. pp. 174–196, 2009.

[51] M. Takahashi, Y. Omoric, and T. Watanabeb, "Estimating stochastic volatility models using daily returns and realized volatility simultaneously," *Comput. Statist. Data Anal.*, vol. 53, no. 6, pp. 2404–2426, Apr. 2009.

[52] J. Nakajima and Y. Omori, "Stochastic volatility model with leverage and asymmetrically heavy-tailed error using GH skew students t-distribution," *Comput. Stat. Data Anal.*, vol. 56, pp. 3690–3704, 2012.



Jianan Han received the Bachelor's degree in network engineering from Hebei Normal University, Shijiazhuang, China, and the Master's degree of Applied Science in electrical and computer engineering from Ryerson University, Toronto, ON, Canada, in 2010 and 2015, respectively. He is currently a Software and Algorithm Developer at the EidoSearch Inc. From 2012 to 2015, he was with the Electrical and Computer Engineering Department, Ryerson University. His research interests include machine learning, signal processing, and data visualization.

Xiao-Ping Zhang (M'97–SM'02) received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China both in electronic engineering, in 1992 and 1996, respectively, and the MBA (Hons.) degree in finance, economics, and entrepreneurship from the University of Chicago Booth School of Business, Chicago, IL, USA.

Since Fall 2000, he has been with the Department of Electrical and Computer Engineering, Ryerson University, where he is currently a Professor, Director of Communication and Signal Processing Applications Laboratory. He has served as the Program Director of Graduate Studies. He is cross appointed to the Finance Department at the Ted Rogers School of Management at Ryerson University. He was Visiting Scientist at Research Laboratory of Electronics (RLE), Massachusetts Institute of Technology, in 2015. His research interests include statistical signal processing, multimedia content analysis, sensor networks and electronic systems, computational intelligence, and applications in big data, finance, and marketing. He is the Cofounder and CEO for EidoSearch, an Ontario-based company offering a content-based search and analysis engine for financial big data.

Dr. Zhang is a registered Professional Engineer in Ontario, Canada, and a member of Beta Gamma Sigma Honor Society. He is the General Cochair for ICASSP2021. He is the General chair for MMSP2015. He is the Publicity Chair for ICME'06 and the Program Chair for ICIC'05 and ICIC'10. He served as a Guest Editor for *Multimedia Tools and Applications*, and the *International Journal of Semantic Computing*. He is a Tutorial Speaker in ACMMM2011, ISCAS2013, ICIP2013, and ICASSP2014. He is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE SIGNAL PROCESSING LETTERS and for *Journal of Multimedia*.

Fang Wang received the Ph.D. degree in Management Information Systems and the MBA degree in finance. He is currently an Associate Professor of Marketing in the Lazaridis School of Business & Economics, Wilfrid Laurier University, Canada. Her research interests include data mining, e-commerce, firm strategy, and long-term firm productivity. Her work has appeared in *Information & Management, Journal of Marketing, International Journal of Research in Marketing, Journal of the Academy of Marketing Science*, among others.