Gaussian Process-Mixture Conditional Heteroscedasticity

Emmanouil A. Platanios and Sotirios P. Chatzis

Abstract—Generalized autoregressive conditional heteroscedasticity (GARCH) models have long been considered as one of the most successful families of approaches for volatility modeling in financial return series. In this paper, we propose an alternative approach based on methodologies widely used in the field of statistical machine learning. Specifically, we propose a novel nonparametric Bayesian mixture of Gaussian process regression models, each component of which models the noise variance process that contaminates the observed data as a separate latent Gaussian process driven by the observed data. This way, we essentially obtain a Gaussian process-mixture conditional heteroscedasticity (GPMCH) model for volatility modeling in financial return series. We impose a nonparametric prior with power-law nature over the distribution of the model mixture components, namely the Pitman-Yor process prior, to allow for better capturing modeled data distributions with heavy tails and skewness. Finally, we provide a copula-based approach for obtaining a predictive posterior for the covariances over the asset returns modeled by means of a postulated GPMCH model. We evaluate the efficacy of our approach in a number of benchmark scenarios, and compare its performance to state-of-the-art methodologies.

Index Terms—Gaussian process, Pitman-Yor process, mixture model, conditional heteroscedasticity, copula, volatility modeling

1 Introduction

Statistical modeling of asset values in financial markets requires taking into account the tendency of assets towards asymmetric temporal dependence [1]. Besides, the data generation processes of the returns of financial market indexes may be non-linear, non-stationary and/or heavy-tailed, while the marginal distributions may be asymmetric, leptokurtic and/or show conditional heteroscedasticity. Hence, there is a need to construct flexible models capable of incorporating these features. The generalized autoregressive conditional heteroscedasticity (GARCH) family of models has been used to address conditional heteroscedasticity and excess kurtosis (see [2], [3]).

The time-dependent variance in series of returns on prices, also known as volatility, is of particular interest in finance, as it impacts the pricing of financial instruments, and it is a key concept in market regulation. GARCH approaches are commonly employed in modeling financial return series that exhibit time-varying volatility clustering, i.e. periods of swings followed by periods of relative calm, and have been shown to yield excellent performance in these applications, consistently defining the state-of-the-art in the field in the last decade. GARCH models represent the variance by a function of the past squared returns and the past variances, which facilitates model estimation and computation of the prediction errors. They have been extremely successful in both volatility prediction based on daily returns, as well as on predictions using intraday information (realized volatility), where they offer state-of-the-art performance.

Gaussian process (GP) models comprise one of the most popular Bayesian methods in the field of machine learning for regression, function approximation, and predictive density estimation [4]. Despite their significant flexibility and success in many application domains, GPs do also suffer from several limitations. In particular, GP models are faced with difficulties when dealing with tasks entailing non-stationary covariance functions, multi-modal output, or discontinuities. Several approaches that entail using ensembles of fractional GP models defined on subsets of the input space have been proposed as a means of resolving these issues (see [5]–[7]).

In this work, we propose a novel GP-based approach for volatility modeling in financial time series (return) data. Our proposed approach provides a viable alternative to GARCH models, that allows for effectively capturing the clustering effects in the variability or volatility. It is based on the introduction of a novel nonparametric Bayesian mixture model, the component distributions of which constitute GP regression models; the noise variance processes of the model component GPs are considered as input-dependent latent variable processes which are also modeled by imposition of appropriate GP priors. This way, our novel approach allows for learning both the observation-dependent nature of asset volatility, as well as the underlying volatility clustering mechanism, modeled as a latent model component switching procedure. In out work, we focus on volatility prediction based on daily returns. Even

- *Emmanouil A. Platanios is with the Department of Machine Learning, Carnegie Mellon University, Pittsburgh, PA 15213 USA. E-mail: e.a.platanios@gmail.com.*
- *Sotirios P. Chatzis is with the Department of Electrical Engineering, Computer Engineering, and Informatics, Cyprus University of Technology, Limassol 3036, Cyprus. E-mail: soteri0s@me.com.*

Manuscript received 23 Jan. 2013; revised 23 Aug. 2013; accepted 10 Sep. 2013. Date of publication 26 Sep. 2013. Date of current version 29 Apr. 2014. Recommended for acceptance by A. J. Storkey.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TPAMI.2013.183

though realized volatility measures have been proven to be more accurate, we opt here for working with daily return series due to the easier access to training data for our algorithms. However, our method is directly applicable to realized volatility as well, without modifications. We dub our approach the Gaussian process-mixure conditional heteroscedasticity (GPMCH) model.

Nonparametric Bayesian modeling techniques, especially Dirichlet process mixture (DPM) models, have become very popular in statistics over the last few years, for performing nonparametric density estimation [8]–[10]. Briefly, a realization of a DPM can be seen as an infinite mixture of distributions with given parametric shape (e.g., Gaussian). An interesting alternative to the Dirichlet process prior for nonparametric Bayesian modeling is the Pitman-Yor process prior [11]. Pitman-Yor processes produce a large number of sparsely populated clusters and a small number of highly populated clusters [12]. Indeed, the Pitman-Yor process prior can be viewed as a generalization of the Dirichlet process prior, and reduces to it for a specific selection of its parameter values. Consequently, the Pitman-Yor process turns out to be more promising as a means of modeling complex real-life datasets that usually comprise a high number of clusters which comprise only few data points, and a low number of clusters which are highly frequent, thus dominating the entire population.

Inspired by these advances, the component switching mechanism of our model is obtained by means of a Pitman-Yor process prior imposed over the component GP latent allocation variables of our model. We derive a computationally efficient inference algorithm for our model based on the variational Bayesian framework, and obtain the predictive density of our model using an approximation technique. We examine the efficacy of our approach considering volatility prediction in a number of financial return series.

Currently, there is an extensive corpus of existing work on conditionally heteroscedastic (Gaussian) mixture models put forward in the financial econometrics literature, which bear some relationship with our proposed approach. In particular, several authors have proposed mixture processes where in each component the conditional variance is driven by GARCH-type dynamics (e.g., [13]–[16]). Such approaches have been shown to yield excellent out-of-sample volatility and density forecasts in a multitude of scenarios. We shall provide comparisons of our approach against a popular such method in the experimental section of our paper.

The remainder of this paper is organized as follows: In Section 2, we provide a brief presentation of the theoretical background of the proposed method. Initially, we present the Pitman-Yor process and its function as a prior in nonparametric Bayesian models; further, we provide a brief summary of Gaussian process regression. In Section 3, we introduce the proposed Gaussian process-mixture conditional heteroscedasticity (GPMCH) model, and derive efficient model inference algorithms based on the variational Bayesian framework. We also propose a copula-based method for learning the interdependencies between the returns of multiple assets jointly modeled by means of an GPMCH model. In Section 4, we conduct the experimental evaluation of our proposed model, considering a number of applications dealing with volatility modeling in financial return series. In the final section, we summarize and discuss our results.

2 Preliminaries

2.1 The Pitman-Yor Process

Dirichlet process models were first introduced by Ferguson [17]. A DP is characterized by a base distribution G_0 and a positive scalar α, usually referred to as the innovation parameter, and is denoted as $\mathrm{DP}(\alpha, G_0)$. Essentially, a DP is a distribution placed over a distribution. Let us suppose we randomly draw a sample distribution G from a DP, and, subsequently, we independently draw M random variables $\{\Theta_m^*\}_{m=1}^M$ from G:

$$G|\alpha, G_0 \sim \mathrm{DP}(\alpha, G_0) \tag{1}$$

$$\Theta_m^*|G \sim G, \quad m = 1, \ldots M. \tag{2}$$

Integrating out G, the joint distribution of the variables $\{\Theta_m^*\}_{m=1}^M$ can be shown to exhibit a clustering effect. Specifically, given the first $M-1$ samples of G, $\{\Theta_m^*\}_{m=1}^{M-1}$, it can be shown that a new sample Θ_M^* is either (a) drawn from the base distribution G_0 with probability $\frac{\alpha}{\alpha+M-1}$, or (b) is selected from the existing draws, according to a multinomial allocation, with probabilities proportional to the number of the previous draws with the same allocation [18]. Let $\{\Theta_c\}_{c=1}^C$ be the set of distinct values taken by the variables $\{\Theta_m^*\}_{m=1}^{M-1}$. Denoting as ν_c^{M-1} the number of values in $\{\Theta_m^*\}_{m=1}^{M-1}$ that equal to Θ_c, the distribution of Θ_M^* given $\{\Theta_m^*\}_{m=1}^{M-1}$ can be shown to be of the form [18]

$$p\left(\Theta_M^*|\{\Theta_m^*\}_{m=1}^{M-1}, \alpha, G_0\right) = \frac{\alpha}{\alpha+M-1}G_0 + \sum_{c=1}^{C}\frac{\nu_c^{M-1}}{\alpha+M-1}\delta_{\Theta_c}, \tag{3}$$

where δ_{Θ_c} denotes the distribution concentrated at a single point Θ_c. These results illustrate two key properties of the DP scheme. First, the innovation parameter α plays a key-role in determining the number of distinct parameter values. A larger α induces a higher tendency of drawing new parameters from the base distribution G_0; indeed, as $\alpha \to \infty$ we get $G \to G_0$. On the contrary, as $\alpha \to 0$ all $\{\Theta_m^*\}_{m=1}^M$ tend to cluster to a single random variable. Second, the more often a parameter is shared, the more likely it will be shared in the future.

The Pitman-Yor process (PYP) [11] functions similar to the Dirichlet process. Let us suppose we randomly draw a sample distribution G from a PYP, and, subsequently, we independently draw M random variables $\{\Theta_m^*\}_{m=1}^M$ from G:

$$G|\delta, \alpha, G_0 \sim \mathrm{PY}(\delta, \alpha, G_0) \tag{4}$$

with

$$\Theta_m^*|G \sim G, \quad m = 1, \ldots M\,, \tag{5}$$

where $\delta \in [0, 1)$ is the discount parameter of the Pitman-Yor process, $\alpha > -\delta$ is its innovation parameter, and G_0 the base distribution. Integrating out G, similar to Eq. (3), we now yield

$$p\left(\Theta_M^*|\{\Theta_m^*\}_{m=1}^{M-1},\delta,\alpha,G_0\right)=\frac{\alpha+\delta C}{\alpha+M-1}G_0+\sum_{c=1}^{C}\frac{\nu_c^{M-1}-\delta}{\alpha+M-1}\delta_{\Theta_c}. \quad (6)$$

As we observe, the PYP yields an expression for $p(\Theta_M^*|\{\Theta_m^*\}_{m=1}^{M-1},G_0)$ quite similar to that of the DP, also possessing the rich-gets-richer clustering property, i.e., the more samples have been assigned to a draw from G_0, the more likely subsequent samples will be assigned to the same draw. Further, the more we draw from G_0, the more likely a new sample will again be assigned to a new draw from G_0. These two effects together produce a *power-law distribution* where many unique Θ_m^* values are observed, most of them rarely [11], thus allowing for better modeling observations with heavy-tailed distributions. In particular, for $\delta>0$, the number of unique values scales as $\mathcal{O}(\alpha M^{\delta})$, where M is the total number of draws. Note also that, for $\delta=0$, the Pitman-Yor process reduces to the Dirichlet process, in which case the number of unique values grows more slowly at $\mathcal{O}(\alpha\log M)$ [12].

A characterization of the (unconditional) distribution of the random variable G drawn from a PYP, $\mathrm{PY}(\delta,\alpha,G_0)$, is provided by the stick-breaking construction of Sethuraman [19]. Consider two infinite collections of independent random variables $\boldsymbol{v}=(v_c)_{c=1}^{\infty}$, $\{\Theta_c\}_{c=1}^{\infty}$, where the v_c are drawn from a Beta distribution, and the Θ_c are independently drawn from the base distribution G_0. The stick-breaking representation of G is then given by [12]

$$G=\sum_{c=1}^{\infty}\varpi_c(\boldsymbol{v})\delta_{\Theta_c}, \quad (7)$$

where

$$p(v_c)=\mathrm{Beta}(1-\delta,\alpha+\delta c) \quad (8)$$

$$\varpi_c(\boldsymbol{v})=v_c\prod_{j=1}^{c-1}(1-v_j)\quad\in[0,1] \quad (9)$$

and

$$\sum_{c=1}^{\infty}\varpi_c(\boldsymbol{v})=1. \quad (10)$$

Under the stick-breaking representation of the Pitman-Yor process, the atoms Θ_c, drawn independently from the base distribution G_0, can be seen as the parameters of the component distributions of a mixture model comprising an unbounded number of component densities, with mixing proportions $\varpi_c(\boldsymbol{v})$.

2.2 Gaussian Process Models

Let us consider an observation space $\mathcal{X}$. A Gaussian process $f(\boldsymbol{x})$, $\boldsymbol{x}\in\mathcal{X}$, is defined as *a collection of random variables, any finite number of which have a joint Gaussian distribution* [20]. A Gaussian process is completely specified by its mean function and covariance function. We define the mean function $m(\boldsymbol{x})$ and the covariance function $k(\boldsymbol{x},\boldsymbol{x}')$ of a real process $f(\boldsymbol{x})$ as

$$\begin{aligned}m(\boldsymbol{x})&=\mathbb{E}[f(\boldsymbol{x})]\\ k(\boldsymbol{x},\boldsymbol{x}')&=\mathbb{E}[(f(\boldsymbol{x})-m(\boldsymbol{x}))(f(\boldsymbol{x}')-m(\boldsymbol{x}'))]\end{aligned} \quad (11)$$

and we will write the Gaussian process as

$$f(\boldsymbol{x})\sim\mathcal{N}(m(\boldsymbol{x}),k(\boldsymbol{x},\boldsymbol{x})). \quad (12)$$

Usually, for notational simplicity, and without any loss of generality, the mean of the process is taken to be zero, $m(\boldsymbol{x})=0$, although this is not necessary. Concerning selection of the covariance function, a large variety of kernel functions $k(\boldsymbol{x},\boldsymbol{x}')$ might be employed, depending on the application considered [20]. This way, a postulated Gaussian process eventually takes the form

$$f(\boldsymbol{x})\sim\mathcal{N}(0,k(\boldsymbol{x},\boldsymbol{x})). \quad (13)$$

Let us suppose a set of independent and identically distributed (i.i.d.) samples $\mathcal{D}=\{(\boldsymbol{x}_i,y_i)|i=1,\ldots,N\}$, with the d-dimensional variables $\boldsymbol{x}_i$ being the observations related to a modeled phenomenon, and the scalars y_i being the associated target values. The goal of a regression model is, given a new observation $\boldsymbol{x}_*$, to predict the corresponding target value y_*, based on the information contained in the training set $\mathcal{D}$. The basic notion behind Gaussian process regression consists in the assumption that the observable (training) target values y in a considered regression problem can be expressed as the superposition of a Gaussian process over the input space $\mathcal{X}$, $f(\boldsymbol{x})$, and an independent white Gaussian noise

$$y=f(\boldsymbol{x})+\epsilon, \quad (14)$$

where $f(\boldsymbol{x})$ is given by (12), and

$$\epsilon\sim\mathcal{N}\left(0,\sigma^2\right). \quad (15)$$

Under this regard, the joint normality of the training target values $\boldsymbol{y}=[y_i]_{i=1}^N$ and some unknown target value y_*, approximated by the value f_* of the postulated Gaussian process evaluated at the observation point $\boldsymbol{x}_*$, yields [20]

$$\begin{bmatrix}\boldsymbol{y}\\ f_*\end{bmatrix}\sim\mathcal{N}\left(\boldsymbol{0},\begin{bmatrix}\boldsymbol{K}(X,X)+\sigma^2\boldsymbol{I}_N & \boldsymbol{k}(\boldsymbol{x}_*)\\ \boldsymbol{k}(\boldsymbol{x}_*)^T & k(\boldsymbol{x}_*,\boldsymbol{x}_*)\end{bmatrix}\right), \quad (16)$$

where

$$\boldsymbol{k}(\boldsymbol{x}_*)\triangleq[k(\boldsymbol{x}_1,\boldsymbol{x}_*),\ldots,k(\boldsymbol{x}_N,\boldsymbol{x}_*)]^T \quad (17)$$

$X=\{\boldsymbol{x}_i\}_{i=1}^N$, $\boldsymbol{I}_N$ is the $N\times N$ identity matrix, and $\boldsymbol{K}$ is the matrix of the covariances between the N training data points *(design matrix)*, i.e.

$$\boldsymbol{K}(X,X)\triangleq\begin{bmatrix}k(\boldsymbol{x}_1,\boldsymbol{x}_1) & k(\boldsymbol{x}_1,\boldsymbol{x}_2)\ldots & k(\boldsymbol{x}_1,\boldsymbol{x}_N)\\ k(\boldsymbol{x}_2,\boldsymbol{x}_1) & k(\boldsymbol{x}_2,\boldsymbol{x}_2)\ldots & k(\boldsymbol{x}_2,\boldsymbol{x}_N)\\ \vdots & \vdots & \vdots\\ k(\boldsymbol{x}_N,\boldsymbol{x}_1) & k(\boldsymbol{x}_N,\boldsymbol{x}_2)\ldots & k(\boldsymbol{x}_N,\boldsymbol{x}_N)\end{bmatrix}. \quad (18)$$

Then, from (16), and conditioning on the available training samples, we can derive the expression of the model predictive distribution, yielding

$$p(f_*|\boldsymbol{x}_*,\mathcal{D})=\mathcal{N}\left(f_*|\mu_*,\sigma_*^2\right) \quad (19)$$

$$\mu_*=\boldsymbol{k}(\boldsymbol{x}_*)^T\left(\boldsymbol{K}(X,X)+\sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{y} \quad (20)$$

$$\sigma_*^2=\sigma^2-\boldsymbol{k}(\boldsymbol{x}_*)^T\left(\boldsymbol{K}(X,X)+\sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{k}(\boldsymbol{x}_*)+k(\boldsymbol{x}_*,\boldsymbol{x}_*). \quad (21)$$

Regarding optimization of the hyperparameters of the employed covariance function (kernel), say θ, and the noise variance σ^2 of a GP model, this is usually conducted by type-II maximum likelihood, that is by maximization of the model marginal likelihood (evidence). Using (16), it is easy to show that the evidence of the GP regression model yields

$$\log p\left(y|X;\theta,\sigma^2\right) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log\left|\boldsymbol{K}(X,X)+\sigma^2\boldsymbol{I}_N\right| - \frac{1}{2}\boldsymbol{y}^T\left(\boldsymbol{K}(X,X)+\sigma^2\boldsymbol{I}_N\right)^{-1}\boldsymbol{y}. \quad (22)$$

It is interesting to note that the GP regression model considers that the noise that contaminates the modeled output variables does not depend on the observations themselves, but rather that it constitutes an additive white noise term with constant variance, which bears no correlation between observations, and no dependence on the values of the observations. Nevertheless, in many real-world applications, with financial return series modeling being a characteristic example, this assumption of constant noise variance is clearly implausible.

To ameliorate this issue, an heteroscedastic GP regression approach was proposed in [21], where the noise variance is considered to be a function of the observed data, similar to previously proposed *heteroscedastic regression* approaches applied to econometrics and statistical finance, e.g., [22], [23]. A key drawback of the approach of [21] is that their heteroscedastic regression approach does not allow for capturing the clustering effects in the variability or volatility, which is apparent in the vast majority of financial return series data, and is effectively captured by GARCH-type models with integrated mixture-model-type clustering mechanisms (e.g., [13]–[16]). Our approach addresses these issues under a nonparametric Bayesian mixture modeling scheme, as discussed next.

3 Proposed Approach

In this section, we first define the proposed GPMCH model, considering a generic modeling problem that comprises the input variables $\boldsymbol{x}\in\mathbb{R}^p$, and the output variables $\boldsymbol{y}\in\mathbb{R}^D$. Further, we derive an efficient inference algorithm for our model under the variational Bayesian inference paradigm, and we obtain the expression of its predictive density. Finally, we show how we can obtain a predictive distribution for the covariances between the modeled output variables $\{y_i\}_{i=1}^D$, by utilization of the statistical tool of *copulas*.

3.1 Model Definition

Let $f_d(\boldsymbol{x})$ be a *latent function* modeling the dth output variable y_d as a function of the model input $\boldsymbol{x}$. We consider that the expression of y_d as a function of $\boldsymbol{x}$ is not uniquely described by the latent function $f_d(\boldsymbol{x})$, but $f_d(\boldsymbol{x})$ is only an instance of the (possibly infinite) set of possible latent functions $f_d^c(\boldsymbol{x})$, $c=1,\ldots,\infty$. To determine the association between input samples and latent functions, we impose a Pitman-Yor process prior over this set of functions. In addition, we consider that each one of these latent functions $f_d^c(\boldsymbol{x})$ has a prior distribution of the form of a Gaussian process over the whole space of input variables $\boldsymbol{x}\in\mathbb{R}^p$. At this point, we make a *further key-assumption*: We assume that the noise variance σ^2 of the postulated GPs is *not* a constant, but rather that *it varies with the input variables* $\boldsymbol{x}\in\mathbb{R}^p$. In other words, we consider the noise variance as a *latent process*, different for each model output variable, and exhibiting a clustering effect, as described by the dynamics of the postulated PYP mixing prior.

Let us consider a set of input/output observation pairs $\{\boldsymbol{x}_n,\boldsymbol{y}_n\}_{n=1}^N$, comprising N samples. Let us also introduce the set of variables $\{z_{nc}\}_{n,c=1}^{N,\infty}$, with $z_{nc}=1$ if the function relating $\boldsymbol{x}_n$ to $\boldsymbol{y}_n$ is considered to be expressed by the set $\{f_d^c(\boldsymbol{x})\}_{d=1}^D$ of postulated Gaussian processes, $z_{nc}=0$ otherwise. Then, based on the previous description, we essentially postulate the following model:

$$p\left(\boldsymbol{y}_n|\boldsymbol{x}_n,z_{nc}=1\right)=\prod_{d=1}^{D}\mathcal{N}\left(y_{nd}|f_d^c(\boldsymbol{x}_n),\sigma_d^c(\boldsymbol{x}_n)^2\right) \quad (23)$$

$$p(z_{nc}=1|\boldsymbol{v})=\varpi_c(\boldsymbol{v}) \quad (24)$$

$$\varpi_c(\boldsymbol{v})=v_c\prod_{j=1}^{c-1}(1-v_j)\in[0,1] \quad (25)$$

with

$$\sum_{c=1}^{\infty}\varpi_c(\boldsymbol{v})=1 \quad (26)$$

$$p(v_c)=\text{Beta}\left(1-\delta,\alpha+\delta c\right) \quad (27)$$

and

$$p\left(\boldsymbol{f}_d^c|X\right)=\mathcal{N}\left(\boldsymbol{f}_d^c|\boldsymbol{0},\boldsymbol{K}^c(X,X)\right), \quad (28)$$

where y_{nd} is the dth element of $\boldsymbol{y}_n$, we define $X\triangleq\{\boldsymbol{x}_n\}_{n=1}^N$, $Y\triangleq\{\boldsymbol{y}_n\}_{n=1}^N$, and $Z\triangleq\{z_{nc}\}_{n,c=1}^{N,\infty}$, $\boldsymbol{f}_d^c$ is the vector of the $f_d^c(\boldsymbol{x}_n)$ $\forall n$, i.e., $\boldsymbol{f}_d^c\triangleq[f_d^c(\boldsymbol{x}_n)]_{n=1}^N$, and $\boldsymbol{K}^c(X,X)$ is the following *design matrix*

$$\boldsymbol{K}^c(X,X)\triangleq\begin{bmatrix}k^c(\boldsymbol{x}_1,\boldsymbol{x}_1) & k^c(\boldsymbol{x}_1,\boldsymbol{x}_2)\ldots & k^c(\boldsymbol{x}_1,\boldsymbol{x}_N)\\ k^c(\boldsymbol{x}_2,\boldsymbol{x}_1) & k^c(\boldsymbol{x}_2,\boldsymbol{x}_2)\ldots & k^c(\boldsymbol{x}_2,\boldsymbol{x}_N)\\ \vdots & \vdots & \vdots\\ k^c(\boldsymbol{x}_N,\boldsymbol{x}_1) & k^c(\boldsymbol{x}_N,\boldsymbol{x}_2)\ldots & k^c(\boldsymbol{x}_N,\boldsymbol{x}_N)\end{bmatrix}. \quad (29)$$

Regarding the latent processes $\sigma_d^c(\boldsymbol{x}_n)^2$, we choose to also impose a GP prior over them. Specifically, to accommodate the fact that $\sigma_d^c(\boldsymbol{x}_n)^2\geq 0$ (by definition), we postulate

$$\sigma_d^c(\boldsymbol{x}_n)^2=\exp\left[g_d^c(\boldsymbol{x}_n)\right] \quad (30)$$

and

$$p\left(\boldsymbol{g}_d^c|X\right)=\mathcal{N}\left(\boldsymbol{g}_d^c|\tilde{m}_d^c\boldsymbol{1},\Lambda^c(X,X)\right). \quad (31)$$

where $\boldsymbol{g}_d^c$ is the vector of the $g_d^c(\boldsymbol{x}_n)$ $\forall n$, i.e., $\boldsymbol{g}_d^c\triangleq[g_d^c(\boldsymbol{x}_n)]_{n=1}^N$, and $\Lambda^c(X,X)$ is a *design matrix*, similar to $\boldsymbol{K}^c(X,X)$, but with (possibly) different kernel functions $\lambda(\cdot,\cdot)$.

Finally, due to the effect of the innovation parameter α on the number of effective mixture components, we also impose a Gamma prior over it:

$$p(\alpha)=\mathcal{G}\left(\alpha|\eta_1,\eta_2\right). \quad (32)$$

This completes the definition of our proposed GPMCH model.

3.2 Inference Algorithm

Inference for nonparametric models can be conducted under a Bayesian setting, typically by means of variational Bayes (e.g., [24]), or Monte Carlo techniques (e.g., [25]). Here, we prefer a variational Bayesian approach, due to its considerably better scalability in terms of computational costs, which becomes of major importance when having to deal with large data corpora [26], [27].

Our variational Bayesian inference algorithm for the GPMCH model comprises derivation of a family of variational posterior distributions $q(.)$ which approximate the true posterior distribution over the infinite sets Z, $\boldsymbol{v} = (v_c)_{c=1}^{\infty}$, $\{\boldsymbol{f}^c\}_{c=1}^{\infty}$, and $\{\boldsymbol{g}^c\}_{c=1}^{\infty}$, and the innovation parameter α. Apparently, Bayesian inference is not tractable under this setting, since we are dealing with an infinite number of parameters.

For this reason, we employ a common strategy in the literature of Bayesian nonparametrics, formulated on the basis of a truncated stick-breaking representation of the PYP [24]. That is, we fix a value C and we let the variational posterior over the v_i have the property $q(v_C = 1) = 1$. In other words, we set $\varpi_c(\boldsymbol{v})$ equal to zero for $c > C$. Note that, under this setting, the treated GPMCH model involves a full PYP prior; truncation is not imposed on the model itself, but only on the variational distribution to allow for tractable inference. Hence, the truncation level C is a variational parameter which can be freely set, and not part of the prior model specification.

Let $W \triangleq \{\boldsymbol{v}, \alpha, Z, \{\boldsymbol{f}^c\}_{c=1}^{C}, \{\boldsymbol{g}^c\}_{c=1}^{C}\}$ be the set of all the parameters of the GPMCH model over which a prior distribution has been imposed, and Ξ be the set of the hyperparameters of the model priors and kernel functions. Variational Bayesian inference introduces an arbitrary distribution $q(W)$ to approximate the actual posterior $p(W|\Xi, X, Y)$ which is computationally intractable, yielding [28]

$$\log p(X, Y) = \mathcal{L}(q) + \mathrm{KL}(q||p), \tag{33}$$

where

$$\mathcal{L}(q) = \int \mathrm{d}W q(W) \log \frac{p(X, Y, W|\Xi)}{q(W)} \tag{34}$$

and $\mathrm{KL}(q||p)$ stands for the Kullback-Leibler (KL) divergence between the (approximate) variational posterior, $q(W)$, and the actual posterior, $p(W|\Xi, X, Y)$. Since KL divergence is nonnegative, $\mathcal{L}(q)$ forms a strict lower bound of the log evidence, and would become exact if $q(W) = p(W|\Xi, X, Y)$. Hence, by maximizing this lower bound $\mathcal{L}(q)$ (variational free energy) so that it becomes as tight as possible, not only do we minimize the KL-divergence between the true and the variational posterior, but we also implicitly integrate out the unknowns W.

Due to the considered conjugate exponential prior configuration of the GPMCH model, the variational posterior $q(W)$ is expected to take the same functional form as the prior, $p(W)$ [29]:

$$q(W) = q(Z) q(\alpha) \left(\prod_{c=1}^{C-1} q(v_c) \right) \prod_{c=1}^{C} \prod_{d=1}^{D} q\left(\boldsymbol{f}_d^c\right) q\left(\boldsymbol{g}_d^c\right) \tag{35}$$

with

$$q(Z) = \prod_{n=1}^{N} \prod_{c=1}^{C} q(z_{nc} = 1). \tag{36}$$

Then, the variational free energy of the model reads (ignoring constant terms)

$$\begin{aligned}
\mathcal{L}(q) = & \sum_{c=1}^{C} \sum_{d=1}^{D} \int \mathrm{d}\boldsymbol{f}_d^c q\left(\boldsymbol{f}_d^c\right) \log \frac{p\left(\boldsymbol{f}_d^c|\boldsymbol{0}, \boldsymbol{K}^c(X, X)\right)}{q\left(\boldsymbol{f}_d^c\right)} \\
& + \sum_{c=1}^{C} \sum_{d=1}^{D} \int \mathrm{d}\boldsymbol{g}_d^c q\left(\boldsymbol{g}_d^c\right) \log \frac{p\left(\boldsymbol{g}_d^c|\tilde{m}_d^c \boldsymbol{1}, \Lambda^c(X, X)\right)}{q\left(\boldsymbol{g}_d^c\right)} \\
& + \int \mathrm{d}\alpha q(\alpha) \Bigg\{ \log \frac{p\left(\alpha|\eta_1, \eta_2\right)}{q(\alpha)} \\
& \quad + \sum_{c=1}^{C-1} \int \mathrm{d}v_c q(v_c) \log \frac{p\left(v_c|\alpha\right)}{q(v_c)} \Bigg\} \\
& + \sum_{c=1}^{C} \sum_{n=1}^{N} q(z_{nc} = 1) \Bigg\{ \int \mathrm{d}\boldsymbol{v} q(\boldsymbol{v}) \log \frac{p(z_{nc} = 1|\boldsymbol{v})}{q(z_{nc} = 1)} \\
& \quad + \sum_{d=1}^{D} \int \int \mathrm{d}\boldsymbol{f}_d^c \mathrm{d}\boldsymbol{g}_d^c q(\boldsymbol{f}_d^c) q(\boldsymbol{g}_d^c) \\
& \qquad \log p\left(y_{nd}|f_d^c(\boldsymbol{x}_n), \sigma_d^c(\boldsymbol{x}_n)^2\right) \Bigg\}.
\end{aligned} \tag{37}$$

Derivation of the variational posterior distribution $q(W)$ involves maximization of the variational free energy $\mathcal{L}(q)$ over each one of the factors of $q(W)$ in turn, holding the others fixed, in an iterative manner [30]. By construction, this iterative, consecutive updating of the variational posterior distribution is guaranteed to monotonically and maximally increase the free energy $\mathcal{L}(q)$ [29].

Let us denote as $\langle . \rangle$ the posterior expectation of a quantity. From (37), we obtain the following variational (approximate) posteriors over the parameters of our model:

1) Regarding the PYP stick-breaking variables v_c, we have

$$q(v_c) = \mathrm{Beta}(v_c|\beta_{c,1}, \beta_{c,2}), \tag{38}$$

where

$$\beta_{c,1} = 1 - \delta + \sum_{n=1}^{N} q(z_{nc} = 1) \tag{39}$$

$$\beta_{c,2} = \langle \alpha \rangle + c\delta + \sum_{c'=c+1}^{C} \sum_{n=1}^{N} q(z_{nc'} = 1). \tag{40}$$

2) The innovation parameter α *approximately* yields

$$q(\alpha) = \mathcal{G}(\alpha|\hat{\eta}_1, \hat{\eta}_2), \tag{41}$$

where

$$\hat{\eta}_1 = \eta_1 + C - 1 \tag{42}$$

$$\hat{\eta}_2 = \eta_2 - \sum_{c=1}^{C-1} \left[\psi(\beta_{c,2}) - \psi(\beta_{c,1} + \beta_{c,2}) \right]. \tag{43}$$

$\psi(.)$ denotes the Digamma function, and

$$\langle \alpha \rangle = \frac{\hat{\eta}_1}{\hat{\eta}_2}. \tag{44}$$

3) Regarding the posteriors over the latent functions $\boldsymbol{f}_d^c$, we have

$$q(\boldsymbol{f}_d^c) = \mathcal{N}(\boldsymbol{f}_d^c | \mu_d^c, \Sigma_d^c), \tag{45}$$

where

$$\Sigma_d^c = \left(\left(\boldsymbol{K}^c(X,X) \right)^{-1} + \boldsymbol{B}_d^c \right)^{-1} \tag{46}$$

$$\mu_d^c = \Sigma_d^c \boldsymbol{B}_d^c \boldsymbol{y}_d \tag{47}$$

$$\boldsymbol{B}_d^c \triangleq \operatorname{diag}\left(\left[\frac{1}{\left\langle \sigma_d^c(\boldsymbol{x}_n)^2 \right\rangle} q(z_{nc}=1) \right]_{n=1}^N \right) \tag{48}$$

and $\boldsymbol{y}_d \triangleq [y_{nd}]_{n=1}^N$.

4) Similar, regarding the posteriors over the latent noise variance processes $\boldsymbol{g}_d^c$, we have

$$q(\boldsymbol{g}_d^c) = \mathcal{N}(\boldsymbol{g}_d^c | \boldsymbol{m}_d^c, \boldsymbol{S}_d^c), \tag{49}$$

where

$$\boldsymbol{S}_d^c = \left(\left(\Lambda^c(X,X) \right)^{-1} + \boldsymbol{Q}_d^c \right)^{-1} \tag{50}$$

$$\boldsymbol{m}_d^c = \Lambda^c(X,X) \left(\boldsymbol{Q}_d^c - \frac{1}{2} \operatorname{diag}\left[q\left(z_{nc}=1\right) \right]_{n=1}^N \right) \mathbf{1} + \tilde{m}_d^c \mathbf{1} \tag{51}$$

and $\boldsymbol{Q}_d^c$ is a positive semi-definite diagonal matrix, whose components comprise variational parameters that can be freely set. Note that, from this result, it follows

$$\left\langle \sigma_d^c(\boldsymbol{x}_n)^2 \right\rangle = \exp\left([\boldsymbol{m}_d^c]_n - \frac{1}{2} [S_d^c]_{nn} \right). \tag{52}$$

5) Finally, the posteriors over the latent variables Z yield

$$q(z_{nc}=1) \propto \exp\left(\left\langle \log \varpi_c(\boldsymbol{v}) \right\rangle \right) \exp\left(r_{nc} \right), \tag{53}$$

where

$$\left\langle \log \varpi_c(\boldsymbol{v}) \right\rangle = \sum_{c'=1}^{c-1} \left\langle \log(1-v_{c'}) \right\rangle + \left\langle \log v_c \right\rangle \tag{54}$$

and

$$r_{nc} \triangleq -\frac{1}{2} \sum_{d=1}^D \left\{ \frac{1}{\left\langle \sigma_d^c(\boldsymbol{x}_n)^2 \right\rangle} \left[\left(\boldsymbol{y}_{nd} - [\mu_d^c]_n \right)^2 + [\Sigma_d^c]_{nn} \right] + [\boldsymbol{m}_d^c]_n \right\}, \tag{55}$$

where $[\xi]_n$ is the nth element of vector ξ, $[\Sigma_d^c]_{nn}$ is the (n,n)th element of Σ_d^c, and it holds

$$\left\langle \log v_c \right\rangle = \psi(\beta_{c,1}) - \psi(\beta_{c,1} + \beta_{c,2}) \tag{56}$$

$$\left\langle \log(1-v_c) \right\rangle = \psi(\beta_{c,2}) - \psi(\beta_{c,1} + \beta_{c,2}). \tag{57}$$

As a final note, estimates of the values of the model hyperparameters set Ξ, which comprises the hyperparameters of the model priors and the kernel functions $k(\cdot,\cdot)$ and $\lambda(\cdot,\cdot)$, are obtained by maximization of the model variational free energy $\mathcal{L}(q)$ over each one of them. For this purpose, in this paper we resort to utilization of the limited memory variant of the BFGS algorithm (L-BFGS) [31].

3.3 Predictive Density

Let us consider the predictive distribution of the dth model output variable corresponding to $\boldsymbol{x}_*$. To obtain it, we begin by deriving the predictive posterior distribution over the latent variables $\boldsymbol{f}$. Following the relevant derivations of Section 2.2, we have

$$q\left(\boldsymbol{f}_*\right) = \sum_{c=1}^C \left\langle \varpi_c\left(\boldsymbol{v}\right) \right\rangle \prod_{d=1}^D \mathcal{N}\left(f_{*d}^c | a_{*d}^c, \left(\sigma_{*d}^c\right)^2 \right), \tag{58}$$

where

$$a_{*d}^c = \boldsymbol{k}^c(\boldsymbol{x}_*)^T \left(\boldsymbol{K}^c(X,X) + \left(\boldsymbol{B}_d^c\right)^{-1} \right)^{-1} \boldsymbol{y}_d \tag{59}$$

$$\left(\sigma_{*d}^c\right)^2 = -\boldsymbol{k}^c(\boldsymbol{x}_*)^T \left(\boldsymbol{K}^c(X,X) + \left(\boldsymbol{B}_d^c\right)^{-1} \right)^{-1} \boldsymbol{k}^c(\boldsymbol{x}_*) + k^c(\boldsymbol{x}_*, \boldsymbol{x}_*) \tag{60}$$

$$\left\langle \varpi_c(\boldsymbol{v}) \right\rangle = \left\langle v_c \right\rangle \prod_{j=1}^{c-1} \left(1 - \left\langle v_j \right\rangle \right) \tag{61}$$

$$\left\langle v_c \right\rangle = \frac{\beta_{c,1}}{\beta_{c,1} + \beta_{c,2}} \tag{62}$$

and

$$\boldsymbol{k}(\boldsymbol{x}_*) \triangleq [k(\boldsymbol{x}_1, \boldsymbol{x}_*), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}_*)]^T. \tag{63}$$

Further, we proceed to the predictive posterior distribution over the latent variables $\boldsymbol{g}$; we yield

$$q\left(g_{*d}^c\right) = \mathcal{N}\left(g_{*d}^c | \tau_{*d}^c, \varphi_{*d}^c \right), \tag{64}$$

where

$$\tau_{*d}^c = \lambda^c(\boldsymbol{x}_*)^T \left(\boldsymbol{Q}_d^c - \frac{1}{2} \right) \mathbf{1} + \tilde{m}_d^c \tag{65}$$

$$\varphi_{*d}^c = \lambda^c(\boldsymbol{x}_*, \boldsymbol{x}_*)^T - \lambda^c(\boldsymbol{x}_*)^T \left(\Lambda_d^c + (\boldsymbol{Q}_d^c)^{-1} \right)^{-1} \lambda^c(\boldsymbol{x}_*) \tag{66}$$

and

$$\lambda(\boldsymbol{x}_*) \triangleq [\lambda(\boldsymbol{x}_1, \boldsymbol{x}_*), \ldots, \lambda(\boldsymbol{x}_N, \boldsymbol{x}_*)]^T. \tag{67}$$

Based on these results, the predictive posterior of our model output variables yields

$$q(y_{*d}) = \int \mathcal{N}\Bigg(y_{*d} \Bigg| \sum_{c=1}^C \left\langle \varpi_c\left(\boldsymbol{v}\right) \right\rangle a_{*d}^c, \sum_{c=1}^C \left\langle \pi_c\left(\boldsymbol{v}\right) \right\rangle^2 \left[\left(\sigma_{*d}^c\right)^2 + \exp\left(g_{*d}^c\right) \right] \Bigg) \mathrm{d}g_{*d}^c. \tag{68}$$

We note that this expression does not yield a Gaussian predictive posterior. However, it is rather straightforward to compute the predictive means and variances of y_{*d}. It holds

$$\hat{y}_{*d} = \mathbb{E}\left[y_{*d} | \boldsymbol{x}_*; \mathcal{D} \right] = \sum_{c=1}^C \left\langle \varpi_c\left(\boldsymbol{v}\right) \right\rangle a_{*d}^c \tag{69}$$

and

$$\mathbb{V}\left[y_{*d} | \boldsymbol{x}_*; \mathcal{D} \right] = \sum_{c=1}^C \left\langle \pi_c\left(\boldsymbol{v}\right) \right\rangle^2 \left[\left(\sigma_{*d}^c\right)^2 + \psi_{*d}^c \right], \tag{70}$$

where

$$\psi_{*d}^c \triangleq \mathbb{E}[\exp(g_{*d}^c) | \boldsymbol{x}_*; \mathcal{D}] = \exp\left(\tau_{*d}^c + \frac{1}{2} \varphi_{*d}^c \right). \tag{71}$$

3.4 Learning the Covariances between the Modeled Output Variables

As one can observe from (23), a characteristic of our proposed GPMCH model is its assumption that the distribution of the modeled output vectors $\boldsymbol{y} \in \mathbb{R}^D$ factorizes over their component variables $\{y_d\}_{d=1}^D$. Indeed, this type of modeling is largely the norm in Gaussian process-based modeling approaches [20]. This construction in essence implies that, under our approach, the modeled output variables are considered independent, i.e. their covariance is always assumed to be zero. However, when jointly modeling the return series of various assets, the modeled output variables (asset returns) are rather strongly correlated, and it is desired to be capable of predicting the values of their covariances for any given input value.

Existing approaches for resolving these issues of GP-based models are based on the introduction of an additional kernel-based modeling mechanism that allows for capturing this latent covariance structure [32]–[36]. For example, in [32] the authors propose utilization of a convolution process to induce correlations between two output components. In [35], a generalization of the previous method is proposed for the case of more than two modeled outputs combined under a convolved kernel. Along the same lines, multitask learning approaches for resolving these issues are presented in [33] and [34], where separate GPs are postulated for each output, and are considered to share the same prior in the context of a multitask learning framework.

A drawback of the aforementioned existing approaches is that, in all cases, learning entails employing a tedious optimization procedure to estimate a large number of hyperparameters of the used kernel functions. As expected, such a procedure is, indeed, highly prone to getting trapped to bad local optima, a fact that might severely undermine model performance.

In this work, to avoid being confronted with such optimization issues, and inspired by the financial research literature, we devise a novel way of capturing the interdependencies between the modeled output variables $\{y_d\}_{d=1}^D$, expressed in the form of their covariances: specifically, we use the statistical tool of *copulas* [37].

The copula, introduced in the seminal work of Sklar [37], is a model of statistical dependence between random variables. A copula is defined as a multivariate distribution with standard uniform marginal distributions, or, alternatively, as a function (with some restrictions mentioned for example in [38]) that maps values from the unit hypercube to values in the unit interval.

3.4.1 Copulas: An Introduction

Let $\boldsymbol{y} = [y_d]_{d=1}^D$ be a D-dimensional random variable with joint cumulative distribution function (cdf) $F\left([y_d]_{d=1}^D\right)$, and marginal cdf's $F_d(y_d),\ d = 1, \ldots, D$, respectively. Then, according to Sklar's theorem, there exists a D-variate copula cdf $C(\cdot, \ldots, \cdot)$ on $[0,1]^D$ such that

$$F\left(y_1, \ldots, y_D\right) = C\left(F_1(y_1), \ldots, F_D(y_D)\right) \tag{72}$$

for any $\boldsymbol{y} \in \mathbb{R}^D$. Additionally, if the marginals $F_d(\cdot),\ d = 1, \ldots, D$, are continuous, then the D-variate copula $C(\cdot, \ldots, \cdot)$ satisfying (72) is unique. Conversely, if $C(\cdot, \ldots, \cdot)$ is a D-dimensional copula and $F_i(\cdot),\ i = 1, \ldots, D$, are univariate cdf's, it holds

$$C\left(u_1, \ldots, u_D\right) = F\left(F_1^{-1}(u_1), \ldots, F_D^{-1}(u_D)\right), \tag{73}$$

where $F_d^{-1}(\cdot)$ denotes the inverse of the cdf of the dth marginal distribution $F_d(\cdot)$, i.e. the quantile function of the dth modeled variable y_d.

It is easy to show that the corresponding probability density function of the copula model, widely known as the *copula density function*, is given by

$$\begin{aligned} c\left(u_1, \ldots, u_D\right) &= \frac{\partial^D}{\partial u_1 \ldots \partial u_D} C\left(u_1, \ldots, u_D\right) \\ &= \frac{\partial^D}{\partial u_1 \ldots \partial u_D} F\left(F_1^{-1}(u_1), \ldots, F_D^{-1}(u_D)\right) \\ &= \frac{p\left(F_1^{-1}(u_1), \ldots, F_D^{-1}(u_D)\right)}{\prod_{i=1}^D p_i\left(F_i^{-1}(u_i)\right)}, \end{aligned} \tag{74}$$

where $p_i(\cdot)$ is the probability density function of the ith component variable y_i.

Let us now assume a parametric class for the copula $C(\cdot, \ldots, \cdot)$ and the marginal cdf's $F_i(\cdot),\ i = 1, \ldots, D$, respectively. In particular, let ζ denote the (trainable) parameter (or set of parameters) of the postulated copula. Then, the joint probability density of the modeled variables $\boldsymbol{y} = [y_i]_{i=1}^D$ yields

$$p(y_1, \ldots, y_D|\zeta) = \left[\prod_{i=1}^D p_i(y_i)\right] c\left(F_1(y_1), \ldots, F_D(y_D)|\zeta\right). \tag{75}$$

Since the emergence of the concept of copula, several copula families have been constructed, e.g., Gaussian, Clayton, Frank, Gumbel, Joe, etc, that enable capturing of any form of dependence structure. By coupling different marginal distributions with different copula functions, copula-based models are able to model a wide variety of marginal behaviors (such as skewness and fat tails), and dependence properties (such as clusters, positive or negative tail dependence) [38]. Selection of the best-fit copula has been a topic of rigorous research efforts during the last years, and motivating results have already been achieved [39] (for excellent and detailed discussions on copulas, c.f. [38], [40]).

3.4.2 Proposed Approach

In this work, to capture the interdependencies (covariances) between the GPMCH-modeled output variables, we propose a *conditional copula*-based dependence modeling framework. Specifically, for the considered D-dimensional output vectors $\boldsymbol{y} = [y_d]_{d=1}^D$, we postulate *pairwise parametric conditional models* for each output pair $(y_i, y_j)_{i,j=1, i\neq j}^D$, with cdf's defined as follows:

$$F(y_i, y_j|\boldsymbol{x}) = C(F_i(y_i|\boldsymbol{x}), F_j(y_j|\boldsymbol{x})|\boldsymbol{x}), \tag{76}$$

where the marginals $F_d(y_d|\boldsymbol{x})$ are the cdf's that correspond to the predictive posteriors $q(y_{*d})$ given by (68), and the used input-conditional copulas are defined under a parametric construction as

$$C(u_i, u_j|\boldsymbol{x}) \triangleq C(u_i, u_j|\zeta_{ij}(\boldsymbol{x})) \tag{77}$$

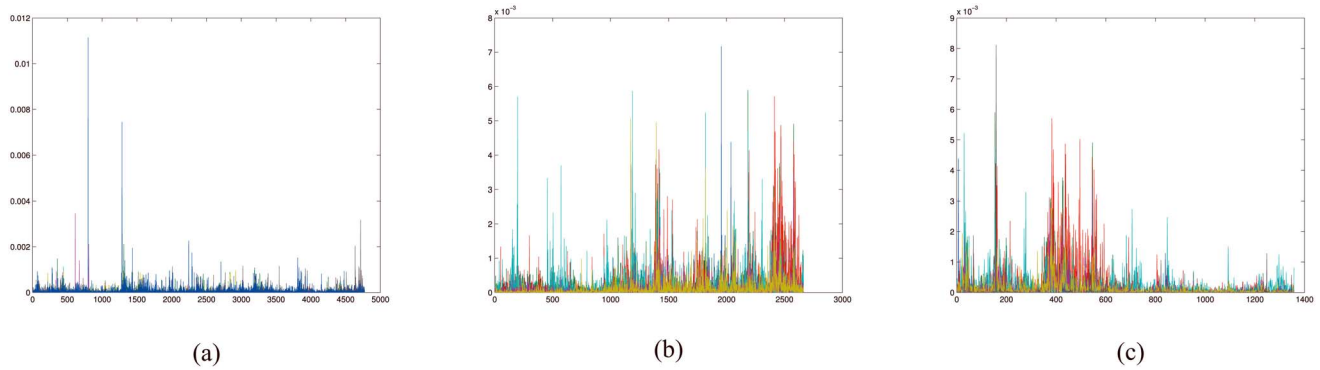

Fig. 1. Squared return series considered in our experiments. (a) First scenario. (b) Second scenario. (c) Third scenario.

and we consider that the $\zeta_{ij}(\boldsymbol{x})$ are given by

$$\zeta_{ij}(\boldsymbol{x}) = \xi(\gamma_{ij}(\boldsymbol{x})), \tag{78}$$

where the $\gamma_{ij}(\boldsymbol{x})$ are trainable real-valued models, and $\xi(\cdot)$ is a link function ensuring that the values of $\zeta_{ij}(\boldsymbol{x})$ will always be within the range allowed by the copula model employed each time. For instance, if a Clayton copula $C(\cdot)$ is employed, it is required that its parameter be positive, i.e. $\zeta_{ij}(\boldsymbol{x}) > 0$ [38]; in such a case, $\xi(\cdot)$ may be defined as the exponential function, i.e. $\xi(\alpha) = \exp(\alpha)$.

Note that the predictive posteriors $q(y_{*d})$ are difficult to compute analytically, since (68) does not yield a Gaussian distribution. For this reason, and in order to facilitate efficient training of the postulated pairwise conditional copula models, in the following we approximate (68) as a Gaussian with mean and variance given by (69) and (70), respectively.

Further, we consider the functions $\gamma_{ij}(\boldsymbol{x})$ to be linear basis functions models. Specifically, we postulate

$$\gamma_{ij}(\boldsymbol{x}) = \boldsymbol{w}_{ij}^T \boldsymbol{h}(\boldsymbol{x}), \tag{79}$$

where the $\boldsymbol{w}_{ij}$ are trainable model parameters, and the basis functions $\boldsymbol{h}(\boldsymbol{x})$ are defined using a small set of basis input observations $\{\boldsymbol{x}_i\}_{i=1}^I$, and an appropriate kernel function $\tilde{k}$:

$$\boldsymbol{h}(\boldsymbol{x}) \triangleq [\tilde{k}(\boldsymbol{x}, \boldsymbol{x}_i)]_{i=1}^I. \tag{80}$$

Training for the postulated pairwise conditional copula models can be performed by optimizing the logarithm of the copula density function that corresponds to the parametric conditional model (77), given a set of training data $\mathcal{D} = (\boldsymbol{x}_n, \boldsymbol{y}_n)_{n=1}^N$, which yields

$$\mathcal{P}_{ij} = \sum_{n=1}^{N} \log c\left(F_i(y_{ni}|\boldsymbol{x}_n), F_j(y_{nj}|\boldsymbol{x}_n)|\xi\left(\boldsymbol{w}_{ij}^T \boldsymbol{h}(\boldsymbol{x}_n)\right)\right) \tag{81}$$

with respect to the parameter vectors $\boldsymbol{w}_{ij}$. To effect this procedure, in this paper we resort to the L-BFGS algorithm [31].

TABLE 1
First Scenario: Obtained RMSEs Based on the Percentage Returns

Prediction Horizon	1-step	7-step	30-step	Average
GARCH	0.0705	0.0706	0.0715	0.0709
mixGARCH	0.0625	0.0625	0.0628	0.0625
VHGP	0.0146	0.0147	0.0147	0.0146
GPMCH	0.0121	0.0121	0.0122	0.0121

The procedure we use to perform model training is widely known as "inference function for margins" (IFM) [41]; it generally comprises two steps: on the first step, the marginal model is maximized with respect to its entailed (marginal) parameters, while, in the second step, the copula model is maximized with respect to the entailed (copula) parameters, using the marginal estimates obtained from the first step. This way, model estimation becomes computationally efficient, while comparison of different copulas can also be conducted in a convenient way, by means of standard methodologies for assumption testing. Of course, such an approximate training procedure naturally results in some information loss. However, IFM has been shown to yield good quality results for very attractive computational costs, especially when working with large training datasets [41].

$$\begin{aligned}\mathbb{V}\left[y_{*i}, y_{*j}|\boldsymbol{x}_*; \mathcal{D}\right] &= \int\int \Big[C\left(F_i(\kappa|\boldsymbol{x}_*), F_j(\kappa'|\boldsymbol{x}_*)|\xi\left(\boldsymbol{w}_{ij}^T \boldsymbol{h}(\boldsymbol{x}_*)\right)\right) \\ &\quad - F_i(\kappa|\boldsymbol{x}_*)F_j(\kappa'|\boldsymbol{x}_*)\Big] \mathrm{d}\kappa \mathrm{d}\kappa'.\end{aligned} \tag{82}$$

After training the postulated pairwise models $C(u_i, u_j|\zeta_{ij}(\boldsymbol{x}))$ $\forall i \neq j$, computation of the predictive covariance $\mathbb{V}\left[y_{*i}, y_{*j}|\boldsymbol{x}_*; \mathcal{D}\right]$ between the ith and the jth model output given the input observation $\boldsymbol{x}_*$ can be conducted using the corresponding conditional copula model and marginal predictive densities. Specifically, from Hoeffding's lemma [42]–[44], we directly obtain [Eq. (82)]; this latter integral can be approximated by means of numerical analysis methods.

Note that, in deriving the proposed approach, we were interested in modeling bivariate marginals separately. As such, there is no enforcement that the result is a coherent probabilistic model. This lack of our approach could result in a loss of statistical efficiency by not conditioning on the constraints enforcing coherence. On the other hand, there are significant computational gains; in addition, the resulting estimator might still be better than a fully coherent model if there is severe mispecification.

4 Experimental Evaluation

In this section, we elaborate on the application of our GPMCH approach to volatility modeling for financial return series data. We perform an experimental evaluation of its performance in volatility modeling, and examine how

TABLE 2
Second Scenario: Obtained RMSEs Based on the Percentage Returns

Prediction Horizon	1-step	7-step	30-step	Average
GARCH	0.2785	0.28	0.2863	0.2824
mixGARCH	0.2623	0.2636	0.2690	0.2656
VHGP	0.0552	0.0554	0.0563	0.0558
GPMCH	0.0360	0.0362	0.0368	0.0364

TABLE 3
Third Scenario: Obtained RMSEs Based on the Percentage Returns

Prediction Horizon	1-step	7-step	30-step	Average
GARCH	0.0552	0.0567	0.0613	0.0586
mixGARCH	0.0550	0.0556	0.0560	0.0556
VHGP	0.0542	0.0549	0.0562	0.0554
GPMCH	0.0345	0.0349	0.0354	0.0351

it compares to state-of-the-art competitors. We also assess the efficacy of the proposed copula-based approach for learning the predictive covariances between the modeled output variables of the GPMCH model.

For this purpose, we consider modeling the daily return series of various financial indices, including currency exchange rates, global large-cap equity indices, and Euribor rates. We note that, in this work, asset return $r(t)$ is defined as the difference between the logarithm of the prices $p(t)$ in two subsequent time points, i.e., $r(t) \triangleq \log p(t) - \log p(t-1)$. All our source codes were developed in MATLAB R2012a.

4.1 Volatility Prediction Using the GPMCH Model: Real Assets

In this set of experiments, we consider three application scenarios:

- In the first scenario, we model the return series pertaining to the following *currency exchange rates*, over the period December 31, 1979 to December 31, 1998 (daily closing prices):
  1) (AUD) Australian Dollar / U.S. $
  2) (GBP) U.K. Pound / U.S. $
  3) (CAD) Canadian Dollar / U.S. $
  4) (DKK) Danish Krone / U.S. $
  5) (FRF) French Franc / U.S. $
  6) (DEM) German Mark / U.S. $
  7) (JPY) Japanese Yen / U.S. $
  8) (CHF) Swiss Franc / U.S. $.
- In the second scenario, we model the return series pertaining to the following *global large-cap equity indices*, for the business days over the period April 27, 1993 to July 14, 2003 (daily closing prices):
  1) (TSX) Canadian TSX Composite
  2) (CAC) French CAC 40
  3) (DAX) German DAX
  4) (NIK) Japanese Nikkei 225
  5) (FTSE) U.K. FTSE 100
  6) (SP) U.S. S&P 500.
- Finally, in the third scenario, we model the return series pertaining to the following seven *global large-cap equity indices* and *Euribor rates*, for the business days over the period February 7, 2001 to April 24, 2006 (daily closing prices for the first 6 indices, and annual percentage rate converted to daily effective yield for the last index):
  1) (TSX) Canadian TSX Composite
  2) (CAC) French CAC 40
  3) (DAX) German DAX
  4) (NIK) Japanese Nikkei 225
  5) (FTSE) U.K. FTSE 100
  6) (SP) U.S. S&P 500
  7) (EB3M) Three-month Euribor rate.

These series have become standard benchmarks for assessing the performance of volatility prediction algorithms [23], [45], [46]. We provide an illustration of the squares of the considered return series in Fig. 1.

In all the considered scenarios, the proposed GPMCH model is trained using as input data, $\boldsymbol{x}(t)$, vectors containing the daily returns of all the assets considered in each scenario. The corresponding training output data $\boldsymbol{y}(t)$ essentially comprise the same series of input vectors shifted one-step ahead. In other words, the output series are defined as $\boldsymbol{y}(t) \triangleq \boldsymbol{r}(t+1),\ t > 0$, and the input series as $\boldsymbol{x}(t) \triangleq \boldsymbol{r}(t),\ t < T$, where T is the total duration of the modeled return series, and the vectors $\boldsymbol{r}(t)$ contain the return values of all the considered indices at time t.

In our experiments, we evaluate the GPMCH model using zero kernels for the mean process, i.e. $k^c(\boldsymbol{x}, \boldsymbol{x}') = 0\ \forall c$; this construction allows for our model to remain consistent with the existing literature, where it is typically considered that the modeled return series constitute a zero-mean noise-only process, i.e. $\langle f_d^c(\boldsymbol{x}) \rangle = 0\ \forall d, c$. Note though that our approach can seamlessly deal with learning the mean process $f_d^c(\boldsymbol{x})$, if a model for its covariance is available. Further, we consider autoregressive kernels of order one for the noise variance process of the model, of the form

$$\lambda^c(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma_0^2}{(1-\phi^2)} \phi^{||x-x'||}, \tag{83}$$

where the ϕ and σ_0^2 are model hyperparameters, estimated by means of free energy optimization (using the L-BFGS algorithm).

To obtain some comparative results, we also evaluate: (i) a common baseline approach from the field of financial engineering and econometrics, namely the GARCH(1,1) model [3], that is a GARCH model with volatility terms of order one and residual terms of order one; (ii) the Mixed Normal Conditional Heteroscedasticity (mix-GARCH(1,1)) approach of [14]; and (iii) the recently proposed VHGP approach of [21]. All these approaches have been shown to be very competitive in the task of volatility prediction in financial return series [21], [47]. Note that the GARCH(1,1) and mix-GARCH(1,1) models use as input the time variable, while the VHGP model is trained similar to GPMCH.

In our experiments, we follow an evaluation protocol similar to [46]: all the evaluated methods are trained using a rolling window of the previous 120 days of returns to

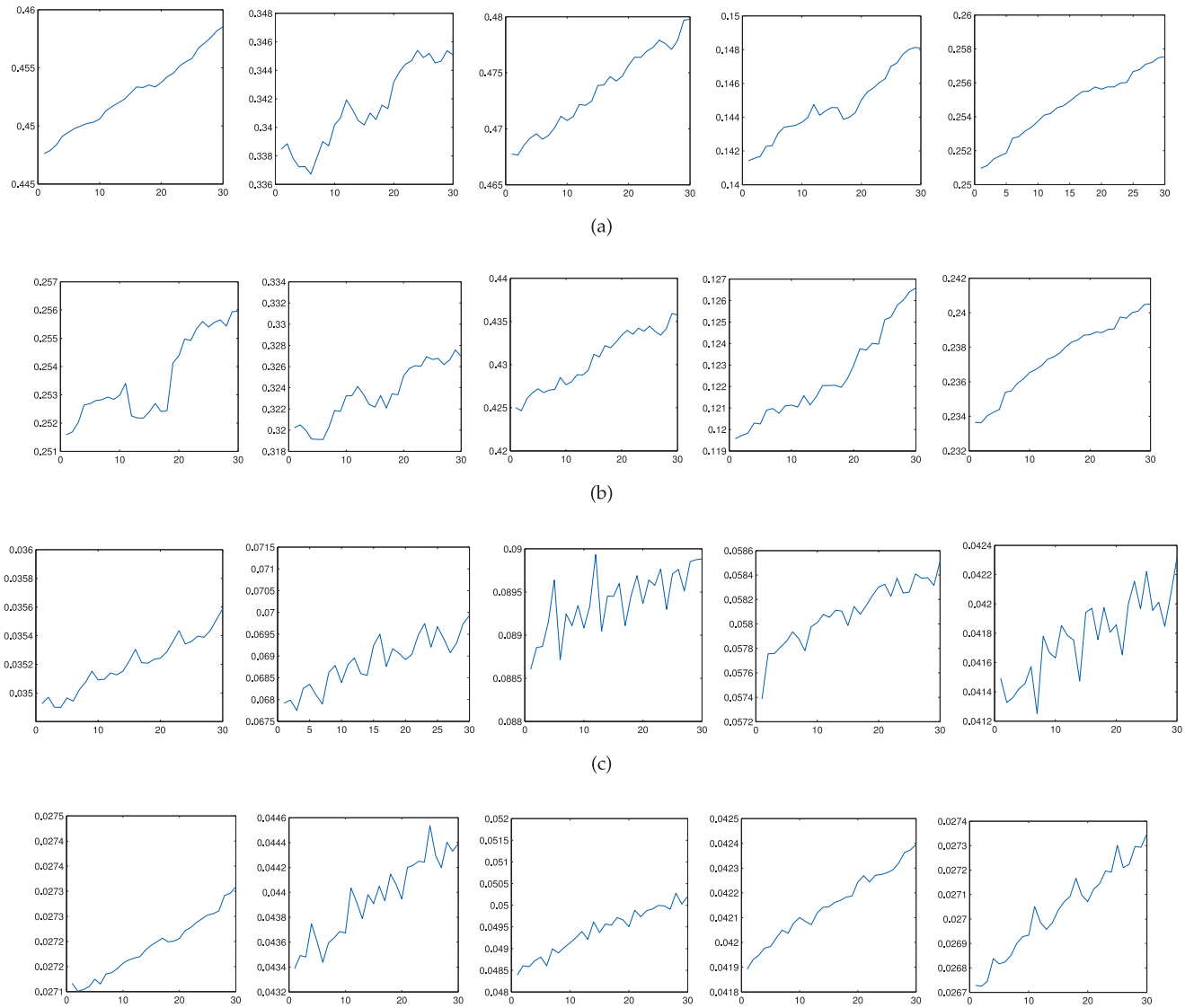

Fig. 2. Second scenario: Obtained RMSEs based on the percentage returns. (a) GARCH. (b) mixGARCH. (c) VHGP. (d) GPMCH.

make volatility forecasts up to 30 days ahead; we retrain the models every 7 days. As our performance metric used to evaluate the considered algorithms, we consider the root mean squared error (RMSE) between the model-estimated volatilities and the squared returns of the modeled return series. As discussed in [48], this groundtruth measurement constitutes one of the few consistent ways of volatility measuring; the same performance measure was employed in [14].

In Tables 1–3, we provide the obtained results for the three considered scenarios. These results are computed over all the assets modeled in each scenario. In addition, to show how performance fluctuates as a function of the prediction horizon, we further elaborate on the second scenario: in Fig. 2, we depict the obtained RMSEs as a function of the prediction horizon, separately for each asset. As we observe, our method works clearly better than the competition for all the modeled assets throughout the

TABLE 4
First Scenario: Obtained RMSEs Considering Comparison Against the Asset Pair Return Products

Prediction Horizon	1-step	7-step	30-step	Average
CCC-MVGARCH	0.0345	0.0345	0.0346	0.0346
BEKK	0.0341	0.0341	0.0342	0.0341
GPMCH: *Clayton*	0.0341	0.0342	0.0342	0.0342
GPMCH: *Frank*	0.0341	0.0341	0.0342	0.0342
GPMCH: *Gumbel*	0.0341	0.0341	0.0341	0.0341

TABLE 5
Second Scenario: Obtained RMSEs Considering Comparison Against the Asset Pair Return Products

Prediction Horizon	1-step	7-step	30-step	Average
CCC-MVGARCH	0.1183	0.1183	0.1183	0.1183
BEKK	0.1304	0.1304	0.1304	0.1304
GPMCH: *Clayton*	0.0557	0.0557	0.0557	0.0557
GPMCH: *Frank*	0.0566	0.0566	0.0566	0.0566
GPMCH: *Gumbel*	0.0557	0.0557	0.0557	0.0557

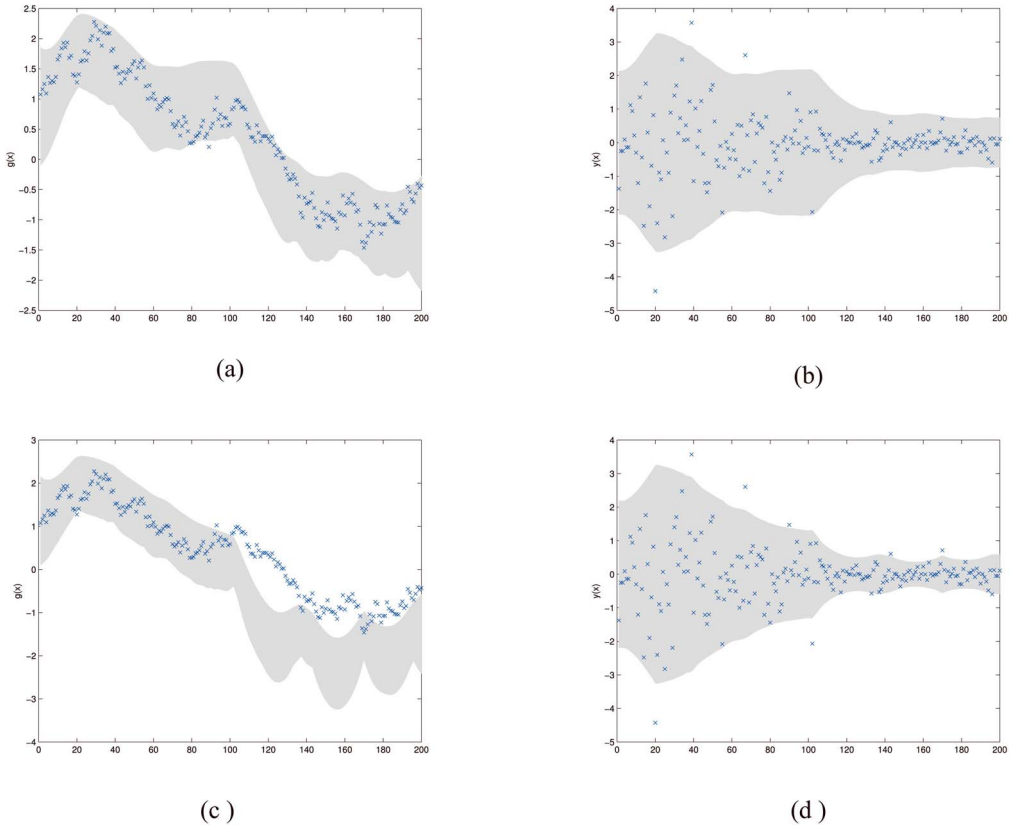

Fig. 3. Volatility prediction: Synthetic experiments: (a) GPMCH: posteriors over *g(x)*. (b) GPMCH: posteriors over *y*. (c) VHGP: posteriors over *g(x)*. (d) VHGP: posteriors over *y*.

considered 30-step prediction horizon. We also observe a clear trend of prediction error increase for longer prediction horizons in all assets. Similar results are obtained for the rest of the considered experimental cases (omitted for brevity).

4.2 Volatility Prediction Using the GPMCH model: Synthetic Experiments

In this experiment, we evaluate our method using synthetic data with availability of groundtruth predictions. This way, we allow for gaining better insights into the modeling capacity of our approach, and how it compares to existing alternatives. Specifically, for this purpose, we use a synthetic dataset of 200 data points generated by Girolami and Calderhead in [49]. This dataset was obtained by sampling from a VHGP model (i.e., an GPMCH with one mixture component), where the latent function is set to zero, i.e., $k(x, x') = 0$, the mean of $g(x)$ is of the form $\tilde{m} = 2\log\beta$, and the kernel of $g(x)$ is of the form $\lambda(x, x') = \frac{\sigma_0^2}{1-\phi^2}\phi^{|x-x'|}$. Specifically, in the considered datasets, the chosen values of the model hyperparameters were $\sigma_0 = 0.15,\ \phi = 0.98,\ \beta = 0.65$.

To make the task harder and more realistic, we further contaminated this dataset with white noise. We used the so-obtained distorted data to perform training of our GPMCH model as well as the related VHGP model. In Figs. 3(a)–(d), we provide the obtained posteriors over $g(x)$ and the outputs y of the model; in these figures, the shaded area illustrates the variance of the depicted posterior distributions. As we observe, our model obtains much better accuracy, especially in terms of the obtained posteriors over $g(x)$. Finally, in Fig. 4 we provide the posterior over model components obtained by our method; specifically, we show how many data points are effectively assigned to each component based on the MAP criterion. As we observe, we eventually obtained 4 (effective) model components; all the rest were empty.

4.3 Copula-Based Modeling of the Covariances between Asset Returns

Here, we evaluate the performance of the proposed copula-based approach for learning a predictive model of the covariances between the GPMCH-modeled asset returns.

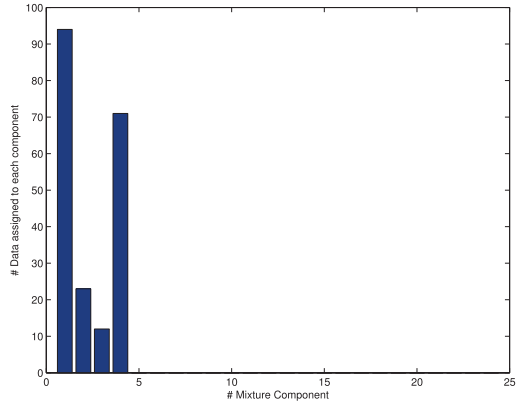

Fig. 4. Volatility prediction: Synthetic experiments. Posterior over model components.

	1	2	3	4	5	6	7	8
1	1	0.2307	0.1929	0.1808	0.1871	0.1852	0.1987	0.1758
2	0.2307	1	0.1983	0.7349	0.7148	0.7372	0.4657	0.7024
3	0.1929	0.1983	1	0.1616	0.1675	0.1641	0.0915	0.1546
4	0.1808	0.7349	0.1616	1	0.9406	0.9888	0.5995	0.9162
5	0.1871	0.7148	0.1675	0.9406	1	0.9464	0.5801	0.8789
6	0.1852	0.7372	0.1641	0.9888	0.9464	1	0.6084	0.9225
7	0.1987	0.4657	0.0915	0.5995	0.5801	0.6084	1	0.6142
8	0.1758	0.7024	0.1546	0.9162	0.8789	0.9225	0.6142	1

(a)

	1	2	3	4	5	6
1	1	0.4403	0.4735	0.1736	0.4334	0.6855
2	0.4403	1	0.7597	0.2233	0.7809	0.4343
3	0.4735	0.7597	1	0.2237	0.6978	0.4926
4	0.1736	0.2233	0.2237	1	0.2375	0.1047
5	0.4334	0.7809	0.6978	0.2375	1	0.4289
6	0.6855	0.4343	0.4926	0.1047	0.4289	1

(b)

	1	2	3	4	5	6	7
1	1	0.4813	0.5058	0.1854	0.4573	0.6526	0.0555
2	0.4813	1	0.8485	0.2261	0.8575	0.5102	0.0653
3	0.5058	0.8485	1	0.2001	0.7650	0.6136	0.0476
4	0.1854	0.2261	0.2001	1	0.2295	0.1439	0.0570
5	0.4573	0.8575	0.7650	0.2295	1	0.4617	0.0576
6	0.6526	0.5102	0.6136	0.1439	0.4617	1	0.0365
7	0.0555	0.0653	0.0476	0.0570	0.0576	0.0365	1

(c)

Fig. 5. Pearson correlation coefficient between the modeled assets in the considered experimental scenarios. (a) First Scenario. (b) Second Scenario. (c) Third Scenario.

For this purpose, we repeat the previous experimental scenarios, with the goal now being to obtain predictions regarding the covariances between the assets modeled each time. To obtain an impression of how strongly correlated the modeled assets are in each scenario, we provide the Pearson correlation coefficient over the modeled assets in Fig. 5. As we observe, most of the modeled assets in our scenarios are found to be moderately to strongly correlated.

In our experiments, we consider application of three popular Archimedean copula types, namely *Clayton, Frank,* and *Gumbel* copulas [38]. The employed GPMCH models are trained similar to the previous experiments. The postulated conditional-copula pairwise models use a basis set of input observations (to compute the $\boldsymbol{h}(\boldsymbol{x})$ in (80)) that comprises the 10% of the available training data points, i.e. 12 data points sampled at regular time intervals (one sample every 10 days).

To obtain some comparative results, we also evaluate the performance of two state-of-the-art methods used for modeling dynamic covariance matrices (multivariate volatility) for high-dimensional vector-valued observations; specifically, we consider the CCC-MVGARCH(1,1) approach of [50], and the GARCH-BEKK(1,1) method of [51]. As our evaluation metric, we use the products of the returns of the corresponding asset pairs at each time point. Our obtained results are depicted in Tables 4–6. We observe that our approach yields a very competitive result: specifically, in two out of the three considered scenarios, the yielded improvement was equal to or exceeded one order of magnitude, while, in one case, all methods yielded comparable results. We also observe that switching the employed Archimedean copula type had only marginal effects on model performance, in all our experiments.

5 Conclusions

In this paper, we proposed a novel nonparametric Bayesian approach for modeling conditional heteroscedasticity in financial return series. Our approach consists in the postulation of a mixture of Gaussian process regression models, each component of which models the noise variance process that contaminates the observed data as a separate latent Gaussian process driven by the observed data. We imposed a nonparametric prior with power-law nature over the distribution of the model mixture components, namely the Pitman-Yor process prior, to allow for better capturing modeled data distributions with heavy tails and skewness. In addition, in order to provide a predictive posterior for the covariances over the modeled asset returns, we devised a copula-based covariance modeling procedure built on top of our model. To assess the efficacy of our approach, we applied it to several asset return series, and compared its performance to several state-of-the-art methods in the field, on the grounds of standard evaluation metrics. As we observed, our approach yields a clear performance improvement over its competitors in all the considered scenarios.

TABLE 6
Third Scenario: Obtained RMSEs Considering Comparison Against the Asset Pair Return Products

Prediction Horizon	1-step	7-step	30-step	Average
CCC-MVGARCH	6.63	6.71	6.86	6.71
BEKK	7.41	7.54	7.55	7.50
GPMCH: *Clayton*	0.9905	0.9905	0.9925	0.9915
GPMCH: *Frank*	0.991	0.991	0.991	0.991
GPMCH: *Gumbel*	0.9905	0.991	0.991	0.991

References

[1] L. Chollete, A. Heinen, and A. Valdesogo, "Modeling international financial returns with a multivariate regime switching copula," *J. Financ. Econ.*, vol. 7, no. 4, pp. 437–480, 2009.

[2] R. Engle, "Autoregressive conditional heteroskedasticity models with estimation of variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, Jul. 1982.

[3] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econ.*, vol. 94, pp. 238–276, Apr. 1986.

[4] D. Gu and H. Hu, "Spatial Gaussian process regression with mobile sensor networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1279–1290, Aug. 2012.

[5] C. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts," in *Proc. NIPS 14*, 2002, pp. 881–888.

[6] E. Meeds and S. Osindero, "An alternative infinite mixture of Gaussian process experts," in *Proc. NIPS 18*, 2006, pp. 883–890.

[7] L. Xu, M. I. Jordan, and G. E. Hinton, "Analternative model for mixtures of experts," in *Proc. NIPS 7*, 1995, pp. 633–640.

[8] S. Walker, P. Damien, P. Laud, and A. Smith, "Bayesian nonparametric inference for random distributions and related functions," *J. Roy. Statist. Soc.*, ser. B, vol. 61, no. 3, pp. 485–527, 1999.

[9] R. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *J. Comput. Graph. Statist.*, vol. 9, no. 2, pp. 249–265, 2000.

[10] P. Muller and F. Quintana, "Nonparametric Bayesian data analysis," *Statist. Sci.*, vol. 19, no. 1, pp. 95–110, 2004.

[11] J. Pitman and M. Yor, "The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator," *Ann. Probab.*, vol. 25, no. 2, pp. 855–900, Apr. 1997.

[12] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proc. 21st Int. Conf. ACL*, Stroudsburg, PA, USA, 2006, pp. 985–992.

[13] C. Alexander and E. Lazar, "Normal mixture GARCH(1,1): Applications to exchange rate modelling," *J. Appl. Econ.*, vol. 21, no. 3, pp. 307–336, 2006.

[14] M. Haas, S. Mittnik, and M. Paolella, "Mixed normal conditional heteroskedasticity," *J. Financ. Econ.*, vol. 2, no. 2, pp. 211–250, 2004.

[15] L. Bauwens, C. M. Hafner, and J. V. K. Rombouts, "Multivariate mixed normal conditional heteroskedasticity," *Comput. Stat. Data Anal.*, vol. 51, no. 7, pp. 3551–3566, Apr. 2007.

[16] M. Haas, S. Mittnik, and M. Paolella, "Asymmetric multivariate normal mixture garch," *Comput. Stat. Data Anal.*, vol. 53, no. 6, pp. 2129–2154, Apr. 2009.
[17] T. Ferguson, "A Bayesian analysis of some nonparametric problems," *Ann. Stat.*, vol. 1, no. 2, pp. 209–230, Mar. 1973.
[18] D. Blackwell and J. MacQueen, "Ferguson distributions via Pólya urn schemes," *Ann. Stat.*, vol. 1, no. 2, pp. 353–355, Mar. 1973.
[19] J. Sethuraman, "A constructive definition of the Dirichlet prior," *Statistica Sinica*, vol. 2, no. 4, pp. 639–650, 1994.
[20] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[21] M. Lázaro-Gredilla and M. Tsitsias, "Variational heteroscedastic Gaussian process regression," in *Proc. 28th Int. Conf. Machine Learning*, Bellevue, WA, USA, 2011.
[22] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian processes regression," in *Proc. ICML*, New York, NY, USA, 2007, pp. 393–400.
[23] C. Brooks, S. Burke, and G. Persand, "Benchmarks and the accuracy of GARCH model estimation," *Int. J. Forecasting*, vol. 17, pp. 45–56, Jan.–Mar. 2001.
[24] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Anal.*, vol. 1, no. 1, pp. 121–144, 2006.
[25] Y. Qi, J. W. Paisley, and L. Carin, "Music analysis using hidden Markov mixture models," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5209–5224, Nov. 2007.
[26] W. Fan, N. Bouguila, and D. Ziou, "Variational learning for finite Dirichlet mixture models and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 762–774, May 2012.
[27] A. Penalver and F. Escolano, "Entropy-based incremental variational Bayes learning of Gaussian mixtures," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 534–540, May 2012.
[28] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, Nov. 1999.
[29] S. Chatzis, D. Kosmopoulos, and T. Varvarigou, "Signal modeling and classification using a robust latent space model based on *t* distributions," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 949–963, Mar. 2008.
[30] D. Chandler, *Introduction to Modern Statistical Mechanics*. New York, NY, USA: Oxford University Press, 1987.
[31] D. Liu and J. Nocedal, "On the limited memory method for large scale optimization," *Math. Prog.*, ser. B, vol. 45, no. 3, pp. 503–528, 1989.
[32] P. Boyle and M. Frean, "Dependent Gaussian processes," in *Proc. NIPS*, 2005, pp. 217–224.
[33] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *Proc. 22nd ICML*, Bonn, Germany, 2005, pp. 1012–1019.
[34] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *Proc. NIPS*, 2008.
[35] M. Alvarez and N. Lawrence, "Sparse convolved multiple output Gaussian processes," in *Proc. NIPS*, 2008, pp. 57–64.
[36] L. Bo and C. Sminchisescu, "Twin Gaussian processes for structured prediction," *Int. J. Comput. Vision*, vol. 87, no. 1–2, pp. 28–52, Mar. 2010.
[37] A. Sklar, "Functions de repartition à n dimensions et luers marges," *Publications de l'Institut de Statistique de l'Universitè de Paris*, vol. 8, pp. 229–231, 1959.
[38] R. Nelsen, *An Introduction to Copulas*. New York, NY, USA: Springer, 2006.
[39] F. Abegaz and U. Naik-Nimbalkar, "Modeling statistical dependence of Markov chains via copula models," *J. Stat. Plan. Inference*, vol. 138, pp. 1131–1146, Apr. 2008.
[40] H. Joe, *Multivariate Models and Dependence Concepts*. London, U.K.: Chapman and Hall, 1997.
[41] H. Joe, "Asymptotic efficiency of the two-stage estimation method for copula-based models," *J. Multivar. Anal.*, vol. 94, no. 2, pp. 401–419, Jun. 2005.
[42] W. Hoeffding, "Masstabinvariante korrelations theorie," *Schr. Math. Inst. Univ. Berlin*, vol. 5, no. 3, pp. 179–233, 1940.
[43] H. Block and Z. Fang, "A multivariate extension of Hoeffding's lemma," *Ann. Probab.*, vol. 16, pp. 1803–1820, 1988.
[44] C. M. Cuadras, "Correspondence analysis and diagonal expansions in terms of distribution functions," *J. Stat. Plan. Inference*, vol. 103, pp. 137–150, Apr. 2002.
[45] B. McCullough and C. Renfro, "Benchmarks and software standards: A case study of GARCH procedures," *J. Econ. Social Meas.*, vol. 25, no. 2, pp. 59–71, 1998.
[46] A. G. Wilson and Z. Ghahramani, "Copula processes," in *Proc. NIPS*, Vancouver, BC, USA, 2010.
[47] P. R. Hansen and A. Lunde, "A forecast comparison of volatility models: Does anything beat a GARCH(1,1)," *J. Appl. Econ.*, vol. 20, no. 7, pp. 873–889, 2005.
[48] C. T. Brownlees, R. F. Engle, and B. T. Kelly. (2009). *A Practical Guide to Volatility Forecasting Through Calm and Storm* [Online]. Available http://ssrn.com/abstract=1502915
[49] M. Girolami and B. Calderhead, "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *J. Roy. Stat. Soc.*, ser. B, vol. 73, no. 2, pp. 1–37, 2011.
[50] R. F. Engle, "Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models," *J. Bus. Econ. Stat.*, vol. 20, no. 3, pp. 339–350, Jul. 2002.
[51] R. F. Engle and K. F. Kroner, "Multivariate simultaneous generalized ARCH," *Econ. Theory*, vol. 11, no. 1, pp. 122–150, Mar. 1995.

Emmanouil A. Platanios conducted this work during the Internship with the Department of Electrical Engineering, Computer Engineering, and Informatics, Cyprus University of Technology, Limassol, Cyprus. He is currently pursuing the Ph.D. degree with the Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA. He received the M.E. degree in electrical and electronic engineering in 2013, with first-class honours, from Imperial College London, U.K. He is the co-founder of Holic Inc., a start-up company in Greece, the main product of which is an intelligent news reader application. He has served as a research director of Holic Inc., directing machine learning-related research. He has been with novel algorithms for news articles classification and clustering, and on specialized ranking of news providers.

Sotirios P. Chatzis is a lecturer (U.S. equivalent: assistant professor) with the Department of Electrical Engineering, Computer Engineering and Informatics, Cyprus University of Technology, Limassol, Cyprus. His current research interests include machine learning theory and methodologies with a special focus on hierarchical Bayesian models, Bayesian nonparametrics, quantum statistics, and neuroscience. He received the M.E. degree in electrical and computer engineering with distinction from the National Technical University of Athens, Greece, in 2005, and the Ph.D. degree in machine learning, in 2008, from the same institution. From January 2009 until June 2010, he was a post-doctoral fellow with the University of Miami, Miami, FL, USA. From June 2010 until August 2012, he was a post-doctoral researcher with the Department of Electrical and Electronic Engineering, Imperial College London, London, U.K. His Ph.D. research was supported by the Bodossaki Foundation and the Greek Ministry for Economic Development. He was also awarded the Dean's Scholarship for Ph.D. studies, being the best performing Ph.D. student in his class. In his first seven years as a researcher, he has authored more than 40 papers in the most prestigious journals and conferences of his research field.