RECURRENT LATENT VARIABLE CONDITIONAL HETEROSCEDASTICITY

Sotirios P. Chatzis

Department of Electrical Eng., Computer Eng., and Informatics
Cyprus University of Technology
Limassol 3036, Cyprus

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 signals. However, this family of methods employ quite rigid assumptions regarding the evolution of the variance. In this paper, we address these issues by introducing a recurrent latent variable model, capable of capturing highly flexible functional relationships for the variances. We derive a fast, scalable, and robust to overfitting Bayesian inference algorithm, by relying on amortized variational inference. This avoids the need to compute per-data point variational parameters, but can instead compute a set of global variational parameters valid for inference at both training and test time. We evaluate the efficacy of our approach in a number of benchmarks, and compare its performance to state-of-the-art methodologies.

***Index Terms*—** Amortized variational inference, conditional heteroscedasticity, latent variable models, volatility prediction.

1. INTRODUCTION

Statistical modeling of asset value signals in financial markets requires taking into account the tendency of assets towards asymmetric temporal dependence [1]. Indeed, it has been well-established that the data generation processes of the *returns* of financial market indices (i.e., the changes in the log price over a specified period) may be non-linear, non-stationary and/or heavy-tailed, while the marginal distributions may be asymmetric, leptokurtic and/or show conditional heteroscedasticity. The heteroscedastic nature of financial return signals refers to the intrinsic property of their variance *(volatility)* to be time-dependent: large returns (either positive or negative) are often followed by returns that are also large in size.

The generalized autoregressive conditional heteroscedasticity (GARCH) family of models is the most popular and extensively examined means of capturing heteroscedasticity in financial return signals [2, 3]. 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 successful in both volatility prediction based on daily returns, as well as on predictions using intraday information (realized volatility). However, the GARCH family of methods is plagued by one fundamental design problem, that cannot be addressed unless a new paradigm is sought: GARCH-type models make a specific assumption of what the functional dynamics of the volatility look like. In reality, this functional form is completely unknown. Hence, a new modeling paradigm is needed, that will be capable of *inferring this functional form* from the data.

Recently, few researchers have attempted to address these issues by resorting to methodologies developed by the machine learning community. Specifically, [4] introduced a Gaussian Process (GP)-volatility model, where a GP prior [5] is employed to infer the relationship between the time-varying variance, σ_t^2, and the previous variance values and return time-series values. In the same vein, [6] introduced a GP-mixture conditional heteroscedasticity (GPMCH) model, where a nonparametric mixture of GPs is employed. Both approaches have been shown to completely outperform GARCH-type models; this corroborates the need of relaxing the restrictive assumptions of GARCH.

Despite these advances, GP-based models are also well-known for several shortcomings, namely: (i) The kernel function employed by the imposed GP priors gives rise to a rigid assumption on the form of the dependencies between the training data points. Hence, a suboptimal selection of the kernel function might result in eventually yielding a poor trained model. (ii) Posterior computation for a GP-based model using N data points imposes computational costs of $\mathcal{O}(N^3)$; these stem from the inversion of a large *gram* matrix. Similar computations result in the predictive density imposing a complexity of $\mathcal{O}(N^2)$. Such a complexity may be prohibitive in real-world application scenarios. In addition, *sparse approximations* of the GP prior, e.g. FITC [7], developed for alleviating these issues, introduce a high number of extra (hyper-)parameters that must be optimized as part of the model training procedure. Naturally, this increases the tendency of the model to *overfitting*, and may undermine the eventually obtained predictive performance.

To address these issues, in this work, for the first time in the literature, we introduce a generative, recurrent latent variable model for conditional heteroscedasticity modeling. The proposed Recurrent Latent Variable Conditional Heteroscedasticity (ReLaVaCH) model is a generative model that postulates a conditional dependency of the return time-series upon a set of latent variables. We impose an intricate prior distribution over the vector of these latent variables that is driven from a high-dimensional nonlinear representation of the observed return values and latent variable values at the previous time points. On the basis of this construction, we obtain a flexible latent variable posterior, which does *not* rely on restrictive assumptions, and imposes computational costs *linear* to the training data (i.e., $\mathcal{O}(N)$).

Specifically, to this end, we exploit recent advances in the field of *amortized variational inference* (AVI) [8, 9, 10]. AVI represents the sought (approximate) variational posterior distribution over the model latent variables via an inference network, which learns an inverse map from observations to latent variables. This allows for capturing much more complex functional forms of the variational posteriors than standard approaches. In addition, it also alleviates the need to compute per data point variational parameters; instead, we compute a set of global variational parameters, valid for inference at both training and test time. Thus, the cost of inference is amortized by generalizing between the posterior estimates for all latent variables through the parameters of the inference network, under a simple feedforward computation scheme with complexity $\mathcal{O}(N)$.

The remainder of this paper is organized as follows: In Section 2, we provide an overview of the theoretical foundation of our work. In Section 3, we present the proposed ReLaVaCH model, and derive its inference and learning algorithm expressions. In Section 4, we perform the experimental evaluation of our approach, under a variety of real-world modeling scenarios, and obtain some comparative results against the state-of-the-art. Finally, Section 5 concludes this paper.

2. AMORTIZED VARIATIONAL INFERENCE

Let us consider a dataset $X = \{\boldsymbol{x}_n\}_{n=1}^N$ consisting of N samples of some observed random variable $\boldsymbol{x}$. We assume that the observed random variable is generated by some random process, involving an unobserved continuous random variable $\boldsymbol{z}$. In this context, we introduce a conditional independence assumption for the observed variables $\boldsymbol{x}$ given the corresponding latent variables $\boldsymbol{z}$; we adopt the conditional likelihood function $p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})$. To perform Bayesian inference for the postulated model, we impose some prior distribution $p(\boldsymbol{z};\boldsymbol{\varphi})$. Under this formulation, the log-marginal likelihood of the model w.r.t. the dataset X yields the following lower bound expression (evidence lower bound, ELBO)

$$\log p(X) \geq \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{\phi}|X) = \sum_{i=1}^{N} \Big\{ -\mathrm{KL}\big[q(\boldsymbol{z}_i;\boldsymbol{\phi})||p(\boldsymbol{z}_i;\boldsymbol{\varphi})\big] + \mathbb{E}_{q(\boldsymbol{z}_i;\boldsymbol{\phi})}[\log p(\boldsymbol{x}_i|\boldsymbol{z}_i;\boldsymbol{\theta})] \Big\} \tag{1}$$

where $\mathrm{KL}\big[q||p\big]$ is the KL divergence between the distribution $q(\cdot)$ and the distribution $p(\cdot)$, $q(\boldsymbol{z};\boldsymbol{\phi})$ is the sought approximate (variational) posterior over the latent variable $\boldsymbol{z}$, while $\mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\phi})}[\cdot]$ is the (posterior) expectation of a function w.r.t. the random variable $\boldsymbol{z}$, the distribution of which is taken to be the posterior $q(\boldsymbol{z};\boldsymbol{\phi})$.

AVI assumes that the adopted likelihood and prior distributions come from a parametric family, and that their probability density functions (pdf's) are differentiable almost everywhere w.r.t. the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$, and the (latent) variables $\boldsymbol{z}$. Specifically, AVI assumes that the likelihood function of the model, as well as the resulting latent variable posterior, $q(\boldsymbol{z};\boldsymbol{\phi})$, are parameterized via deep neural networks. This yields a non-conjugate model construction, which does not allow to analytically derive the expression of $\mathbb{E}_{q(\boldsymbol{z}_i;\boldsymbol{\phi})}[\log p(\boldsymbol{x}_i|\boldsymbol{z}_i;\boldsymbol{\theta})]$, and, hence, of the derivative of $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{\phi}|X)$. Besides, attempting to resolve this issue by means of a naive Monte Carlo gradient estimator is not an option in our context, due to its entailed prohibitively high variance that renders it completely impractical [11].

AVI resolves these issues by reparameterizing the random samples of $\boldsymbol{z} \sim q(\boldsymbol{z};\boldsymbol{\phi})$ using an appropriate differentiable transformation of an (auxiliary) random noise variable $\boldsymbol{\epsilon}$. Specifically, by drawing L samples, the ELBO expression becomes

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{\phi}|X) = \sum_{i=1}^{N} \Big\{ -\mathrm{KL}\big[q(\boldsymbol{z}_i;\boldsymbol{\phi})||p(\boldsymbol{z}_i;\boldsymbol{\varphi})\big] + \frac{1}{L}\sum_{l=1}^{L} \log p(\boldsymbol{x}_i|\boldsymbol{z}_i^{(l)};\boldsymbol{\theta}) \Big\} \tag{2}$$

where, considering a Gaussian posterior of the form

$$q(\boldsymbol{z}_i;\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_i), \mathrm{diag}\ \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x}_i)) \tag{3}$$

we have:

$$\boldsymbol{z}_i^{(l)} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_i) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}_i) \cdot \boldsymbol{\epsilon}_i^{(l)} \tag{4}$$

In Eq. (4), $\boldsymbol{\epsilon}_i^{(l)}$ is white random noise with unitary variance, i.e. $\boldsymbol{\epsilon}_i^{(l)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, the $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$ are parameterized via deep neural networks, and $\mathrm{diag}\ \boldsymbol{\chi}$ is a diagonal matrix with $\boldsymbol{\chi}$ on its main diagonal.

As we observe, the key difference between AVI and, say, a naive Monte Carlo estimator, is that the drawn samples of $\boldsymbol{z}$, used to approximate the intractable posterior expectation $\mathbb{E}_{q(\boldsymbol{z}_i;\boldsymbol{\phi})}[\log p(\boldsymbol{x}_i|\boldsymbol{z}_i;\boldsymbol{\theta})]$, are now taken as *functions of the parameters $\boldsymbol{\phi}$ of the posterior $q(\boldsymbol{z}_i;\boldsymbol{\phi})$ that we seek to optimize.* As proven in [8], this formulation of the inference

algorithm allows for yielding low variance estimators, under some mild conditions.

One limitation of AVI consists in the fact that, to allow for computational efficiency, the variational posterior distribution $q(\boldsymbol{z}_i;\boldsymbol{\phi})$ is assumed to be a diagonal Gaussian. Indeed, an ideal family of variational distributions $q(\boldsymbol{z}_i;\boldsymbol{\phi})$ would be one that is flexible enough to contain the true posterior as one solution. The principle of *normalizing flows* is a recently proposed feasible path towards this end [12]. It consists in: (i) postulating the *auxiliary* latent variables $\boldsymbol{z}'_i$, for which we consider that the Gaussian assumption regarding their posterior is accurate; and (ii) performing a series of *invertible* transforms, $\{f_k(\cdot)\}_{k=1}^K$, that converts the *auxiliary* latent variables $\boldsymbol{z}'_i$ to the originally postulated ones, $\boldsymbol{z}_i$, while obtaining a valid posterior distribution over them, $q(\boldsymbol{z}_i)$. The latter procedure is effected by application of the variable change rule, which eventually yields the posterior:

$$\log q(\boldsymbol{z}_i) = \log q(\boldsymbol{z}'_i) - \sum_k \log\,\det|\nabla f_k| \tag{5}$$

In Eq. (5), $\log\,\det|\nabla f_k|$ constitutes the log-determinant of the Jacobian of the (invertible) transform $f_k(\cdot)$. Since this computation may turn out to be of high complexity, [13] proposed a class of invertible transforms $f_k(\cdot)$ that alleviate the need of computing the Jacobian; these are referred to as *planar flows*, and read:

$$f(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{u}h(\boldsymbol{w}^T\boldsymbol{z} + b) \tag{6}$$

where the $\{\boldsymbol{u}, \boldsymbol{w}, b\}$ constitute a set of (trainable) hyperparameters, and $h(\cdot)$ is a smooth element-wise nonlinearity, with derivative $h'(\cdot)$. Under the scheme (6), the logdet-Jacobian term reduces to a simple linear time operation, which consists in computation of the quantity:

$$\xi(\boldsymbol{z}) = |1 + \boldsymbol{u}^T\psi(\boldsymbol{z})| \tag{7}$$

where

$$\psi(\boldsymbol{z}) = h'(\boldsymbol{w}^T\boldsymbol{z} + b)\boldsymbol{w} \tag{8}$$

3. PROPOSED APPROACH

Inspired from these advances in the field of AVI, as well as the recent ideas on variational inference using *planar normalizing flows*, we now proceed to the definition of the proposed ReLaVaCH model, and derivation of its inference and learning algorithm expressions. Let us consider a time-series signal of asset returns, $\{x_t\}_{t=1}^T$. We postulate a conditional independence assumption, where the conditioning variables $\boldsymbol{z}_t$ are some latent variables defined in a D-dimensional space with support in $\mathbb{R}$. Specifically, we postulate the conditional likelihood

$$x_t|\boldsymbol{z}_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_t^2) \tag{9}$$

where

$$\sigma_t^2 = g_{\boldsymbol{\theta}}(\boldsymbol{z}_t) \tag{10}$$

Table 1. Average predictive log-likelihood of the evaluated methods (the higher the better).

Equity Index	GARCH	GJR	GPMCH	ReLaVaCH
A	-1.328	-1.298	-1.280	**-1.264**
AA	-1.215	-1.223	-1.213	**-1.201**
AAPL	-1.222	-1.211	-1.211	**-1.198**
ABC	-1.352	-1.340	-1.322	**-1.311**
ABT	**-1.283**	**-1.283**	**-1.283**	**-1.283**
ACE	-1.070	-1.074	-1.067	**-1.060**
ADBE	-1.352	-1.393	-1.293	**-1.282**
ADI	-1.357	-1.334	-1.331	**-1.317**
ADM	-1.210	-1.210	-1.210	**-1.206**
ADP	-1.235	-1.219	-1.215	**-1.198**
ADSK	-1.028	-1.042	**-1.020**	-1.022
AEE	-1.283	-1.269	-1.159	**-1.140**
AEP	-1.138	-1.131	-1.130	**-1.121**
AES	-1.215	-1.215	-1.199	**-1.182**
AET	-1.268	-1.260	-1.243	**-1.228**
AFL	-1.044	-1.046	-1.109	**-1.024**
AGN	-1.257	-1.253	-1.256	**-1.249**
AIG	-1.142	-1.173	-1.055	**-1.005**
AIV	-1.021	-1.032	-1.003	**-1.002**
AIZ	-1.304	-1.336	-1.264	**-1.227**
AKAM	-1.343	-1.329	-1.342	**-1.302**
AKS	-1.211	-1.240	-1.182	**-1.158**
ALL	-1.250	-1.183	**-1.182**	-1.186
ALTR	-1.070	-1.067	-1.056	**-1.044**
AMAT	-1.223	-1.218	-1.235	**-1.211**

and $g_{\boldsymbol{\theta}}(\cdot)$ is a deep neural network (DN) comprising *rectified* linear units [14], with parameters set $\boldsymbol{\theta}$. Turning to the latent variables vector of the postulated generative model, we impose over it a prior density that allows for capturing the temporal dynamics of volatility in financial return series. We consider

$$\boldsymbol{z}_t \sim \mathcal{N}(\tilde{\boldsymbol{m}}_t, \text{diag}(\tilde{\boldsymbol{s}}_t^2)) \tag{11}$$

where

$$[\tilde{\boldsymbol{m}}_t; \tilde{\boldsymbol{s}}_t^2] = g_{\boldsymbol{\varphi}}(\boldsymbol{\rho}_{t-1}) \tag{12}$$

$[\boldsymbol{\alpha};\boldsymbol{\beta}]$ is the concatenation of two vectors, while $g_{\boldsymbol{\varphi}}(\cdot)$ is a DN comprising *rectified* linear units, with parameters set $\boldsymbol{\varphi}$. On the other hand, $\boldsymbol{\rho}_{t-1}$ is a *state vector* that encodes the history of observed return values, $\{x_\tau\}_{\tau=1}^{t-1}$, and inferred latent vector values, $\{\boldsymbol{z}_\tau\}_{\tau=1}^{t-1}$, in the form of a high-dimensional representation. Specifically, this high-dimensional state vector $\boldsymbol{\rho}_\tau$ is obtained as the state variable of a postulated recurrent neural network (RNN), with

$$\boldsymbol{\rho}_\tau = r([r_x(x_\tau);\ r_z(\boldsymbol{z}_\tau);\ \boldsymbol{\rho}_{\tau-1}]) \tag{13}$$

where $r(\cdot)$, $r_x(\cdot)$, and $r_z(\cdot)$ are DNs composed of *rectified* linear units, the (trainable) parameters of which constitute part of the vector $\boldsymbol{\varphi}$.

Based on the above formulation of ReLaVaCH, the variational posterior over the postulated latent variables $\boldsymbol{z}_t$ will

be a function of both the current observation, x_t, as well as the RNN-generated high-dimensional history representation, $\boldsymbol{\rho}_{t-1}$, $\forall t$. Nevertheless, the nonlinear expression (10) of the variance σ_t^2 as a function of the latent variables $\boldsymbol{z}_t$ makes it apparent that a Gaussian posterior assumption for the latent variables $\boldsymbol{z}_t$ is less than relevant. Based on this motivation, and to combine accuracy with computational efficiency, we elect to perform inference by utilizing the *normalizing flows*-driven variant of AVI, described in Section 2.

To this end, we postulate the *auxiliary latent variables* $\boldsymbol{z}'_t \in \mathbb{R}^D$, which we assume that yield an (accurate) Gaussian variational posterior of the form:

$$p(\boldsymbol{z}'_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1}; \boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{z}'_t|\hat{\boldsymbol{m}}_t, \mathrm{diag}(\hat{\boldsymbol{s}}_t^2)) \tag{14}$$

where

$$[\hat{\boldsymbol{m}}_t; \hat{\boldsymbol{s}}_t^2] = g_{\boldsymbol{\phi}}([\boldsymbol{x}_t; \boldsymbol{\rho}_{t-1}]) \tag{15}$$

and $g_{\boldsymbol{\phi}}(\cdot)$ is a DN comprising *rectified* linear units. Then, we assume that the original postulated latent variables, $\boldsymbol{z}_t \in \mathbb{R}^D$, can be obtained by transforming the auxiliary ones, $\boldsymbol{z}'_t$, by application of a series of transforms of the form

$$f_k(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{u}_k h(\boldsymbol{w}_k^T \boldsymbol{z} + b_k) \tag{16}$$

i.e. exploitation of the theory of planar normalizing flows. This way, the resulting posterior over $\boldsymbol{z}_t \in \mathbb{R}^D$ yields

$$\begin{aligned} \log p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1}; \boldsymbol{\phi}) =& \log p(\boldsymbol{z}'_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1}; \boldsymbol{\phi}) \\ & - \sum_k \log|1 + \boldsymbol{u}_k^T \psi_k(\boldsymbol{z}_t^k)| \end{aligned} \tag{17}$$

where $\boldsymbol{z}_t^k \triangleq f_k \circ f_{k-1} \cdots \circ f_1(\boldsymbol{z}'_t)$, and the form of $\psi_k(\boldsymbol{z})$ is given by (8). Introduction of the ReLaVaCH likelihood and prior assumptions, given by Eqs. (9) and (17), into Eq. (2), yields the expression of the ELBO of the model, optimization of which obtains an effective model inference scheme. To the latter end, in this work we resort to the popular Adagrad stochastic optimization algorithm [15], similar to [8].

4. EXPERIMENTAL EVALUATION

To evaluate the predictive performance of our approach, we consider 25 datasets, comprising the daily closing prices of 25 Equity indices from the New York Stock Exchange (NYSE), taken from January 2008 to January 2011. We convert these price time-series, p_t, into series of logarithmic returns, given by $x_t = \log\frac{p_t}{p_{t-1}}$, which we standardize to have zero mean and unit standard deviation. Each of the resulting time-series contains a total of $T = 780$ observations.

Initially, our method is trained on the first 100 data points from the obtained return signals, $x_{1:100}$. The resulting model is evaluated on the basis of one-step-ahead prediction; specifically, we compute the prediction for the σ_{100}^2 value, and evaluate our model on the basis of the resulting test-data log-likelihood pertaining to the corresponding data point, x_{100}. Subsequently, we add x_{100} to the training set, and rerun training and evaluation of our model. This procedure is repeated, one step ahead at a time, until no further data is available.

To obtain some comparative results, apart from our method, we also evaluate GPMCH, GARCH(1,1), and GJR-GARCH(1,1) under the same experimental setup. GPMCH hyperparameter and kernel selection is adopted from the corresponding paper. We implemented GARCH and GJR-GARCH using source code from Kevin Sheppard[1]. We used a MATLAB implementation of GPMCH, provided by its authors. We implemented ReLaVaCH in Python, making use of the Theano[2] [16], Lasagne[3], and Parmesan[4] libraries.

The inference DNs, parameterizing the distributions of ReLaVaCH, comprised two layers of 100 hidden units each. The dimensionality of the latent variables $\boldsymbol{z}_t$ was set to $D = 50$. We used a cascade of $K = 5$ planar transforms in the employed normalizing flows. The obtained results are provided in Table 1; these results comprise the average predictive log-likelihood of each evaluated model, over the executed runs of model training and one-step-ahead evaluation. For readers convenience, the best performance obtained in each case is typed therein in bold. As we observe, ReLaVaCH yields the highest predictive performance in most cases.

Finally, to establish the statistical significance of the observed performance differences, we use the multiple comparison approach proposed by [17]; specifically, in this context, we employ the Friedman rank sum statistical test. This procedure results in the employed test rejecting the hypothesis that all methods have equivalent performance, with p-values below 10^{-12} in all cases.

5. CONCLUSIONS

The aim of this paper was to exploit the latest advances in the field of AVI so as to alleviate the major shortcomings of existing conditional heteroscedasticity models for financial return series. Specifically, we proposed a method that obviates the need of econometric models to introduce a specific assumption regarding the functional form of the volatility dynamics in asset return series. This was effected by postulating a recurrent latent variable model with very flexible assumptions, that can learn to extract complex underlying dynamics in the modeled data. Our approach alleviates the high computational complexity of related GP-based models, that can also infer the dependency structure in the modeled data, by resorting to AVI combined with the normalizing planar flows technique. We performed an extensive experimental evaluation of our approach, considering several real-world asset return series. In most cases, we showed that our method yields a statistically significant performance improvement over the competition.

[1] `http:///www.kevinsheppard.com/wiki/UCSD_GARCH/`

[2] Available: `http://deeplearning.net/software/theano/`

[3] Available: `https://github.com/Lasagne/Lasagne`.

[4] Available: `https://github.com/casperkaae/parmesan`.

6. REFERENCES

[1] L. Chollete, A. Heinen, and A. Valdesogo, "Modeling international financial returns with a multivariate regime switching copula," *Journal of Financial Econometrics*, 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, pp. 987–1007, 1982.

[3] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 94, pp. 238–276, 1986.

[4] Yue Wu, José Miguel Hernández Lobato, and Zoubin Ghahramani, "Gaussian process volatility model," in *Proc. NIPS*, 2014.

[5] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[6] Emmanouil A. Platanios and Sotirios P. Chatzis, "Gaussian process-mixture conditional heteroscedasticity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 888–900, 2014.

[7] Andrew Naish-Guzman and Sean Holden, "The generalized FITC approximation," in *Pro. NIPS*, 2008.

[8] D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014.

[9] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. ICML*, 2014.

[10] S. J. Gershman and N. D. Goodman, "Amortized inference in probabilistic reasoning," in *Proc. Annual Conference of the Cognitive Science Society*, 2014.

[11] David M Blei, Michael I Jordan, and John W Paisley, "Variational Bayesian inference with stochastic search," in *Proc. ICML*, 2012, pp. 1367–1374.

[12] E. G. Tabak and C. V. Turner, "A family of nonparametric density estimation algorithms," *Communications on Pure and Applied Mathematics*, vol. 66, no. 2, pp. 145–164, 2013.

[13] Danilo Jimenez Rezende and Shakir Mohamed, "Variational inference with normalizing flows," in *Proc. ICML*, 2015.

[14] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.

[15] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121– 2159, 2010.

[16] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[17] J. Demsar, "Statistical comparisons of classifiers over multiple data sets.," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.