Cryptocurrency Volatility Modelling Using Gaussian Processes

Landon Buechner
Texas A&M University

Abstract

Standard stochastic volatility models such as GARCH have strong assumptions such as linearity and homogeneous effects of positive and negative returns. In attempt to address those shortcomings, I explore the application of Gaussian processes while also surveying the main ideas behind inference and model selection. I conclude by discussing an online Gaussian process regression algorithm for forecasting stochastic volatility and test the performance on BTC/USD returns.

1 Introduction

Every year billions dollars of financial instruments are traded based on forecasts generated by stochastic volatility models. These models are important tools for optimizing investment portfolios, pricing options, and hedging risk. Definitively, the most popular volatility model in finance is the Generalized Auto-Regressive Conditional Heteroscedasticity model (GARCH). It is a simple yet powerful parametric model that is able to capture volatility clustering effects expressed by financial time series. Introduced by Bollerslev in 1986, the GARCH(p,q) model assumes that asset returns are generated by a white noise process $r_t \sim \mathcal{N}(0, \sigma_t^2)$ and that the conditional variance σ_t^2 given p past values of itself and q previously observed squared returns is of the form

$$\sigma_t = \omega + \sum_{i=1}^{q} \alpha_i r_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

Despite it's widespread use, a primary limitation of the model is that does not distinguish between potential heterogeneous effects of positive and negative returns. Alternative models such as EGARCH and GJR-GARCH have been proposed in attempt to address this issue but these too are limited by the assumption of linearity. While there are a wide range of parametric non-linear models available, one way to address this is through the use of Gaussian Processes.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

2 Gaussian Processes

A Gaussian process is defined as a collection of random variables with the characteristic property that the joint distribution of any finite subset of them is Gaussian (Rasmussen 2006). Similar to how regular Gaussian random variables defined over a real support are uniquely specified by a given mean and covariance, Gaussian processes are characterized by mean $m(t)$ and kernel $k(t, x')$ which specify a Gaussian distribution over *functions* and are written as

$$f(x) \sim \mathcal{GP}\big(m(x),\, k(x, x')\big)$$

In general the domain of these functions can defined on inputs such as $\mathbf{x} \in \mathbb{R}^d$ but from here on out inputs $t \in \mathbb{R}$ are used since a stochastic volatility function f over time is the subject of interest. Additionally, if the mean function is constant a Gaussian process with mean zero can be used to model the shifted data. Note that k can be any positive-definite kernel of choice. Properties of kernel functions are not discussed here but they are important to understand in order to smartly design Gaussian process models. That being said there is rich theory behind them and interesting parallels to be drawn between Gaussian processes and other kernel based models.

2.1 Regression

Note that the notation used in this section parallels that of (Rasmussen 2006). Consider observing data $\mathbf{t} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ where the values of the underlying function $\boldsymbol{f}$ are corrupted by noise $y_i = f(t_i) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Jointly we know that these define a Gaussian process, but what if we wanted to predict the function at an unobserved point $t_* \in \mathbb{R}$? Recalling that any finite collection of Gaussian random variables is again Gaussian we have the joint distribution

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{t},\mathbf{t}) & K(\mathbf{t},t_*) \\ K(t_*,\mathbf{t}) & k(t_*,t_*) \end{bmatrix}\right)$$

where $K = K(\mathbf{t},\mathbf{t}) + \sigma^2\mathbf{I}$ is the $n \times n$ covariance matrix evaluated on the observed data set plus noise, $\mathbf{k}_* = K(\mathbf{t},t_*) = K(t_*,\mathbf{t})^T$ is the $n \times 1$ vector containing the kernel evaluated at all pairs of observed inputs and t_*, and $k(t_*,t_*)$ is just the scalar output of the kernel. Note that this construction generalizes for multiple out of sample inputs. Now all that must be done in order to find the predictive distribution of f_* is condition on $\mathbf{y}$, a straight forward operation for multivariate Gaussians.

$$f_* \sim \mathcal{N}\big(\mathbf{k}_*^T K^{-1}\mathbf{y} \, , \, k(t_*,t_*) - \mathbf{k}_*^T K^{-1}\mathbf{k}_*\big)$$

For high dimensions, the main bottleneck with Gaussian processes regression is inverting the kernel matrix but this can be addressed with standard inversion techniques for positive definite symmetric matrices.

At this point, it is still not known what the optimal hyperparameters $\boldsymbol{\theta}$ of the kernel function are. The marginal likelihood is a key object of interest in Gaussian process model selection as it allows us to find the probability of the observed data as a function of the model hyperparameters after marginalizing over the possible values of the Gaussian process prior.

$$p(\mathbf{y}|\mathbf{t},\boldsymbol{\theta}) = \int p(\mathbf{y}|\boldsymbol{f},\mathbf{t})p(\boldsymbol{f}|\mathbf{t})d\boldsymbol{f}$$

Given that the residuals of the model are i.i.d, the likelihood is the fully factorized Gaussian density corresponding to $\mathbf{y}\,|\,\boldsymbol{f} \sim \mathcal{N}\big(\boldsymbol{f},\sigma^2\mathbf{I}\big)$. As discussed, the prior over functions is the multivariate Gaussian specified by the observed data $\boldsymbol{f}\,|\,\mathbf{t} \sim \mathcal{N}\big(\mathbf{0},K\big)$. Using standard techniques for integrating products of Gaussian densities the marginal likelihood is easily found.

$$\log p\big(\mathbf{y}|\mathbf{t},\boldsymbol{\theta}\big) = -\frac{1}{2}\left(\mathbf{y}^T K^{-1}\mathbf{y} - \log|K| - n\log 2\pi\right)$$

Note that the first term measures goodness of fit whereas the second term penalizes model complexity. Given the observed data, the optimal hyperparameters can be selected by maximizing the log marginal likelihood which is straight forward using gradient based methods. Letting $\boldsymbol{\alpha} = K^{-1}\mathbf{y}$, this reduces to only needing to compute partial derivatives of the kernel matrix with respect to the hyper parameters.

$$\frac{\partial}{\partial\theta_j}\log p\big(\mathbf{y}|\mathbf{t},\boldsymbol{\theta}\big) = \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T K^{-1})\frac{\partial K}{\partial\theta_j}\right)$$

In practice it is recommended to randomly initialize the optimizer and select the best parameter in order to combat local optima.

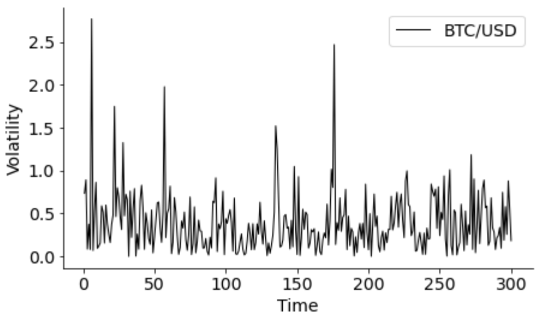
3 Volatility of Cryptocurrencies

3.1 Data

Transaction level data is obtained using the API made available by FTX, one of the main cryptocurrency exchanges. In order to recover the underling price sequence of BTC/USD, the currency used for this particular study, the data is partitioned into 30 minute windows and order-book snapshots are computed. Specifically, the average of the maximum bid and minimum ask prices are used as an estimate of the true price $p(t)$. Despite the transaction sizes contained in order-book data offering rich insights into the behavior of market participants, it is not considered in this particular study. Instead, the returns are of primary interest and are defined as

$$r(t) = 100(\log(p(t)) - \log(p(t-1)))$$

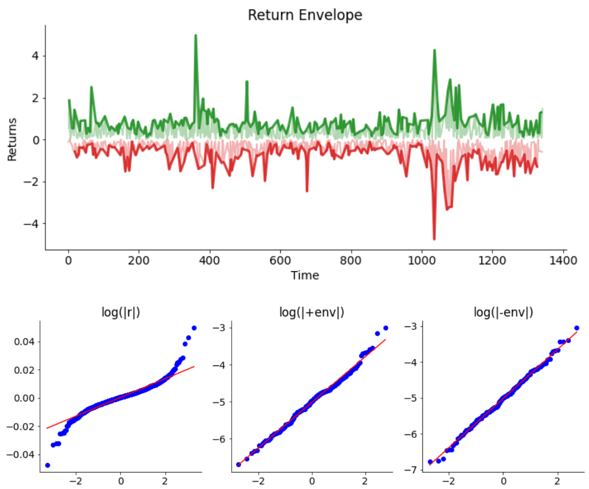
where the scale factor of 100 is so that they can be interpreted as percent change in the price. Constructing models on the first difference of the log prices is necessary because of the non-stationary nature of financial time series. When modelling stochastic volatility it is common to use either the squared or absolute returns as a surrogate for their underlying variance. I opted for working with the absolute returns $v(t) = |r(t)|$



Furthermore, as per (Rizvi 207) I separate the returns into positive and negative sequences r_+ and r_- then find the upper and the lower envelope of them respectively. Whether or not a point of any general sequence x belongs to an upper or lower envelope is

$$\begin{aligned} &\mathrm{U} \text{ , if } x(t) \geq x(t-1) \text{ and } x(t) \geq x(t+1) \\ &\mathrm{L} \text{ , if } x(t) \leq x(t-1) \text{ and } x(t) \leq x(t+1) \end{aligned}$$

For notational simplicity, from here on out I denote the volatility of these envelopes as $v_+ = |\mathrm{U}[r_+]|$ and $v_- = |\mathrm{L}[r_-]|$. It is important to note that unlike the original return sequence that takes on values for every value of t, these envelopes may have missing observations for multiple time steps. When thought about in terms of a Gaussian process regression model defined over time, this space between observations translates into increased uncertainty in predictions the longer that either positive or negative returns have not been observed. This behavior is desirable as the predictions will revert to the mean ultimately reflecting the lack of knowledge about the underlying volatility. Furthermore, taking the envelope of this helps the data better satisfy the normality assumption discussed in the introduction. The final transformation needed before regressing on the the envelopes is to convert them into log-space. This maintains the positivity constraint on the volatility predictions while also giving asymmetric confidence intervals after exponentiating.



3.2 Methodology

As discussed, this particular model set up independently models the effect of positive and negative returns on volatility. In order to predict the underlying volatility at time t, the preceding 100 observations of both transformed envelopes and are conditioned on in order to obtain their respective posterior means $\bar{f}_+(t)$ and $\bar{f}_-(t)$. These are then averaged after being exponentiated as follows

$$\bar{v}(t) = \frac{1}{2}(e^{\bar{f}_+(t)} + e^{\bar{f}_-(t)})$$

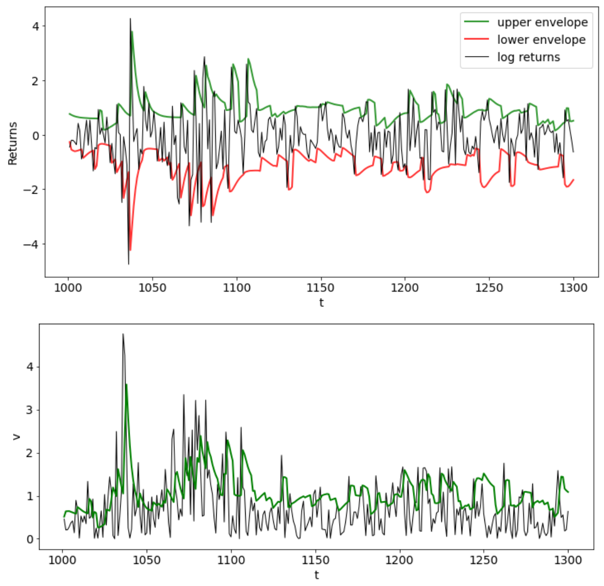
After observing the true volatility at t, the tail of the training set is dropped and the new observation is retained. Given this new training set, the log marginal likelihood is then re-maximized using gradient ascent with 10 random initializations in order to obtain the optimal hyperparameters of the kernel function. At first glance, the online nature of this model might seem computationally prohibitive in practice but it is in fact not since the time until the next observation is made is 30 minutes, a relatively high-frequency data set.

In order to evaluate the performance of the model I selected 10 disjoint return sequences of length 1100 of BTC/USD data and average the mean squared error (MSE) across all splits. Note that my choice of using a rolling window of length 100 for the training data is ad-hoc. I observed that increasing the length does not provide increased performance. Naturally, any longer would be pointless due to the asymptotic nature of kernel functions evaluated at disparate inputs. Unfortunately, I did not compare my results to the base GARCH model though this would be the first thing to do upon revisiting this analysis.

Interestingly, after some experimentation I found that the kernel defined by the sum of the Matérn-$\frac{3}{2}$ and periodic kernels offers the best performance. In the following results I provide the average performance of these kernels individually as well as results from the RBF kernel as a benchmark.

Kernel	MSE
RBF	0.436
Matérn-3/2	0.413
Matérn-3/2 + Periodic	0.329

Below is an example of out of sample predictions for the volatility envelope itself as well as their aggregate predictions defined as the average of the two. Notice that the predictions of the volatility envelope consistently track the upper bound of the volatility as desired.



References

Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning., vol. 14. the MIT Press (2006)

Rizvi, Syed and Roberts, Stephen and Osborne, Michael and Nyikosa, Favour. A Novel Approach to Forecasting Financial Volatility with Gaussian Process Envelopes, 2017.