Using Gaussian Processes to Address the Issues of Predicting Financial Time Series

Student ID:

Abstract

In this report, the statistical modelling technique *Gaussian process regression* is exploited to generate predictions over a financial time series (logarithmic returns of the cryptocurrency pair *ETH/USD*). The method is shown to overcome challenges of complex systems, such as non-linear dependencies between variables and non-stationarity, producing a model that captures the highly volatile trend of this financial time series and generalises well to future observation times.

1 Introduction

Developing a statistical model to describe and interpret the distribution of values in a financial market as it evolves is a challenging task, as this data is part of a complex system comprised of many connections and dependencies between features and variables, and requires us to formulate a multivariate probability distribution that accurately captures the properties of the dataset. Because of this complexity, using such a model to predict how outcomes will evolve—forecasting the value that these variables will take in future—is an even more challenging task.

Data from financial markets, such as prices of a given cryptocurrency, can be easily represented as a *time series*: a sequential ordering of data points where each value y is observed at a time t, with the collection of observation values Y (the dependent variable) being ordered by a chronological index of observation times X (e.g. seconds, days, or years). For example, this report focuses on the evolution of *closing prices* of the cryptocurrency pair *ETH/USD* (i.e. how the final value of Ethereum at the end of each five-minute trading window changes over time based upon exchanges between it and the US Dollar); in this case, the dependent variable $Y\colon y_0, y_1, \ldots, y_n$ is a sequence of these closing prices, with $X\colon x_0, x_1, \ldots, x_n$ denoting the times at which these prices were observed.

Hence, to make predictions of what value the variables in a time series will take in future we want to find a mapping $Y = f(X) + \varepsilon$, such that the relation between times X and values Y is represented with the smallest possible residual error ε, and extrapolate observations Y^* at unseen time points X^*. This process is known as *regression*, and can be learned *parametrically* (where we assume the parameters of the probability distribution the data is drawn from a priori) or *non-parametrically* (where no such assumption is relied upon). Due to the complex non-linear interdependencies between variables, performing regression on systems such as financial market time series can generate limited success when using traditional methods such as *linear regression* (Lin et al. 2007) or *logistic regression* (Baidoo & Priestley 2016), resulting in regressions that still have a large residual and thus cannot be relied upon to make accurate predictions.

In this report, *Gaussian process regression* (GPR) will be explored as an alternative to typical regression approaches, and its utility evaluated in the context of probabilistic prediction of financial market time series. GPR is a kernel-based non-parametric regression technique based upon *Gaussian processes* (GPs) that use Bayesian statistics (where variables are assumed to be drawn from some unknown probability distribution) to learn a regression and model a predictive distribution. This approach has been successfully used to model time series such as the Mackey-Glass differential equations (Girard et al. 2002), respiratory rate patterns (Brahim-Belhouari & Bermak 2004), and weather sensor data (Roberts et al. 2013). Limited research has also been conducted on the utilisation of GPR for financial time series, such as the short-term volatility forecasting of Liu et al. (2020) and the combination of GPs and the Monte-Carlo method to model stochastic volatility by Han et al. (2016), however,

this field has not yet been extensively explored and the potential of GPR for financial predictions showcased.

Hence, this report will explore the utilisation of GPR for the prediction of financial returns of the cryptocurrency pair ETH/USD, evaluating the benefit of exploiting GPs in this domain. This exploration will be structured as follows: initially, the challenges of financial datasets will be highlighted, and the failing points of traditional methods on these challenges identified; secondly, GPs and GPR will be introduced, exploring the methods underlying this report; finally, the results generated by GPR on our dataset will be discussed, with the benefits and issues associated with this model evaluated and future improvements considered.

2 Methodology and Data

2.1 Challenges of the Dataset

2.1.1 The Dataset

As discussed, this report focuses on the evolution of closing prices of the cryptocurrency pair ETH/USD. Specifically, over a global time horizon of 500 minutes, the price of Ethereum (ETH) in US Dollars (USD) was observed at 5-minute intervals, with the price at the end of each interval recorded as the closing price at that time. This is represented through the time series $T_{price} = (X, Y)$ where $X : x_0, x_1, \ldots, x_{99}$ are indices representing the 100-time steps over which observations were taken, and $Y : y_0, y_1, \ldots, y_{99}$ the closing prices at each of these times. Figure 1 shows the ETH/USD exchange over the 100 data points in question, starting at *13:45* on *01-01-2022* and ending at *22:00* on *01-01-2022*. The figure illustrates how an upwards global trend is seen in price over this time horizon, but fluctuations in price around this trend are common, often exhibiting large jumps in value in the positive and negative directions. These fluctuations are known as the *volatility* of the prices, which is a statistical measure of dispersion.

2.1.2 Stationarity

When developing a model for a time series, one must confirm that the underlying statistical properties of that series are consistent for the entirety of the time horizon over which we are modelling to ensure that the model remains valid. Thus we must validate that the series is statistically identical over the length of the time series, with parameters of the distribution, for example, the standard deviation, remaining constant. This is known as the *stationarity* of the data. Upon visual inspection of our ETH/USD dataset, it appears to exhibit *non-stationarity*—meaning the statistical properties do change over time—as an approximately linear trend of increasing value can be observed. This non-stational hypothesis can be checked quantitively through the *Augmented Dickey-Fuller test* (Mushtaq 2011). The test defines the null hypothesis H_0 that the time series being tested exhibits some non-zero trend, and is thus non-stationary. We then attempt to falsify this hypothesis, in turn leading us to believe the alternative hypothesis H_1 that the time series is stationary. To check the stationarity of the ETH/USD closing prices used in this report, the Augmented Dickey-Fuller test is conducted using the *adfuller* method provided by the *statsmodels* library (Seabold & Perktold 2010) which outputs a p-value (the probability under H_0 that any other series of observations is more extreme, and hence fits H_0 less than the given series). When evaluating our ETH/USD prices, a p-value of 0.5939 was output. Using the p-value threshold of 5%, we cannot reject the null hypothesis H_0 that the prices are non-stationary (as $0.5939 > 0.05$) and hence we must assume non-stationarity in our data.

Non-stationarity poses an issue for building a model of the data, hence, to accurately model this domain it must be removed. Typically, to detrend a financial dataset and remove non-stationarity, the *returns* over the closing prices are computed and used as an alternative time series that demonstrates the same information whilst being stationary (Liu et al. 2020). The return is computed as the forward difference between consecutive prices in the time series, and the logarithm of this is taken to simplify the computation process, generating the *logarithmic return* (or *log-return*); this is shown mathematically as the computation of the return R_t in Equation 1. This method is exploited over our dataset before

a model is constructed, reformulating the time series as $T_{return} = (X, Y)$ where Y: $r_0, r_1, \ldots, r_{98}$ represents the log-returns indexed by X (note as returns represent a difference, the time series will contain one less element). The Augmented Dickey-Fuller test can again be implemented to establish the stationarity of this dataset; evaluating ETH/USD log-returns generates a p-value of 8.306×10^{-13}, thus (as $8.306 \times 10^{-13} < 0.05$) the null hypothesis that the time series T_{return} is non-stationary can be rejected and the alternative hypothesis that T_{return} is stationary accepted. Hence, the subsequent work exploited the time series T_{return} over the log-returns of ETH/USD prices to ensure stationarity and build a valid model.

$$R_t = \ln\left[\frac{price\,(t)}{price\,(t-1)}\right] = \ln\left[price\,(t)\right] - \ln\left[price\,(t-1)\right] \tag{1}$$

2.1.3 Dependencies and Correlations

Beyond the stationarity of the dataset, the dependencies between variables along the time series pose a challenge for developing a model, as to infer accurate predictions the interrelations between elements have to be accurately captured. Statistically modelling dependencies is inherent to the regression process, as the mapping $y = f\,(X) + \varepsilon$ exists if and only if a relation between the observation times X and the dependent variable Y exists. However, when conducting regression over a time series, the dependencies between the variables in this series also have to be considered; namely, we must take into account that the value of variable y_i at time i is likely to be highly dependent on the variable y_{i-1} that preceded it. This is especially the case when considering financial markets where prices at consecutive times are highly dependent and hence have complex interrelations that must be correctly modelled to make accurate predictions. Furthermore, the dependencies between prices are often non-linear, meaning they cannot be represented by a simple linear map $A = p + qB + \varepsilon$. The presence of such a dependency between variables means certain regression models cannot be used, such as linear regression; additionally, non-linear dependencies are known to exacerbate biases within the dataset (Phillips & Yu 2007), posing a challenge for any remaining model.

For those variables that are linearly related, the strength of their linear dependency can be measured through their *covariance* and *correlation*. If two variables A and B are linearly independent, it's given that the *joint probability distribution* $P(A, B)$ of A and B (i.e. the distribution over conjunctions of the two variables) is simply the product of the probability distributions of each variable $P(A)$ and $P(B)$ (known as the *marginal probability distributions*). This is not the case if the variables are linearly dependent, as the joint distribution is given by $P(A, B) = P(A) * P(B \mid A)$ (where $P(B \mid A)$ is the probability of B conditional on the value of A). Hence, to quantify the linear dependency between two variables, we can measure the distance between the value of their joint distribution $P(a, b)$ and the value that the joint distribution would take if the variables were independent (i.e. $P(a) * P(b)$) over all values $a \in A, b \in B$. This is known as the covariance $Cov(A, B)$ and is computed through Equation 2.

$$Cov(A, B) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[P\,(a, b) - P\,(a)\,P\,(b)\right]dxdy \tag{2}$$

Since the size of the covariance between variables A and B depends upon the individual scales of these variables, we can a more general measure of dependency can be constructed by normalising the covariance by the product of each variable's standard deviation σ. This is known as the *correlation* between variables A and B and is shown in Equation 3. Because of this normalisation, correlation ranges between -1 and 1: $Corr(A, B) = -1$ indicates the variables are perfectly anti-correlated, $Corr(A, B) = 0$ indicates there's no linear correlation, and $Corr(A, B) = 1$ indicates a perfect correlation.

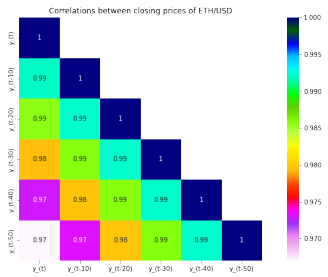
$$Corr(A, B) = \frac{Cov(A, B)}{\sigma_A \sigma_B} \tag{3}$$

Typically variables in a model dataset have to be *independent and identically distributed* (i.i.d)—i.e. independent variables drawn from the same probability distribution—to ensure modelling can be conducted. For example, both linear and logistic regression requires the assumption that all observation

variables are independent of each other, and the residual errors of each are completely uncorrelated. This means that these methods cannot be used on correlated data, as they would produce an unrepresentative distribution and significant errors in predictions. For simple datasets, dependency relations between variables can be mitigated by *shuffling* the data, randomising the map between variables. However, this is not possible for time series, as the sequential nature of variables is intrinsic to their distribution. Hence, correlations between variables can pose a significant challenge to modelling time series, especially in complex systems such as financial markets that can exhibit many complex linear and non-linear dependencies.

To exemplify the correlations between time-series data, and inform the model selection for this experimentation, the dependencies within the utilised ETH/USD dataset were checked using the *corr* method provided by the *pandas* library (Wes McKinney 2010) that produces a correlation matrix representing the dependencies between variables of the time series (produced though shifting the time series to different time lags). Figure 1 demonstrates this correlation matrix for the ETH/USD closing prices; the variable on each row and column of the matrix represents a time-lagged version of the time series, beginning with no lag (i.e. the original series) and incrementing in lags of 10 elements (e.g. y_{t-10} compares the correlation between variables and their predecessors 10 time steps previously in the series). The matrix demonstrates that even over 10 time step intervals the variables in the same series are highly positively correlated, with all correlation coefficients near 1, indicating there is a strong dependency between sequential elements of the time series, even significant time intervals between them (e.g. the correlation between each variable and the variable that precedes it by 50 time steps, or 250 minutes, is 0.97 indicating a high dependency). This interrelation is expected in financial time series, especially when explicitly considering prices, as even in extremely volatile markets the price of established assets is unlikely to repeatedly change by a significant margin (with respect to the price at the previous time step) on a minute-by-minute basis.

Figure 1: test



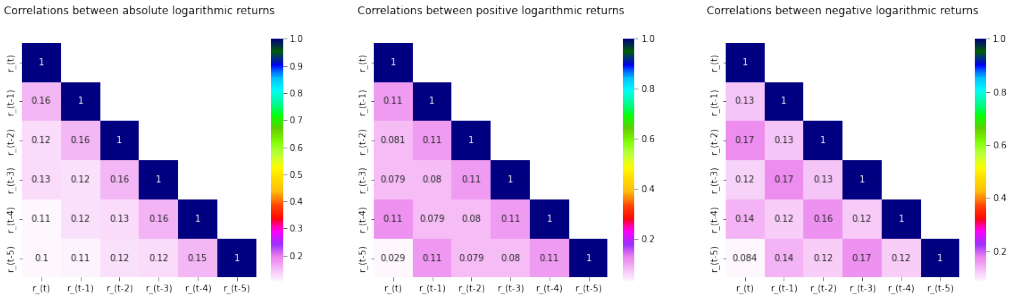
Similarly to the issue of stationarity, the linear correlation between variables in a financial time series can be reduced by focussing on the log-returns of the asset instead of closing prices. This is because taking the logarithmic returns de-trends the dataset, reducing the extent of the linear relationship between variables, and making the development of models less challenging. The *pandas* library can similarly be used to exemplify the correlations between log-returns; the correlation matrix for the log-returns, positive log-returns, and negative log-returns is shown in Figure 2 (note time lags up to 5 time steps are used, a significant reduction from when closing prices were inspected as the interdependence of variables stretches much further into the past in that case). Upon inspecting the correlation matrices for the variables of this time series, it is clear the magnitude of the correlation between log-returns is significantly less than that of closing prices, and stretches less far back into the past, although a small positive correlation is still visible meaning the variables are still dependent upon each other. It is important to remember that despite this correlation being small, it is not zero, and so the variables within the log-return time series still exhibit a linear (and possibly non-linear) dependency. Hence, as our dataset is still not in i.i.d simple models like linear and logistic regression still cannot be used in this domain.

2.2 Gaussian Process Regression

2.2.1 Bayesian Statistics and Non-Parametric Models

A common practice when building a predictive probability distribution is to assume the type of distribution the data comes from *a priori* and then learn the optimal parameters of this distribution from the dataset; as discussed, this is known as *parametric modelling*. However, this requires knowing

Figure 2: test



the distribution that our data comes from beforehand, or an extensive validation procedure to select the collect distribution. Thus, in many cases, it is preferable to construct a predictive model non-parametrically. This can be done by directly modelling the test points x^* from the testing dataset to predict the probability distribution over the possible test observations y^* without making any assumption about the underlying model of the dataset.

Bayesian statistics tells us that all variables v are drawn from some underlying probability distribution. Thus, we can predict the value a variable will take based on what prior information we know about it. For instance, say we have some probability distribution $p(v)$ specifying what we think we know about the variable v; this is known as the *prior distribution*. We can update the prior distribution based upon new information we get from observing training data points $D = (X, Y)$ that tell us more about our variable, inferring a new *posterior distribution* $p(v \mid D)$. This exact process can be done to model the predictive distribution over testing data in our time series; namely, we form the probability distribution $P(y^* \mid x^*, D)$ over possible observations y^* given the training data D and the observation time x^*. Hence, to make predictions of the next value y^* in a time series at the observation time x^* we can simply evaluate $P(y^* \mid x^*, D)$ to find the most likely value for our variable.

2.2.2 Gaussian Processes

A *Gaussian process* is a non-parametric model that uses Bayesian statistics to describe a probability distribution over a set of functions. This is done through the aforementioned concept of prior and posterior distributions. Initially, we begin with some prior distribution $p(w)$ over all possible Gaussian functions $f_{w \in W}$, where the function $y = f_w(x)$ is evaluated over a given domain through the Gaussian probability function shown in Equation 4. From Equation 4, it is clear how the function $f_w(x)$ can be manipulated by taking different values for the mean μ and standard deviation σ as parameter set $w = (\mu, \sigma)$. Hence, the prior distribtuion $p(w) = p\big(f_w\big(x \mid w = (\mu, \sigma)\big)\big)$ is constructed over all possible parameter values, with some intial probability associated with each function (e.g. a uniform distribution over all functions).

$$f\left(x \mid \mu, \sigma\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4}$$

To make predictions of the dependent variable y^* at test observation times x^* using this prior distribution, we must first construct the posterior distribution $P(w \mid X, Y)$ that describes a conditional probability distribution over the potential functions f_w that fit the data, given the dependent observation variables Y and observation times X from the time series in the training dataset. Specifically, under the distribution $P(w \mid X, Y)$ the probability of each possible function f_w is found through *Bayes' rule*, conditioning the dataset onto the parameterisation. This posterior distribution enumerates the probability that each function f_w, parameterised by the specific parameter set $w \in W, w = (\mu, \sigma)$, generated the time series (X, Y). This posterior distribution specifies the Gaussian process, describing a conditional distribution over the underlying Gaussian functions that explain a given set of observations.

References

Baidoo, E. & Priestley, J. L. (2016), An analysis of accuracy using logistic regression and time series, *in* 'Grey Literature from PhD Candidates'.

Brahim-Belhouari, S. & Bermak, A. (2004), 'Gaussian process for nonstationary time series prediction', *Computational Statistics and Data Analysis* **47**(4), 705–712.

Girard, A., Rasmussen, C. E., Candela, J. Q. n. & Murray-Smith, R. (2002), Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting, *in* 'Proceedings of the 15th International Conference on Neural Information Processing Systems', NIPS'02, MIT Press, Cambridge, MA, USA, pp. 545–552.

Han, J., Zhang, X.-P. & Wang, F. (2016), 'Gaussian process regression stochastic volatility model for financial time series', *IEEE Journal of Selected Topics in Signal Processing* **10**(6), 1015–1028.

Lin, K., Lin, Q., Zhou, C. & Yao, J. (2007), Time Series Prediction Based on Linear Regression and SVR, *in* 'Third International Conference on Natural Computation (ICNC 2007)', Vol. 1, pp. 688–691.

Liu, B., Kiskin, I. & Roberts, S. (2020), 'An overview of gaussian process regression for volatility forecasting', *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* pp. 681–686.

Mushtaq, R. (2011), 'Augmented dickey fuller test', *Econometrics: Mathematical Methods & Programming eJournal* .

Phillips, P. & Yu, J. (2007), 'Maximum likelihood and gaussian estimation of continuous time models in finance', *Handbook of Financial Time Series* .

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N. & Aigrain, S. (2013), 'Gaussian Processes for Time-Series Modelling', *Philosophical Transactions of the Royal Society (Part A* .

Seabold, S. & Perktold, J. (2010), statsmodels: Econometric and statistical modeling with python, *in* '9th Python in Science Conference'.

Wes McKinney (2010), Data Structures for Statistical Computing in Python, *in* Stéfan van der Walt & Jarrod Millman, eds, 'Proceedings of the 9th Python in Science Conference', pp. 56 – 61.