Multivariate Gaussian Process Regression for Portfolio Risk Modeling: Application to CVA

Matthew F. Dixon*
Department of Applied Mathematics
Illinois Institute of Technology

and

Stéphane Crépey†
Department of Mathematics
University of Evry

September 25, 2018

Abstract

Modeling counterparty risk is computationally challenging because it requires the simultaneous evaluation of all the trades with each counterparty under both market and credit risk. We present a multi-Gaussian process regression for estimating portfolio risk, which is well suited for OTC derivative portfolios, in particular CVA computation. Our spatio-temporal modeling approach avoids nested MC simulation by learning a 'kernel pricing layer'. The pricing layer is flexible - we model the joint posterior of the derivatives as a Gaussian over function space, with the spatial covariance structure imposed only on the risk factors. Monte-Carlo (MC) simulation is then used to simulate the dynamics of the risk factors. Our approach quantifies uncertainty in portfolio risk arising from uncertainty in point estimates. Numerical experiments demonstrate the accuracy and convergence properties of our approach for CVA estimation.

1 Overview

Post the global financial crisis of 2007-2008, banks have been subject to much stricter regulation and conservative capital and liquidity requirements. Pricing,

*Matthew Dixon is an Assistant Professor in the Department of Applied Mathematics, Illinois Institute of Technology, Chicago. E-mail: matthew.dixon@iit.edu.

†Stéphane Crépey is a Professor in the Department of Mathematics, University of Evry, Paris. E-mail: stephane.crepey@univ-evry.fr.

valuing and managing over-the-counter (OTC) derivatives has been substantially revised to more robustly capture counter-party credit risk. Pricing now includes valuation adjustments collectively known as xVAs (Abbas-Turki et al., 2018; Kenyon and Green, 2014; Crépey et al., 2014). Since the xVAs must be hedged, first-order sensitivities, such as delta and vega, are also computed.

The BCBS pointed out that 2/3 of total credit losses during the 2007-2009 crisis were CVA losses, but this risk was not capitalized under Basel II. A first CVA regulatory framework was introduced in December 2010 as part of the initial phase of the Basel III framework.

Modeling counterparty risk is computationally challenging because it requires the evaluation of all the trades with each counterparty under market and credit simulation. In practice, CVA computation requires pricing an option for each counterparty portfolio under simulated market moves, with counterparty default modeled separately. There has been much progress towards real-time CVA estimation using adjoint algorithmic differentiation to reduce the computational work for xVA sensitivities (Giles and Glasserman, 2005; Capriotti and Peacock, 2011; Capriotti, 2011; Antonov et al., 2018). The main source of computational complexity in CVA computation arises from portfolio holdings in exotic derivative contracts such as path dependent and early exercise options. Nested Monte-Carlo simulations may then be needed to evaluate the various valuation adjustments and sensitivities. Nevertheless, nested Monte Carlo simulations are still unsuited to real-time computations and, in particular, do not lead themselves to real-time what-if analysis, under which a particular market risk factor is perturbed. An alternative is nonlinear regressions in the form of least squares Monte Carlo methods a la (Longstaff and Schwartz, 2001). We note that the computational complexity is exacerbated for computation of CVA Expected Shortfall and VaR, although a full explanation of this is beyond the scope of this paper.

Spiegeleer et al. (2018) note, in the general context of derivative pricing, that many of the calculations for pricing a wide array of complex instruments, are often similar. Furthermore, the market conditions affecting OTC derivatives may often only slightly vary between observations by a few variables, such as interest rates. Rather than simulate a derivative price or Greeks, Spiegeleer et al. (2018) propose learning the pricing function, through Gaussian Process regression. Specifically, the authors configure the training set over a grid and then use the GP to interpolate at the test points. The advantage of this approach, compared to regression on historical option prices, is the ability to estimate options prices over a larger domain. On the other hand, the GP estimates depend on option pricing models, rather than just market data - somewhat counter the motivation for adopting machine learning.

Gaussian process regression, or simply Gaussian Processes (GPs), is a Bayesian kernel learning method which has demonstrated much success in spatio-temporal applications outside of finance. Their adoption in financial modeling is less widely and typically under the name of 'kriging' (see e.g. (Liu and Staum, 2009)). We refer to the reader to (Rasmussen and Williams, 2005) for an excellent general introduction to GPs. In additional to a number of favorable statisti-

cal and mathematical properties, such as universality (Micchelli et al., 2006), the implementation support infrastructure is mature - provided by `scikit-learn`, `Edward`, `STAN`, `gpTorch` and other open source machine learning packages.

Spiegeleer et al. (2018) demonstrate the speed up of GPs relative to Monte-Carlo methods and tolerable accuracy loss applied to pricing and Greek estimation with a Heston model, in addition to approximating the implied volatility surface. The increased expressibility of GPs compared to cubic spline interpolation, a popular numerical approximation techniques useful for fast point estimation, is also demonstrated.

The applications shown in (Spiegeleer et al., 2018) are limited to single instrument pricing and do not consider risk modeling aspects. In particular, their study is limited to univariate GPs (i.e. with a single response), without consideration of multivariate GPs (a.k.a. multi-GPs).

This paper presents a multivariate generalization of GPs for learning the posterior distribution of a portfolio value prediction[1]. Multi-GPs learn the joint posterior distribution of each derivative price in the portfolio, given a training set of, say, risk factors, time to maturities and derivative prices. In a single-response GP setting, individual GPs are used to model the posterior of each predicted derivative price under the assumption that the derivative prices are independent, conditional on the training data and test input. Given that either the derivatives may share common underlyings, or the underlyings are different but correlated, this assumption is clearly too restrictive.

In this context, the multi-GP model is both a theoretical and a practical innovation. Multi-GPs directly model the uncertainty in the prediction of a vector of derivative prices (responses) with spatial covariance matrices specified by kernel functions. Thus the amount of error in a portfolio value prediction, at any point in space and time, can be more comprehensively modeled using multi-GPs than single -GPs.

The need for uncertainty in the prediction is the primary practical motivation for using GPs, as opposed to frequentist machine learning techniques such as support vector machines or neural networks etc, which provide point estimates. In practice, a high uncertainty in a prediction might result in a GP model estimate being rejected in favor of either retraining the model or even using derivative model pricing.

Overview Our goal is to develop a methodology and provide numerical evidence in favor of using multi-GPs to estimate the CVA of a simple portfolio holding multiple derivatives. Note that the use of multi-GPs compared to single-GPs provides a robust approach to aggregating uncertainty in point estimates over a portfolio - accounting for the joint posterior over the options in the portfolio.

Our approach is based on training to model rather than training to data due to limitations of OTC derivative historical data. However, if historical data

[1]Through out this paper, we will refer to 'prediction' as out-of-sample point estimation. For avoidance of doubt, the test point need not be in the future as the terminology suggests.

is available, we emphasize that the methodology presented here could just as easily train to data, as demonstrated in Section 6.3.

This paper begins by reviewing GPs in the simpler setting of a single response, providing the minimal necessary terminology for the remainder of the paper. Section 3 introduces a multi-response generalization of GPs and demonstrates the application to prediction of a toy portfolio holding a call and put option. Section 4 develops the approach for portfolio risk modeling, introducing a transition density function with the view towards Monte-Carlo simulation of the risk factors. Section 5 reviews the formulation of a CVA model which uses our MC-MGP approach. Numerical experiments demonstrating the accuracy and convergence properties of the approach are presented in Section 6. Section 7 concludes.

2 Gaussian Processes

Statistical inference involves learning a function $Y = f(X)$ of the data, $(X, Y) := \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, \ldots, n\}$. The idea of Gaussian processes (GPs) is to, without parameterizing[2] $f(X)$, place a prior directly on the space of functions (MacKay, 1997). The GP is hence a Bayesian nonparametric model that generalizes the Gaussian distributions from finite dimensional vector spaces to infinite dimensional function spaces.

Before describing GPs in more detail, it is instructive to contrast GPs with classical financial modeling. In a Black-Scholes framework, noise is modeled as a Gaussian distribution in a vector space and linear diffusion of asset prices is modeled with multi-variate Geometric Brownian motion (GBM). Under the risk neutral measure, the implied drift and covariance of the GBM can be calibrated to observed pairs of asset and option prices. It well known that since derivative prices are not generated by the Black Scholes model, the calibrated parameters violate the assumption of spatial-temporal independence.

GPs do not assume a data generation process and learn a parameterized covariance function of the input through maximum likelihood estimation over all input and output pairs. GPs learn the priors over the output space without necessarily knowing the functional form of the map between input and output. So, for example, if the data is observed pairs of asset and option prices, then the GP learns the functional relationship between them. If the option prices are generated by an option pricing model, then the GP will learn the relationship between the input variables and the model option prices, without knowledge of the model.

GPs are an example of a more general class of supervised machine learning techniques referred to as 'kernel learning', which model the covariance matrix from a set of parametrized kernels over the input, rather than from the joint expectation of GBMs. The approach can consequently be referred to as 'model-free' if the data is learned without relying on an option pricing and asset dynam-

[2]This is in contrast to nonlinear regressions commonly used in finance, which attempt to parameterize a non-linear function with a set of weights.

ics model. However, in this paper, we will mainly train our GPs on simulated data.

The basic theory of prediction with Gaussian processes dates back to at least as far as the time series work of Wiener [1949] and Kolmogorov [1941] in the 1940's (Whittle and Sargent, 1983). Examples of applying GPs to financial time series prediction are presented in (Roberts et al., 2013). The same authors helpfully note that $AR(p)$ processes are discrete time equivalents of GP models with a certain class of covariance functions, known as Matérn covariance functions. Hence, GPs can be viewed as a Bayesian non-parametric generalization of well known econometrics techniques.

GPs are not new in portfolio risk modeling; da Barrosa et al. (2016) present a GP method for optimizing financial asset portfolios which allows for approximating the risk surface. Other examples of GPs include meta-modeling for Expected Shortfall through nested simulation (Liu and Staum, 2009), where GPs are used to infer portfolio values in a scenario based on inner-level simulation of nearby scenarios. This significantly reduces the required computational effort by avoiding inner-level simulation in every scenario and naturally takes account of the variance that arises from inner-level simulation.

Spiegeleer et al. (2018) demonstrate how GPs can be applied to many classical problems in derivate pricing, with speed-ups of several orders of magnitude through pricing function estimation. GPs are found to be much more accurate than spline fitting techniques commonly used in derivative modeling. Examples demonstrate the pricing of American options and the pricing of exotic options under models beyond the Black–Scholes setting.

2.1 Preliminaries

More formally, we say that a random function f is drawn from a GP with a mean function μ and a covariance kernel k, $f \sim \mathcal{GP}(\mu, k)$, if for any vector of inputs, $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, the corresponding vector of function values is Gaussian:

$$[f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)] \sim \mathcal{N}(\mu, K_{X,X}),$$

with mean μ, such that $\mu_i = \mu(\mathbf{x}_i)$, and covariance matrix $K_{X,X}$ that satisfies $(K_{X,X})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. GPs can be seen as distributions over the reproducing kernel Hilbert space (RKHS) of functions which is uniquely defined by the kernel function, k (Scholkopf and Smola, 2001). GPs with RBF kernels are known to be universal approximators with prior support to within an arbitrarily small epsilon band of any continuous function (Micchelli et al., 2006).

Assuming additive Gaussian noise, $y \mid \mathbf{x} \sim \mathcal{N}(f(\mathbf{x}), \sigma^2)$, and a GP prior on $f(\mathbf{x})$, given training inputs $\mathbf{x} \in X$ and training targets $y \in Y$, the predictive distribution of the GP evaluated at an arbitrary test point $\mathbf{x}_* \in X_*$ is:

$$\mathbf{f}_* \mid X, Y, \mathbf{x}_* \sim \mathcal{N}(\mathbb{E}[\mathbf{f}_* | X, Y, \mathbf{x}_*], \mathbb{V}[\mathbf{f}_* | X, Y, \mathbf{x}_*]), \tag{1}$$

where the moments are

$$\begin{aligned} \mathbb{E}[\mathbf{f}_* | X, Y, X_*] &= \mu_{X_*} + K_{X_*,X}[K_{X,X} + \sigma^2 I]^{-1}\mathbf{y}, \\ \mathbb{V}[\mathbf{f}_* | X, Y, X_*] &= K_{X_*,X_*} - K_{X_*,X}[K_{X,X} + \sigma^2 I]^{-1}K_{X,X_*}. \end{aligned} \tag{2}$$

Here, $K_{X_*,X}$, K_{X,X_*}, $K_{X,X}$, and K_{X_*,X_*} are matrices that consist of the kernel, $k : \mathbf{R}^p \times \mathbf{R}^p \mapsto \mathbb{R}$, evaluated at the corresponding points, X and X_*, and μ_{X_*} is the mean function evaluated on the test inputs X_*.

2.2 Hyper-parameter tuning

GPs are fit to the data by optimizing *the evidence*-the marginal probability of the data given the model with respect to the learned kernel hyperparameters.

The evidence has the form:

$$\log p(\mathbf{y} \mid \mathbf{x}, \lambda) = -\left[\mathbf{y}^\top (K + \sigma^2 I)^{-1}\mathbf{y} + \log\det(K + \sigma^2 I)\right] - \frac{n}{2}\log 2\pi, \quad (3)$$

where we use a shorthand K for $K_{X,X}$, and K implicitly depends on the kernel hyperparameters $\lambda = [\ell, \sigma]$ and ℓ is the length-scale of the Radial Basis Function (RBF) kernel:

$$cov(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') = \exp\{-\frac{1}{2\ell^2}||\mathbf{x} - \mathbf{x}'||^2\}. \quad (4)$$

This objective function consists of a *model fit* and a *complexity penalty* term that results in an automatic Occam's razor for realizable functions (Rasmussen and Ghahramani, 2001). By optimizing the evidence with respect to the kernel hyperparameters, we effectively learn the structure of the space of functional relationships between the inputs and the targets:

$$\lambda^* = \arg\max_{\lambda} \log p(\mathbf{y} \mid \mathbf{x}, \lambda) = -\left[\mathbf{y}^\top (K + \sigma^2 I)^{-1}\mathbf{y} + \log\det(K + \sigma^2 I)\right] - \frac{n}{2}\log 2\pi. \quad (5)$$

The gradient of the log likelihood is given analytically:

$$\partial_\lambda \log p(\mathbf{y} \mid \mathbf{x}, \lambda) = tr\left(\alpha\alpha^T - (K + \sigma^2 I)^{-1})\partial_\lambda (K + \sigma^2 I)^{-1}\right) \quad (6)$$

where $\alpha := (K + \sigma^2 I)^{-1}\mathbf{y}$ and

$$\partial_\ell (K + \sigma^2 I)^{-1} = -(K + \sigma^2 I)^{-2}\partial_\ell K, \quad (7)$$

$$\partial_\sigma (K + \sigma^2 I)^{-1} = -2\sigma(K + \sigma^2 I)^{-2}. \quad (8)$$

and

$$\partial_\ell k(\mathbf{x}, \mathbf{x}') = \ell^{-3}||\mathbf{x} - \mathbf{x}'||^2 k(\mathbf{x}, \mathbf{x}'). \quad (9)$$

Computational properties: Training time, which is required for maximizing (5) numerically, scales poorly with the number of observations n. This complexity stems from the need to solve linear systems and compute log determinants involving an $n \times n$ symmetric positive definite covariance matrix K. This task is commonly performed by computing the Cholesky decomposition of K incurring $\mathcal{O}(n^3)$ complexity. Prediction, however, is fast and can be performed in $\mathcal{O}(1)$, and hence the primary motivation for using GPs is real-time risk estimation performance.

Note that fast massively scalable Gaussian processes (MSGP) (Gardner et al., 2018) are a significant extension of the basic kernel interpolation framework described above. The core idea of the framework is to improve scalability by combining GPs with 'inducing point methods'. The concept is similar to multi-grid methods. A small set of inducing points are extracted from the original training points. The covariance matrix has Kronecker and Toeplitz structure which is exploited by FFT. Finally, output over the original input points is interpolated from the output at the inducing points. In this paper, we use the basic interpolation approach and leave the application of MSGP for future work.

3 Multi-response Gaussian Processes

A multivariate Gaussian process is a collection of random vector-valued variables, any finite number of which have matrix-variate Gaussian distribution. We define a multivariate Gaussian process as follows.

Definition 3.0.1 (MV-GP)**.** $\mathbf{f}$ *is a multivariate Gaussian process on* $\mathbf{R}^p$ *with vector-valued mean function* $\boldsymbol{\mu} : \mathbf{R}^p \mapsto \mathbb{R}^d$*, kernel* $k : \mathbf{R}^p \times \mathbf{R}^p \mapsto \mathbb{R}$ *and positive semi-definite parameter covariance matrix* $\Omega \in \mathbb{R}^{d\times d}$ *if the vectorization of any finite collection of vector-valued variables have a joint multi-variate Gaussian distribution,*

$$\mathrm{vec}([\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_n)]) \sim \mathcal{N}(\mathrm{vec}(M), \Sigma \otimes \Omega),$$

where $\mathbf{f}, \boldsymbol{\mu} \in \mathbb{R}^d$ *are column vectors whose components are the functions* $\{\mathbf{f}_i\}_{i=1}^d$ *and* $\{\mu_i\}_{i=1}^d$ *respectively. Furthermore,* $M \in \mathbb{R}^{d\times n}$ *with* $M_{ij} = \mu_j(\mathbf{x}_i)$*, and* $\Sigma \in \mathbb{R}^{n\times n}$ *with* $\Sigma_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$*. Sometimes* Σ *is called the column covariance matrix while* Ω *is the row covariance matrix. We denote* $\mathbf{f} \sim \mathcal{MGP}(\boldsymbol{\mu}, k, \Omega)$*.*

3.1 Multivariate Gaussian process regression

Given n pairs of observations $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n, \mathbf{x}_i \in \mathbb{R}^p, \mathbf{y}_i \in \mathbb{R}^d$, we assume the following model

$$\begin{aligned} \mathbf{f} &\sim \mathcal{MGP}(\boldsymbol{\mu}, k', \Omega), \\ \mathbf{y}_i &= \mathbf{f}(\mathbf{x}_i), i \in \{1, \ldots, n\} \end{aligned}$$

where $k' = k(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}\sigma_n^2$, and σ_n^2 is the variance of the additive Gaussian noise. With loss of generality, we follow the convention in the literature of assuming $\boldsymbol{\mu} = \mathbf{0}$.

By the definition of multivariate Gaussian process, it yields that the vectorization of the collection of functions $[\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_n)]$ follow a multivariate Gaussian distribution

$$\mathrm{vec}([\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_n)]) \sim \mathcal{N}(\mathbf{0}, K' \otimes \Omega),$$

where K' is the $n \times n$ covariance matrix of which the (i, j)-th element $[K']_{ij} = k'(\mathbf{x}_i, \mathbf{x}_j)$. See Appendix A for further details of prediction with the multi-GP model.

In the next section, we shall consider the general application of GPs to portfolio value estimation and market risk modeling. The scope of the methodology is therefore more general than CVA modeling.

4 Portfolio Value and Market Risk Estimation

The value of a portfolio of financial derivative contracts can be expressed as a linear combination of the components of $\mathbf{f}$, 'kernel pricing' functions, on a set of underlying risk factors $\mathbf{x}$

$$\pi(\mathbf{x}) = \sum_{i=1}^{N} w_i \mathbf{f}_i(\mathbf{x}) \tag{10}$$

We estimate the moments of the predictive distribution, $p(\pi_*|X, Y, X_*)$, where $\pi_* := \pi(X_*)$:

$$\begin{aligned} \mathbb{E}[\pi_*|X, Y, X_*] &= \mathbf{w}^T \hat{M}, & (11)\\ \mathrm{cov}(\pi_*|X, Y, X_*) &= \mathbf{w}^T \hat{\Sigma} \otimes \Omega \mathbf{w} - \mathbf{w}^T \hat{M} \otimes \hat{M} \mathbf{w}. & (12) \end{aligned}$$

where

$$\begin{aligned} \hat{M} &= K'(X_*, X)^T K'(X, X)^{-1} Y, & (13)\\ \hat{\Sigma} &= K'(X_*, X_*) - K'(X_*, X)^{\mathrm{T}} K'(X, X)^{-1} K'(X_*, X). & (14) \end{aligned}$$

We therefore have an expression for estimating the value of a portfolio, given the underlying risk factors, which accounts for the dependence between the financial derivative contracts. In general financial derivative contracts share common risk factors in the portfolio and the risk factors are correlated.

The integral of the marginal distribution of π over $\mathbf{x}_* \in X_*$ gives

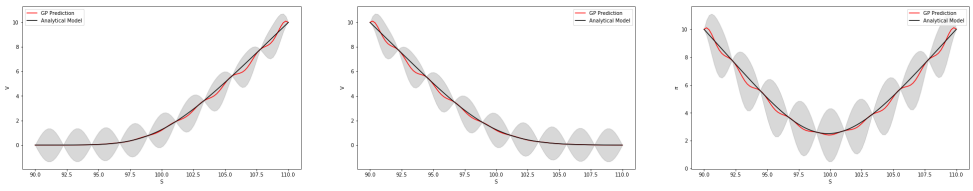
$$p(\pi|X, Y) = \int p(\pi|X, Y, \mathbf{x}_*) p(\mathbf{x}_*) d\mathbf{x}_* \tag{15}$$

where $p(\mathbf{x}_*)$ is the prior over $\mathbf{x}_*$ and π is now a scalar value, depending on the training set, and not a function of $\mathbf{x}*$. We shall see in Section 4.2 that such a distribution will be useful in portfolio risk estimation.

Example The above concepts are illustrated in Figure 1 using a equally weighted portfolio consisting of a long position in both a call option (left) and a put option (center). For ease of exposition, the time to maturity of each option is the same and assumed fixed here. In this example, there is one risk factor which is common to both options - the underlying instrument S. Each GP has been trained to (Black-Scholes) model as a function of S on a small number of training points. We use a RBF kernel for the GP.

The multi-GP subsequently estimates the price of the options at a number of test points. Some of these test points have been chosen to coincide with the

training set and others are not in the set. The test points which are also in the training set are observed to exhibit a zero width 95% confidence band, whereas test points far from observed points exhibit a wide band. The value of the portfolio at the training and test points is shown in the right hand graph. Note that the uncertainty in the point estimates is an aggregate of the uncertainty in the point estimate of each option price *and* the cross-terms in the covariance matrix in Equations 11 and 12. We emphasize, that if single GPs are used separately for the put and the call price, then the uncertainty in the point estimate would neglect the cross-terms in the covariance matrix. Multi-GPs do not, however, provide any methodology improvement in estimation of the mean.



(a) call price (b) put price (c) portfolio price

Figure 1: *Using a set of training points, the predicted mean (red line) and variance of the posterior are estimated from Equations 11 and 12 over all S_* for the (left) call option (center) put option and portfolio (right). The gray shaded envelope represents the 95% confidence interval about the mean of the posterior. The exact result, using the Black-Scholes pricing formula is given by the black line. Note that the time to maturity of the options are fixed to one year.*

4.1 Discussion

Our approach learns a kernel representation of the joint posterior distribution over the estimated derivative prices in a portfolio. This posterior is used in conjunction with a parameterized covariance function over the input space. It is important to emphasize that our approach does not fit a parameterized covariance function over the derivative prices, only to the risk factors.

The GP model, as illustrated here, is entirely spatial and financial model based. Specifically, the training set of the GP is a grid of risk factors and corresponding model option prices. We then estimate the option price at a test point, not necessary in the training set, and evaluate the means and covariance of the posterior. Kernel learning is sufficiently flexible to allow for the function to be non-smooth, as observed at, say, the maturity of the option.

The example here has no time dependency - we only learned a snapshot in time of an option surface, with a time to maturity of one year. In Section 6, we consider learning option prices as a function of underlying prices, volatility and time to maturity - fixing time to maturity for each GP and then stepping backward in time to give a sequence of GPs.

We also clarify that the underlying price and volatility dynamics are kept separate from the GP model. In using an option pricing model to train the GP, we have tacitly assumed a data generation process for the underlying and volatility dynamics. For example, in Section 6, we shall train a GP from a Heston model and evaluate the CVA by simulating the price and volatility under Heston dynamics using Monte-Carlo.

Once the 'pricing kernel layer' - consisting of a kernel representation of the prices of all options in the portfolio- has been learned, there is no need to evaluate derivate prices with a numerical pricing formula. Hence the practical utility of our multi-GP approach is the ability to quickly predict new option prices and, hence, portfolio values together with an error estimate which accounts for covariance of the derivative prices over the test points.

Moreover, the weights of the portfolio can change as the pricing kernel layer allows for dynamic weights. Thus the predictive distribution of the portfolio remains valid even when the portfolio composition changes. The one caveat is that the kernels must be relearned if a new option is added to the portfolio.

In principal, a financial 'model-free' alternative approach could be formulated by only using past observed underlying prices and option prices at different maturities. In practice, however, it is difficult to decouple the effects of each observed variable on the observed option prices, e.g. fixing price, implied volatility, and varying only time to maturity. Moreover, many OTC derivatives do not have comparable exchange traded instruments and can be illiquid. Hence we have chosen to pursue an option model based approach. However, we evaluate the potential for a model free approach in Section 6.3.

4.2 Portfolio Risk

In this section we combine our spatial kernel option pricing layer with a temporal model for the risk factors. We hence arrive at a spatio-temporal model for portfolio risk which accounts for the joint uncertainty in point estimation of the financial derivative contracts.

Under a Markovian stochastic process $(X_t)_{t\geq 0}$, the marginal distribution of the portfolio value π_{t+h} at time $t+h$, given $X_t = \mathbf{x}$, is

$$p(\pi_{t+h}|X, Y, X_t = \mathbf{x}) = \int p(\pi(\mathbf{x}_*)|X, Y, \mathbf{x}_*)p(X_{t+h} = \mathbf{x}_*|X_t = \mathbf{x})d\mathbf{x}_* \quad (16)$$

where the multi-variate transition density function $p(X_{t+h}|X_t)$ for $(X_t)_{t\geq 0}$ is determined by a diffusion model or estimated from historical data. The distribution of the future portfolio value depends on the uncertainty from the point distribution $p(\pi_*|X, Y, \mathbf{x}_*)$. Note that if $X_{t+h} = \mathbf{x}_* \in X$, then the uncertainty in the estimate π_* is zero.

Note if the risk factor is not an observable, or the risk manager simply seeks to express uncertainty in the current risk factor value $p(X_t)$, then the more

general form of Equation 16 can be used

$$p(\pi_{t+h}|X,Y) = \int\int p(\pi(\mathbf{x}_*)|X,Y,\mathbf{x}_*)p(X_{t+h}=\mathbf{x}_*|X_t=\mathbf{x})p(X_t=\mathbf{x})d\mathbf{x}_*.d\mathbf{x} \tag{17}$$

In Section 6, we use Equation 17 to estimate the expected future exposure of a portfolio and associated kernel approximation error for CVA estimation. However, our kernel approach described above is general and valid for any portfolio risk measure such as VaR, Expected Shortfall and techniques such as stress testing.

Computational aspects We emphasize that the benefit of using GPs is primarily computational. The training time of each GP is $\mathcal{O}(n^3)$, where n is the number of observations. If the option depends on several risk factors, then $n = \prod_i n_i$, where n_i are the number of grid points per risk factor. Note that although each kernel matrix $K_{X,X}$ is $n \times n$, we only store the n-vector α for each option, which brings reduced memory requirements.

4.2.1 Simulation

Typically $p(X_{t+h} = \mathbf{x}_*|X_t = \mathbf{x})$ is not known in closed form and must be estimated with Monte-Carlo simulation. Hence, our approach combines Monte-Carlo simulation with MGP pricing to estimate the portfolio risk. We refer to this approach as MC-MGP.

5 CVA

As an example of a portfolio risk application, we consider the estimation of counter-party credit risk on a client portfolio. In this case the weights w_i in (10) are typically 0 or 1. The expected loss to the investor, associated with the counterparty defaulting, is given by the unilateral CVA. CVA is the expected cost of the default risk, or equivalently, the expected cost of counterparty risk. Taking expectations with respect to the risk neutral measure for a numeraire N_t at time t, the loss from recovery on the market value of the portfolio is given by

$$\text{CVA}_0 = (1-R)\int_0^T \mathbb{E}[\pi_u^+ N_u^{-1}\delta(u,\tau)du] \tag{18}$$

where π_t is the value of the portfolio and τ is the default time. If the default is independent of both the portfolio value and the numeraire then the above expression simplifies to

$$\text{CVA}_0 = (1-R)\int_0^T \mathbb{E}[\pi_u^+ N_u^{-1}]p(u)du, \tag{19}$$

where $\pi(X)_t$ is assumed to only depend on time only through the market risk factors X_t (i.e. the portfolio weights are fixed in time), and $p(t)$ is default probability density function. To hedge the CVA, a set of n dates $t_1, \ldots, T = t_n$ is chosen over which to evaluate the expected positive exposure $\mathbb{E}[\pi_t^+ N_t^{-1}]$. A multivariate stochastic process $(\mathbf{X}_t)_{t \geq 0}$ determines the market risk factors, such as asset price processes, which govern the portfolio's market value. The discounted asset price processes are martingales with respect to an equivalent martingale measure. The contingent claims, such as options, are priced with respect to this same measure.

The credit risk component of CVA can be modeled in reduced form with a Poisson default model. Under a deterministic hazard rate, interval default probabilities in period $[t_i, t_{i+1})$ for the counterparty are given by the difference of the exponential survival probabilities

$$P(t_i \leq \tau < t_{i+1}) = \exp\{-\int_{s=0}^{t_i} \lambda(s)ds\} - \exp\{-\int_{s=0}^{t_{i+1}} \lambda(s)ds\} \tag{20}$$

which can be approximated, for example, by the discrete time evolution of piecewise linear hazard rate $\lambda(s)$

$$\Delta p_i := p_i - p_{i+1} := \exp\{-\sum_{j=0}^{i-1}(t_{j+1} - t_j)\lambda_j\} - \exp\{-\sum_{j=0}^{i}(t_{j+1} - t_j)\lambda_j\}. \tag{21}$$

5.1 Multi Gaussian Process Regression estimation of CVA

Starting with a Monte-Carlo estimate of the CVA over M paths, along which the market risk factors are sampled:

$$\text{CVA}_M = \frac{(1-R)}{M} \sum_{j=1}^{M} \sum_{i=1}^{n} \pi(X_{t_i}^{(j)})^+ (N_{t_i}^{(j)})^{-1} \Delta p_i \tag{22}$$

we replace the exact derivative prices with the mean of the posterior function conditioned on the simulated market risk factors $\mathbf{X}_{t_i}$:

$$\widehat{\text{CVA}}_M = \frac{(1-R)}{M} \sum_{j=1}^{M} \sum_{i=1}^{n} |\mathbb{E}[\pi_* | X, Y, \mathbf{x}_* = \mathbf{X}_{t_i}^{(j)}]|^+ (N_{t_i}^{(j)})^{-1} \Delta p_i \tag{23}$$

and MGP error estimate, based on the covariance of the posterior of π_*, evaluated over each sample path:

$$\epsilon_M = \frac{(1-R)}{M} \sum_{j=1}^{M} \sum_{i=1}^{n} \mathbf{1}_{(\mathbb{E}[\pi_* | \cdots] > 0)} \text{cov}(\pi_* | X, Y, \mathbf{x}_* = \mathbf{X}_{t_i}^{(j)}) (N_{t_i}^{(j)})^{-1} \Delta p_i. \tag{24}$$

The above approximation uses Gaussian Process regression to estimate the potential future exposure of the portfolio. We note that the pricing models

are still fitted to model generated data, assuming a data generation process for the risk factors. However, we have used machine learning to learn derivative exposure as a function of the underlying and other parameters such as time to maturity, by slicing in time. In this way, we avoid nested Monte-Carlo simulations, which are computationally intractable for large portfolios. Moreover, the multi-GP regressions provides an estimation of the amount of error in the point estimation of the portfolio value.

6 Numerical Experiments

In the following example, we use our MC-MGP simulation to estimate the CVA of the portfolio from Equations 23 and 24. For simplicity, we continue the example shown in Section 4 - the portfolio holds a long position in both a European call and a put option struck on the same underlying. The only difference between the earlier example is that we now assume that the underlying follows Heston dynamics:

$$\frac{dS_t}{S_t} = \mu dt + \sqrt{V_t} dW_t^1, \tag{25}$$

$$dV_t = \kappa(\theta - V_t)dt + \sigma\sqrt{V_t} dW_t^2, \tag{26}$$

$$\mathbb{E}[dW_t^1 \cdot dW_t^2] = \rho dt. \tag{27}$$

where the notation and fixed parameter values used for experiments are given in Table 1 under $\mu = r_0$. We use a Fourier Cosine method (Fang and Oosterlee, 2008) to generate the European Heston option price training and testing data for the GP. We also use this method to compare the GP Greeks, obtained by differentiating the kernel function.

Parameter description	Symbol	Value
Mean reversion rate	κ	0.1
Mean reversion level	θ	0.15
Vol. of Vol.	σ	0.1
Risk free rate	r_0	0.002
Strike	K	100
Maturity	T	1.0
Correlation	ρ	-0.9

Table 1: This table shows the values of the parameters for the Heston dynamics and terms of the European Call and Put option contracts.

For the corresponding intervals used for the CVA estimate, we simultaneously fit a multi-GP to both gridded call and put prices over price and volatility, keeping time to maturity fixed. Figure 2 shows the gridded call (top) and put (bottom) price surfaces at various time to maturities, together with the GP

estimate. Within each column in the figure, the same GP model has been simultaneously fitted to both the call and put price surfaces over a 30×30 grid $\Omega_h \subset \Omega := [0,1] \times [0,1]$ of prices and volatilities[3], fixing the time to maturity. The scaling to the unit domain is not essential. However, we observed superior numerical stability when scaling.

Across each column, corresponding to different time to maturities, a different GP model has been fitted. The GP is then evaluated out-of-sample over a 40×40 grid $\Omega_{h'} \subset \Omega$, so that many of the test samples are new to the model. This is repeated over various time to maturities corresponding to, say, the hedging periods in a CVA model. The option model versus GP model are observed to produce very similar results.

Table 1 lists the values of the parameters for the Heston dynamics and terms of the European Call and Put option contract used in our numerical experiments. Tables 2 and 3 show the values for the Euler time stepper used for simulating Heston dynamics and the credit risk model.

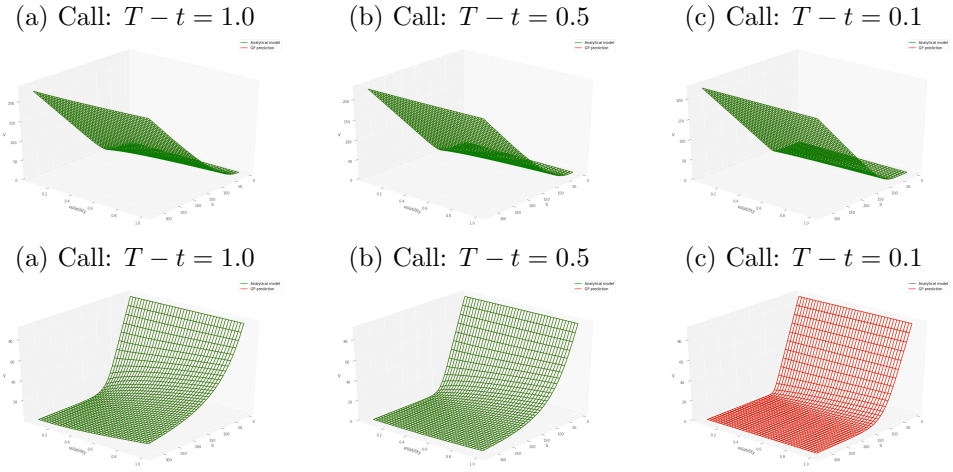


Figure 2: *This figure shows the gridded call (top) and put (bottom) price surfaces at various time to maturities, together with the GP estimate. Within each column in the figure, the same GP model has been simultaneously fitted to both the call and put price surfaces over a* 30×30 *grid of prices and volatilities, fixing the time to maturity. Across each column, corresponding to different time to maturities, a different GP model has been fitted. The GP is then evaluated out-of-sample over a* 40×40 *grid, so that many of the test samples are new to the model. This is repeated over various time to maturities corresponding to hedging periods.*

Figure 3 compares the (left) full-MC and MC-MGP estimate of the expected positive exposure of the portfolio over time. The error in the MC-MGP estimate

[3]Note that the plot uses the original coordinates and not the re-scaled co-ordinates.

Parameter description	Symbol	Value
Number of simulation	M	1000
Number of time steps	n_s	100
Initial stock price	S_0	100
Initial variance	V_0	0.1

Table 2: *This table shows the values for the Euler time stepper used for market risk factor simulation.*

Constant hazard rate	λ	0.1
Number of default horizons	n	10
Recover rate	R	0.4

Table 3: *This table shows the parameters of the reduced form credit risk model used for estimating the CVA in our numerical experiments.*

and 95% uncertainty band, exclusive of the MC sampling error, is also shown against time (right).

Figure 4 shows how the error in the MC-MGP CVA estimate versus MC with full portfolio evaluation decays against the number of training samples used for each GP model. The 95% confidence band of the MC-MGP prediction, exclusive of the MC sampling error, is also shown. Note that while the training samples are varied, the 40×40 testing set remains fixed during the experiment.

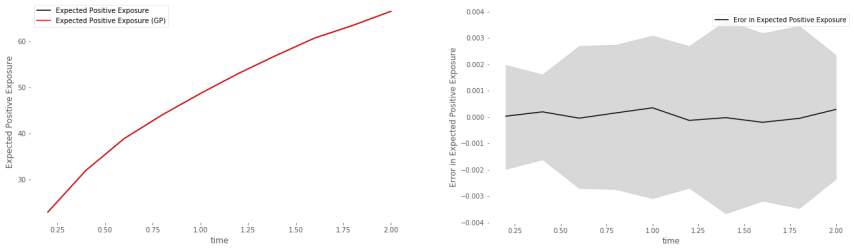


Figure 3: *(Left) Full-MC and MC-MGP estimate of the expected positive exposure of the portfolio over time. The two graphs are practically indistinguishable, with one graph superimposed over the other. (Right) The error in the MC-MGP estimate and 95% uncertainty band (exclusive of the MC sampling error) is also shown against time.*

6.1 CVA VaR

In this section, we demonstrate the application of GPs to the estimation of the Value-at-risk (VaR) of a one year incremental CVA. The purpose of the calculation is to estimate, at a given confidence level, the extent which to the

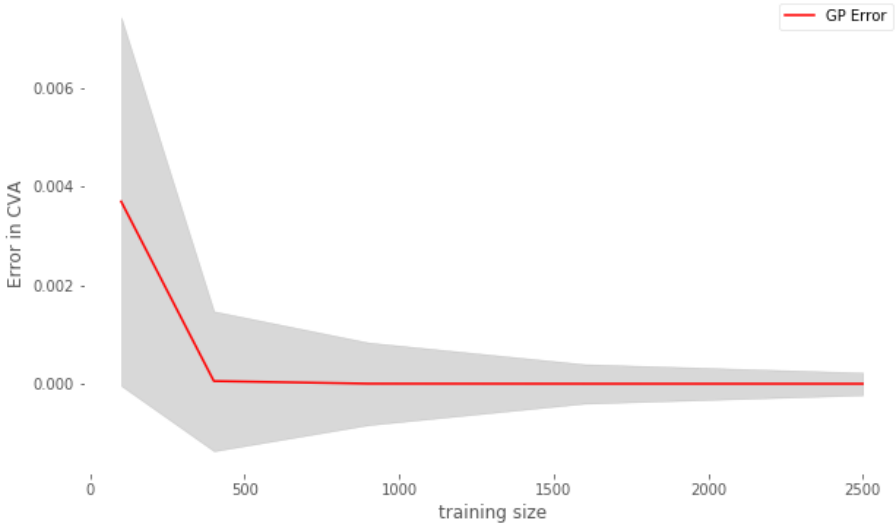


Figure 4: *This figure shows how the percentage error in the MC-MGP CVA estimate versus MC with full portfolio evaluation decays against the number of training samples used for each GP model. The 95% confidence band in the GP prediction is also shown centered about MC-MGP CVA estimation error.*

CVA will increase over the next year. More precisely, we estimate the loss distribution corresponding to the one year ahead CVA minus the CVA_0 estimate.

We model the CVA process as

$$\mathbf{1}_{t<\tau} CVA(t, \mathbf{X}_t) = \mathbf{1}_{t<\tau} \mathbf{E}[\mathbf{1}_{\tau<T} C(\tau, S_t) \mid S_t, t < \tau] \tag{28}$$

where the intensity λ is assumed constant, under zero interest rates. Figure 5 shows the CVA VaR, as estimated with a full MC and a MC-MGP method. In order to isolate the effect of the MGP approximation, we use identical random numbers for each method. The left hand plots compares the full-MC and MC-MGP out-of-sample estimate of the CVA loss distribution; the two graphs are practically indistinguishable, with one graph superimposed over the other. Note that the reason for the sharp approximation is two-fold: (i) the statistical experiment has been configured as an interpolation problem, with many of the gridded training points close to the gridded test points; and (ii) the training sample size of 900 is relatively large to approximate smooth surfaces (with no outliers). The right hand plot shows the error between the full MC and MC-MGP estimate as a distribution of the CVA loss.

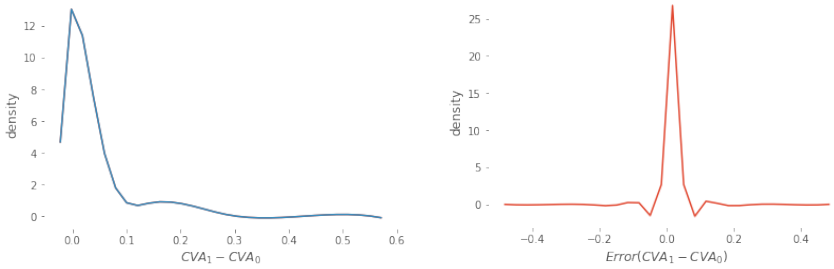


Figure 5: *(Left) full-MC and MC-MGP out-of-sample estimate of the CVA loss distribution; the two graphs are practically indistinguishable, with one graph superimposed over the other. Note that the reason for the sharp approximation is two-fold: (i) the statistical experiment has been configured as an interpolation problem, with many of the gridded training points close to the gridded test points; and (ii) the training sample size of 900 is relatively large to approximate smooth surfaces (with no outliers). (Right) The error between the full MC and the MC-MGP estimate as a distribution of the CVA loss.*

6.2 Derivatives

The GP provides analytic derivatives with respect to the input variables

$$\partial_{X_*}\mathbb{E}[\mathbf{f}_*|X, Y, X_*] = \partial_{X_*}\mu_{X_*} + \partial_{X_*}K_{X_*,X}\alpha, \tag{29}$$

where $\partial_{X_*}K_{X_*,X} = \frac{1}{\ell^2}(X - X_*)K_{X_*,X}$ and recall that $\alpha = [K_{X,X} + \sigma^2 I]^{-1}\mathbf{y}$. Note that α is already calculated at training time (for pricing) by Cholesky matrix factorization of $[K_{X,X} + \sigma^2 I]$ with $\mathcal{O}(n^3)$ complexity, so there is no significant computational overhead from greek estimation. Once the GP has learned the derivative prices, Equation 29 is used to evaluate the first order Greeks with respect to the input variables over the test set. Example source code illustrating the implementation of this calculation using given in Section B.

Figure 6 shows the GP estimate of the call option's vega ν (left), having trained on the volatility and Heston Cosine option model prices, fixing the underlying asset price for simplicity of computations. For avoidance of doubt, the model is not trained on the Heston Cosine model vegas. For comparison in the figure, the Heston Cosine estimate of ν is also shown. The two graphs are practically indistinguishable, with one graph superimposed over the other. The error in the vega estimate is observed to converge with the number of training samples for the GP. Note that the number of training samples is relatively small compared to other experiments in this section on account of our choice to fix S and train on volatility and option prices.

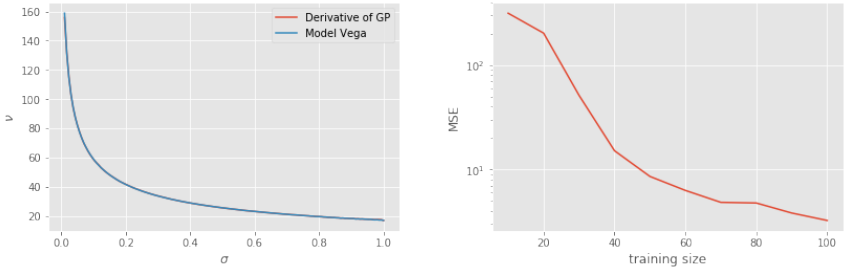


Figure 6: *This figure shows the GP estimate of the call option's vega ν (left), having trained on the volatility and Heston Cosine option model prices, fixing the underlying asset price for simplicity of computations. For avoidance of doubt, the model is not trained on the Heston Cosine model vegas. For comparison in the figure, the Heston Cosine estimate of ν is also shown. The two graphs are practically indistinguishable, with one graph superimposed over the other. The error in the vega estimate is observed to converge with the number of training samples for the GP.*

6.3 Model free price estimation

In this section, we estimate equity option prices from historical observations of underlying price, time-to-maturity, strike, volatility, option type and option prices. In our dataset[4] each option chain is observed over four snapshots in time. For each chain, we separate calls and puts and construct a training set from the moneyness, volatility, time-to-maturity and option price using three of the snapshots (approximately 1300 observations). The most recent snapshot is reserved for testing.

Figure 7 compares the (left) GP estimate of the call prices (blue), having trained from the joint observations of the moneyness, maturity and volatility, with the observed out-of-sample call prices (red). The training data is shown with gray points. Note that the volatility is not shown in the figure. (Right) The error in the GP estimate, with and without volatility as an input variable, is compared with the observed call prices in the test set against moneyness for a fixed maturity (2 years). We note that the figure shows the importance of including volatility as an input variable. In particular, the uncertainty in the GP estimate is observed to be large if the volatility is excluded.

[4]The dataset has been downloaded from https://mamamomama.org on September 20th, 2018.

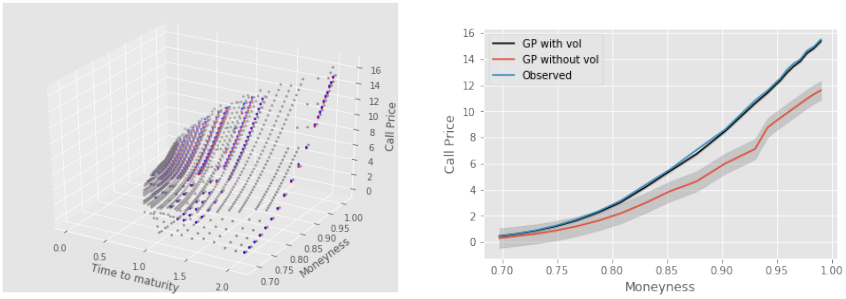


Figure 7: *This figure compares the (left) GP estimate of the call prices (blue), having trained from the joint observations of the moneyness, maturity and volatility, with the observed out-of-sample call prices (red). The training data is shown with gray points. Note that the volatility is not shown in the figure. (Right) The error in the GP estimate, with and without volatility as an input variable, is compared with the observed call prices in the test set against moneyness for a fixed maturity (2 years).*

7 Conclusion

This paper introduces a MC-MGP approach for fast evaluation of derivative portfolios and their risk. The approach is demonstrated by estimating the CVA on a simple portfolio with numerical studies of accuracy and convergence of MC-MGP estimates. The primary advantage of kernel learning over Monte-Carlo with full repricing is computational - there is no need to use expensive derivative pricing functions for risk point estimates once the kernels have been learned. The kernels permit a closed form approximation for the sensitivity of the portfolio risk to the risk factors and the approach preserves the flexibility to rebalance the portfolio. However, the advantage is more than just computational. The risk estimation approach is Bayesian - the uncertainty in a point estimate which the model hasn't seen in the training data is quantified and can be factored into the risk estimate. Additionally, derivatives of the pricing kernel layer are given analytical and hence avoid the use of numerical differentiation.

A Prediction with Multi-GPs

To predict a new variable $\mathbf{f}_* = [\mathbf{f}_{*1}, \ldots, \mathbf{f}_{*m}]$ at the test locations $X_* = [\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}]$, the joint distribution of the training observations $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ and the predictive targets $\mathbf{f}_*$ are given by

$$\begin{bmatrix} Y \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{MN}\left(\mathbf{0}, \begin{bmatrix} K'(X,X) & K'(X_*,X)^T \\ K'(X_*,X) & K'(X_*,X_*) \end{bmatrix}, \Omega\right), \tag{30}$$

where $K'(X,X)$ is an $n \times n$ matrix of which the (i,j)-th element $[K'(X,X)]_{ij} = k'(x_i, x_j)$, $K'(X_*,X)$ is an $m \times n$ matrix of which the (i,j)-th element $[K'(X_*,X)]_{ij} = k'(x_{n+i}, x_j)$, and $K'(X_*,X_*)$ is an $m \times m$ matrix with the (i,j)-th element $[K'(X_*,X_*)]_{ij} = k'(x_{n+i}, x_{n+j})$. Thus, taking advantage of conditional distribution of multivariate Gaussian process, the predictive distribution is

$$p(\mathrm{vec}(\mathbf{f}_*)|X, Y, X_*) = \mathcal{N}(\mathrm{vec}(\hat{M}), \hat{\Sigma} \otimes \hat{\Omega}), \tag{31}$$

where

$$\begin{aligned} \hat{M} &= K'(X_*,X)^T K'(X,X)^{-1} Y, & (32)\\ \hat{\Sigma} &= K'(X_*,X_*) - K'(X_*,X)^{\mathrm{T}} K'(X,X)^{-1} K'(X_*,X), & (33)\\ \hat{\Omega} &= \Omega. & (34) \end{aligned}$$

Additionally, the expectation and the covariance are obtained,

$$\begin{aligned} \mathbb{E}[\mathbf{f}_*|X, Y, X_*] &= \hat{M}, & (35)\\ \mathrm{cov}(\mathrm{vec}(\mathbf{f}_*)|X, Y, X_*) &= \hat{\Sigma} \otimes \hat{\Omega}. & (36) \end{aligned}$$

The hyperparameters and elements of the covariance matrix Ω are found by minimizing the negative log marginal likelihood of observations:

$$\mathcal{L}(Y|X, \lambda, \Omega) = \frac{nd}{2} \ln(2\pi) + \frac{d}{2} \ln|K'| + \frac{n}{2} \ln|\Omega| + \frac{1}{2} tr((K')^{-1} Y \Omega^{-1} Y^T). \tag{37}$$

B GP Greeks

This Python 3.0 code, using `scikit-learn` excerpt illustrates how to calculate the derivative of the option by differentiating the GP price model. If x are gridded volatilities, then f_prime is the estimate of the vega. If x were gridded underlying prices, then f_prime is the estimate of the delta.

```
x = np.linspace(0.01,1.0, training_number)
x_train = np.array(x, dtype='float32').reshape(training_number, 1)

y_train = []

for idx in range(len(x_train)):
  y_train.append(PyHeston.HestonCall(S0, x_train[idx], K, time, r, lmbda, meanV, sigma, rho, 0.4))
y_train = np.array(y_train)
gp = gaussian_process.GaussianProcessRegressor(kernel=sk_kernel, n_restarts_optimizer=20)
gp.fit(x_train,y_train)
y_pred, sigma_hat = gp.predict(x_test, return_std=True)

k_s = rbf(x_test, x_train)

k_s_prime = np.zeros([len(x_test), len(x_train)])
for i in range(len(x_test)):
    for j in range(len(x_train)):
            k_s_prime[i,j]=(1.0/l**2)*(x_train[j]-x_test[i])*k_s[i,j]

f_prime = np.dot(k_s_prime, alpha_p)
```

References

Abbas-Turki, L. A., S. Crépey, and B. Diallo (2018, February). XVA Principles, Nested Monte Carlo Strategies, and GPU Optimizations. working paper or preprint.

Antonov, A., S. Issakov, A. McClelland, and S. Mechkov (2018). Pathwise XVA Greeks for early-exercise products. *Risk Magazine* (January).

Capriotti, L., J. L. and M. Peacock (2011). Real-time counterparty credit risk management in monte carlo. *Risk 24*(6).

Capriotti, L. (2011). Fast greeks by algorithmic differentiation. *Journal of Computational Finance 14*(3), 3–35.

Crépey, S., T. Bielecki, and D. Brigo (2014). *Counterparty Risk and Funding*. New York: Chapman and Hall/CRC.

da Barrosa, M. R., A. V. Salles, and C. de Oliveira Ribeiro (2016). Portfolio optimization through kriging methods. *Applied Economics 48*(50), 4894–4905.

Fang, F. and C. W. Oosterlee (2008). A novel pricing method for european options based on fourier-cosine series expansions. *SIAM J. SCI. COMPUT.*

Gardner, J., G. Pleiss, R. Wu, K. Weinberger, and A. Wilson (2018). Product kernel interpolation for scalable gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1407–1416.

Giles, M. and P. Glasserman (2005). Smoking adjoints: fast evaluation of greeks in monte carlo calculations. Technical report.

Kenyon, C. and A. Green (2014). Efficient xva management: Pricing, hedging, and attribution using trade-level regression and global conditioning. Papers, arXiv.org.

Liu, M. and J. Staum (2009). Stochastic kriging for efficient nested simulation of expected shortfall. *Journal of Risk*.

Longstaff, F. A. and E. S. Schwartz (2001). Valuing american options by simulation: A simple least-squares approach. *The Review of Financial Studies 14*(1), 113–147.

MacKay, D. J. (1997). Gaussian processes - a replacement for supervised neural networks?

Micchelli, C. A., Y. Xu, and H. Zhang (2006, December). Universal kernels. *J. Mach. Learn. Res. 7*, 2651–2667.

Rasmussen, C. E. and Z. Ghahramani (2001). Occam's razor. In *In Advances in Neural Information Processing Systems 13*, pp. 294–300. MIT Press.

Rasmussen, C. E. and C. K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Roberts, S., M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain (2013). Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 371*(1984).

Scholkopf, B. and A. J. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Spiegeleer, J. D., D. B. Madan, S. Reyners, and W. Schoutens (2018). Machine learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance 0*(0), 1–9.

Whittle, P. and T. J. Sargent (1983). *Prediction and Regulation by Linear Least-Square Methods* (NED - New edition ed.). University of Minnesota Press.